Oracle® Communications

Cloud Native Core Automated Testing Suite Guide



Release 25.1.203
G33884-07
October 2025

Oracle Communications Cloud Native Core Automated Testing Suite Guide, Release 25.1.203

G33884-07

Copyright © 2019, 2025, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Documentation Accessibility i
Diversity and Inclusion i
Conventions i

1 Introduction

1.1 Overview 1
1.2 Deployment Model 2
1.3 References 3

2 ATS Framework Features

2.1 ATS API 3
2.1.1 Generating an API Token for a User 3
2.1.2 Use the RESTful Interfaces 5
2.1.2.1 Configuring Host 5
2.1.2.2 Starting Jobs 6
2.1.2.3 Monitoring Jobs 15
2.1.2.4 Stopping Jobs 17
2.1.2.5 Getting Test Suite Artifacts 19
2.2 ATS Custom Abort 22
2.3 ATS Feature Activation and Deactivation 25
2.4 ATS GUI Enhancements 27
2.5 ATS Health Check 29
2.6 ATS Jenkins Job Queue 31
2.7 Application Log Collection 32
2.7.1 Application Log Collection and Parallel Test Execution Integration 34
2.8 ATS Maintenance Scripts 35
2.8.1 ATS Scripts 36
2.8.2 Updating ats_install.sh 37
2.8.3 Restarting Jenkins without Restarting Pod 37
2.8.4 Updating Stub Scripts 38
2.8.5 Running ATS and Stub Deployment Scripts 39

Copyright © 2019, 2025, Oracle and/or its affiliates.

2.9 ATS System Name and Version Display on the ATS GUI 39
2.10 ATS Tagging Support 39
2.10.1 Combination of Tags and their Results 41
2.11 Custom Folder Implementation 44
2.12 Health Check 45
2.13 Individual Stage Group Selection 54
2.14 Lightweight Performance 55
2.14.1 Configure <NF>-Performance Pipeline 55
2.15 Managing Final Summary Report, Build Color, and Application Log 56
2.16 Modifying Login Password 63
2.17 Multiselection Capability for Features and Scenarios 64
2.17.1 Feature Level Selection 64
2.17.2 Scenario Selection 65
2.18 Parallel Test Execution 66
2.18.1 ATS GUI Page Changes 67
2.18.2 ATS Console Log Changes 68
2.18.3 Downloading or Viewing Individual Group Logs 69
2.19 Parameterization 72
2.19.1 Running Test Cases 73
2.20 PCAP Log Collection 75
2.20.1 Application Log Collection and Parallel Test Execution Integration 75
2.21 Persistent Volume for 5G ATS 76
2.21.1 Processing Flow 76
2.21.2 Deploying Persistent Volume 76
2.21.3 Backward Porting 81
2.22 Single Click Job Creation 81
2.22.1 Configuring Single Click Job 81
2.23 Support for ATS Deployment in OCI 84
2.23.1 Accessing ATS GUI in OCI 84
2.23.2 Adding Ingress and Egress Rules to Access the OCI Console 85
2.24 Support for Transport Layer Security 89
2.24.1 Deploy ATS with TLS Enabled 90
2.24.1.1 Generate JKS File for Jenkins Server 90
2.24.1.2 Enable ATS GUI with HTTPS 95
2.25 Test Results Analyzer 97
2.25.1 Accessing Test Results Analyzer Feature 97
2.26 Support for Test Case Mapping and Count 100
2.26.1 Access Test Case Mapping and Count Feature 100

3 Installing ATS for Different Network Functions

3.1 Installing ATS for BSF 1

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.1.1 Resource Requirements 1
3.1.2 Downloading the ATS Package 4
3.1.3 Deploy ATS with TLS Enabled 5
3.1.3.1 Generate JKS File for Jenkins Server 5
3.1.3.2 Enable TLS on Python Stubs 9
3.1.3.3 Enable ATS GUI with HTTPS 11
3.1.4 Pushing the Images to Customer Docker Registry 12
3.1.5 Configuring ATS 14
3.1.5.1 Enabling Static Port 14
3.1.5.2 Enable Static API Node Port 14
3.1.5.3 Service Account Requirements 15
3.1.5.4 Enabling Aspen Service Mesh 15
3.1.5.5 Enabling Health Check 18
3.1.5.6 Enabling Persistent Volume 20
3.1.5.7 ATS-BSF API Extended Support 22
3.1.6 Deploying ATS and Pods 23
3.1.6.1 Deploying ATS in Kubernetes Cluster 23
3.1.6.2 Deploying Stub Pod in Kubernetes Cluster 24
3.1.6.3 Deploying DNS Stub in Kubernetes Cluster 29
3.1.6.4 Deploying ocdiam Simulator in Kubernetes Cluster 30
3.1.7 Post-Installation Steps 32
3.2 Installing ATS for NRF 34
3.2.1 Resource Requirements 34
3.2.2 Downloading the ATS Package 35
3.2.3 Pushing the Images to Customer Docker Registry 36
3.2.4 Configuring ATS 37
3.2.4.1 Enabling Static Port 37
3.2.4.2 Enabling Aspen Service Mesh 37
3.2.4.3 Enabling Persistent Volume 39
3.2.4.4 Enabling NF FQDN Authentication 40
3.2.5 Deploying ATS and Stub in Kubernetes Cluster 40
3.2.6 Verifying ATS Deployment 41
3.2.7 Post-Installation Steps (if Persistent Volume is Used) 43
3.3 Installing ATS for NSSF 43
3.3.1 Resource Requirements 43
3.3.2 Locating and Downloading ATS and Simulator Images 45
3.3.3 Deploying ATS in Kubernetes Cluster 48
3.4 Installing ATS for Policy 58
3.4.1 Resource Requirements 58
3.4.2 Downloading the ATS Package 62
3.4.3 Deploy Policy ATS with TLS enabled 64
3.4.3.1 Generate JKS File for Jenkins Server 64

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.4.3.2 Enable TLS on Python Stubs 68
3.4.3.3 Enable TLS on ocams Stubs 70
3.4.3.4 Enable TLS on Diam-Sim 72
3.4.3.5 Enable ATS GUI with HTTPS 73
3.4.4 Pushing the Images to Customer Docker Registry 75
3.4.5 Configuring ATS 77
3.4.5.1 Enabling Static Port 77
3.4.5.2 Enabling Static API Node Port 77
3.4.5.3 Service Account Requirements 77
3.4.5.4 Enabling Aspen Service Mesh 78
3.4.5.5 Enabling Persistent Volume 81
3.4.5.6 Enabling Health Check 84
3.4.5.7 ATS-Policy API Extended Support 85
3.4.6 Deploying ATS and Pods 86
3.4.6.1 Deploying ATS in Kubernetes Cluster 86
3.4.6.2 Deploying Stub Pod in Kubernetes Cluster 86
3.4.6.3 Deploying DNS Stub in Kubernetes Cluster 93
3.4.6.4 Deploying AMF Stub in Kubernetes Cluster 95
3.4.6.5 Deploying LDAP Stub in Kubernetes Cluster 96
3.4.6.6 Deploying ocdiam Simulator in Kubernetes Cluster 96
3.4.7 Post-Installation Steps 99
3.5 Installing ATS for SCP 101
3.5.1 Prerequisites 101
3.5.1.1 Software Requirements 101
3.5.1.2 Environment Setup Requirements 101
3.5.1.3 Resource Requirements 102
3.5.1.4 Downloading the ATS Package 104
3.5.1.5 Pushing the Images to Customer Docker Registry 113
3.5.1.6 Preinstall Preparation of SCP for SCP-ATS 114
3.5.1.7 Preinstallation Preparation for SCP-ATS 119
3.5.2 Configuring SCP-ATS and SCP 123
3.5.2.1 Updating ATS Configuration 123
3.5.3 Deploying ATS and Stub in the Kubernetes Cluster 139
3.5.4 Post Installation and Deployment Steps 140
3.5.4.1 Verifying ATS Deployment 140
3.5.4.2 Modifying the scpc-alternate-resolution Microservice 140
3.5.4.3 ATS Testcase Parametrization on User Input 141
3.5.4.4 ATS Tagging Support for SCP 150
3.5.5 Appendix 151
3.5.5.1 Creating Custom Service Account 151
3.5.5.2 Adding New Stubs to SCP-ATS 151
3.6 Installing ATS for SEPP 152

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.6.1 Resource Requirements 152
3.6.1.1 SEPP ATS Compatibility Matrix 155
3.6.2 Downloading the ATS Pakage 155
3.6.3 Pushing the Images to Customer Docker Registry 156
3.6.4 Creating Secrets and Supporting TLS 1.2 and TLS 1.3 157
3.6.4.1 Configuring Root Certificate 157
3.6.4.2 Generating ATS Certificate 159
3.6.4.3 Creating ATS Secret and Configuring Helm chart 163
3.6.5 Create ATS Health Check secret 165
3.6.6 Configuring ATS 166
3.6.6.1 Enabling Aspen Service Mesh 166
3.6.6.2 ATS API 168
3.6.6.3 Enabling Static Port 168
3.6.6.4 Enabling Roaming Hub Mode 169
3.6.6.5 Enabling Hosted SEPP Mode 169
3.6.6.6 Configuring Egress Rate Limiting Feature 169
3.6.6.7 Configuring Ingress Rate Limiting Feature 170
3.6.6.8 Configuring Single Service Account 170
3.6.6.9 Configuring the Cache Refresh Timeout Value 170
3.6.6.10 Configuring the Topology Cache Refresh Timeout Value 171
3.6.6.11 Configuring the Security Counter Measure Cache Refresh Timeout 171
3.6.6.12 Configuring n32cHandshakePlmnIdListValidationEnabled 172
3.6.6.13 Configuring Request Timeout 175
3.6.6.14 Idle Timeout 176
3.6.6.15 Configuring Polling Interval in N32 Egress Gateway 176
3.6.6.16 EvictSanHeaderCacheDelay 177
3.6.6.17 Configuring the Reroute Attempts for Egress Gateway 177
3.6.6.18 Configuring Alternate Routing based on the DNS SRV Record for Home Network Functions 177
3.6.6.19 Configuring Load Sharing among Multiple Remote SEPP Nodes 179
3.6.6.20 Configuring Dual Stack 180
3.6.6.21 Enabling Persistant Volume Storage 180
3.6.6.22 ATS Tagging Support 181
3.6.6.23 Configuring LCI and OCI Header Support 183
3.6.7 Deploying ATS in Kubernetes Cluster 184
3.6.8 Verifying ATS Deployment 185
3.6.9 Post Installation Steps (If persistent volume is used) 185
3.7 Installing ATS for UDR 185
3.7.1 Resource Requirements 185
3.7.2 Locating and Downloading ATS Image 201
3.7.3 Preparing to Deploy ATS and Stub Pods 203
3.7.4 Loading UDR ATS Image 203

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.7.5 Loading UDR Stub Images in the SLF-NewFeatures or SLF-Regression Pipeline 206
3.7.6 Loading UDR Stub Images in the UDR-NewFeatures or UDR-Regression Pipeline 207
3.7.7 Loading EIR Stub Images in the EIR-NewFeatures or EIR-Regression Pipeline 210
3.7.8 Configuring ATS 211
3.7.8.1 Configuring Docker Registry 212
3.7.8.2 Enabling Static Port 212
3.7.8.3 Enabling Persistent Volume 212
3.7.8.4 Settings for OAuth2 Test Cases 213
3.7.8.5 Enabling IPv6 on ATS 214
3.7.8.6 Configuring ATS to Run Health-Check Pipeline 214
3.7.8.7 Creating Service Account 215
3.7.8.8 Enabling Service Mesh 216
3.7.8.9 Configuring Stubs 218
3.7.9 Deploying ATS in Kubernetes Cluster 223
3.7.9.1 Deploying ATS Pod and Stubs in Kubernetes Cluster 223
3.7.10 Verifying ATS Deployment 224
3.7.11 Post Installation Steps 229

4 Running NF Test Cases Using ATS

4.1 Running BSF Test Cases using ATS 1
4.1.1 Prerequisites 1
4.1.2 Logging into ATS 6
4.1.3 Running BSF_NewFeatures Pipeline 7
4.1.4 Running BSF_HealthCheck Pipeline 13
4.1.5 BSF_NewFeatures Documentation 13
4.1.6 Running BSF Regression Pipeline 15
4.1.7 BSF_Regression Documentation 16
4.2 Running NRF Test Cases using ATS 17
4.2.1 Prerequisites 17
4.2.2 Logging into ATS 26
4.2.3 NRF-NewFeatures Pipeline 28
4.2.4 NRF-NewFeatures Documentation 38
4.2.5 NRF-Regression Pipeline 40
4.2.6 NRF-Regression Documentation 41
4.3 Running NSSF Test Cases using ATS 44
4.3.1 Prerequisites 44
4.3.2 Logging into ATS 44
4.3.3 NSSF-NewFeatures Pipeline 46
4.3.4 NSSF-NewFeatures Documentation 56
4.3.5 NSSF-Regression Pipeline 58

Copyright © 2019, 2025, Oracle and/or its affiliates.

4.3.6 NSSF-Regression Documentation 66
4.4 Running Policy Test Cases using ATS 68
4.4.1 Prerequisites 69
4.4.2 Logging into ATS 75
4.4.3 Running Policy-HealthCheck Pipeline 76
4.4.4 Running Policy-NewFeatures Pipeline 77
4.4.5 Policy-NewFeatures Documentation 84
4.4.6 Running Policy-Regression Pipeline 85
4.4.7 Policy-Regression Documentation 87
4.5 Running SCP Test Cases using ATS 88
4.5.1 Logging in to ATS 89
4.5.2 SCP-NewFeatures Pipeline 91
4.5.2.1 Configuring New Feature Pipelines 91
4.5.2.2 Running SCP New Feature Pipelines 96
4.5.3 SCP-NewFeatures Documentation 97
4.5.4 SCP-Regression Pipeline 99
4.5.4.1 Configuring Regression Pipelines 99
4.5.4.2 Running Regression Pipelines 102
4.5.5 SCP-Regression Documentation 103
4.5.5.1 Renaming of Regression Feature Files 105
4.6 Running SEPP Test Cases using ATS 113
4.6.1 Prerequisites 113
4.6.2 Logging into ATS 114
4.6.3 SEPP NewFeatures Pipeline 116
4.6.3.1 Single Click Job Creation for SEPP New Features 120
4.6.4 NewFeatures - Documentation 122
4.6.5 SEPP Regression Pipeline 123
4.6.5.1 Single Click Job Creation for SEPP Regression Features 124
4.6.6 SEPP Regression Documentation 126
4.6.7 Running SEPP Health Check Pipeline 127
4.7 Running UDR Test Cases using ATS 129
4.7.1 Prerequisites 129
4.7.1.1 Prerequisites for SLF Pipelines 130
4.7.1.2 Prerequisites for UDR Pipelines 140
4.7.1.3 Prerequisites for EIR Pipelines 153
4.7.2 UDR Golden Configuration Support 159
4.7.3 UDR Application Log Collection 163
4.7.4 Parallel Test Execution 166
4.7.5 Logging into ATS 196
4.7.6 Running SLF Pipelines 197
4.7.6.1 Running SLF-NewFeatures Pipeline 198
4.7.6.2 Running SLF Regression Pipeline - Single Site Deployment 203

Copyright © 2019, 2025, Oracle and/or its affiliates.

4.7.6.3 Running SLF Regression Pipeline - Multiple Site Deployment 207
4.7.6.4 Running SLF HealthCheck Pipeline 211
4.7.6.5 Running SLF-Performance Pipeline: Model 2: Single Site Deployment 213
4.7.6.6 SLF NewFeatures Documentation - Single Site Scenarios 215
4.7.6.7 SLF Regression Documentation - Single Site Scenarios 216
4.7.6.8 SLF Regression Documentation - Multiple Site Scenarios 217
4.7.7 Running UDR Pipelines 217
4.7.7.1 Running UDR NewFeatures Pipeline 218
4.7.7.2 Running UDR Regression Pipeline 224
4.7.7.3 Running UDR HealthCheck Pipeline 232
4.7.7.4 UDR NewFeatures Documentation 234
4.7.7.5 UDR Regression Documentation 235
4.7.8 Running EIR Pipelines 235
4.7.8.1 Running EIR NewFeatures Pipeline 235
4.7.8.2 Running EIR Regression Pipeline 240
4.7.8.3 Running EIR HealthCheck Pipeline 244
4.7.8.4 EIR NewFeatures Documentation 245
4.7.8.5 EIR Regression Documentation 246

Copyright © 2019, 2025, Oracle and/or its affiliates.

Preface

- Documentation Accessibility
- Diversity and Inclusion
- Conventions

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Copyright © 2019, 2025, Oracle and/or its affiliates.

My Oracle Support

My Oracle Support (https://support.oracle.com) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. When calling, make the selections in the sequence shown below on the Support telephone menu:

- For Technical issues such as creating a new Service Request (SR), select **1**.
- For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.
- For Hardware, Networking and Solaris Operating System Support, select **3**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Acronyms

The following table lists the acronyms and the terminologies used in the document:

Table 1 Acronyms

Acronym	Meaning
ATS	Oracle Communications Cloud Native Core, Automated Test Suite
BDD	Behavior Driven Development
BSF	Oracle Communications Cloud Native Core, Binding Support Function
CI	Continuous Integration
CSR	Certificate Signing Request
EIR	Equipment Identity Register
Policy	Oracle Communications Cloud Native Core, Converged Policy
NRF	Oracle Communications Cloud Native Core, Network Repository Function
NSSF	Oracle Communications Cloud Native Core, Network Slice Selection Function
CNE	Oracle Communications Cloud Native Core, Cloud Native Environment
OL	Oracle Linux is a Linux distribution packaged and freely distributed by Oracle, available partially under the GNU (General Public License) since late 2006. It is compiled from Red Hat Enterprise Linux source code, replacing Red Hat branding with Oracle's.
PCF	Oracle Communications Cloud Native Core, Policy Control Function
PVC	Persistent Volume Claim
cnPCRF	Oracle Communications Cloud Native Core, Policy and Charging Rules Function
SAN	Subject Alternative Name
SCP	Oracle Communications Cloud Native Core, Service Communication Proxy
SEPP	Oracle Communications Cloud Native Core, Security Edge Protection Proxy
SLF	Subscriber Location Function
UDR	Oracle Communications Cloud Native Core, Unified Data Repository

Copyright © 2019, 2025, Oracle and/or its affiliates.

What's New in This Guide

This section introduces the documentation updates for release 25.1.2xx.

ATS Framework Release 25.1.203 - G33884-07, October 2025

SEPP ATS Release 25.1.203

- ATS release tag for SEPP is changed to 25.1.203.
- Updated the images according to release 25.1.203.

NRF Release 25.1.203

- The ATS release tag for NRF is changed to 25.1.203.
- Updated the following sections with 25.1.203 release updates:
  - Prerequisites
  - Logging into ATS
  - NRF-NewFeatures Pipeline
  - NRF-Regression Pipeline
- **Number of new Test Cases added to pipeline**:
  - **New Features**: **12** feature files with **32** scenarios.
  - **Regression**: **632** feature files with **1,777** scenarios.
    - "2-Site-Geo" cases: 7 feature files with 19 scenarios.
    - "3-Site-Geo" cases: 10 feature files with 14 scenarios.
    - "NF-FQDN-Authentication-Feature" cases: 6 feature files with 28 scenarios.
    - "CCA-Header" cases: 6 feature files with 19 scenarios.
    - "NRF-Growth" cases: 22 feature files with 30 scenarios.
  - **Total regression count**: **683** feature files with **1,887** scenarios.

SCP Release 25.1.202

- ATS release tag for SCP is changed to 25.1.202.
- Number of test cases in SCP_NewFeatures pipeline: No additional feature files from the last release.
- Number of test cases in the SCP_Regression pipeline: No additional feature files from the last release.
- Number of sanity features: No sanity features.

NSSF ATS Release 25.1.201

- ATS release tag for NSSF is changed to 25.1.201.
- Updated the images according to release 25.1.201.
- Number of test cases in NSSF_NewFeatures pipeline: 3 new feature files with 25 new scenarios.
- Number of test cases in the NSSF_Regression pipeline: 80 feature files with 812 scenarios.

Copyright © 2019, 2025, Oracle and/or its affiliates.

ATS Framework Release 25.1.202 - G33884-06, October 2025

SEPP ATS Release 25.1.202

- ATS release tag for SEPP is changed to 25.1.202.
- Updated the images according to release 25.1.202.
- Added the Configuring LCI and OCI Header Support section.
- Number of test cases in SEPP_NewFeatures pipeline (in SEPP mode): 11 new feature files and 104 new scenarios.

ATS Framework Release 25.1.202 - G33884-05, October 2025

NRF Release 25.1.202

- The ATS release tag for NRF is changed to 25.1.202.
- Updated the following sections with 25.1.202 release updates:
  - Prerequisites
  - Logging into ATS
  - NRF-NewFeatures Pipeline
  - NRF-Regression Pipeline
- **Number of new Test Cases added to pipeline**:
  - **New Features**: **12** feature files with **32** scenarios.
  - **Regression**: **632** feature files with **1,777** scenarios.
    - "2-Site-Geo" cases: 7 feature files with 19 scenarios (4 scenarios added newly).
    - "3-Site-Geo" cases: 10 feature files with 14 scenarios.
    - "NF-FQDN-Authentication-Feature" cases: 6 feature files with 19 scenarios. (1 new feature file added)
    - "CCA-Header" cases: 6 feature files with 19 scenarios (1 scenario added newly).
    - "NRF-Growth" cases: 22 feature files with 30 scenarios.
  - **Total regression count**: **683** feature files with **1,887** scenarios.

ATS Framework Release 25.1.202 - G33884-03, September 2025

Updated the following sections with the details of ATS 25.1.202:

SCP Release 25.1.201

- ATS release tag for SCP is changed to 25.1.201.
- Number of test cases in SCP_NewFeatures pipeline: No additional feature files from the last release.
- Number of test cases in the SCP_Regression pipeline: No additional feature files from the last release.
- Number of sanity features: No sanity features.

BSF Release 25.1.200

- Updated the ATS package versions for BSF.
- Updated the Prerequisites section with details to update the BSF custom values.yaml file.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Policy Release 25.1.200

- Updated the ATS package versions for Policy.
- Updated the Prerequisites section with details to update the Policy custom values.yaml file.

ATS Framework Release 25.1.202 - G33884-02, July 2025

Updated the following sections with the details of ATS 25.1.200:

SEPP ATS Release 25.1.201

- ATS release tag for SEPP is changed to 25.1.201.
- Updated multiple images according to release 25.1.201.

ATS Framework Release 25.1.202 - G33884-01, July 2025

Updated the following sections with the details of ATS 25.1.200:

Policy Release 25.1.200

- ATS release tag for Policy is changed to 25.1.200.
- Updated the following sections with 25.1.200 release updates under **Installing ATS for Policy**:
  - Downloading the ATS Package
  - Pushing the Images to Customer Docker Registry
- Updated the following sections with 25.1.200 release updates under **Running Policy Test Cases using ATS**:
  - Logging into ATS
  - Running Policy-HealthCheck Pipeline
  - Running Policy-NewFeatures Pipeline
  - Policy-NewFeatures Documentation
  - Running Policy-Regression Pipeline
  - Policy-Regression Documentation
- Number of new Test Cases added to pipeline (PCF): Provides a total of **11 feature files** and **24 scenarios** of Policy_NewFeatures pipeline.
- Number of new Test Cases added to pipeline (cnPCRF): Provides a total of **1 feature file** and **2 scenarios** of Policy_NewFeatures pipeline.
- Number of new Test Cases added to pipeline (Converged Policy): Provides a total of **19 feature files** and **54 scenarios** of Policy_NewFeatures pipeline.
- Previous Release Test Cases (PCF): Provides a total of **220 feature files** and **643 scenarios** of Policy_Regression pipeline.
- Previous Release Test Cases (cnPCRF): Provides a total of **28 feature files** and **109 scenarios** of Policy_Regression pipeline.
- Previous Release Test Cases (Converged Policy): Provides a total of **282 feature files** and **859 scenarios** of Policy_Regression pipeline.

BSF Release 25.1.200

- ATS release tag for BSF is changed to 25.1.200.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- Updated the following sections with 25.1.200 release updates under **Installing ATS for BSF**:
  - Downloading the ATS Package
  - Pushing the Images to Customer Docker Registry
- Updated the following sections with 25.1.200 release updates under **Running BSF Test Cases using ATS**:
  - Logging into ATS
  - Running BSF_HealthCheck Pipeline
  - Running BSF_NewFeatures Pipeline
  - BSF_NewFeatures Documentation
  - Running BSF Regression Pipeline
  - BSF_Regression Documentation
- Number of new Test Cases added to pipeline: Provides a total of **1 feature file** and **10 scenario** of BSF_NewFeatures pipeline.
- Previous Release Test Cases: Provides a total of **29 feature files** and **102 scenarios** of BSF_Regression pipeline.

NRF Release 25.1.200

- ATS release tag for NRF is changed to 25.1.200.
- Updated the following sections with 25.1.200 release updates:
  - Prerequisites
  - Logging into ATS
  - NRF-NewFeatures Pipeline
  - NRF-NewFeatures Documentation
  - NRF-Regression Pipeline
  - NRF-Regression Documentation
- **Number of new Test Cases added to pipeline**:
  - **New Features**: 11 feature files with 29 scenarios.
  - **Regression**: 633 feature files with 1777 scenarios.
    - 2-Site-Geo" cases: 7 feature files with 19 scenarios (4 scenarios added newly)
    - "3-Site-Geo" cases: 10 feature files with 14 scenarios.
    - "NF-FQDN-Authentication-Feature" cases: 5 feature files with 25 scenarios.
    - "CCA-Header" cases: 6 feature files with 18 scenarios.
    - "NRF-Growth": 22 feature files with 30 scenarios.
  - **Total regression count**: 684 feature files with 1894 scenarios.

NSSF Release 25.1.200

- ATS release tag for NSSF is changed to 25.1.200.
- Number of test cases in NSSF_NewFeatures pipeline: 3 new feature files with 25 new scenarios.
- Number of test cases in the NSSF_Regression pipeline: 80 feature files with 812 scenarios.

Copyright © 2019, 2025, Oracle and/or its affiliates.

SCP Release 25.1.200

- ATS release tag for SCP is changed to 25.1.200.
- Number of test cases in SCP_NewFeatures pipeline: 44 new feature files with 96 new scenarios for Release 16 and no new feature file with no new scenarios for Release 15 SCP deployments.
- Number of test cases in the SCP_Regression pipeline: 461 feature files with more than 3000 scenarios for Release 16 and no new feature file with no new scenarios for Release 15 SCP deployments.
- Number of sanity features: 27 sanity features and 46 test cases for Release 16. Sanity test cases can be run to perform a quick sanity check on SCP SUT features.
- Updated the following sections with 25.1.200 release updates:
  - ATS Tagging Support for SCP
  - Logging in to ATS
  - Configuring New Feature Pipelines
  - Running SCP New Feature Pipelines
  - Configuring Regression Pipelines
  - SCP-Regression Documentation

SEPP Release 25.1.200

- ATS release tag for SEPP is changed to 25.1.200.
- Updated multiple images according to release 25.1.200.
- Updated the SEPP ATS Compatibility Matrix section.
- The ATS Tagging Support section is updated to support the ATS Tagging Support feature.
- Added the Single Click Job Creation for SEPP New Features
- Number of test cases in SEPP_NewFeatures pipeline (in SEPP mode): 10 new feature files and 100 new scenarios.
- Number of test cases in SEPP_Regression pipeline (in SEPP mode): 178 feature files and 1270 scenarios.
- Number of test cases in SEPP_NewFeatures pipeline (in Roaming Hub mode): 4 new feature files and 19 new scenarios.
- Number of test cases in SEPP_Regression pipeline (in Roaming Hub mode): 149 feature files and 794 scenarios.

UDR Release 25.1.200

- ATS release tag for UDR is changed to 25.1.200.
- Updated multiple images according to release 25.1.200.
- **Number of Test Cases in UDR-NewFeatures pipeline:** Provides a total of **46** test cases.
- **Number of Test Cases in EIR-NewFeatures pipeline:** No new test cases added.
- **Number of Test Cases in SLF-NewFeatures pipeline:** Provides a total of **1** test cases.
- **Number of Test Cases in SLF-Regression pipeline: 410** test cases with **21** MultiSite scenarios.
- **Number of Test Cases in UDR-Regression pipeline: 1425** test cases with **22** MultiSite scenarios.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- **Number of Test Cases in EIR-Regression pipeline: 336** test cases.
- Updated the Table 4-14 to add SLF_Subscriber_Export_File_Cleanup.feature.
- Updated the UDR NewFeatures Documentation to add the following features:
  - UDR_Subscriber_Policy_Data_Export.feature
  - UDR_PNR_Multisite.feature
  - UDR_Bug_Fix_25_1_100.feature
  - UDR_Default_Headers.feature

Copyright © 2019, 2025, Oracle and/or its affiliates.

1 Introduction

This document provides information about Automated Testing Suite (ATS) deployment model for 5G Network Functions (NFs).

ATS allows you to run software test cases using an automated testing tool. It compares the actual results with the expected or predicted results.

1.1 Overview

Using ATS, you can deploy and test 5G NFs.

This document provides information to implement ATS for the following 5G NFs:

- Oracle Communications Cloud Native Core, Binding Support Function (BSF)
- Oracle Communications Cloud Native Core, Network Repository Function (NRF)
- Oracle Communications Cloud Native Core, Network Slice Selection Function (NSSF)
- Oracle Communications Cloud Native Core, Converged Policy (Policy)
- Oracle Communications Cloud Native Core, Service Communication Proxy (SCP)
- Oracle Communications Cloud Native Core, Security Edge Protection Proxy (SEPP)
- Oracle Communications Cloud Native Core, Unified Data Repository (UDR)

ATS for 5G Network Functions

For 5G NFs, ATS is developed using Oracle Linux 8-slim as the base image. Jenkins is a part of the ATS image, and it provides a graphical user interface (GUI) to test either a single NF or multiple NFs independently in the same network environment.

ATS comprises NF docker images, ATS image, simulator images, and test cases of the specific NF. All these images and test cases constitute a fully automated suite to deploy and test NFs. You can combine it with any other Continuous Integration (CI) pipeline with minimal changes because 5G ATS uses Jenkins as GUI.

The ATS package contains the following elements:

- Test scripts and docker images of test container. The docker images have complete framework and libraries installed, which are common for all NFs working with the Behavior Driven Development (BDD) framework.
- Docker image of HTTP Server simulator.
- Helm chart to deploy the ATS (delivered as a tar file).
- Readme text file (.txt file).

ATS provides basic environment, framework, and a GUI (Jenkins) to run all the functional test cases.

Copyright © 2019, 2025, Oracle and/or its affiliates.

ATS Framework Version with Supporting NF Version

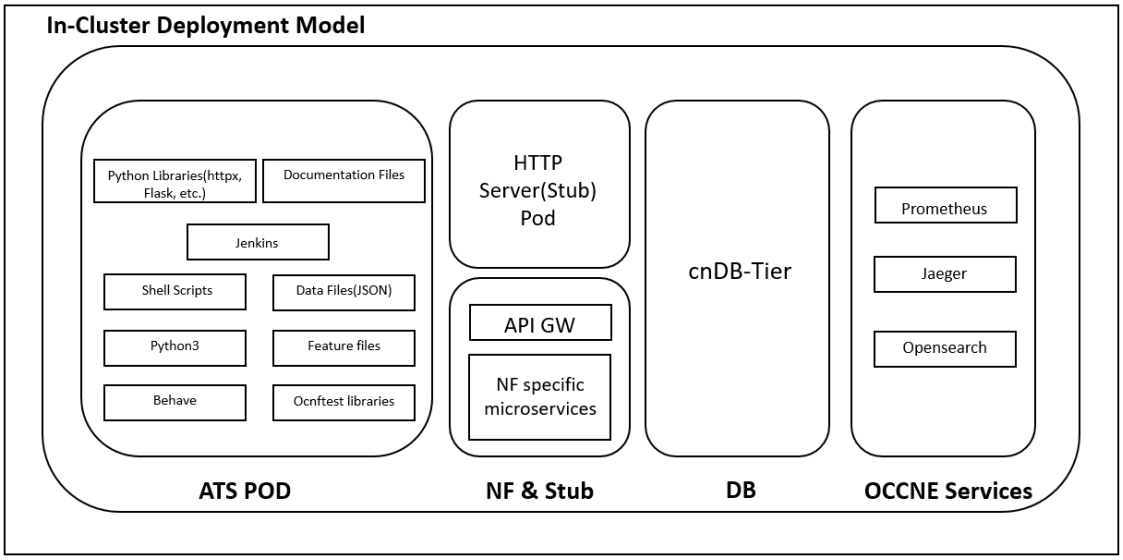
Table 1-1 ATS Framework Version with Supporting NF

ATS Framework Version	BSF	NRF	NSSF	Policy	SCP	SEPP	UDR
25.1.201	25.1.200	-	-	25.1.200	-	-	-
25.1.202	-	25.1.200, 25.1.202	25.1.200	-	25.1.200, 25.1.201	25.1.200, 25.1.201	25.1.200
25.1.203	-	25.1.203	25.1.201	-	25.1.202	-	-

1.2 Deployment Model

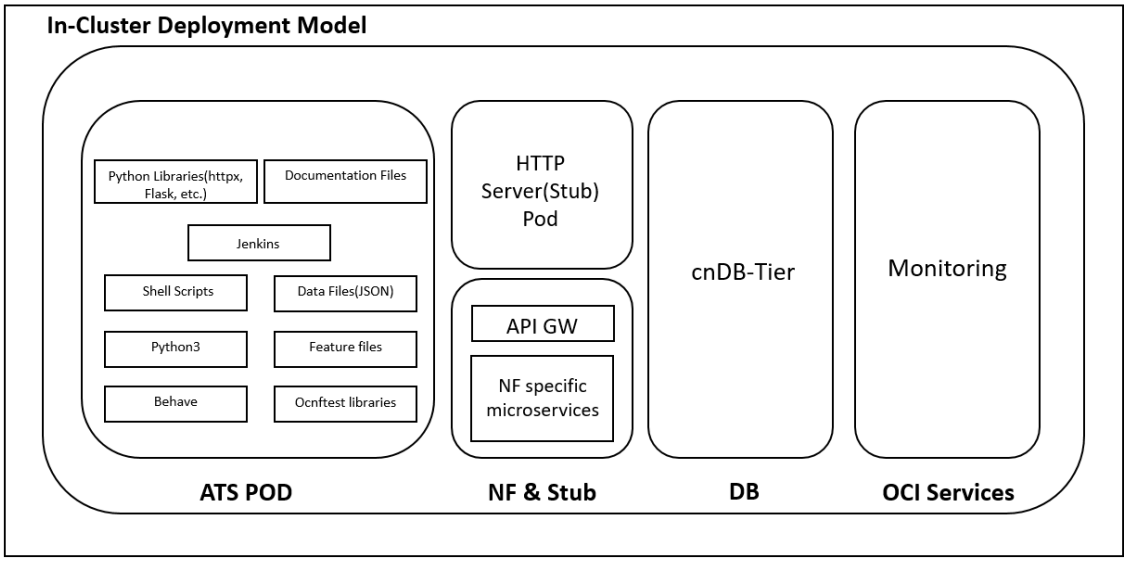
According to the In-Cluster deployment model, ATS can coexist in the same cluster where the NFs are deployed. This deployment model is useful for In-Cluster testing.

Figure 1-1 In-Cluster Deployment Model in OCCNE



Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 1-2 In-Cluster Deployment Model in OCI



1.3 References

For more information about NFs and their deployment processes, refer to the following documents:

- *Oracle Communications Cloud Native Core, Binding Support Function Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Configuration Console Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Network Repository Function Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Provisioning Gateway Guide*
- *Oracle Communications Cloud Native Core, Service Communication Proxy Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Policy Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*
- *Oracle Communications Cloud Native Core, Cloud Native Environment Installation, Upgrade, and Fault Recovery Guide*

Copyright © 2019, 2025, Oracle and/or its affiliates.

2 ATS Framework Features

This chapter describes ATS Framework features.

The following table lists ATS framework features supported by different NFs:

Table 2-1 ATS Framework Features Compliance Matrix

Features	BSF	NRF	NSSF	Policy	SCP	SEPP	UDR
ATS API	Yes	Yes	No	Yes	No	Yes	Partially compliant (In starting jobs, executing all test cases is only supported)
ATS Custom Abort	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ATS Feature Activation and Deactivation	Yes	No	No	Yes	No	Yes	Yes
ATS GUI Enhancements	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ATS Health Check	Yes	No	No	Yes	Yes	Yes	Yes
ATS Jenkins Job Queue	Yes	No	No	Yes	Yes	Yes	Yes
Application Log Collection	Yes	No	No	Yes	Yes	Yes	Yes
ATS Maintenance Scripts	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ATS System Name and Version Display on Jenkins GUI	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ATS Tagging Support	No	No	No	Yes	Yes	Yes	No
Custom Folder Implementation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Health Check	Yes	No	Yes	Yes	No	Yes	Yes
Individual Stage Group Selection	Yes	No	No	Yes	Yes	Yes	No
Lightweight Performance	No	No	No	No	No	No	Yes

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 2-1 (Cont.) ATS Framework Features Compliance Matrix

Features	BSF	NRF	NSSF	Policy	SCP	SEPP	UDR
Managing Final Summary Report, Build Color, and Application Log	Yes	Partially compliant (Application Log is not supported.)	Yes	Yes	Yes	Yes	Yes
Modifying Login Password	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Multiselection Capability for Features and Scenarios	Yes	Partially compliant (Feature level selection is supported)	No	Yes	Partially compliant (Feature level selection is supported)	Partially compliant (Feature level selection is supported)	Partially compliant (Feature level selection is supported)
Parallel Test Execution	Partially compliant (Parallel Test Execution Framework integrated, but only supports sequential execution)	No	Partially compliant (Parallel Test Execution Framework integrated, but only supports sequential execution)	Yes	Yes	Partially compliant (Parallel Test Execution Framework integrated, but ATS test cases need to be organized to utilize the parallel execution)	Yes (UDR, SLF, and EIR)
Parameterization	Yes	Partially compliant (Supports only new features)	No	Yes	Yes	Yes	Yes
PCAP Log Collection	No	No	No	No	Yes	Yes	Yes
Persistent Volume	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Single Click Job Creation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Support for ATS Deployment in OCI	No	No	No	No	Yes	Yes	No
Support for Transport Layer Security	No	No	No	No	Yes	Yes	No
Test Result Analyzer	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Test Case mapping and Count	Yes	No	No	Yes	Yes	Yes	Yes

Copyright © 2019, 2025, Oracle and/or its affiliates.

2.1 ATS API

The Application Programming Interface (API) feature provides APIs to perform routine ATS tasks as follows:

- Start: To initiate one of the three test suites, such as Regression, New Features, or Performance.
- Monitor: To obtain the progress of a test suite's execution.
- Stop: To cancel an active test suite.
- Get Artifacts: To retrieve the JUNIT format XML test result files for a completed test suite.

For more information about configuring the tasks, see Use the RESTful Interfaces.

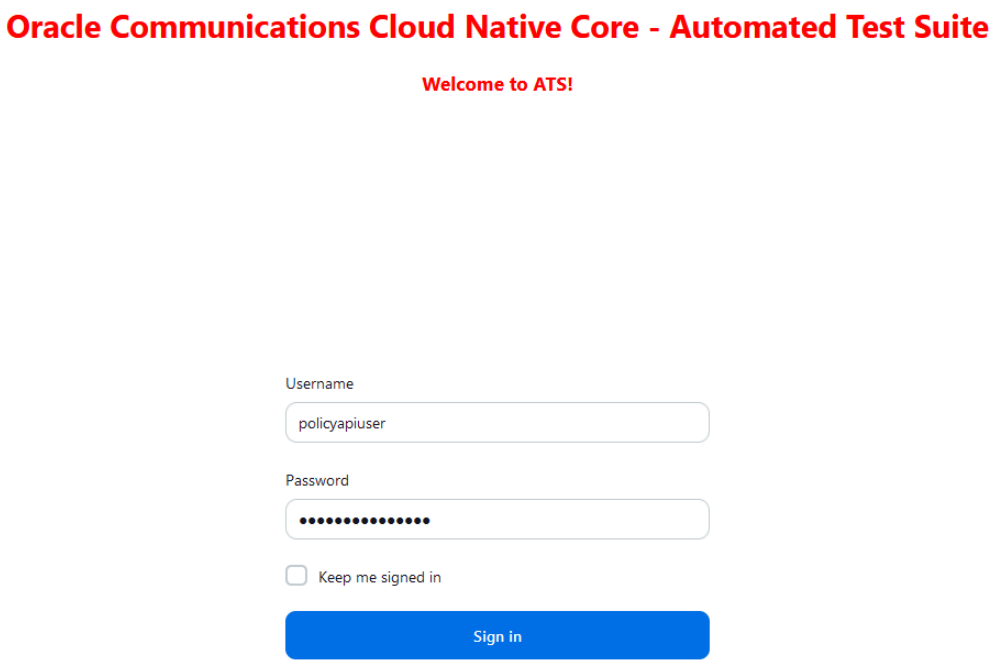
2.1.1 Generating an API Token for a User

An API token that has to be generated for the user to perform routine ATS tasks using the Restful Interfaces API. Any API call requires the use of an API token for authentication. You can generate the API token, and it works until it is revoked or deleted.

Perform the following procedure to generate an API token for a user:

1. Log in to Jenkins as an NF API user to generate an API token.

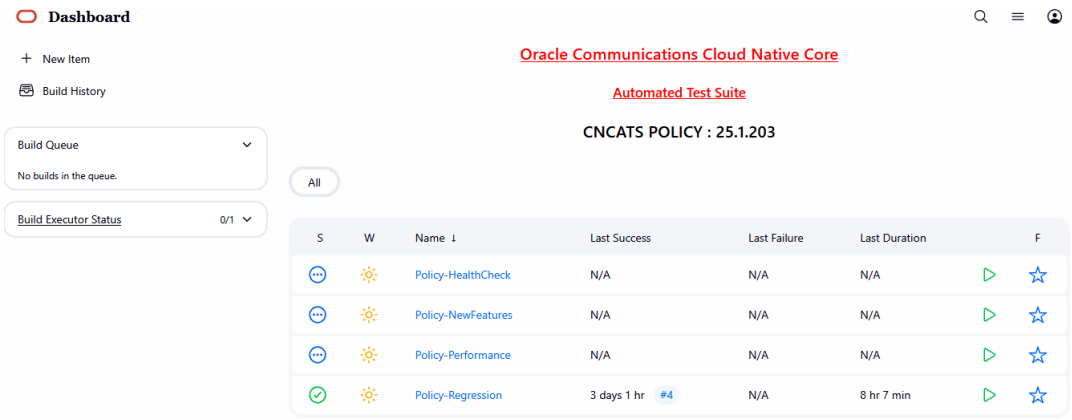
Figure 2-1 ATS Login Page



2. Click the user name in the upper right corner of the GUI, and then click **Security**.

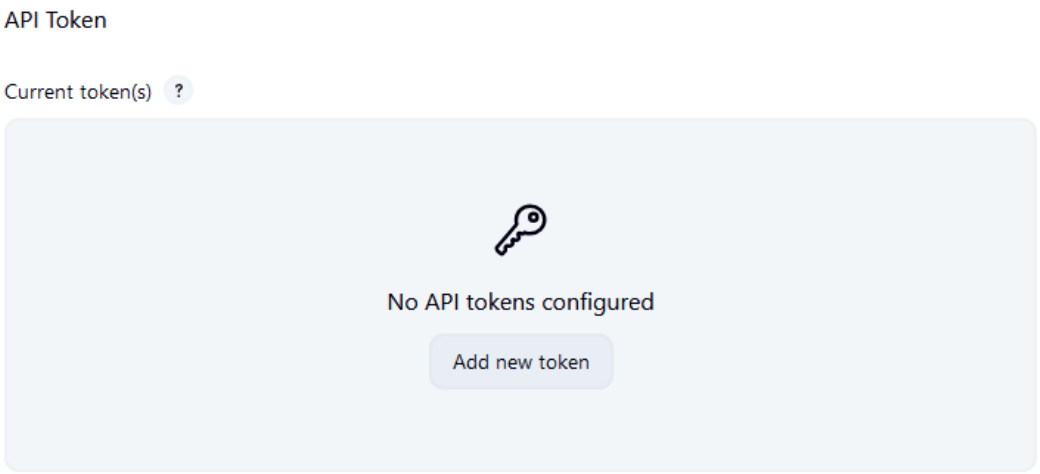
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-2 Add Token



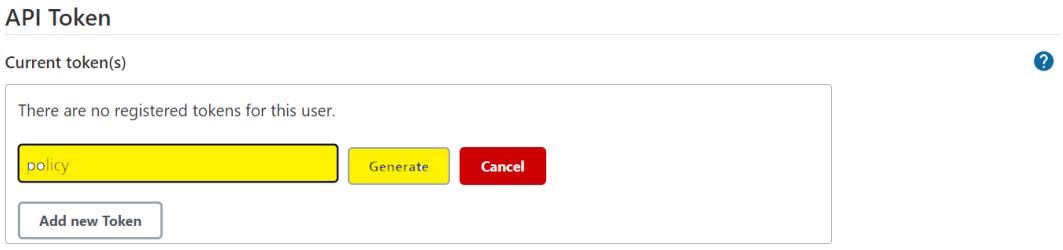
3. In the **API Token** section, click **Add new token**.

Figure 2-3 Add New Token



4. Enter a suitable name for the token, such as policy, and then click **Generate**.

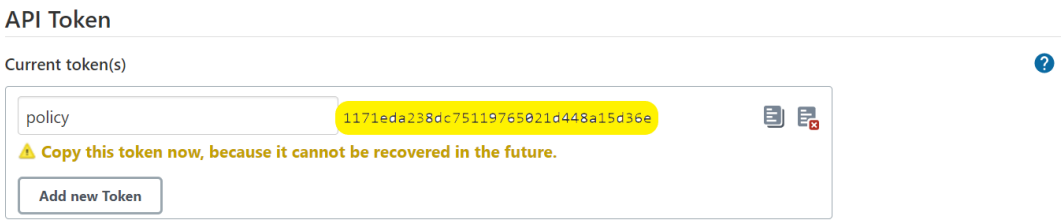
Figure 2-4 Generate Token



5. Copy and save the generated token.
You cannot retrieve the token after closing the prompt.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-5 Save Generated Token



6. Click **Save**.
   An API token is generated and can be used for starting, monitoring, and stopping a job using the REST API.

2.1.2 Use the RESTful Interfaces

This section provides an overview of each RESTful interface.

2.1.2.1 Configuring Host

The host to access the ATS GUI will remain the same in non-OCI setup.

For OCI Setup

The following two ways are supported to access the ATS API in OCI:

Using Loadbalancer IP

1. Add proper Ingress/Egress security rules for ATS API port (5001) and ATS service nodeport corresponding to ATS API port in loadbalancer (nf_lb_subnet) and node subnet (nf_node_subnet). To add ingress and egress security rules, see the Adding Ingress and Egress Rules to Access the OCI Console section.
2. If this step is not performed to access GUI, then insert the following annotations under the **Metadata** section to assign an external IP (Loadbalancer IP):

   ```
   oci-network-load-balancer.oraclecloud.com/security-list-management-mode:
   None
   oci.oraclecloud.com/load-balancer-type: nlb
   ```

3. Edit and save the ATS service after ATS deployment:
   For example,

   ```
   kubectl edit svc ats-service-name -n
         ats-namespace
   ```

4. Access the GUI using URL: `<http/https>://<Loadbalancer IP>:5001`

> **Note**
>
> The assignment of Loadbalancer IP to the ATS service is subject to availability. If the Loadbalancer IP is not assigned to the ATS service even after applying the required annotations, try to debug on the OCI side.

Using Tunneling

Copyright © 2019, 2025, Oracle and/or its affiliates.

1. Add an ingress security rule for the node subnet (nf_node_subnet) to allow TCP traffic on all ports from the operator subnet. To add ingress and egress security rules, see the Adding Ingress and Egress Rules to Access the OCI Console section.
2. Run the following ssh tunneling command from a bash terminal on your local PC:

   ```
   ssh -f -N -i <operator instance private key> -o StrictHostKeyChecking=no -
   o ProxyCommand="ssh -i <bastion private key> -o StrictHostKeyChecking=no -
   W %h:%p <bastion username>@<bastion IP>" <operator instance
   username>@<operator instance IP> -L <desired system port>:<Worker Node
   IP>:<ATS API NodePort> -o ServerAliveInterval=60 -o ServerAliveCountMax=300
   ```

   For example,`ssh -f -N -i id_rsa -o StrictHostKeyChecking=no -o ProxyCommand="ssh -i id_rsa -o StrictHostKeyChecking=no -W %h:%p opc@129.287.66.123" opc@10.1.76.7 -L 5009:10.9.60.118:32018 -o ServerAliveInterval=60 -o ServerAliveCountMax=300`

   Here, ATS GUI URLis http://localhost:5009

TROUBLESHOOTING

If the ATS API returns an error stating "Network is unreachable," ensure that there is a proper ingress security rule in the loadbalancer subnet (nf_lb_subnet) allowing traffic from the system where the ATS API is being utilized.

2.1.2.2 Starting Jobs

To start a job, use the following RESTful interfaces:

- Default Jenkins API: The default Jenkins API to start a pipeline job
- Custom API: To start a job forcibly

Starting a Job using Default Jenkins API
If any other job is already running, the job started with this API goes to Jenkins' job queue.

- Run the following command to start a job (Default Jenkins method):

  ```
  curl --request POST <Jenkins_host_port>/job/<Pipeline_name>/
  buildWithParameters –user
        <username>:<API_token> --verbose
  ```

  When the `atsGuiTLSEnabled` parameter is set to true:

  ```
  curl --request POST <Jenkins_host_port>/job/<Pipeline_name>/
  buildWithParameters --user <username>:<API_token> --verbose --cacert
  <path_to_root_certificate>
  ```

  For example,

  ```
  curl --request POST http://10.123.154.163:30427/job/Policy-NewFeatures/
  buildWithParameters
        --user policyapiuser:111ad02d7471cec9ca689696e9c7a55c62 --verbose
  ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

When the `atsGuiTLSEnabled` parameter is set to true:

```
curl --request POST https://10.75.217.25:30301/job/SCP-Regression/
buildWithParameters --user
    scpapiuser:11c2fde49cea6eb8f332ad23a7877ea2de --verbose --cacert
caroot.cer
```

Starting a Job Forcibly using Custom API

If another job is already running and has not been started by an API user, the running job is aborted, along with all other jobs in the queue that have not been started by an API user, and a new job is started.

If the running job is started by the API user, the new job does not start, and the start job request fails, returning a message in response: `Build <job_id> of pipeline <pipeline_name> is already running, triggered by an API user`.

Builds are aborted gracefully by a forceful API, such as when a running scenario completes its execution and cleanup before the corresponding build is aborted.

The forceful API now returns an `aborted-builds` parameter in response, which contains job IDs for all the aborted builds. It also returns a parameter called `cancelled_builds_in_queue`, which contains queue IDs for all the builds aborted in queue.

If a job ID is assigned to a build in queue, it contains a list of two values: [queueid, jobid] rather than just the queue ID.

Run the following command to start a job forcibly:

```
curl -s --request POST <Startjob_host_port>/build -H "Content-Type:
application/json" -d '{"pipelineName": "<Pipeline_name>", "pageAndQuery":
"<pageAndQuery>"}' --user <username>:<token> --verbose
```

When the `atsGuiTLSEnabled` parameter is set to true:

```
curl -s --request POST
      <Startjob_host_port>/build -H "Content-Type: application/json" -d
'{"pipelineName":  "<Pipeline_name>", "pageAndQuery":
      "<pageAndQuery>"}' --user <username>:<token>   --verbose
      --cacert <path_to_root_certificate>
```

For example,

```
curl --request POST http://10.75.217.25:31170/build  -H "Content-Type:
application/json" -d
    '{"pipelineName": "Policy-Regression",
    "pageAndQuery": "buildWithParameters"}'
    --user policyapiuser:11c1a628f808972c846c510151afa13ba2 --verbose
```

When the `atsGuiTLSEnabled` parameter is set to true:

```
curl --request POST https://10.75.217.25:30170/build  -H "Content-Type:
application/json" -d
    '{"pipelineName": "Policy-Regression",
    "pageAndQuery": "buildWithParameters"}'
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
--user policyapiuser:11c1a628f808972c846c510151afa13ba2 --verbose --cacert
caroot.cer
```

The details of the parameters for the API are as follows:

Table 2-2 API Parameters

Parameters	Mandatory	Default Value	Description
username	YES	NA	This parameter indicates the name of API user.
token	YES	NA	This parameter indicates the API token for API user.
Startjob_host_port	YES	NA	This parameter's format is `<host>:<port>` • `<host>` will be same as Jenkins host • `<port>` will be different(5001 or its nodeport)
pipelineName	YES	NA	This parameter indicates the name of the pipeline for which build is to be triggered.
pageAndQuery	YES	NA	This parameter can have two values: • buildWithParameters: for parametrized pipelines • build: for non-parametrized pipelines
jenkins_wait_time	NO	5	This parameter indicates the wait time for Jenkins in seconds. • If Jenkins is very slow in responding and the API response is not as expected, this wait time can be increased. • It is required when multiple running builds must be aborted before starting a new API build. • This value can be provided along with the parameters "pipelineName" and "pageAndQuery" in a similar way. For example: `{"jenkins_wait_time": "10"}`

Customizing Job Parameters

Both of the Start APIs start the pipeline job with default parameter values. You can provide a different value for a parameter in an API call, such as `paramx=valuex`.

1. Append `paramx=value` to `buildWithParameters?`.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Example 1,

```
curl --request POST 10.75.217.40:31378/job/Policy-NewFeatures/
buildWithParameters?paramx=valuex --user
      policyapiuser:110ed65222b9e63445689314998ff8c3bk -- verbose
```

Example 2,

```
curl --request POST 10.75.217.4:32476/build -H "Content-Type: application/
json" -d '{
      "pipelineName": "Policy-NewFeatures", "pageAndQuery":
"buildWithParameters?paramx=valuex" }'--user <username>:<token>
   --verbose
```

2. To add more than 1 parameter, such as `paramx=valuex` and `paramy=valuey`, append the other parameters to the API call using `&`.
   Example 1,

```
curl --request POST 10.75.217.40:31378/job/Policy-NewFeatures/
buildWithParameters?paramx=valuex&paramy=valuey
--user policyapiuser:110ed65222b9e63445689314998ff8c3bk -- verbose
```

   Example 2,

```
curl --request POST 10.75.217.4:32476/build -H "Content-Type: application/
json" -d '{"pipelineName": "Policy-NewFeatures", "pageAndQuery":
"buildWithParameters?paramx=valuex&paramy=valuey" }' --user
<username>:<token> --verbose
```

3. Replace `buildWithParameters?` with `build` for non-parametrized pipeline jobs.
4. Start the pipeline by using the default Jenkins API or by changing the pageAndQuery parameter's value to build in the following way:

```
curl --request POST <Jenkins_host_port>/job/<Pipeline_name>/build --user
<username>:<API_token> --verbose
```

   Example 1,

```
curl --request POST 10.75.217.40:31378/job/Policy-NewFeatures/build --user
policyapiuser:110ed65222b9e63445689314998ff8c3bk --
      verbose
```

   Example 2,

```
curl --request POST 10.75.217.4:32476/build -H "Content-Type: application/
json" -d '{
      "pipelineName": "Policy-NewFeatures", "pageAndQuery": "build" }'--
user <username>:<token> –verbose
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Currently, the ATS API has enhanced capabilities to include triggering builds for running individual or multiple features, scenarios, stages, groups, and tagged executions.

```
curl --request
 POST <IP>:<PORT>/build -H
 "Content-Type: application/json" -d '{"pipelineName": "<NF PIPELINE>",
"pageAndQuery": "buildWithParameters", "otherBuildParameters": { }}' --user
 <NF>apiuser:<token> --verbose
```

When the value of `atsGuiTLSEnabled` parameter is set to true:

```
curl --request POST <IP>:<PORT>/build \
  -H "Content-Type: application/json" \
  -d '{
    "pipelineName": "<NF PIPELINE>",
    "pageAndQuery": "buildWithParameters",
    "otherBuildParameters": {
    }
  }' \
  --user <NF>apiuser:<Token> \
  --verbose
```

> **Note**
>
> The API continues to support the same functionality as in the previous release. In addition, to provide extended support, a new key "otherBuildParameters" has been introduced. This key can be included in the JSON payload sent to the server.

The following table lists the details of "otherBuildParameters" for the API request:

Copyright © 2019, 2025, Oracle and/or its affiliates.

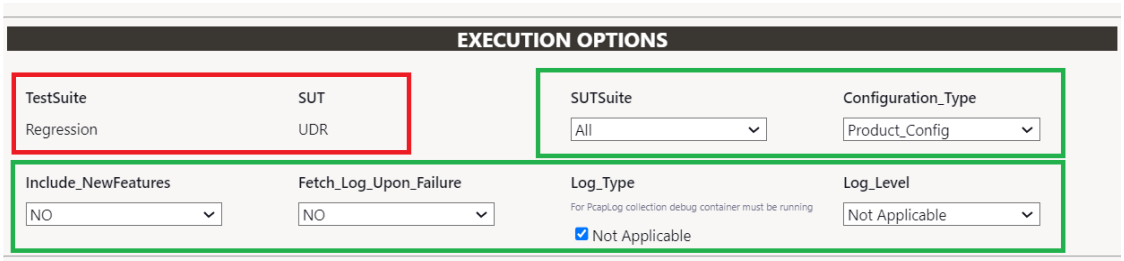
Table 2-3 otherBuildParameters Details

Parameter	Mandatory/Optional	Default Value	Description
otherBuildParameters	Optional	NA	"otherBuildParameters" is a dictionary of build parameters. You can add key-value pairs to customize the build process. For example: `"otherBuildParameters" : { "Features" : "YamlSchema_Import_Export,Custom_Jsons", "Stages" : "stage2,stage3", "Groups" : { "stage1" : "group1,group4", "stage4" : "group4,group5" } }` This dictionary supports the following keys: • Features • Scenarios • FeaturesAndScenarios • Stages • Groups • Feature_Include_Tags • Feature_Exclude_Tags • Scenario_Include_Tags • Scenario_Exclude_Tags

> **Note**
>
> If none of the keys in "otherBuildParameters" are included in the API request, all the test cases with the given execution options will be triggered.

Parameters that are static in the UI should not be included in the "pageAndQuery" parameter of the request. Only parameters with multiple options should be passed. For example, in the UDR pipeline image below, the variables highlighted in red should not be included in the API request, while the parameters highlighted in green can be passed.

Figure 2-6 Execution Option



Copyright © 2019, 2025, Oracle and/or its affiliates.

The following is the format for the API request:

- Example format for executing features, stages, scenarios, or groups through the API request:

```
curl --request POST <IP>:<PORT>/build \
  -H "Content-Type: application/json" \
  -d '{
    "pipelineName": "<NF PIPELINE>",
    "pageAndQuery": "buildWithParameters",
    "otherBuildParameters": {
      "Features": "<Feature List>",
      "Stages": "<stages>",
      "Groups":{ "<stage-n>":"<group-a,group-b>","<stage-m>":"<group-
p,group-q>" },
      "Scenarios": "",
      "Featuresandscenarios": {
        "<Feature 1>": "<Scenario1>",
        "<Feature2>": "<Scenario1>,<Scenario2>"
      }
    }
  }' \
  --user <NF>apiuser:<Token> \
  --verbose
```

- Example format for executing test cases based on provided tags. If tags are specified, other keys such as "Features", "Scenarios", "Stages", and so on, should not be included.

```
curl --request POST <IP>:<PORT>/build \
  -H "Content-Type: application/json" \
  -d '{
    "pipelineName": "<NF PIPELINE>",
    "pageAndQuery": "buildWithParameters",
    "otherBuildParameters": {
      "Feature_Include_Tags": "<tags>",
      "Feature_Exclude_Tags": "<tags>",
      "Scenario_Include_Tags": "<Tags>",
      "Scenario_Exclude_Tags": "<Tags>"
    }
  }' \
  --user <NF>apiuser:<Token> \
  --verbose
```

Starting a job with otherBuildParameters

Execute Features

In the API request, the user can include all parameters such as SUT, Fetch_Log_Upon_Failure, and so on, within the "pageAndQuery" key. Features can be specified in the request using the "otherBuildParameters" dictionary as shown below:

```
"otherBuildParameters" : { "Features" : "<Comma separated FeatureList>" }
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

For example:

```
curl --request POST 10.75.217.25:30100/build \
  -H "Content-Type: application/json" \
  -d '{
    "pipelineName": "Policy-Regression",
    "pageAndQuery": "buildWithParameters",
    "otherBuildParameters": {
      "Features":
"NF_Scoring,ManualGetDeleteSession_DeleteSelectivepcfBindingswithSUPI",
    }
  }' \
  --user policyapiuser:11c1a628f808972c846c510151afa13ba2 \
  --verbose
```

Execute Scenarios

- Using "Scenarios" key
  In the API request, the user can include all parameters such as SUT, Fetch_Log_Upon_Failure, and so on, under the "pageAndQuery" key. Scenarios can be specified in the request using the "otherBuildParameters" dictionary as follows:

  ```
  "otherBuildParameters" : { "Scenarios" : "<Comma separated Scenarios
  List>" }
  ```

  For example:

  ```
  curl --request POST 10.75.217.25:30100/build \
    -H "Content-Type: application/json" \
    -d '{
      "pipelineName": "Policy-Regression",
      "pageAndQuery": "buildWithParameters",
      "otherBuildParameters": {
        "Scenarios":
  "Re_Import_YamlSchema_Verify_SMPolicy,AM_Terminate_Notify_Timeout,AM_Notify
  _With_Header_Timeout"
      }
    }' \
    --user policyapiuser:11c1a628f808972c846c510151afa13ba2 \
    --verbose
  ```

- Using "Featuresandscenarios" key
  The API request allows the user to include all parameters such as SUT, Fetch_Log_Upon_Failure, and so on, using the "pageAndQuery" key. Scenarios can be specified within the "otherBuildParameters" dictionary, as follows:

  ```
  "otherBuildParameters" : { "Featuresandscenarios" : { "<Feature -1>" :
  "<Comma separated scenarios from Feature -1>" , "Feature -2" : "<Comma
  separated scenarios from Feature -2>" } }
  ```

  For example:

  ```
  curl --request POST 10.75.217.25:30100/build \
    -H "Content-Type: application/json" \
  ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    -d '{
      "pipelineName": "Policy-Regression",
      "pageAndQuery": "buildWithParameters",
      "otherBuildParameters": {
        "Featuresandscenarios":{
           "YamlSchema_Import_Export" :
  "Re_Import_YamlSchema_Verify_SMPolicy"
         }
      }
    }' \
    --user policyapiuser:11c1a628f808972c846c510151afa13ba2 \
    --verbose
```

Execute Stages

In the API request, the user can provide all parameters such as SUT, Fetch_Log_Upon_Failure, etc., using the "pageAndQuery" key. Stages can be specified within the "otherBuildParameters" dictionary, as shown below:

```
"otherBuildParameters" : { "Stages" : "<Comma separated Stages list>" }
```

For example:

```
curl --request POST 10.75.217.25:30100/build \
  -H "Content-Type: application/json" \
  -d '{
    "pipelineName": "Policy-Regression",
    "pageAndQuery": "buildWithParameters?Configuration_Type=Custom_Config",
    "otherBuildParameters": {
      "Stages" : "stage2,stage4"
    }
  }' \
  --user policyapiuser:11c1a628f808972c846c510151afa13ba2 \
  --verbose
```

Execute Groups

In the API request, the user can include all parameters such as SUT, Fetch_Log_Upon_Failure, and so on, under the "pageAndQuery" key. Groups can be specified within the "otherBuildParameters" dictionary, as follows:

```
"otherBuildParameters" : { "Groups" : {  "<stage n>" : "<Comma
    separated Groups List from stage n>", "<stage m>" : "<Comma  separated
Group list from
    stage m>"  } }
```

For example:

```
curl --request POST 10.75.217.25:30100/build \
  -H "Content-Type: application/json" \
  -d '{
    "pipelineName": "Policy-Regression",
    "pageAndQuery": "buildWithParameters",
    "otherBuildParameters": {
      "Groups":{"stage1":"group1" , "stage2" : "group3,group6"}
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    }
  }' \
  --user policyapiuser:11c1a628f808972c846c510151afa13ba2 \
  --verbosee
```

Execute with Tags

In the API request, the user can specify all parameters such as SUT, Fetch_Log_Upon_Failure, and so on, using the "pageAndQuery" key. When using tags, it is mandatory to include "FilterWithTags" in the "pageAndQuery". If tags are provided by the user, any other parameters such as features, scenarios, stages, or groups will not be considered. Only tags will be considered as input.

Tags can be specified in the request within the "otherBuildParameters" dictionary, as shown below:

```
"otherBuildParameters": {  "Feature_Include_Tags":"<tags>",
"Feature_Exclude_Tags":"<tags>", "Scenario_Include_Tags":"<Tags>" ,
"Scenario_Exclude_Tags":"<Tags>" }
```

> **Note**
>
> The provided tags must be separated by commas.

for example:

```
curl --request POST 10.75.217.25:30100/build \
  -H "Content-Type: application/json" \
  -d '{
    "pipelineName": "Policy-Regression",
    "pageAndQuery": "buildWithParameters",
    "otherBuildParameters": {
      "Feature_Include_Tags":"cne-common,cm-service",
"Scenario_Include_Tags":"cleanup,sanity"
    }
  }' \
  --user policyapiuser:11c1a628f808972c846c510151afa13ba2 \
  --verbose
```

> **Note**
>
> The API behaves in a manner similar to that of the UI. For instance, when running a set of scenarios, the user selects them, and only those chosen are executed upon triggering the build. If any stages, groups, or features are also selected in the features section, they are ignored. Similarly, if stages, groups, or features are included alongside scenarios in the API request, only the scenarios will be executed.

2.1.2.3 Monitoring Jobs

This Default Jenkins API is used to monitor the progress of the job that was started.

For monitoring, the following APIs are used:

Copyright © 2019, 2025, Oracle and/or its affiliates.

- A `qid` is obtained from the Location header in the response for `starting a job`. The first API uses this `qid` to get queue status about the corresponding job, including its `job_id`.
- The second API uses the `job_id` to obtain further information about the job status.

Monitoring a Job
To monitor jobs, run the following commands in a sequence:

1. `curl --request POST <Jenkins_host_port>/queue/item/<qid>/api/json --user <username>:<API_token> --verbose`
   When the `atsGuiTLSEnabled` parameter is set to true:

   ```
   curl --request POST
         <Jenkins_host_port>/queue/item/<qid>/api/json --user
   <username>:<API_token> --verbose --cacert
         <path_to_root_certificate>
   ```

   For example,

   ```
   curl --request POST http://10.123.154.163:30427/queue/item/5/api/json--
   user policyapiuser:111ad02d7471cec9ca689696e9c7a55c62
            --verbose
   ```

   When the `atsGuiTLSEnabled` parameter is set to true:

   ```
   curl --request POST https://10.75.217.25:30301/queue/item/27/api/json --
   user
         scpapiuser:11c2fde49cea6eb8f332ad23a7877ea2de --verbose --cacert
       caroot.cer
   ```

2. `curl --request POST <Jenkins_host_port>/job/<Pipeline_name>/<job_id>/api/json --user <username>:<API_token> --verbose`
   When the `atsGuiTLSEnabled` parameter is set to true:

   ```
   curl --request POST
         <Jenkins_host_port>/job/<Pipeline_name>/<job_id>/api/json  --user
   <username>:<API_token> --verbose
         --cacert<path_to_root_certificate>
   ```

   For example,

   ```
   curl --request POST http://10.123.154.163:30427/job/Policy-
   NewFeatures/3/api/json--user
   policyapiuser:111ad02d7471cec9ca689696e9c7a55c62
            --verbose
   ```

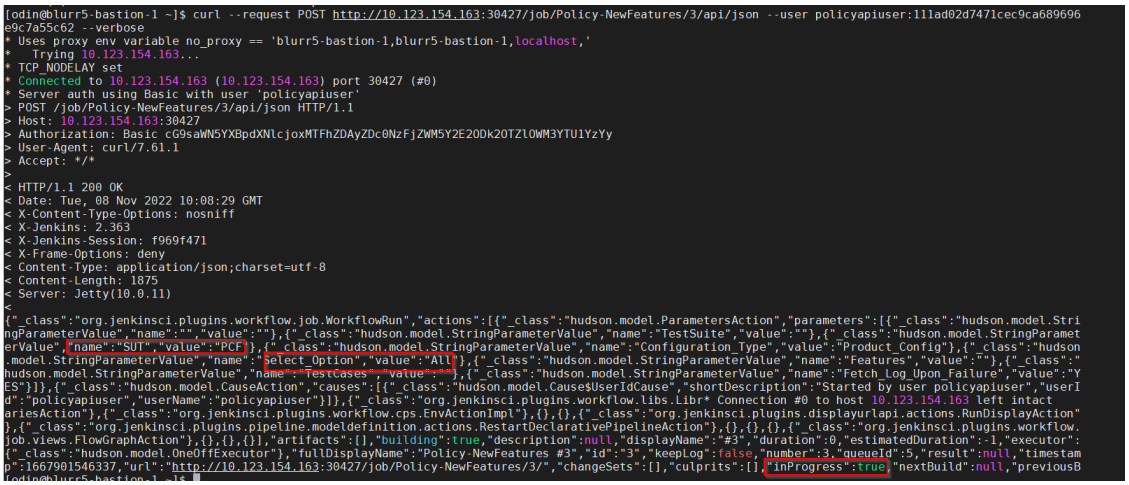
   When the `atsGuiTLSEnabled` parameter is set to true:

   ```
   curl --request POST https://10.75.217.25:30301/job/SCP-Regression/2/api/
   json --user
         scpapiuser:11c2fde49cea6eb8f332ad23a7877ea2de --verbose --cacert
       caroot.cer
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

The following screenshot shows an example of monitoring the progress of the job:

Figure 2-7 Monitoring a Job



2.1.2.4 Stopping Jobs

Stop API is used to stop the currently running job using its job_id. It is also a default Jenkins API.

Stopping a Job

For ATS without Parallel Test Execution framework integrated:

```
curl --request POST
      <Jenkins_host_port>/job/<Pipeline_name>/<job_id>/stop  --user
<username>:<API_token>
    --verbose
```

When the `atsGuiTLSEnabled` parameter is set to true:

```
curl --request POST
      <Jenkins_host_port>/job/<Pipeline_name>/<job_id>/stop  --user
<username>:<API_token> --verbose –cacert
      <path_to_root_certificate>
```

For example,

```
curl --request POST http://10.75.217.4:31881/job/UDR-Regression/21/stop --user
    udrapiuser:1139a72213e0a686972cbff4a2f9333a9f --verbose
```

When the `atsGuiTLSEnabled` parameter is set to true:

```
curl --request POST https://10.75.217.25:30301/job/SCP-Regression/2/stop --
user
    scpapiuser:11c2fde49cea6eb8f332ad23a7877ea2de --verbose --cacert
caroot.cer
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> - If the rerun count is greater than zero, the job must be stopped twice.
> - This Stop API call does not abort the build gracefully.

For ATS with Parallel Test Execution framework integrated:

```
curl --request POST
      <Stopjob_host_port>/job/<Pipeline_name>/<job_id>/stop  --user
<username>:<API_token>
    --verbose
```

When the `atsGuiTLSEnabled` parameter is set to true:

```
curl --request POST
      <Stopjob_host_port>/job/<Pipeline_name>/<job_id>/stop  --user
<username>:<API_token> --verbose --cacert
      <path_to_root_certificate>
```

For example.

```
curl --request POST http://10.75.217.4:32476/job/UDR-Regression/21/stop --user
    udrapiuser:1139a72213e0a686972cbff4a2f9333a9f --verbose
```

When the `atsGuiTLSEnabled` parameter is set to true:

```
curl --request POST https://10.75.217.25:30170/job/SCP-Regression/2/stop --
user
    scpapiuser:11c2fde49cea6eb8f332ad23a7877ea2de --verbose --cacert
caroot.cer
```

The following table lists the parameter details for Stop API:

Table 2-4 Stop API Details

Parameter	Mandatory	Default Value	Description
userName	Yes	NA	Name of API user
token	Yes	NA	The API token for the API user
Stopjob_host_port	Yes	NA	Format is <host>:<port> • <host> is same as Jenkins host • <port> is different such as 5001 or its nodeport
pipelineName	Yes	NA	Name of the pipeline for which build is to be stopped

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 2-4 (Cont.) Stop API Details

Parameter	Mandatory	Default Value	Description

immediate	No	False	To stop the build immediately, send a query parameter ("immediate=true") with API call.
			For example,
			
			`curl --request POST`
			
			`<Stopjob_host_port>/job/<Pipeline_name>/<job_id>/stop?immediate=true --user`
			
			`<username>:<API_token> --verbose`
			
			• `immediate` can also have values such as yes or 1. These values work similar to the true value.

2.1.2.5 Getting Test Suite Artifacts

Default Jenkins API is used to get the JUNIT-formatted XML test result files for a completed test suite.

For getting artifacts for any completed test suite, the following APIs are used:

- For getting an overall build summary
- For getting a JUNIT XML test result file for every feature file that ran

For getting an overall build summary:

```
curl --request POST <Jenkins_host_port>/job/<Pipeline_name>/<job_id>/
testReport/api/xml?exclude=testResult/suite --user <username>:<API_token> --
verbose
```

When the `atsGuiTLSEnabled` parameter is set to true:

```
curl --request POST
      <Jenkins_host_port>/job/<Pipeline_name>/<job_id>/testReport/api/xml?
exclude=testResult/suite  --user
      <username>:<API_token> --verbose --cacert  <path_to_root_certificate>
```

For example,

```
curl --request POST http://10.123.154.163:30427/job/Policy-NewFeatures/4/
testReport/api/xml?exclude=testResult/suite--user
policyapiuser:111ad02d7471cec9ca689696e9c7a55c62 --verbose
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

When the atsGuiTLSEnabled parameter is set to true:

```
curl --request POST https://10.75.217.25:30301/job/SCP-Regression/1/
testReport/api/xml?exclude=testResult/suite
      --user scpapiuser:11c2fde49cea6eb8f332ad23a7877ea2de --verbose --cacert
    caroot.cer
```

For getting Feature-wise XML, Select_Option = All:

```
curl --request POST
      <Jenkins_host_port>/job/<Pipeline_name>/<job_id>/artifact/test-results/
reports/*zip*/test-results.zip
      --user <username>:<API_token> --verbose --output  test-results.zip
```

When the atsGuiTLSEnabled parameter is set to true:

```
curl --request POST
      <Jenkins_host_port>/job/<Pipeline_name>/<job_id>/artifact/test-results/
reports/*zip*/test-results.zip
      --user <username>:<API_token> --verbose --output  test-results.zip --
cacert
      <path_to_root_certificate>
```

For example,

```
curl --request POST http://
        10.75.217.4:31881/job/Policy-NewFeatures/21/artifact/test-results/
reports/*zip*/test-results.zip  --user
policyapiuser:11c3344996c4fda01ded2124bec4f9aa17
      --verbose –-output test-results.zip
```

When the atsGuiTLSEnabled parameter is set to true:

```
curl --request POST https://10.75.217.25:30301/job/SCP-Regression/1/artifact/
test-results/reports/*zip*/test-results.zip
    --user scpapiuser:11c2fde49cea6eb8f332ad23a7877ea2de --verbose --cacert
    caroot.cer
```

For getting Feature-wise XML, Select_Option = Single/MultipleFeatures:

```
curl --request POST
      <Jenkins_host_port>/job/<Pipeline_name>/<job_id>/artifact/test-results/
reports/
      *.<Feature1_name>.xml,*.<Feature2_name>.xml/*zip*/test-results.zip  --
user <username>:<API_token>
      --verbose --output  test-results.zip
```

When the atsGuiTLSEnabled parameter is set to true:

```
curl --request POST
      <Jenkins_host_port>/job/<Pipeline_name>/<job_id>/artifact/test-results/
reports/
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
      *.<Feature1_name>.xml,*.<Feature2_name>.xml/*zip*/test-results.zip  --
user <username>:<API_token>
      --verbose --output  test-results.zip –-cacert
    <path_to_root_certificate>
```

For example,

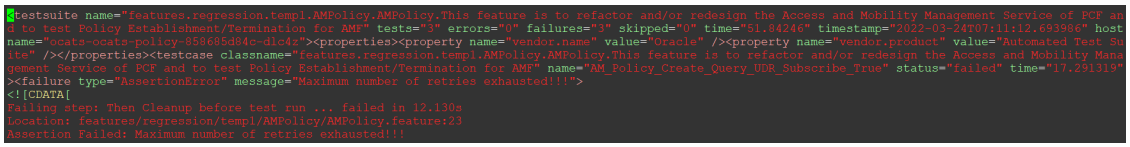
```
curl --request POST http://
      10.75.217.4:31881/job/Policy-NewFeatures/21/artifact/test-results/
reports/*.goldenfeature.xml,*.AMPolicy.xml/*zip*/test-results.zip
      –user policyapiuser:11c3344996c4fda01ded2124bec4f9aa17 --verbose  --
output test-results.zip
```

When the `atsGuiTLSEnabled` parameter is set to true:

```
curl --request POST https://10.75.217.25:30301/job/SCP-Regression/1/artifact/
test-results/reports/*.SCP_Registration_With_PLMNList.xml/*zip*/test-
results.zip
    --user scpapiuser:11c2fde49cea6eb8f332ad23a7877ea2de --verbose --cacert
    caroot.cer
```

API calls for `Select_Option = All and Select_Option = Single/MultipleFeatures` return a zip file with JUNIT XMLs, one XML for each feature.

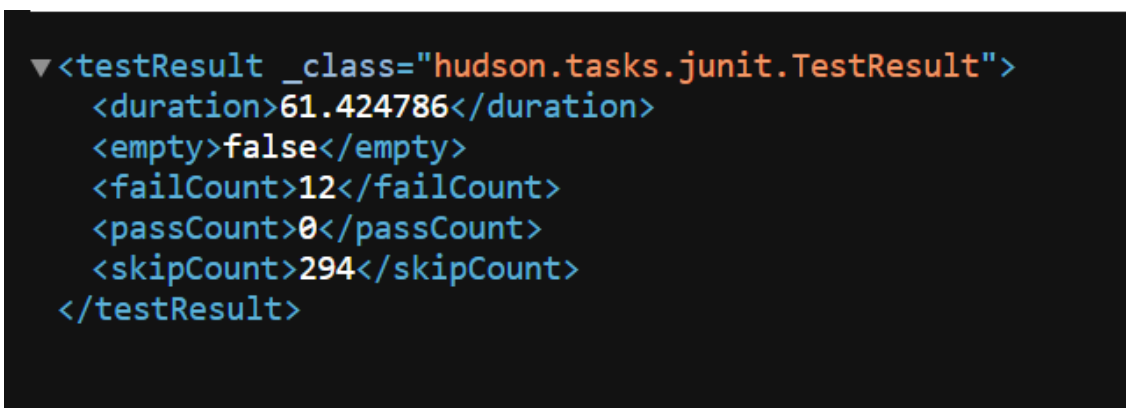
Figure 2-8 Sample XML Output for AMPolicy.feature



In the API call, specify other selected features in comma-separated form as `/*<Feature1_name>.xml,*<Feature2_name>.xml,*<Feature3_name>.xml,*<Feature4_name>.xml/` for Select_Option = Single/MultipleFeatures.

The API call for getting the overall build summary returns an XML with values for `duration`, `failCount`, `passCount`, and `skipCount` for the current build.

Figure 2-9 Sample Output



Copyright © 2019, 2025, Oracle and/or its affiliates.

It is recommended to maintain a gap of at least a few seconds between two API calls. This gap depends on the time Jenkins takes to complete the API request.

2.2 ATS Custom Abort

ATS Custom Abort feature allows you to gracefully abort the ongoing build directly from the Graphical User Interface (GUI).

You can abort the builds in ATS using the following ways:

- **Using the Abort Button on the GUI**: This method, supported by Jenkins, allows you to abort builds directly from the user interface.
- **Using the ATS API**: This is a partially manual method for aborting builds.
  When the ATS API is used to abort a build, ATS will wait for any running scenarios to complete their cleanup process before finalizing the abort. This ensures that there are no issues related to cleanup when using the ATS API.

Manual Abort

By default, Jenkins provides a **cancel** icon for every pipeline on the dashboard whenever a job is running. However, using the manual abort or **cancel** icon has some limitations, especially in parallel execution scenarios.

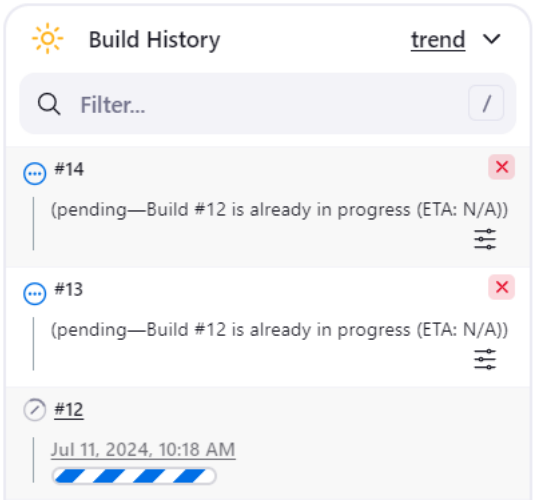
When the manual abort or **cancel** icon is used, Jenkins sends a kill signal to stop all current executions, regardless of whether test cases or other operations are in progress. In cases of parallel execution with multiple stages, the manual abort or **cancel** icon may need to be clicked multiple times due to the presence of multiple stages. This approach can lead to pending cleanup for test cases, which may cause failures in subsequent executions.

To address the issues with manual abort, the release 24.3.0 introduces an **Abort_Build** menu in all new features and regression pipelines, ensuring a more controlled and graceful termination of builds.

The **cancel** icon will not be displayed for currently running builds. However, if there are any builds in the queue, they will still have the **cancel** icon available for use.

For example, builds 13 and 14 can be stopped using the **cancel** icon next to them, while build 12 can only be stopped using the new **Abort_Build** in the left navigation pane.

Figure 2-10 Stopping Builds



Copyright © 2019, 2025, Oracle and/or its affiliates.

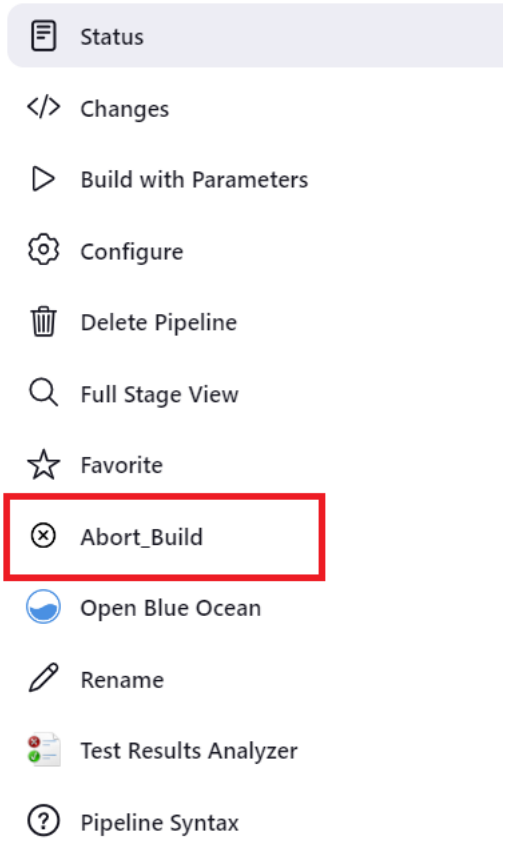
Abort_Build

The **Abort_Build** is available on every regression and new features pipeline. It supports the graceful termination of builds. This **Abort** button available in the GUI triggers a stop API request to the ATS API server. This request ensures that the execution is stopped gracefully, allowing all currently running scenarios to complete.

Using Abort Button

1. Enter the login credentials and click **Sign in**.
   The screens displays preconfigured pipelines for NF individually.
2. Click **NF-NewFeatures** or **NF-Regression** in the **Name** column.
   The NF-NewFeatures or NF-Regression screen appears.
3. Click **Abort_Build** in the left navigation pane.

Figure 2-11 Abort Build

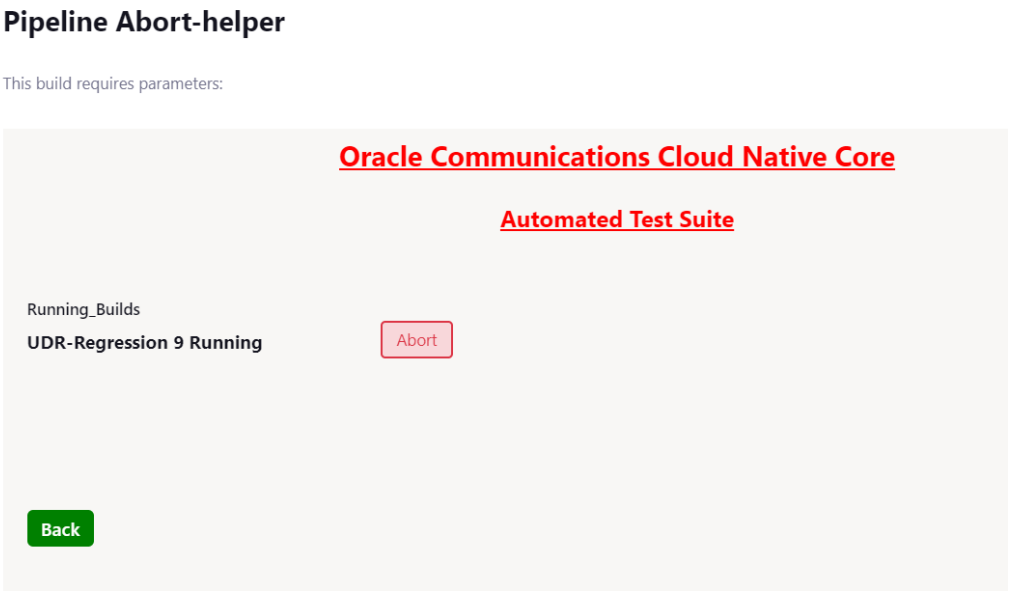


You will be redirected to the Pipeline Abort-helper page.

4. If no builds are currently running or in progress, the page will display the message: "No builds are running."
5. If builds are running, the page will display the following:
   - **Running_Builds**: This section lists the running pipeline names and build numbers.

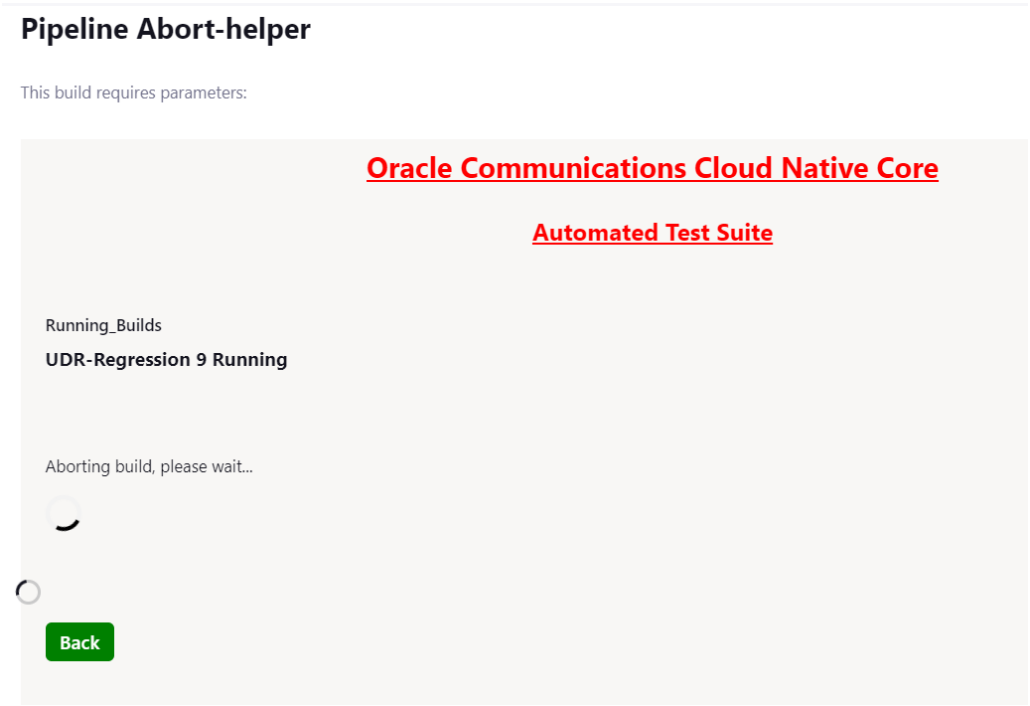
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-12 Running_Builds



- **Abort**: Click the button to start the abort process.

Figure 2-13 Aborting Build



- **Back**: Click the button to return to the pipeline page or remain on the current page while the abort process completes.

Time Taken by Custom Abort to Stop a Build

The time required to stop a build gracefully depends on various factors, including the current stage of the build:

Copyright © 2019, 2025, Oracle and/or its affiliates.

- If the abort is initiated during stages such as "Preparation" or "POST," the build stops within a few minutes.
- If parallel test case execution is not enabled for NFs, and there is only one stage, such as "Execute-tests" or "stage1/group1", the time taken for a graceful abort depends on the duration of the currently running scenarios.
- If parallel test case execution is enabled for NFs, the time required for abort depends on the scenario that takes the longest time to complete among the currently running groups.

On the Console Output page of any running build, for example, http://<IP>:<Node port>/job/<Job Name>/<Build Number>/console, Jenkins will displays a manual abort or **cancel** icon.

Figure 2-14 Manual Abort or Cancel icon



> **Note**
>
> It is recommended to use the new **Abort** button instead of the manual abort or **cancel** icon provided by Jenkins for a more reliable abort process.

2.3 ATS Feature Activation and Deactivation

ATS Feature Activation and Deactivation feature allows users to activate or deactivate specific features within the ATS using Helm charts.

> **Note**
>
> Once these features are removed, they cannot be reinstated in the deployed ATS. However, users have the option to reinstall the ATS to restore the disabled features.

The following table lists the features to enable or disable using Helm charts.

> **Note**
>
> These parameters can be edited in the ATS deployment file (`values.yaml`).

Table 2-5 Enable or Disable ATS Feature

Features	Parameter	Description
Support for Test Case Mapping and Count	testCaseMapping	Set this parameter to `true` to activate the feature in the ATS GUI.
Application Log Collection and PCAP Log Collection	logging	Set this parameter to `true` to collect the ATS logs. If the parameter is set to false, the logs will not be collected.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 2-5 (Cont.) Enable or Disable ATS Feature

Features	Parameter	Description
Lightweight Performance	lightWeightPerformance	Set this parameter to `true` to activate the feature in the ATS GUI. If the parameter is set to false, the performance pipeline will not be accessible.
ATS Health Check	healthcheck	Set this parameter to `true` to activate the feature in the ATS GUI. If the parameter is set to false, the health check pipeline will not be accessible.
ATS API	atsApi	Set this parameter to `true` to activate the feature. If the parameter is set to false, the ATS API feature on the 5001 port will be disabled.
Parameterization	parameterization	Set this parameter to `true` to activate the feature in the ATS GUI. If the parameter is set to false, the Configuration_Type parameter on the GUI will not be available.
Parallel Test Execution	parallelFrameworkChangesIntegrated	Set this parameter to `true` if all parallel test case execution features are picked by NF and changes are made to files. **Note**: Do not change the values provided in the `values.yaml` file for this parameter.
Parallel Test Execution	parallelTestCaseExecution	Set this parameter to `true` to activate the feature in the ATS GUI. If set to false, all the features will be copied into a single stage or group, resulting in sequential execution. **Note**: It is not advisable to edit the default value given in the values.yaml file for this parameter.
Parallel Test Execution	mergedExecution	Set this parameter to `true` to activate the feature in the ATS GUI. If set to false, the option to include other pipelines for mergedExecution will not be available. **Note**: It is not advisable to edit the default value given in the values.yaml file for this parameter.
ATS Support to Execute Scenarios	scenarioSelection	Set this parameter to `true` to activate the feature in the ATS GUI. If set to false, the ability to select single or multiple scenarios will be removed. It is dependent on the testCaseMapping parameter. If this parameter is set to false, the scenarioSelection parameter will be set to false too. **Note**: It is not advisable to edit the default value given in the values.yaml file for this parameter.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 2-5 (Cont.) Enable or Disable ATS Feature

Features	Parameter	Description
ATS Tagging Support	executionWithTagging	Set this parameter to `true` to activate the feature in the ATS GUI. If set to false, the ability to execute test cases based on tags will be removed. It is dependent on the testCaseMapping parameter. If this parameter is set to false, the executionWithTagging parameter will be set to false too.
Stage or Group Level Execution	individualStageGroupSelection	Set this parameter to `true` to activate the feature in the ATS GUI. If set to false, the ability to select all test cases from individual stages or groups using a single checkbox will be removed. **Note**: It is not advisable to edit the default value given in the values.yaml file for this parameter.
Support for Transport Layer Security	atsGUITLSEnabled	Set this parameter to `true` to activate the feature in the ATS GUI. If set to false, ATS will work in HTTP mode.

Note

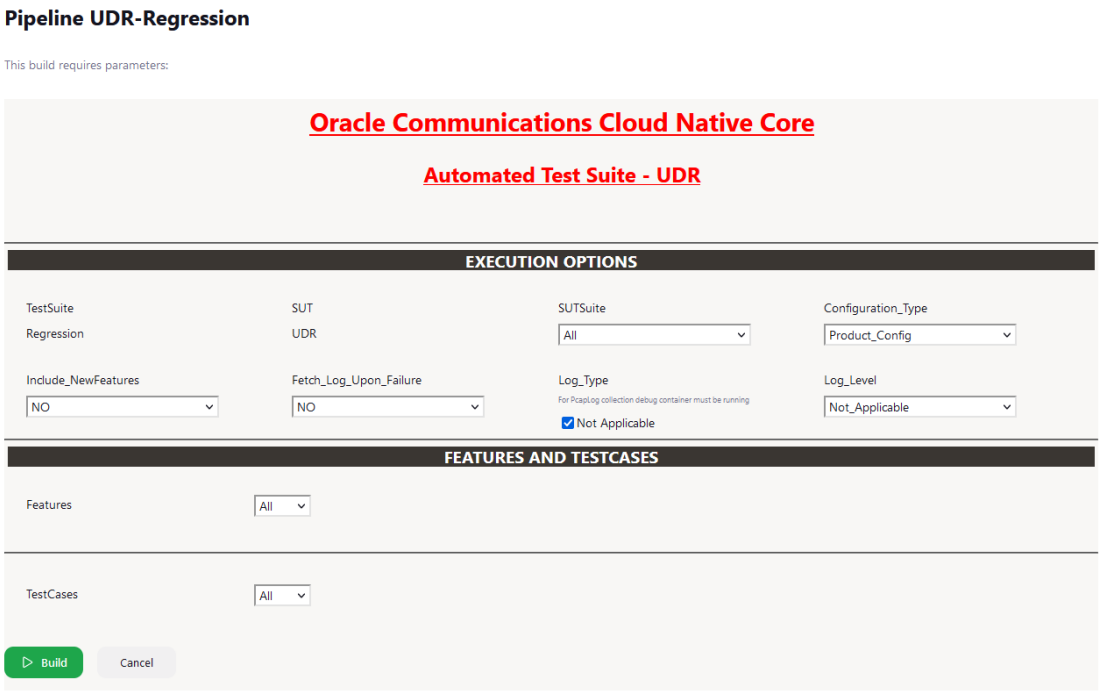
- You can edit the parameters relating to the features that NF supports. Keep the default value for the remaining parameters.
- For the current release, mergedExecution, individualStageGroupSelection, and parallelTestCaseExecution parameters value should not be modified.

2.4 ATS GUI Enhancements

With this enhancement, the ATS Graphical User Interface (GUI) layout is redesigned to streamline user interaction by segregating execution options and features or test cases into distinct compartments, allowing the users to navigate effortlessly between different functionalities.

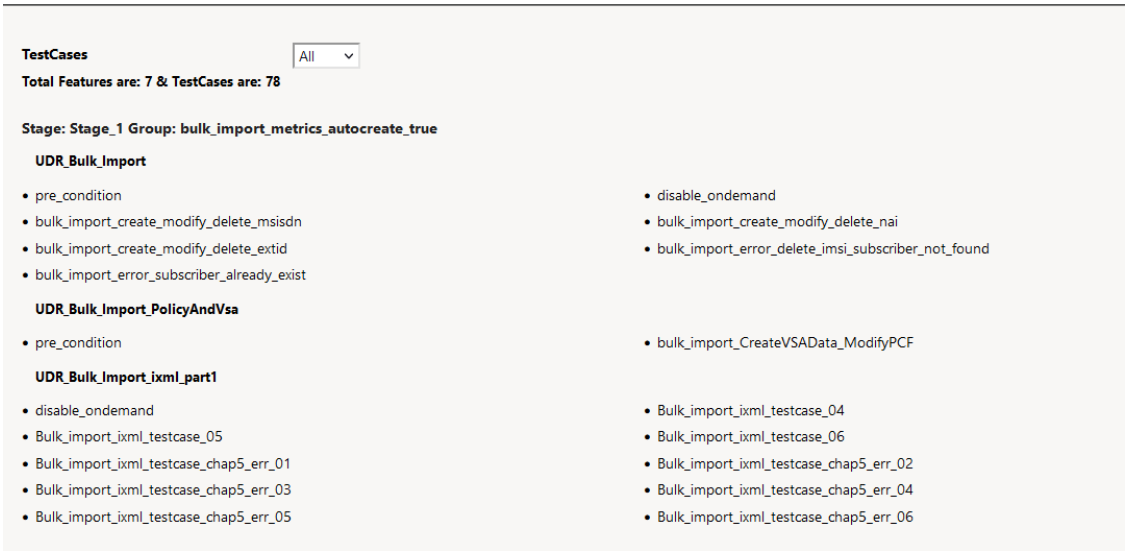
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-15 Layout Enhancement



Test cases now feature a display of fixed length, ensuring that even lengthier test cases remain visually manageable. Additionally, the inclusion of hover-scroll functionality allows for easy access to content beyond the visible area, enhancing readability and user friendliness.

Figure 2-16 Test Cases Visibility



Tooltips have been introduced to allow user to have quick insights into the grouping and staging of individual test cases, thereby improving overall understanding and navigation within the ATS environment.

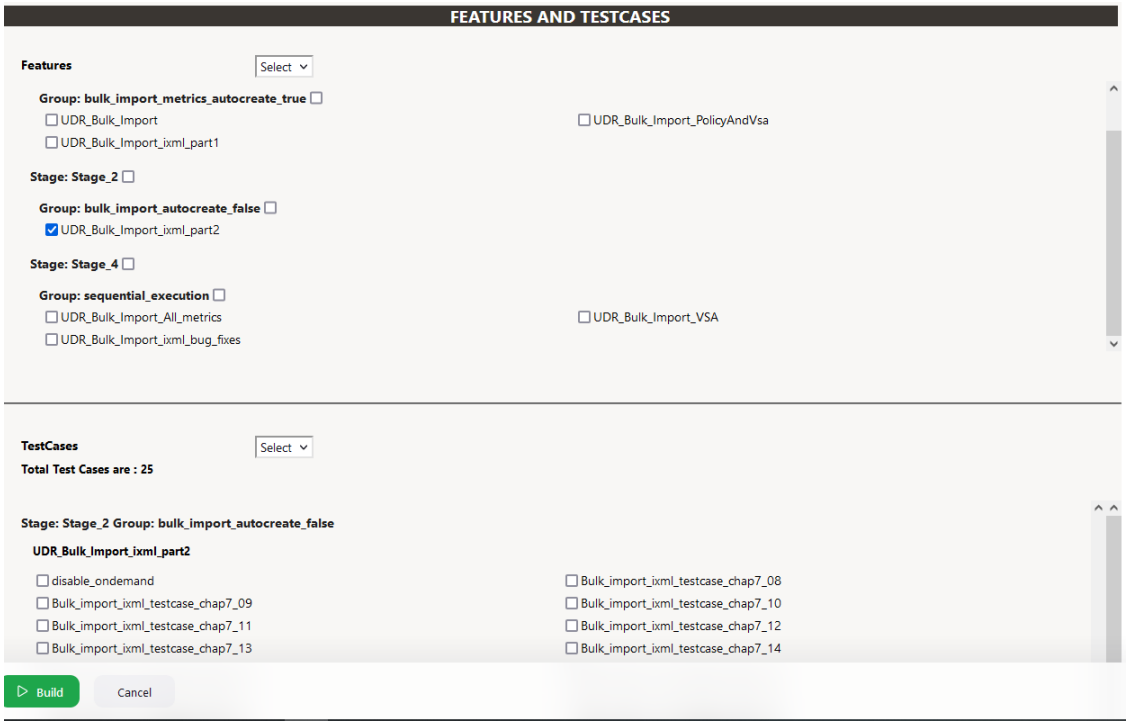
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-17 Tooltips



To facilitate the use of multiselect features and scenarios, distinct drop down menus have been introduced. This enables users to efficiently choose and manage multiple options simultaneously, thereby improving flexibility and productivity in selecting test cases.

Figure 2-18 Multiselect Features and Scenarios



2.5 ATS Health Check

The ATS Health Check feature allows you to evaluate the health of the ATS deployment by conducting a comprehensive series of checks. ATS health checks are performed using the

Copyright © 2019, 2025, Oracle and/or its affiliates.

Health Check tool. After installation, it ensures the health of CNCATS pods, their services, and associated configurations.

Overview of Health Check Tool Functionality

The following provides a summary of its main functions:

1. Initial Setup: After installation, the Health Check tool begins by running a series of predefined tests to establish a baseline.
2. CPU and Memory Verification: The Health check tool verifies whether the CPU and memory allocated to CNCATS pods meet the minimum requirements set. It compares current resource allocations with these thresholds and flags any shortfalls, recommending necessary adjustments.
3. Service Verification: It checks the operational status of the CNCATS services (ATS API and ATS GUI), including verifying service endpoints, ensuring services are running, and confirming they respond as expected.
4. Test Folder Validation: It inspects the test folder to ensure that all necessary test files are present and properly configured.
5. Configuration Checks: The tool reviews the authentication configurations required for running the System Under Test (SUT/NF) health check, verifying that all configurations are correct to facilitate smooth run.
6. PVC Verification: It confirms whether the Persistent Volume Claim (PVC) is in a bound state and properly connected to a Persistent Volume. Any issues with PVC binding are flagged for further investigation.

By performing these checks, the tool ensures that the CNCATS pod and its associated services are functioning correctly, identifying potential issues before they affect system performance.

> **Note**
>
> - This feature is available starting from CNCATS 24.3.0 Build.
> - For initial checks, view the ATS Health Check in pod logs using the command:
>
>   ```
>   kubectl logs <podname> -n <namespace>
>   ```
>
>   .
> - For subsequent checks, rerun the tool through ATS bash with the command:
>
>   ```
>   healthtest
>   ```
>
>   .

The following screenshots provide visual examples of different scenarios:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-19 Success Health Check

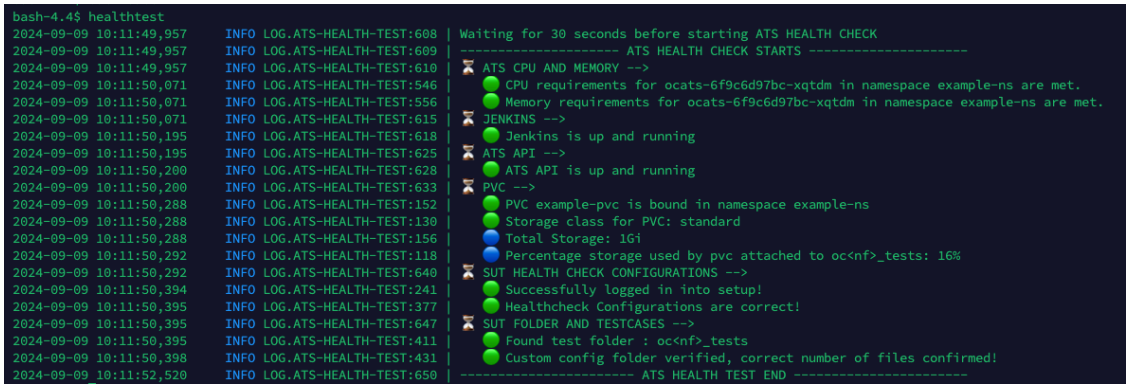
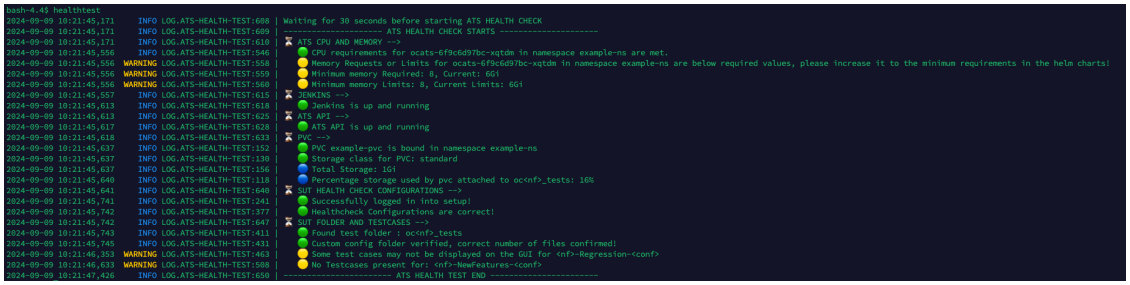


Figure 2-20 Warning and Errors



> **Note**
>
> The highlighted areas illustrate a health check with warnings and errors that require further investigation or action.

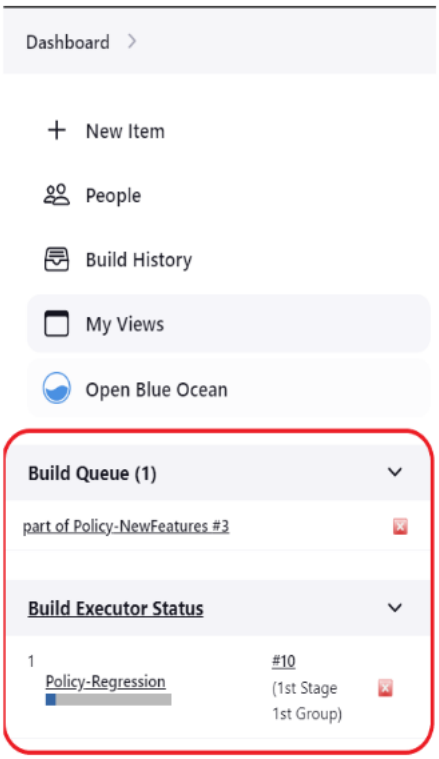
2.6 ATS Jenkins Job Queue

The ATS Jenkins Job Queue feature places the second job in a queue if the current job is already running from the same or different pipelines to prevent jobs from running in parallel to one another.

Job/build queue status can be viewed in the left navigation pane on the ATS home page. The following image shows the build queue status when a user has tried to run the NewFeatures pipeline when the Regression pipeline is already running.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-21 Build Executor Status



2.7 Application Log Collection

Using Application Log Collection, you can debug a failed test case by collecting the application logs for NF System Under Test (SUT). Application logs are collected for the duration that the failed test case was run.

Application Log Collection can be implemented by using OpenSearch or Kubernetes Logs. In both these implementations, logs are collected per scenario for the failed scenarios.

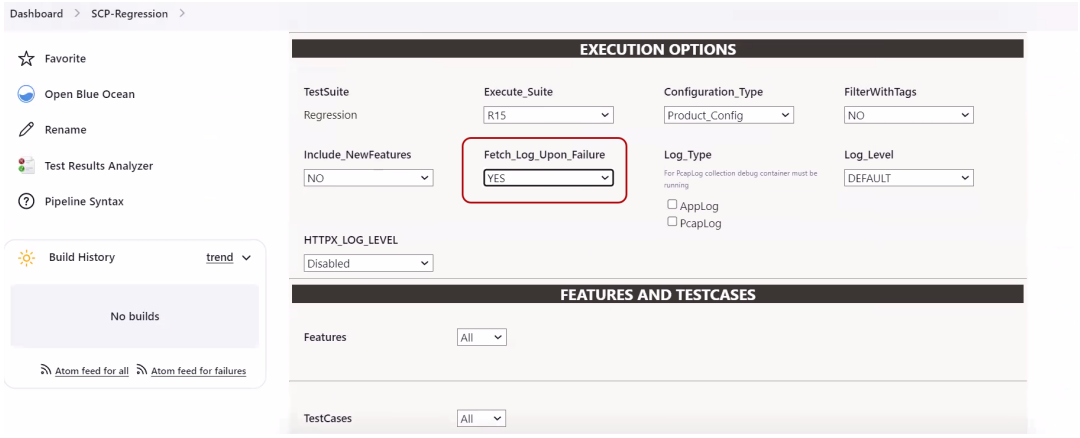
Application Log Collection Using OpenSearch

To access the option to collect logs using OpenSearch:

1. Log in to ATS using respective <NF> login credentials.
2. On the NF home page, click any new feature or regression pipeline, from where you want to collect the logs.
3. In the left navigation pane, click **Build with Parameters**.
4. Select YES or NO from the drop-down menu of **Fetch_Log_Upon_Failure** to select whether the log collection is required for a particular run.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-22 Fetch_Log_Upon_Failure



5. If option **Log_Type** is also available, select value **AppLog** for it.
6. Select the Log Level from the drop-down menu of Log_Level to set the log level for all the microservices. The possible values for Log_Level are as follows:
   - WARN: Designates potentially harmful situations.
   - INFO: Designates informational messages that highlight the progress of the application at coarse-grained level.
   - DEBUG: Designates fine-grained informational events that are most useful to debug an application.
   - ERROR: Designates error events that might still allow the application to continue running.
   - TRACE: The TRACE log level captures all the details about the behavior of the application. It is mostly diagnostic and is more granular and finer than DEBUG log level.

   > **Note**
   >
   > Log_Level values are NF dependent.

7. After the build execution is complete, go into the ATS pod, then navigate to following path to find the applogs:`.jenkins/jobs/<Pipeline Name>/builds/<build number>/` For example,`.jenkins/jobs/SCP-Regression/builds/5/`

   Applogs is present in zip form. Unzip it to get the log files.

The following tasks are carried out in the background to collect logs:

- OpenSearch API is used to access and fetch logs.
- Logs are fetched from OpenSearch for the failed scenarios
- Hooks (after scenario) within the cleanup file initiate an API call to OpenSearch to fetch Application logs.
- Duration of the failed scenario is calculated based on the time stamp and passed as a parameter to fetch the logs from OpenSearch.
- Filtered query is used to fetch the records based on Pod name, Service name, and timestamp (Failed Scenario Duration).

Copyright © 2019, 2025, Oracle and/or its affiliates.

- For OpenSearch, there is no rollover or rotation of logs over time.
- The following configuration parameters are used for collecting logs using OpenSearch:
  - OPENSEARCH_WAIT_TIME: Wait time to connect to OpenSearch
  - OPENSEARCH_HOST: OpenSearch HostName
  - OPENSEARCH_PORT: OpenSearch Port

Application Log Collection Using Kubernetes Logs

To access the option to to collect logs using Kubernetes Logs:

1. On the NF home page, click any new feature or regression pipeline, from where you want to collect the logs.
2. In the left navigation pane, click **Build with Parameters**.
3. Select YES or NO from the drop-down menu of **Fetch_Log_Upon_Failure** to select whether the log collection is required for a particular run.
4. Select the Log Level from the drop-down menu of Log_Level to set the log level for all the microservices. The possible values for Log_Level are as follows:
   - WARN: Designates potentially harmful situations.
   - INFO: Designates informational messages that highlight the progress of the application at coarse-grained level.
   - DEBUG: Designates fine-grained informational events that are most useful to debug an application.
   - ERROR: Designates error events that might still allow the application to continue running.

> **Note**
>
> Log_Level values are NF dependent.

The following tasks are carried out in the background to collect logs:

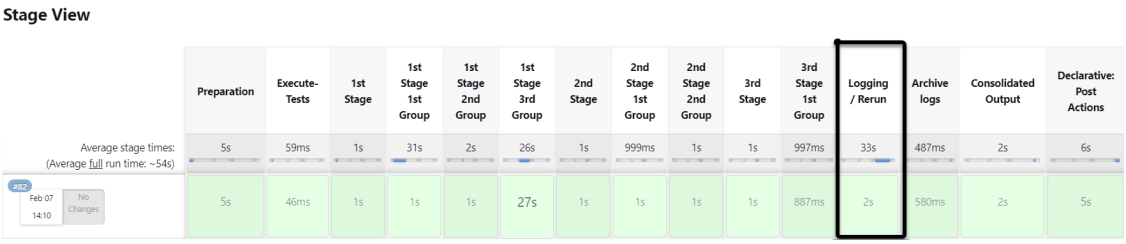
- Kube API is used to access and fetch logs.
- For failed scenarios, logs are directly fetched from microservices.
- Hooks (after scenario) within the cleanup file initiate an API call to Kubernetes Logs to fetch Application logs.
- The duration of the failed scenario is calculated based on the time stamp and passed as a parameter to fetch the logs from microservices.
- Logs roll can occur while fetching the logs for a failed scenario. The maximum loss of logs is confined to a single scenario.

2.7.1 Application Log Collection and Parallel Test Execution Integration

A new stage,"Logging/Rerun", has been added at the end of the Execute-Tests stage to collect rerun logs, such as applog and PCAP logs, by running the failed test cases in a sequence.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-23 Logging/Rerun new stage



If the `Fetch_Log_Upon_Failure` parameter is set to YES and if any test case fails in the initial run, then:

- The failed test case reruns and log collection start in the Logging/Rerun stage after the initial run is completed for all the test cases.
- The logs from the initial execution are collected, but they might be incorrect.
- Even if the `rerun` parameter is set to 0, the failed test case reruns in the Logging/Rerun stage and the log is collected.

> **Note**
>
> Not applicable for all the NFs.

- If the `Fetch_Log_Upon_Failure` parameter is set to NO and if any test case fails in the initial run, then the failed test case rerun starts in the same stage after the initial execution is over for all the test cases in its group.

2.8 ATS Maintenance Scripts

ATS maintenance scripts are used to perform the following operations:

- Taking a backup of the ATS custom folders and Jenkins pipeline.
- Viewing the configuration and restoring the Jenkins pipeline.
- Viewing the configuration and installing or uninstalling ATS and stubs.

ATS maintenance scripts are present in the ATS image at the following path: `/var/lib/jenkins/ocats_maint_scripts`

Run the following command to copy the scripts to a local system (bastion):

```
kubectl cp <NAMESPACE>/<POD_NAME>:/var/lib/jenkins/ocats_maint_scripts
<DESTINATION_PATH_ON_BASTION> pod
```

For example,

```
kubectl cp ocpcf/ocats-ocats-policy-694c589664-js267:/var/lib/Jenkins/
ocats_maint_scripts /home/meta-user/ocats_maint_scripts pod
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

2.8.1 ATS Scripts

ATS maintenance scripts are used to perform various task related to ATS and Jenkin pipeline.

The following are the types of scripts:

- **ats_backup.sh**: This script requires the user's input and takes a backup of the ATS custom folders, Jenkins jobs, and user's folders on the user's system. The backup can be of the Jenkins jobs and user's folder, the custom folders, or both. The custom folders include cust_regression, cust_newfeatures, cust_performance, cust_data, and custom_config. For a Jenkins job or a user's folder, the script only takes a backup of the config.xml file. Also, the script requires the user to store a backup on the user's system (the default path is the location from where the script is being run) and to create a backup folder on the system and take the backup of the chosen folder from the corresponding ATS into the backup folder. The backup folder name can be of the following notation: `ats_<version>_backup_<date>_<time>`.
- **ats_uninstall.sh**: This script requires the user's input and uninstalls the corresponding ATS.
- **ats_install.sh**: This script requires the user's input and installs a new ATS. If `PVEnabled` is set to `true`, the script also reads the PVC name from values.yaml and creates values.yaml before installation. Also, if needed, the script performs the postinstallation steps, such as copying tests and Jenkins jobs' folders from the `ats_data` tar file to the ATS pod when PV is deployed, and then restarts the pod.
- **ats_restore.sh**: This script requires the user's inputs, restores the new release ATS pipeline, and views the configuration by referring to the last release ATS Jenkins jobs and the user's configuration. It depends on the user whether to use the backup folders from the user's system to restore the ATS configuration. If the user instructs the script to use the backup from the system, the script requires the path of the backup and uses the backup to restore. Otherwise, the script requires the last ATS Helm release name to refer to its Jenkins jobs and the user's configuration to restore.
  The script refers to the last release of ATS Jenkins pipelines and sets the `Discard old builds` property if this property is set in the last release of ATS for a pipeline but not in the current release. If this property is set in both releases, the script just updates the values according to the last release. Also, the script restores the `pipeline environment variables` values as per the last release of ATS. If any custom pipeline (created by the user) was present in the last release of ATS, the script restores that as well. It also restores the extra views created by NF users, for example, policy users, SCP users, and NRF users. Moreover, the script displays messages about the pending configuration that the user needs to perform manually. For example, a new pipeline or a new environment variable (for a pipeline) is introduced in the new release.

  While deploying ATS without PV, Jenkins needs to be restarted for the restore process to complete. If the last release ATS contains the `Configuration_Type` parameter, the `Configuration_Type` script needs to be approved with the **In Process Script Approval** setting under **Manage ATS** in Jenkins for the restore process to complete.

Copyright © 2019, 2025, Oracle and/or its affiliates.

2.8.2 Updating ats_install.sh

Currently, the `ats_install.sh` script copies the tests folder and Jenkins jobs folder into the ATS pod and then restarts the pod when deployed with PV.

How to Update ats_install.sh

Other NFs can also use the `ats_install.sh` scripts. However, additional post installation steps may have to be performed manually for a few NFs.

For the additional post installation commands, perform the following steps:

1. In the `ats_install.sh` script, there is a post install section between `####POST_INSTALL_START####` and `#### POST_INSTALL_END ####`.
   a. Add the required post install commands.

   > **Note**
   >
   > These commands are NF-specific.

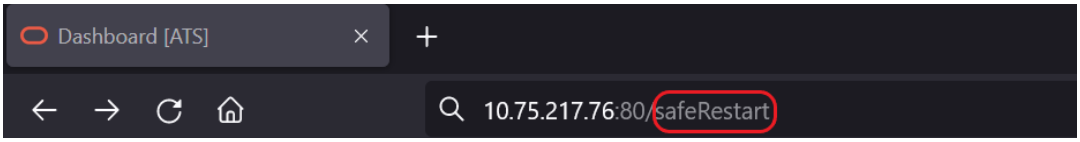
   b. Use the following commands:
      - `$namespace` for the namespace value
      - `$pod_name` for the pod name
      - `$ats_data_path` for the `ats_data` folder path (it has tests folder and Jenkins jobs folder provided as tar file in ATS package)
   c. In the if-else block related to whether PV is enabled or not, add the following:
      - Add a command specific to `PVEnabled=true` in the `if block`.
      - Add a command specific to `PVEnabled=false` in the `else block`.
2. For additional inputs, enter the required code between `#### INPUT_START ####` and `#### INPUT_END ####`.

2.8.3 Restarting Jenkins without Restarting Pod

Perform the following procedure to restart Jenkins without restarting pods:

1. Log in as the Jenkins admin.
2. Go to the `<Jenkins_IP>:<port>/safeRestart`, for example, `10.87.73.32:32156/safeRestart`.

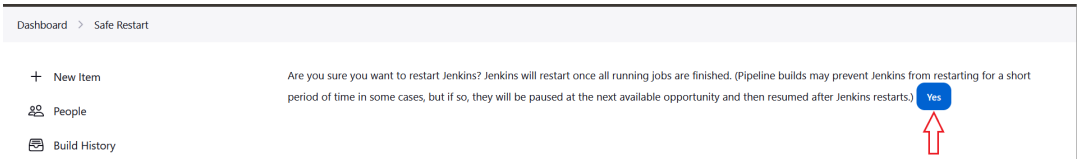
Figure 2-24 Safe Restart



3. Click **Yes**.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-25 Restart Jenkins



2.8.4 Updating Stub Scripts

The following stubs can be updated:

- **stub_uninstall.sh**: This script requires the user's inputs and uninstalls all the stubs.
- **stub_install.sh**: This script requires the user's inputs and installs all the stubs.

> **Note**
>
> Currently, `stub_uninstall.sh` and `stub_install.sh` work.

Perform the following procedure to update the stub scripts for other NFs (NRF in this case):

1. Go to the `stub` folder.
2. From each script:
   a. Remove the CNC Policy-specific stubs inputs (dns, amf, and ldap), and add the input code blocks for NRF-specific stubs.
   b. For the stubs to uninstall, change the value of the `stubUninstallList` variable, and delete the variables for the CNC Policy-specific stubs below it.

      > **Note**
      >
      > `stubUninstallList` contains the Helm release names of the common stubs that are deployed generally.

   c. Declare the variables for the NRF-specific stubs below the `stubUninstallList` line.
   d. Remove the Helm uninstallation commands of the policy-specific stubs, and add the Helm uninstallation commands of the NRF-specific stubs.
   e. For the stubs to install, change the value of the `stubInstallList` variable, and delete the variables for the CNC Policy-specific stubs below it.

      > **Note**
      >
      > `stubInstallList` contains the Helm release names of the common stubs that are deployed generally.

   f. Declare the variables for the NRF-specific stubs below the `stubInstallList` line.
   g. Remove the Helm installation commands of the CNC Policy-specific stubs, and add the Helm installation commands of the NRF-specific stubs.

Copyright © 2019, 2025, Oracle and/or its affiliates.

2.8.5 Running ATS and Stub Deployment Scripts

Perform the following procedure to run ATS and stub deployment scripts:

> **Note**
>
> If you want to take a backup of the custom folders or Jenkins jobs and user's configuration or both, run the `ats_backup.sh` script.

1. Run the `ats_install.sh` script to install the new release ATS (values.yaml of the ATS Helm chart must be updated before this step).
2. Run the `ats_restore.sh` script to restore the new ATS pipeline and view configuration.

> **Note**
>
> - You might perform the manual steps required for the restore script.
> - You must copy all the necessary changes to the new release ATS from the last release ATS. To get the changes in the last release, you must refer to the custom folders in the last release ATS backup on the system with an existing backup using `ats_backup.sh` before this step.
> - You can remove the last release ATS pod using the `ats_uninstall.sh` script while continuing to retain the last release PVC. You can use the last release PVC to port backward. Delete the last release PVC when you do not require the backward porting.

3. Run the `stub_install.sh` script to install all the new release stubs values.yaml of the stub Helm charts must be updated before this step.
4. Run the `stub_uninstall.sh` script to uninstall all the last release stubs.

2.9 ATS System Name and Version Display on the ATS GUI

This feature displays the ATS system name and version on the ATS GUI.

You can log in to the ATS application using the login credentials to view the following:

- ATS system name: Abbreviated product name followed by NF name.
- ATS Version: Release version of ATS.

2.10 ATS Tagging Support

The ATS Tagging Support feature assists in running the feature files after filtering features and scenarios based on tags. Instead of manually navigating through several feature files, the user can save time by using this feature.

The GUI offers the following four options for selecting tag types:

- Feature_Include_Tags: The features that contain either of the tags available in the **Feature_Include_Tags** field are considered for tagging.

Copyright © 2019, 2025, Oracle and/or its affiliates.

  - For example, "cne-common", "config-server". All the features that have either "cne-common" or "config-server" tags are taken into consideration.

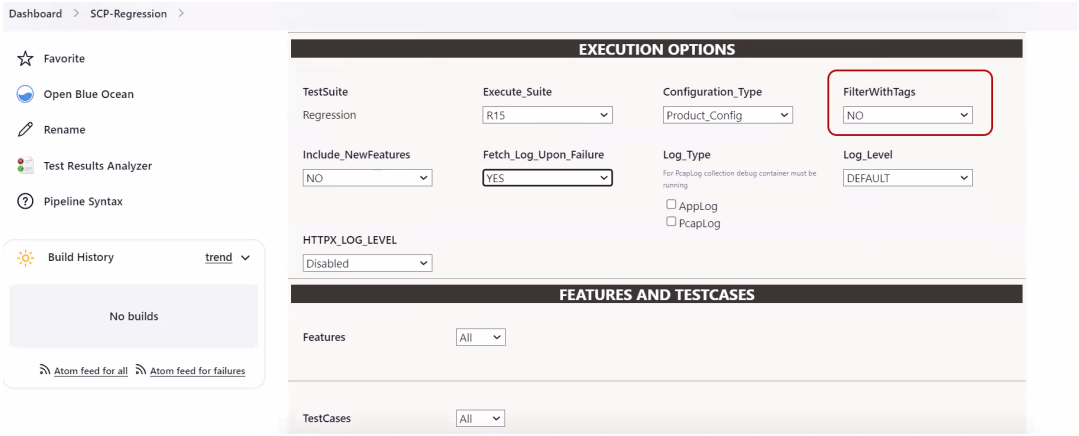
- Feature_Exclude_Tags: The features that contain neither of the tags available in the **Feature_Exclude_Tags** field are considered for tagging.
  - For example, "cne-common","config-server". All the features that have neither "cne-common" nor "config-server" as tags are taken into consideration.
- Scenario_Include_Tags: The scenarios that contain either of the tags available in the **Scenario_Include_Tags** field are considered.
  - For example, "sanity", "cleanup". The scenarios that have either "sanity" or "cleanup" tags are taken into consideration.
- Scenario_Exclude_Tags: The features that contain neither of the tags available in the **Scenario_Exclude_Tags** field are considered.
  - For example, "sanity", "cleanup". The scenarios that have neither "sanity" nor "cleanup" as tags are taken into consideration.

Filter with Tags

The procedure to filter feature files and scenarios based on tags are as follows:

1. On the NF home page, click any new feature or regression pipeline, where you want to use this feature.
2. In the left navigation pane, click **Build with Parameters**. The following image appears.

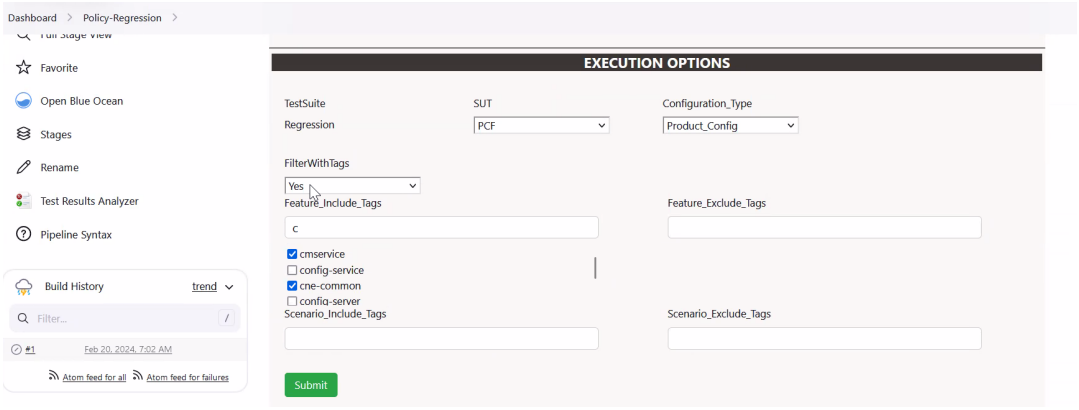
Figure 2-26 Filter with Tags



3. Select **Yes** from the **FilterWithTags** drop-down menu. The result shows four input fields. The default value of **FilterWithTags** field is "No".
4. The input fields serve as a search or filter, displaying all tags that match the prefix entered. You can select one or multiple tags.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-27 Tags Matching with Entered Prefix



5. Select the required tags from the different tags list and click **Submit**.

The specified feature-level tags are used to filter out features that contain any one of the include tags and none of the exclude tags. Here, any or both the fields may be left empty. All features are automatically taken into consideration when both fields are empty.

The scenario level tags are used to filter out the scenarios from the features filtered above. Only scenarios with any of the include tags and none of the exclude tags are considered. Any or both fields can be empty. When both fields are empty, all the scenarios from the above filtered feature files are considered.

> **Note**
>
> - If you select the Select_Option as 'All', all the displayed features and scenarios will run.
> - If you select the Select_Option as 'Single/MultipleFeatures, it enables you to select some features, and only those features and respective scenarios are going to run.

2.10.1 Combination of Tags and their Results

The combination of tags and expected results are as follows.

Table 2-6 Result of Filtered Tags

Feature_Include	Feature_Exclude	Scenario_Include	Scenario_Exclude	Results
-	-	-	-	All the features and scenarios are taken into consideration.
"abc","def"	-	-	-	Features with either "abc" or "def" tags and all scenarios from the filtered features are taken into consideration.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 2-6 (Cont.) Result of Filtered Tags

Feature_Include	Feature_Exclude	Scenario_Include	Scenario_Exclude	Results
-	"abc","def"	-	-	All the features with neither "abc" nor "def" tags and all scenarios from the filtered features are taken into consideration.
-	-	"sanity","cne"	-	Scenarios with either "sanity" or "cne" tags and features having these scenarios are taken into consideration.
-	-	-	"sanity","cne"	Scenarios with neither "sanity" nor "cne" tags and features having these filtered scenarios are taken into consideration.
"abc","def"	"ghi"	-	-	Features with either "abc" or "def" tags but without the "ghi" tag and all scenarios from filtered features are taken into consideration.
"abc","def"	-	"sanity","cne"	-	Scenarios only with either "sanity" or "cne" tags and only features that contain these scenarios and have either "abc" or "def" as feature tags are taken into consideration.
"abc","def"	-	-	"sanity","cne"	Scenarios with neither "sanity" nor "cne" tags and only features that contain the filtered scenarios and have either "abc" or "def" feature tags are taken into consideration.
-	"ghi"	"sanity","cne"	-	Features without the "ghi" tag and scenarios with either "sanity" or "cne" tags from the filtered features are taken into consideration.
-	"ghi"	-	"sanity","cne"	Features without the "ghi" tag and scenarios without the "sanity" and "cne" tags from filtered features are taken into consideration.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 2-6 (Cont.) Result of Filtered Tags

Feature_Include	Feature_Exclude	Scenario_Include	Scenario_Exclude	Results
-	-	"sanity","cne"	"cleanup"	Scenarios with either the "sanity" or "cne" tags and without the "cleanup" tag and features with filtered scenarios are taken into consideration.
"abc","def"	"ghi"	"sanity","cne"	-	Scenarios with either the "sanity" or "cne" tags and features that have these scenarios and have either the "abc" or "def" tags but not the "ghi" tag are taken into consideration.
"abc","def"	-	"sanity","cne"	"cleanup"	Scenarios with either the "sanity" or "cne" tags and without the "cleanup" tag, and features having the filtered scenarios and having the feature tags either "abc" or "def" are taken into consideration.
"abc","def"	"ghi"	-	"cleanup"	Scenarios without the tag "cleanup", and features with filtered scenarios and having either "abc" or "def" as feature tags but not the "ghi" tag are taken into consideration.
-	"ghi"	"sanity","cne"	"cleanup"	Scenarios with either the "sanity" or "cne" tags and without the "cleanup" tag, and features with filtered scenarios and not the tag "ghi," are taken into consideration.
"abc","def"	"ghi"	"sanity","cne"	"cleanup"	Scenarios with either "sanity" or "cne" tags and without the "cleanup" tag, and features with filtered scenarios and feature tags either "abc" or "def" but without the tag "ghi" are taken into consideration.

Copyright © 2019, 2025, Oracle and/or its affiliates.

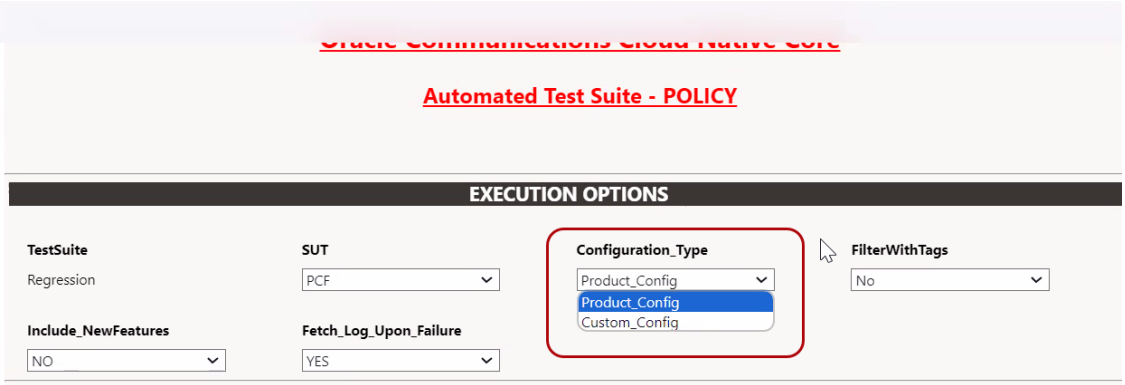
> **Note**
>
> - The tags mentioned in the table are just examples; they may or may not be actually used.
> - The **Replay** option in the Jenkins GUI is not supported for tag-related test case execution. Always trigger builds related to tagging from the **Build with Parameter** step, and do not replay any previous builds.
> -

2.11 Custom Folder Implementation

The Custom Folder Implementation feature allows the user to update, add, or delete test cases without affecting the original product test cases in the new features, regression, and performance folders. The implemented custom folders are cust_newfeatures, cust_regression, and cust_performance. The custom folders contain the newly created, customised test cases.

Initially, the product test case folders and custom test case folders will have the same set of test cases. The user can perform customization in the custom test case folders, and ATS always runs the test cases from the custom test case folders. If the option "Configuration_Type" is present on the GUI,the user needs to set its value to "Custom_Config" to populate test cases from the custom test case folders.

Figure 2-28 Custom Config Folder

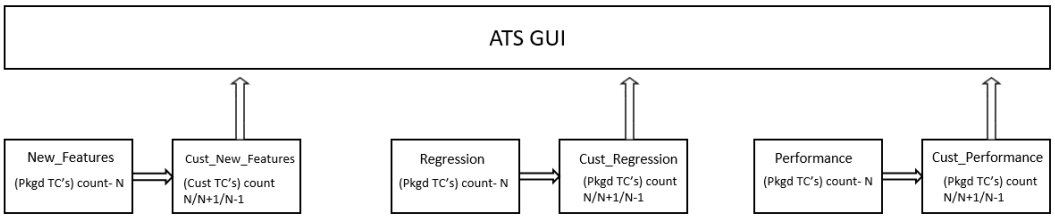


Summary of Custom Folder Implementation

- Separate folders such as cust_newfeatures, cust_regression, and cust_performance are created to hold the custom cases.
- The prepackaged test cases are available in the newfeature and regression Folder.
- The user copies the required test cases to the cust_newfeatures and cust_regression folders, respectively.
- Jenkins always points to the cust_newfeatures and cust_regression folders to populate them in the menu.
  If someone initially launches ATS, they will not see any test cases in the menu if the cust folders are not populated. To avoid this, it is recommended to prepopulate both the folders, cust and original, and ask the user to modify only the cust folder if needed.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-29 Summary of Custom Folder Implementation



2.12 Health Check

Health Check functionality is to check the health of the System Under Test (SUT)

Earlier, ATS used Helm test functionality to check the health of the System Under Test (SUT). With the implementation of the ATS Health Check pipeline, the SUT health check process has been automated. ATS health checks can be performed on webscale and non-webscale environments.

Convert a Value in Base64

The following command can be utilized to convert any value into base64 encoding:

```
echo-n "value"| base64
```

For example,

```
echo-n "126.98.76.43"| base64
```

Deploying Health Check in a Webscale Environment

- Set the `Webscale` to 'true' and the following parameters by encoding them with base64 in the ATS values.yaml file:
- Set the following parameter to encrypted data:

```
webscalejumpserverip: encrypted-data
webscalejumpserverusername: encrypted-data
webscalejumpserverpassword: encrypted-data
webscaleprojectname: encrypted-data
webscalelabserverFQDN: encrypted-data
webscalelabserverport: encrypted-data
webscalelabserverusername: encrypted-data
webscalelabserverpassword: encrypted-data
```

Encrypted data is the value of parameters encrypted in base64. Fundamentally, Base64 is used to encode the parameters.

For example:

```
webscalejumpserverip=$(echo -n '10.75.217.42' | base64), Where Webscale Jump
server ip needs to be provided
webscalejumpserverusername=$(echo -n 'cloud-user' | base64), Where Webscale
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
Jump server Username needs to be provided
webscalejumpserverpassword=$(echo -n '*****' | base64), Where Webscale Jump
server Password needs to be provided
webscaleprojectname=$(echo -n '*****' | base64), Where Webscale Project Name
needs to be provided
webscalelabserverFQDN=$(echo -n '*****' | base64), Where Webscale Lab Server
FQDN needs to be provided
webscalelabserverport=$(echo -n '*****' | base64), Where Webscale Lab Server
Portneeds to be provided
webscalelabserverusername=$(echo -n '*****' | base64), Where Webscale Lab
Server Username needs to be provided
webscalelabserverpassword=$(echo -n '*****' | base64), Where Webscale Lab
Server Password needs to be provided
```

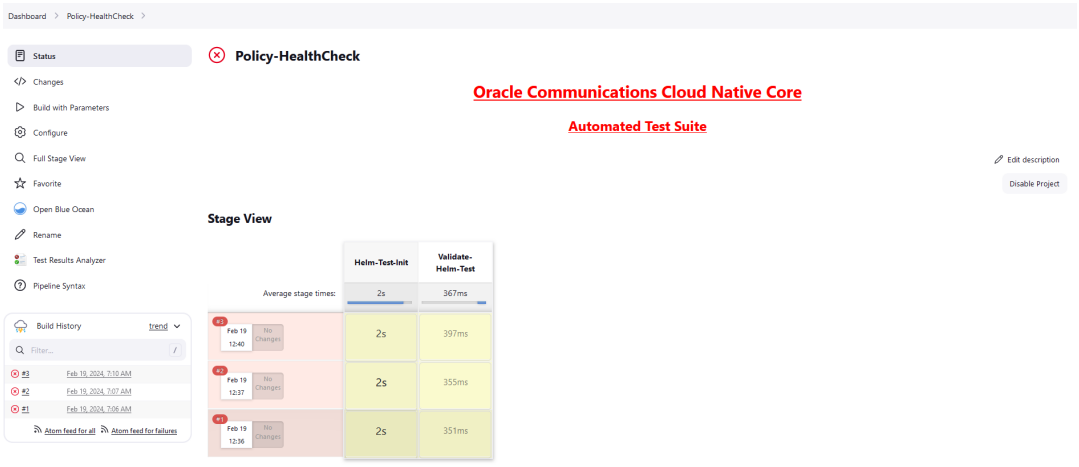
Running Health Check Pipeline in an Webscale Environment

To run Health Check pipeline:

1. Log in to ATS using respective <NF> login credentials.
2. Click <NF>HealthCheck pipeline and then click **Configure**.

> **Note**
>
> <NF> denotes the network function. For example, in Policy, it is called as Policy-HealthCheck pipeline.

Figure 2-30 Configure Healthcheck



3. Provide parameter a with Helm release name deployed. If there are multiple releases, use comma to provide all Helm release names.

   ```
   //a = helm releases [Provide Release Name with Comma Separated if more
   than 1 ]
   ```

   Provide parameter c with the appropriate Helm command, such as helm, helm3, or helm2.

   ```
   //c = helm command name [helm or helm2 or helm3]
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

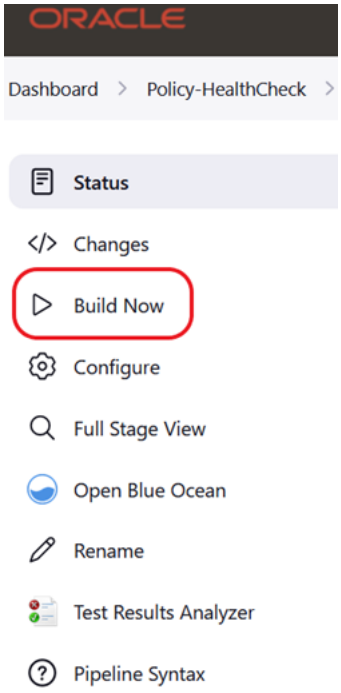
Figure 2-31 Save the Changes



4. Save the changes and click **Build Now**. ATS runs the health check on respective network function.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-32 Build Now



Deploying Health Check Pipeline in an OCI Environment

To use a ssh private key, create healthcheck-oci-secret and set the value of the key "passwordAuthenticationEnabled" to false.

Creating healthcheck-oci-secret

Create healthcheck-oci-secret to use ssh private keys instead of passwords using the following command:

```
kubectl create secret generic healthcheck-oci-secret --from-
file=bastion_key_file='<path of bastion ssh private key file>' --from-
file=operator_instance_key_file='<path of operator instance ssh private key
file>' -n <ATS namespace>
```

For example,

```
kubectl create secret generic healthcheck-oci-secret --from-
file=bastion_key_file='/tmp/bastion_private_key' --from-
file=operator_instance_key_file='/tmp/operator_instance_private_key' -n
seppsvc
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> - Maintain the name of the secret as "healthcheck-oci-secret".
> - Ensure that the '--from-file' keys retain the same names: "bastion_key_file" and "operator_instance_key_file".
> - If the SSH private key is identical for both the bastion and operator instance, you can use the same path for both in the secret creation command.

Perform the following procedure to deploy ATS Health Check in a OCI environment:

Set the `Webscale` parameter set to 'false' and following parameters by encoding it with base64 in the ATS values.yaml file.

- To use password, provide base64 encoded values for key "password" for both bastion and operator instances, and set the value of key **passwordAuthenticationEnabled** to "true".
- Set the following parameter to encrypted data:

```
envtype: encrypted-data
ociHealthCheck:
  passwordAuthenticationEnabled: true or false
  bastion:
    ip: encrypted-data
    username: encrypted-data
    password: encrypted-data
  operatorInstance:
    ip: encrypted-data
    username: encrypted-data
    password: encrypted-data
```

> **Note**
>
> All fields are mandatory except for passwords. When the "passwordAuthenticationEnabled" field is set to true, only the "password" field needs to be updated; otherwise, it can remain with its default value.

Running Health Check Pipeline in an OCI Environment

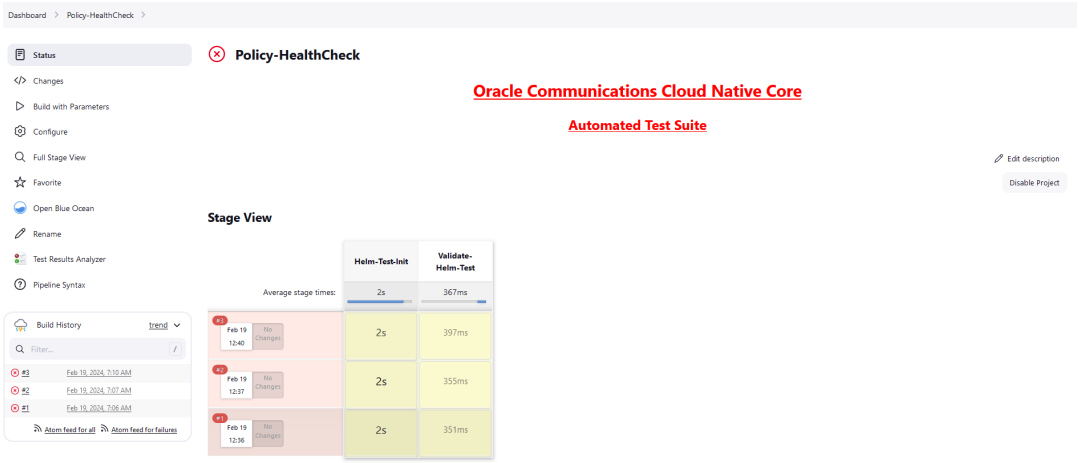
To run ATS Health Check pipeline:

1. Log in to ATS using respective <NF> login credentials.
2. Click <NF>HealthCheck pipeline and then click **Configure**.

> **Note**
>
> <NF> denotes the network function. For example, in Policy, it is called as Policy-HealthCheck pipeline.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-33 Configure Healthcheck



3. Provide parameter a with Helm release name deployed. If there are multiple releases, use comma to provide all Helm release names.

```
//a = helm releases [Provide Release Name with Comma Separated if more
than 1 ]
```

Provide parameter c with the appropriate Helm command, such as helm, helm3, or helm2.

```
//c = helm command name [helm or helm2 or helm3]
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

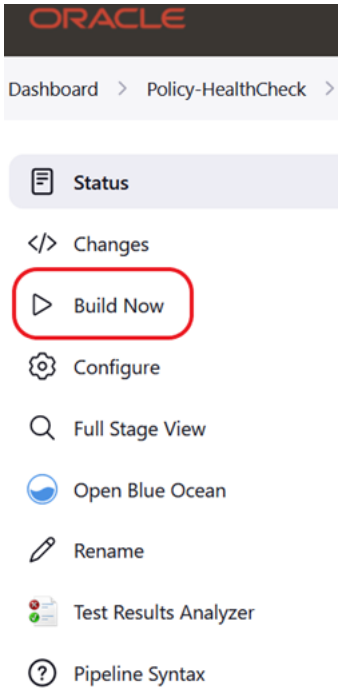
Figure 2-34 Save the Changes



4. Save the changes and click **Build Now**. ATS runs the health check on respective network function.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-35 Build Now



Deploying Health Check in a Non-Webscale or Non-OCI Environment

Perform the following procedure to deploy ATS Health Check in a non-webscale or non-OCI environment such as OCCNE:

Set the `Webscale` parameter set to 'false' and following parameters by encoding it with base64 in the ATS values.yaml file:

```
occnehostip: encrypted-data
occnehostusername: encrypted-data
occnehostpassword: encrypted-data
```

Example:

```
occnehostip=$(echo -n '10.75.217.42' | base64) , Where occne host ip needs to
be provided
occnehostusername=$(echo -n 'cloud-user' | base64), Where occne host username
needs to be provided
occnehostpassword=$(echo -n '****' | base64), Where password of host needs to
be provided
```

Running Health Check Pipeline in a Non-Webscale or Non-OCI Environment

Perform the following procedure to run the ATS Health Check pipeline in a non-webscale or non-OCI environment such as OCCNE:

1. Log in to ATS using respective <NF> login credentials.
2. Click <NF>HealthCheck pipeline and then click **Configure**.
3. Provide parameter a with Helm release name deployed. If there are multiple releases, use comma to provide all Helm release names.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Provide parameter b with SUT deployed namespace name.

Provide parameter c with the appropriate Helm command, such as helm, helm3, or helm2.

```
//a = helm releases [Provide Release Name with Comma Separated if more
than 1 ]
//b = Namespace, If not applicable to WEBSCALE environment then remove the
argument
//c = helm command name [helm or helm2 or helm3]
```

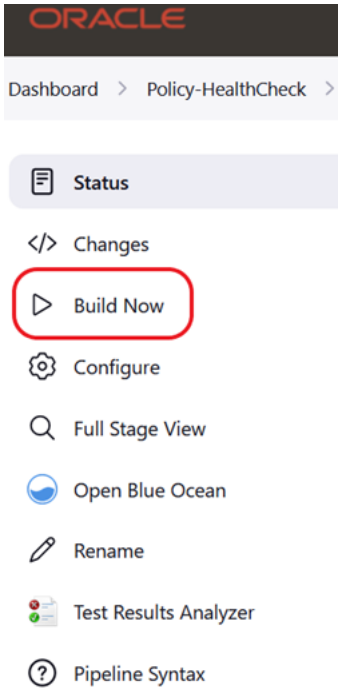
Figure 2-36 Save the Changes



4. Save the changes and click **Build Now**. ATS runs the health check on respective network function.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-37 Build Now



By clicking **Build Now**, you can run the health check on ATS and store the result in the console logs.

2.13 Individual Stage Group Selection

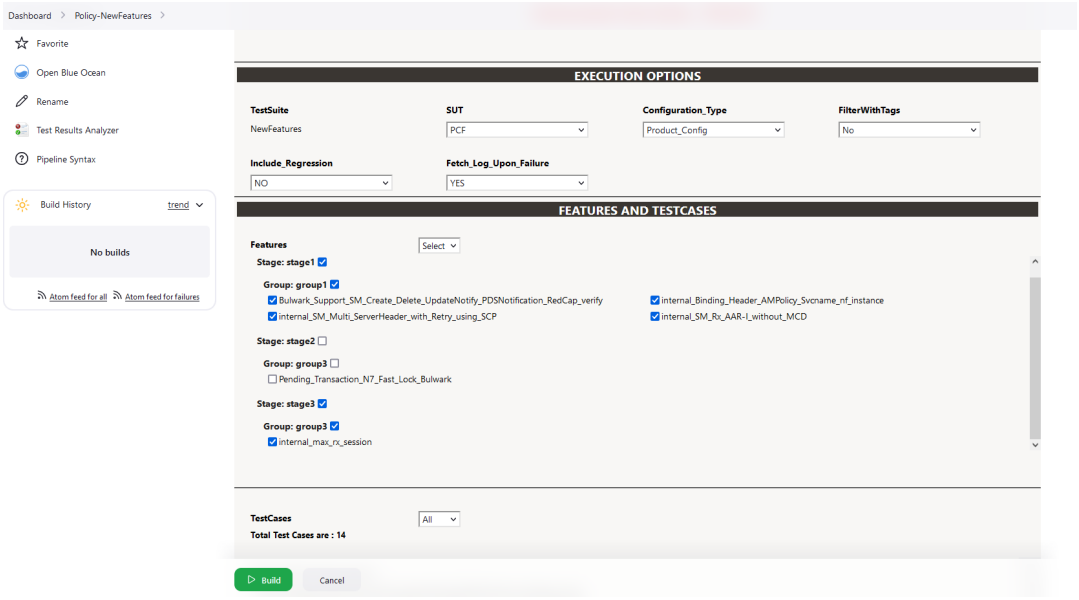
The Individual Stage Group Selection feature allows you to select and execute a single or multiple stages or groups by selecting a check box for the corresponding stage or group.

Follow the steps to select an individual stage or group:

1. Click NF-Regression or NF-NewFeatures, and then click **Build with Parameters**.
2. On the FEATURE and TESTCASES section, click **Select** from the **Features** drop-down menu.
3. Select the corresponding check box to select any number of stages or groups you want to run from the list available for execution.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-38 Stages or Groups Selection

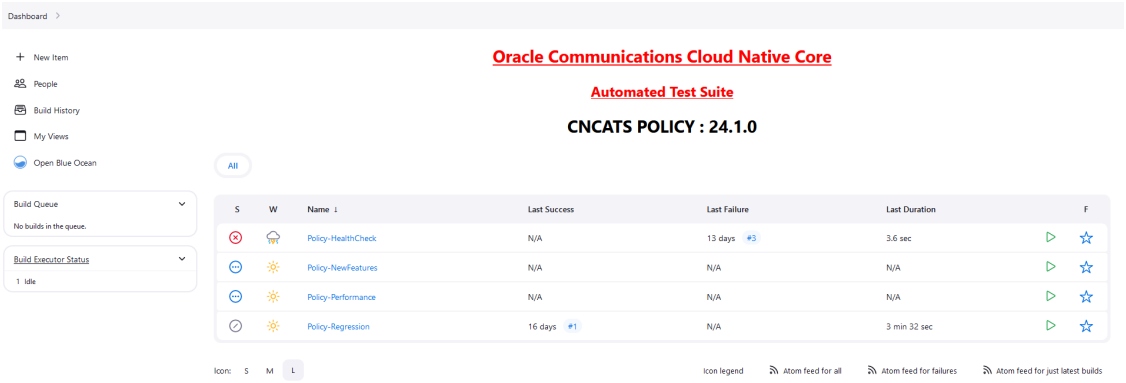


4. Scroll down to click **Build**.

2.14 Lightweight Performance

The Lightweight Performance feature allows you to run performance test cases. In ATS, a new pipeline known as "<NF>-Performance", where NF stands for Network Function, is introduced, for example, Policy-Performance.

Figure 2-39 Sample Screen: Home Page



The <NF>-Performance pipeline verifies from 500 to 1k TPS (Transactions per Second) of traffic using the http-go tool, a tool used to run the traffic on the backend. It also helps to monitor the CPU and memory of microservices while running lightweight traffic.

The duration of the traffic run can be configured on the pipeline.

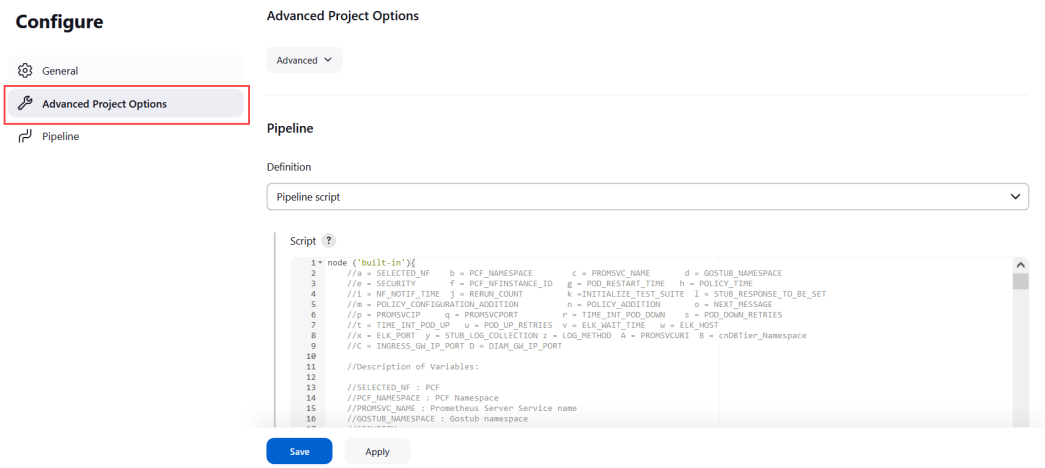
2.14.1 Configure <NF>-Performance Pipeline

Perform the following to configure the performance pipeline:

Copyright © 2019, 2025, Oracle and/or its affiliates.

1. On the NF home page, click **<NF>-Performance** pipeline, and then click **Configure**. The **General** tab appears. The user must wait for the page to load completely.
2. Click the **Advanced Project Options** tab. Scroll down to reach the **Pipeline** configuration section.

Figure 2-40 Advanced Project Options



3. Update the configurations as per your NF requirements and click **Save**. The **Pipeline <NF>-Performance** page appears.
4. Click **Build Now**. This triggers lightweight traffic for the respective network function.

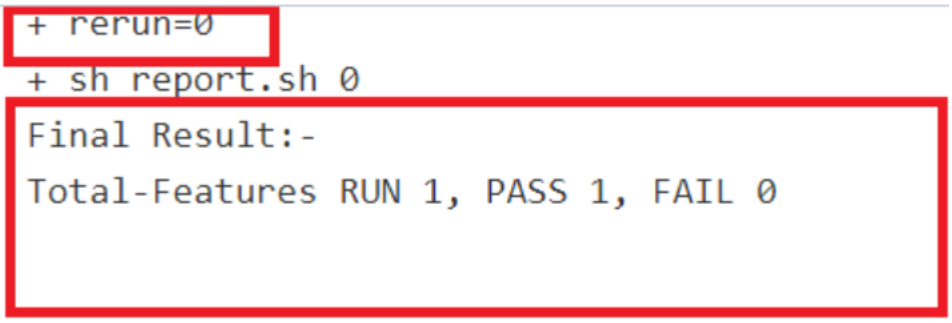
2.15 Managing Final Summary Report, Build Color, and Application Log

This feature displays an overall execution summary, such as the total run count, pass count, and fail count.

Supports Implementation of Total-Features
ATS supports implementation of **Total-Features** in the final summary report. Based on the rerun value set, the **Final Result** section in the final summary report displays the Total-Features output.

- If rerun is set to 0, the test result report shows the following result:

Figure 2-41 Total-Features = 1, and Rerun = 0



Copyright © 2019, 2025, Oracle and/or its affiliates.

- If rerun is set to non-zero, the test result report shows the following result:

Figure 2-42 Total-Features = 1, and Rerun = 2

```
+ rerun=2
+ sh report.sh 2
Initial Run :-
Features RUN 1, PASS 0, FAIL 1

1st Rerun:-
Features RUN 1, PASS 0, FAIL 1

2nd Rerun:-
Features RUN 1, PASS 0, FAIL 1

Final Result:-
Total-Features RUN 1, PASS 0, FAIL 1
```

Changes After Parallel Test Execution Framework Feature Integration
After incorporating the Parallel Test Execution feature, the following results were obtained:

Final Summary Report Implementations

Figure 2-43 Group Wise Results

```
**********Stage: stage1 Group: group3 Result**********
Initial Run :-
Features RUN 1, PASS 0, FAIL 1

1st Rerun:-
Features RUN 1, PASS 0, FAIL 1

2nd Rerun:-
Features RUN 1, PASS 0, FAIL 1

stage1 group3
Final Result:-
Total-Features RUN 1, PASS 0, FAIL 1
```

Figure 2-44 Overall Result When Selected Feature Tests Pass

```
**********Overall Result**********

Took 0h0m50s

Final Result:-
Total-Features RUN 1, PASS 1, FAIL 0
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-45 Overall Result When Any of the Selected Feature Tests Fail

```
**********************************Overall Result****************************************
Failing scenarios:
Stage:stage1 Group:group3
  SCP_Registration_With_PLMNList.feature:45  Static configuration to NRF for all nfType

Took 0h1m59s

Initial Run:-
Features RUN 1, PASS 0, FAIL 1

1st Rerun:-
Featues RUN 1, PASS 0, FAIL 1

2nd Rerun:-
Featues RUN 1, PASS 0, FAIL 1

Final Result:-
Total-Featues RUN 1, PASS 0, FAIL 1
```

Implementing Build Colors

ATS supports implementation of build color. The details are as follows:

Table 2-7 Build Color Details

Rerun Values	Rerun set to zero		Rerun set to non-zero		
Status of Run	**All Passed in Initial Run**	**Some Failed in Initial Run**	**All Passed in Initial Run**	**Some Passed in Initial Run, Rest Passed in Rerun**	**Some Passed in Initial Run, Some Failed Even After Rerun**
Build Status	SUCCESS	FAILURE	SUCCESS	SUCCESS	FAILURE
Pipeline Color	GREEN	Execution Stage where test cases failed shows YELLOW color, rest of the successful stages are GREEN.	GREEN	GREEN	Execution Stage where test cases failed shows YELLOW color, rest of the successful stages are GREEN
Status Color	BLUE	RED	BLUE	BLUE	RED

Changes After Integrating Parallel Test Execution Framework Feature
In sequential execution, the build color or overall pipeline status of any run was mainly dependent on the following parameters:

- the rerun count and the pass or fail status of test cases in the initial run
- the rerun count and the pass or fail status of test cases in the final run

Copyright © 2019, 2025, Oracle and/or its affiliates.

For the parallel test case execution, the pipeline status also depends on another parameter, "Fetch_Log_Upon_Failure," which is given in the **build with parameters** page. If the parameter `Fetch_Log_Upon_Failure` is not there, its default value is considered "NO".

Table 2-8 Pipeline Status When Fetch_Log_Upon_Failure = NO

Rerun Values	Rerun set to zero		Rerun set to non-zero		
Passed/Failed	**All Passed in Initial Run**	**Some Failed in Initial Run**	**All Passed in Initial Run**	**Some Passed in Initial Run, Rest Passed in Rerun**	**Some Passed in Initial Run, Some Failed Even After Rerun**
Status	SUCCESS	FAILURE	SUCCESS	SUCCESS	FAILURE

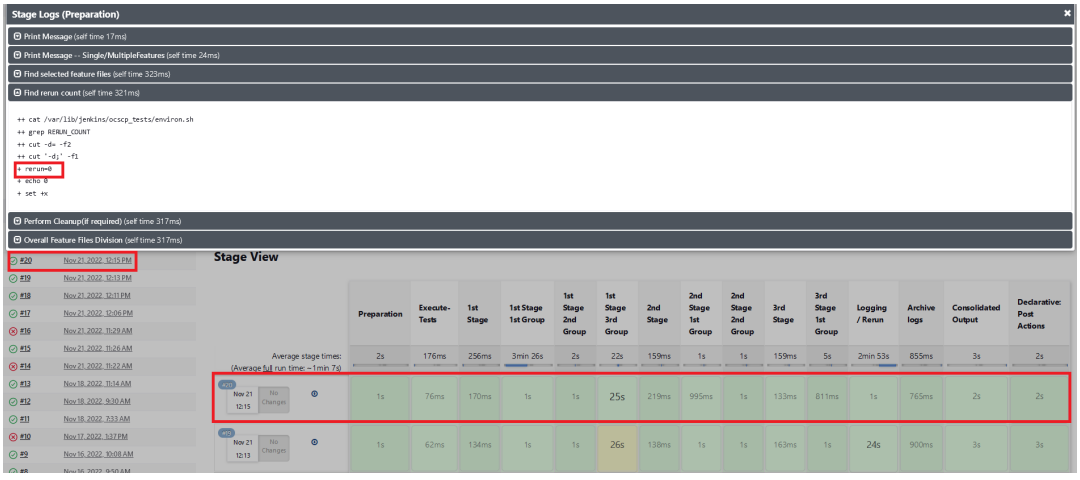
Table 2-9 Pipeline Status When Fetch_Log_Upon_Failure = YES

Rerun Values	Rerun set to zero			Rerun set to non-zero		
Passed/Failed	**All Passed in Initial Run**	**Some Failed in Initial Run and Failed in Rerun**	**Some Failed in Initial Run and Passed in Rerun**	**All Passed in Initial Run**	**Some Passed in Initial Run, Rest Passed in Rerun**	**Some Passed in Initial Run, Some Failed Even After Rerun**
Status	SUCCESS	FAILURE	SUCCESS	SUCCESS	SUCCESS	FAILURE

Some common combinations of these parameters, such as `rerun_count`, `Fetch_Log_Upon_Failure`, and `pass/fail status of test cases in initial and final run` and the corresponding build colors are as follows:

- When `Fetch_Log_Upon_Failure` is set to YES and `rerun_count` is set to 0, test cases pass in the initial run. The pipeline will be green, and its status will show as blue.

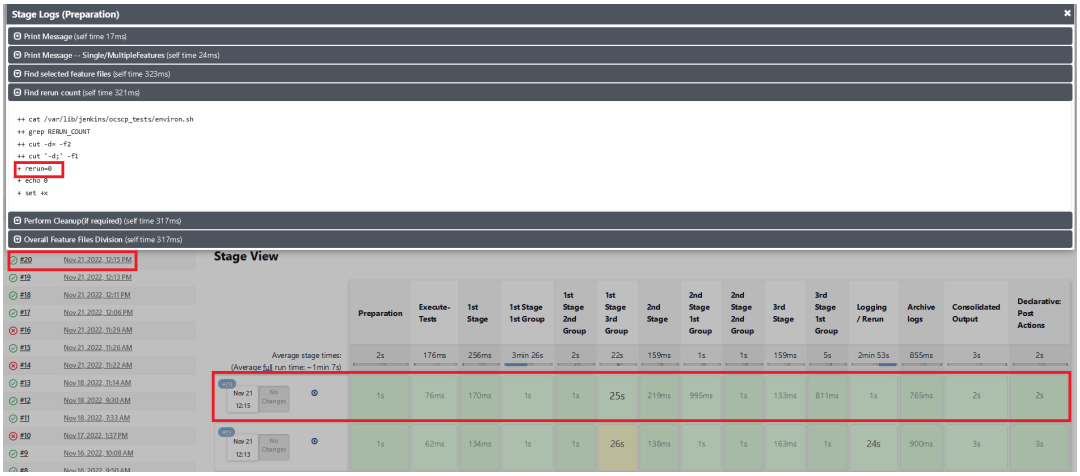
Figure 2-46 Fetch_Log_Upon_Failure is set to YES and rerun_count is set to 0, test cases pass



- When `Fetch_Log_Upon_Failure` is set to YES and `rerun_count` is set to 0, test cases fail on the initial run but pass during the rerun. The initial execution stage is yellow and all subsequent successful stages will be green, and the status will be blue.

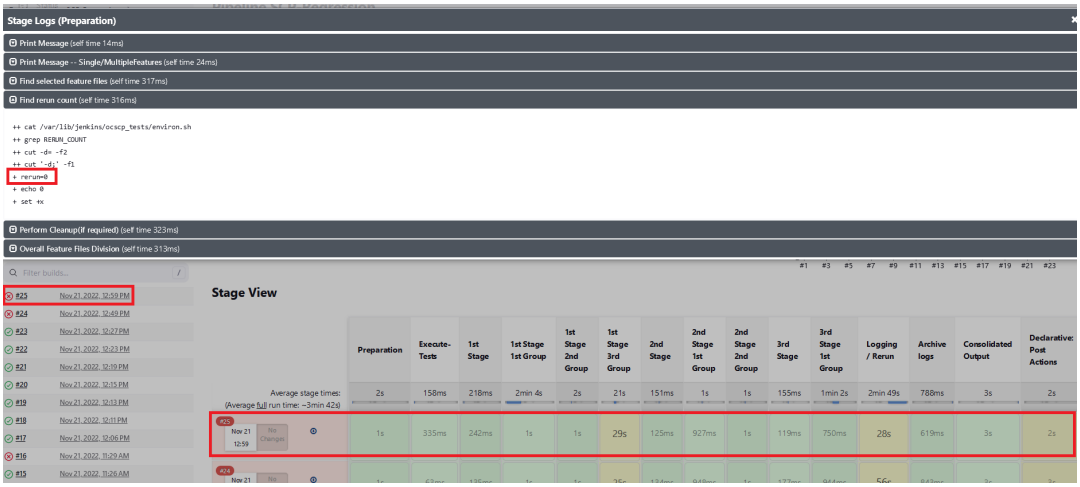
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-47 Test Cases Fail on the Initial Run but Pass in the Rerun



- When `Fetch_Log_Upon_Failure` is set to YES and `rerun_count` is set to 0, test cases fail in both the initial and the rerun. Execution stages will show as yellow, all other successful stages will be shown as green, and the overall pipeline status will be red.

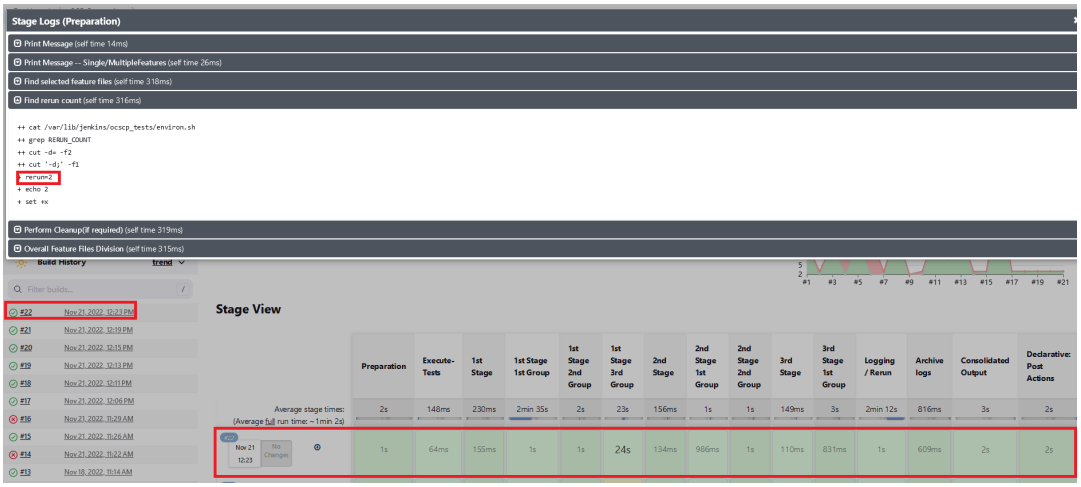
Figure 2-48 Test Cases Fail in Both the initial and the Rerun



- When `Fetch_Log_Upon_Failure` is set to YES and the `rerun count` is set to non-zero. If all of the test cases pass in the first run, no rerun will be initiated because the cases have already been passed. The pipeline will be green, and the status will be indicated in blue.

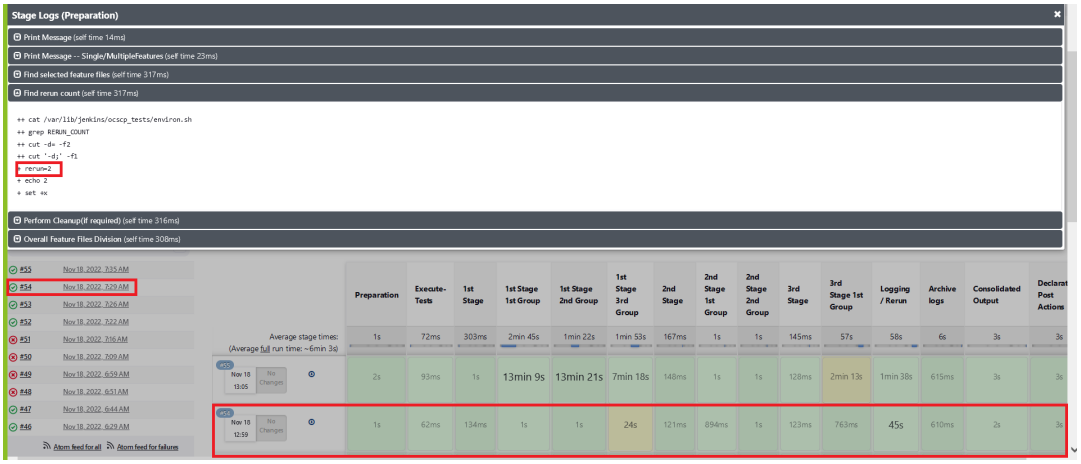
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-49 All of the Test cases Pass in the Initial Run



- When `Fetch_Log_Upon_Failure` is set to YES and the `rerun count` is set to non-zero. If some of the test cases fail in the initial run and the remaining ones pass in one of the remaining reruns, then the initial test case execution stages will show as yellow, the remaining stages as green, and the overall pipeline status as blue.

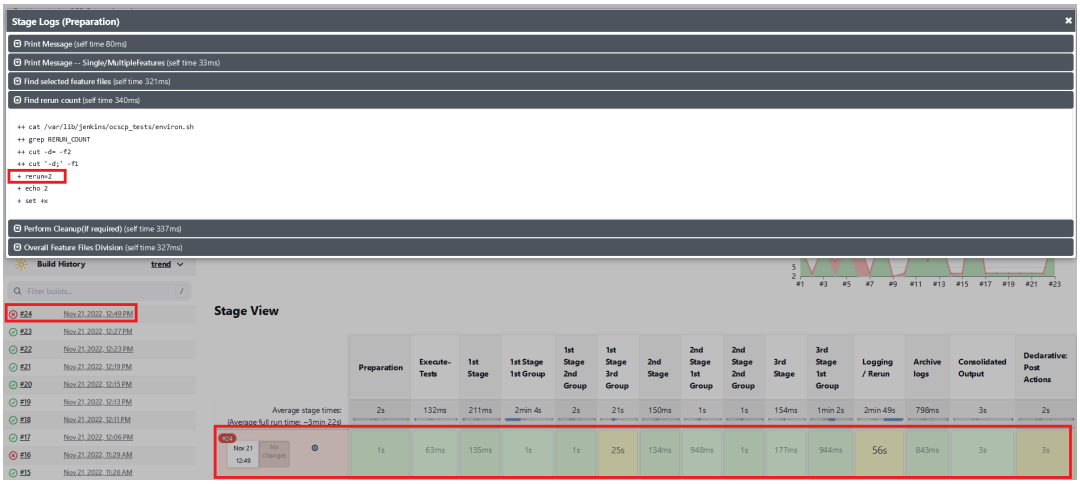
Figure 2-50 Test Cases Fail in the Initial Run and the Remaining Ones Pass



- When `Fetch_Log_Upon_Failure` is set to YES and the `rerun count` is set to non-zero. If some of the test cases fail in the initial run and the remaining ones fail in all the remaining reruns, the stages of test case execution will be shown in yellow, the remaining stages in green, and the overall pipeline status in red.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-51 Test Cases Fail in the Initial and Remaining Reruns



- Whenever any of the multiple Behave processes that are running in the ATS are exited without completion, the stage in which the process exited and the consolidated output stage are shown as yellow, and the overall pipeline status will be yellow. Also in the consolidated output stage, near the respective stage result, the exact run in which the Behave processes exited without completion will be printed.

Figure 2-52 Stage View When Behave Process is Incomplete

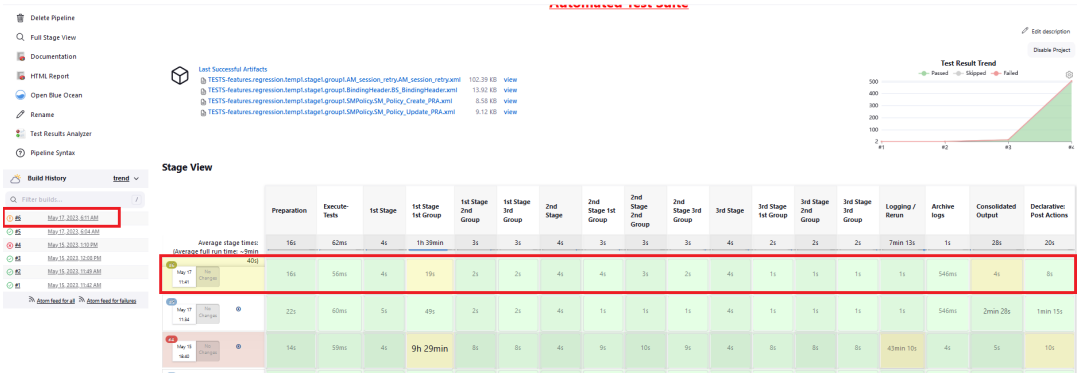
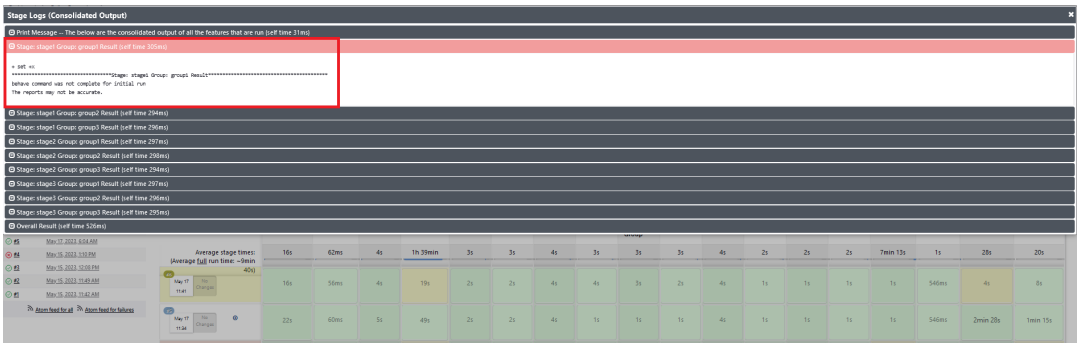


Figure 2-53 Consolidated Report for a Group When a Behave Process was Incomplete



Copyright © 2019, 2025, Oracle and/or its affiliates.

Implementing Application Log

ATS automatically fetches the SUT Debug logs during the rerun cycle if it encounters any failures and saves them in the same location as the build console logs. The logs are fetched for the rerun time duration only using the timestamps. If, for some microservices, there are no log entries in that time duration, it does not capture them. Therefore, the logs are fetched only for the microservices that have an impact or are associated with the failed test cases.

Location of SUT Logs: `/var/lib/jenkins/.jenkins/jobs/PARTICULAR-JOB-NAME/builds/BUILD-NUMBER/date-timestamp-BUILD-N.txt`

> **Note**
>
> The file name of the SUT log is added as a suffix with the date, timestamp, and build number (for which the logs are fetched). These logs share the same retention period as build console logs, set in the ATS configuration. It is recommended to set the retention period to optimal owing to the Persistent Volume Claim (PVC) storage space availability.
>
> .

2.16 Modifying Login Password

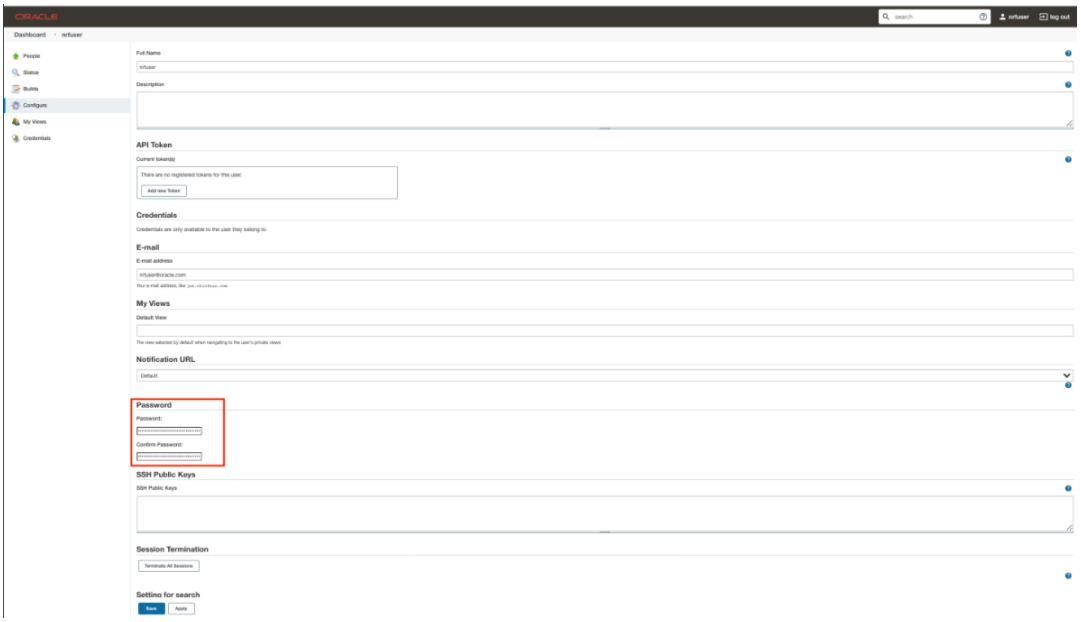
You can log in to the ATS application using the default login credentials. The default login credentials are shared for each NF in the respective chapter of this guide.

Perform the following procedure to modify the default password:

1. Log in to the ATS application using the default login credentials. The home page of the respective NF appears.
2. Click the down arrow next to the user name.
3. Click **Configure**.
4. In the Password section, enter the new password in the **Password** and **Confirm Password** fields..

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-54 Logged-in User Details



5. Click **Save**.
   A new password is set for you.

2.17 Multiselection Capability for Features and Scenarios

ATS allows you to select and run single or multiple features and scenarios by selecting a check box for the corresponding features or scenarios.

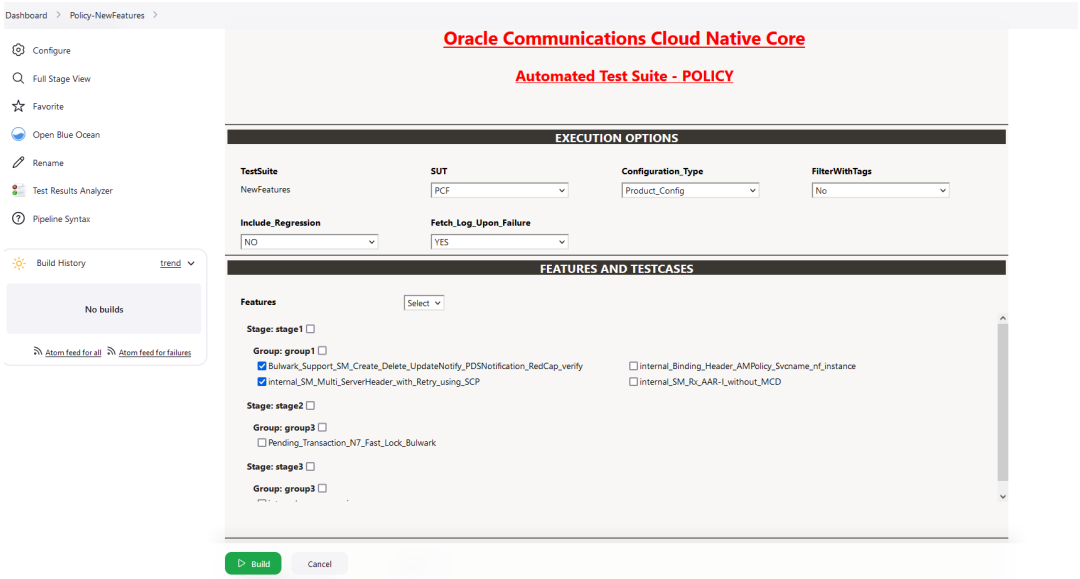
2.17.1 Feature Level Selection

Perform the following procedure to run the single or multiple features:

1. Log in to ATS using the respective <NF> login credentials.
2. On the NF home page, click any new feature or regression pipeline from where you want to run the feature.
3. In the left navigation pane, click **Build with Parameters**.
4. Scroll down to the FEATURES AND TEST CASES section.
5. Click **Select** from the **Features** drop-down.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-55 Feature Selection



6. Select any number of features by selecting the check box for the corresponding feature you want to run from the list available for execution.
7. Click **Build**.

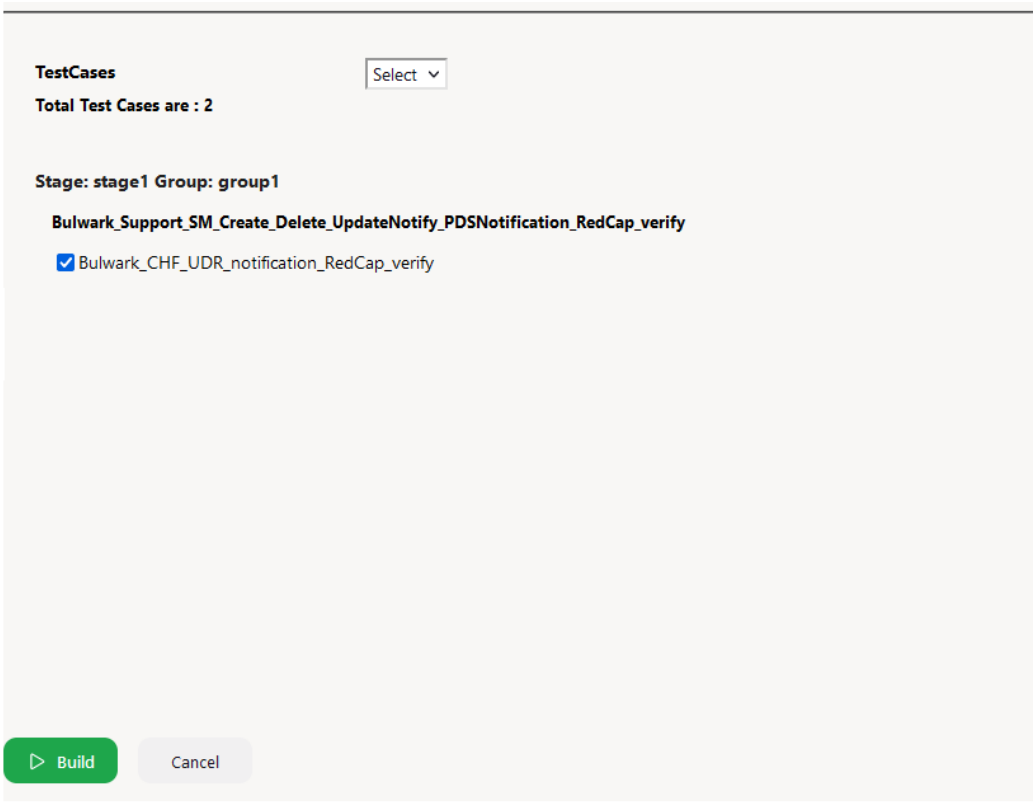
2.17.2 Scenario Selection

Perform the following procedure to run the single or multiple test cases or scenarios related to features:

1. Log in to ATS using the respective <NF> login credentials.
2. On the NF home page, click any new feature or regression pipeline from where you want to run the feature.
3. In the left navigation pane, click **Build with Parameters**.
4. Scroll down to the FEATURES AND TEST CASES section.
5. Click **Select** from the **Features** drop-down.
6. Select any number of features by selecting the check box for the corresponding feature to view the TestCases details mapped to each feature.
7. Click **Select** from the **TestCases** drop-down.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-56 Scenario or Testcase Selection



8. Select the check box for the corresponding test case or scenario to run the test cases mapped to the feature.
9. Click **Build**.

2.18 Parallel Test Execution

Parallel test execution allows you to perform multiple logically grouped tests simultaneously on the same System Under Test (SUT) to reduce the overall execution time of ATS.

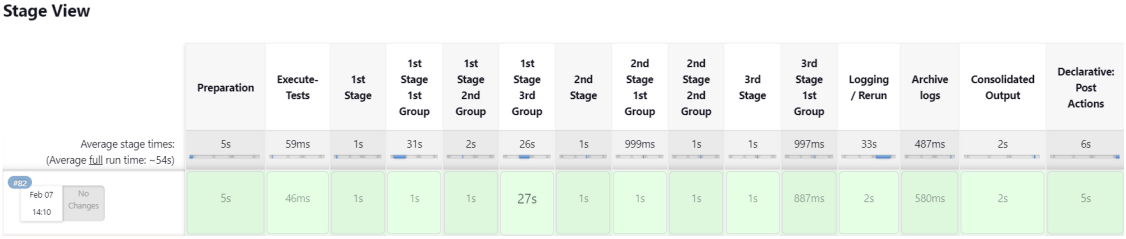
ATS currently runs all its tests in a sequential manner, which is time-consuming. With parallel test execution, tests can be run concurrently rather than sequentially or one at a time. Test cases or feature files are now separated into different folders, such as stages and groups, for concurrent test execution. Different stages, such as stage 1, stage 2, and stage 3, run the test cases in a sequential order, and each stage has its own set of groups. Test cases or feature files available in different groups operate in parallel. When all the groups within one stage have completed their execution, only then the next stage will start the execution.

Pipeline Stage View

The pipeline stage view appears as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

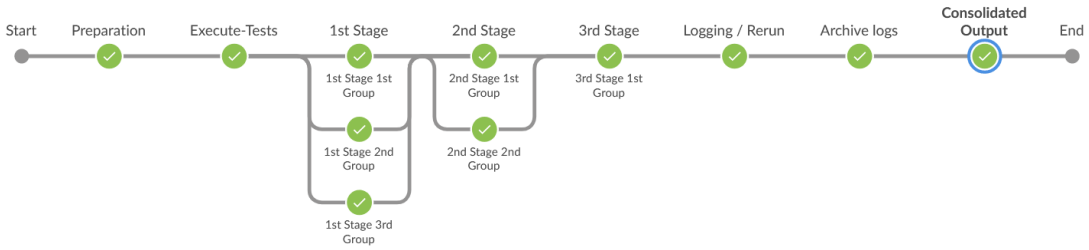
Figure 2-57 Pipeline Stage View



Pipeline Blue Ocean View

Blue Ocean is a Jenkins plugin that provides a better representation of concurrent execution with stages and groups. The pipeline blue ocean view appears as follows:

Figure 2-58 Pipeline Blue Ocean View



Impact on Other Framework Features

The integration of the parallel test framework feature has an impact on the following framework features. See the following sections for more details:

- Application Log Collection
- ATS API
- Managing Final Summary Report, Build Color, and Application Log
- PCAP Log Collection

2.18.1 ATS GUI Page Changes

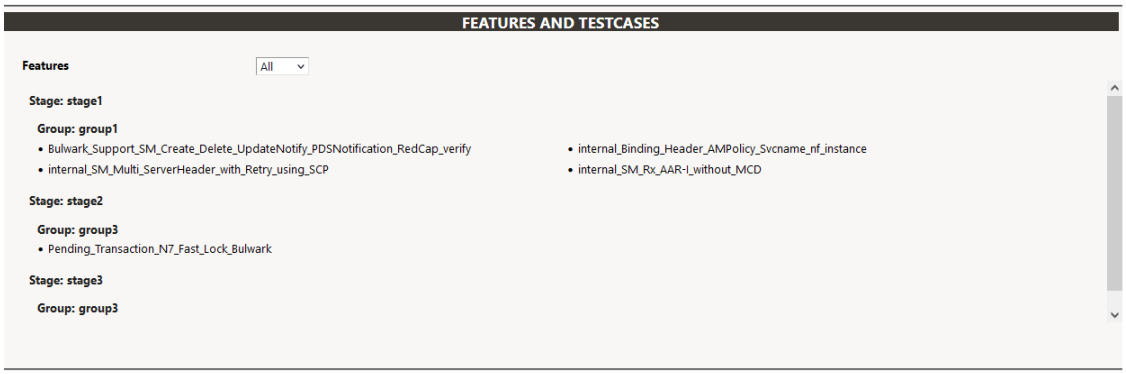
This section describes the changes to the ATS GUI page to trigger a build.

Changes in ATS GUI Page to Trigger a Build

The feature name, file name, and test case name are displayed under their stage and group names.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-59 Displays Stages and Groups



2.18.2 ATS Console Log Changes

This section describes the changes to the ATS console log. The ATS console log contains logs of all the stages and groups. For more details, see Downloading or Viewing Individual Group Logs.

- A test case's stage and group names are listed in the logger statements for that test case.

Figure 2-60 Logger Statement

```
2022-12-06 09:34:53,762[32m     INFO LOG.stage2.group1.STEP:5453 [0m| [32m10.233.109.13[0m
2022-12-06 09:34:53,762[32m     INFO LOG.stage2.group1.STEP:5454 [0m| [32mudm1svc.scpsvc.svc.galaxy[0m
2022-12-06 09:34:53,893[32m     INFO LOG.stage2.group1.STEP:5459 [0m| [32m{"url":"/USEast/nudm-uecm/v1/imsi
```

- When a test case fails, a list of test cases running in parallel gets printed to make the debugging easier. The name of the test case and the absolute path to the feature file it belongs to are listed in this list.

Figure 2-61 Absolute Path of Feature File

```
2022-12-06 11:27:19,253[32m     INFO LOG.stage1.group3.CleanUP:2317 [0m| [32mBelow feature(s)/scenario(s) were found
to be running in parallel at the time of failure:
{
    "feature_path": "/var/lib/jenkins/ocscp_tests/features/regression/release16features/allfeatures/stage1/group3
/SCP_Registration_With_PLMNList.feature",
    "scenario_name": "Static configuration to NRF for all nfType"
}
{
    "feature_path": "/var/lib/jenkins/ocscp_tests/features/regression/release16features/allfeatures/stage1/group1
/Version_support.feature",
    "scenario_name": "Scenario-1_To test Forward route for notification request UECM AMF Registration messages with
3gpp-Sbi-Target-apiRoot and 3gpp-sbi-routing-binding header bl=nfset, when the apiVersionInUri is set to v2 and the
message is also sent to v2"
}
```

- The test result summary contains a summary for each group and an overall summary, along with the details of failing scenarios (stage-groupwise) and the total time taken by any pipeline execution. For further information, see the Managing Final Summary Report, Build Color, and Application Log.

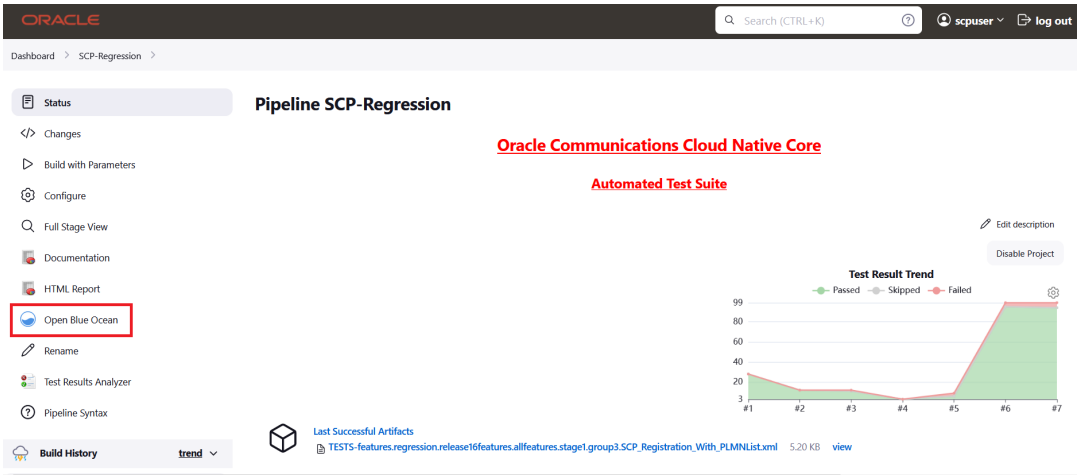
Copyright © 2019, 2025, Oracle and/or its affiliates.

2.18.3 Downloading or Viewing Individual Group Logs

To download individual group logs:

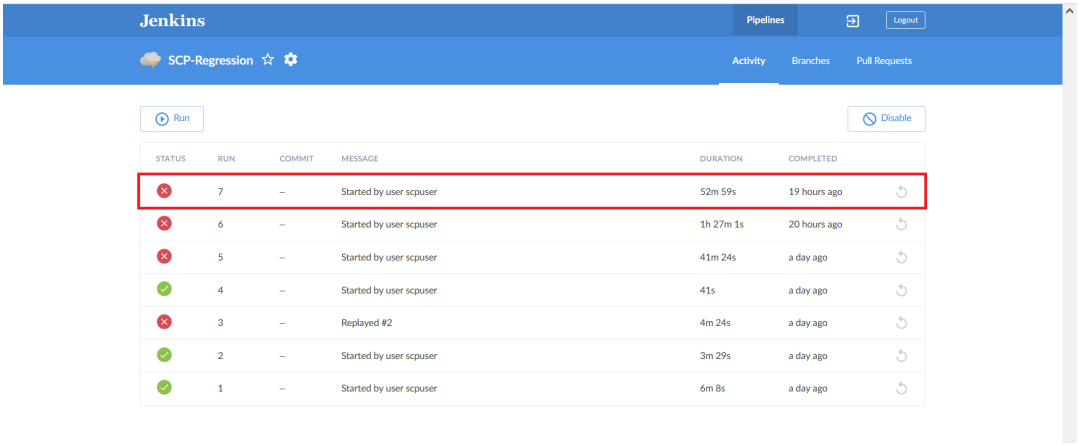
1. On the Jenkins pipeline page, click **Open Blue Ocean** in the left navigation pane.

Figure 2-62 Jenkins Pipeline Page



2. Click the desired build row on the Blue Ocean page.

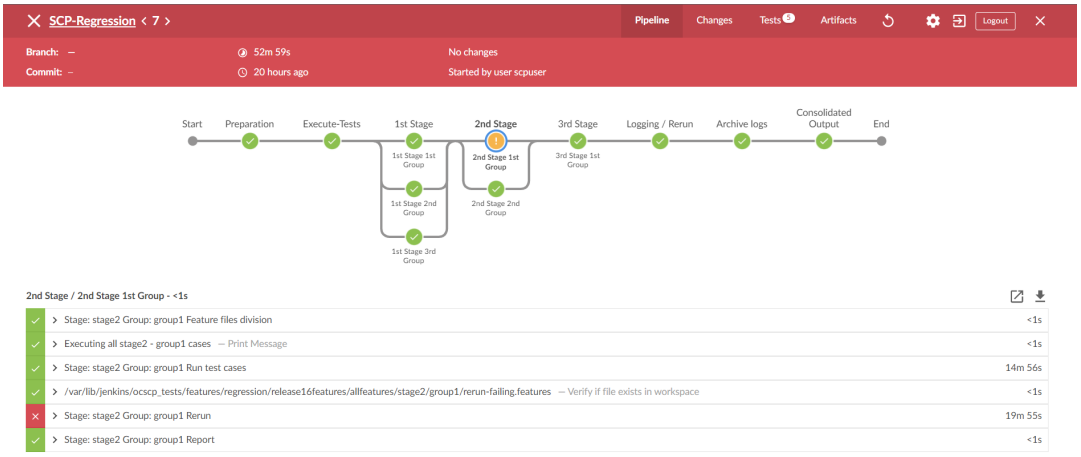
Figure 2-63 Run the Build



3. The selected build appears. The diagram displays the order in which the different stages, or groups, are executed.

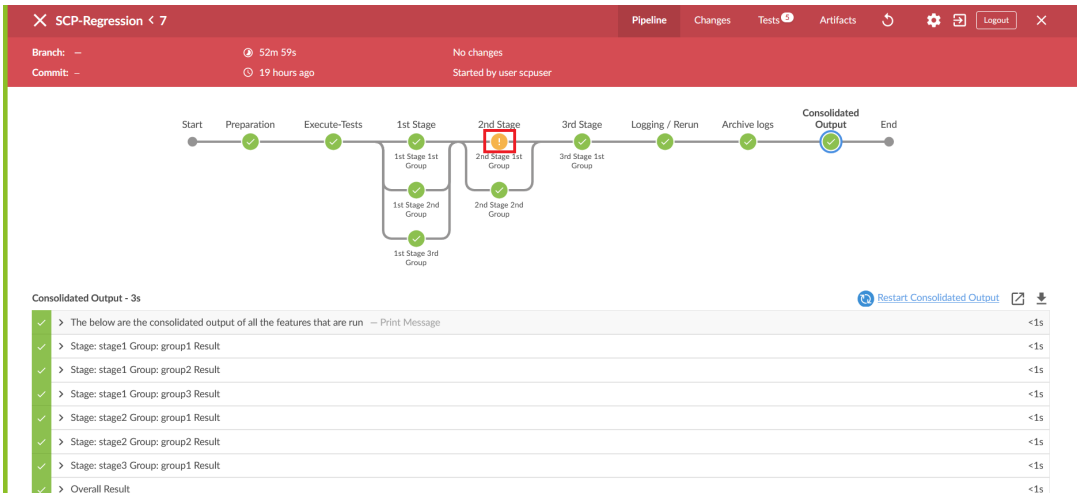
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-64 Stage Execution



4. Click the desired group to download the logs.

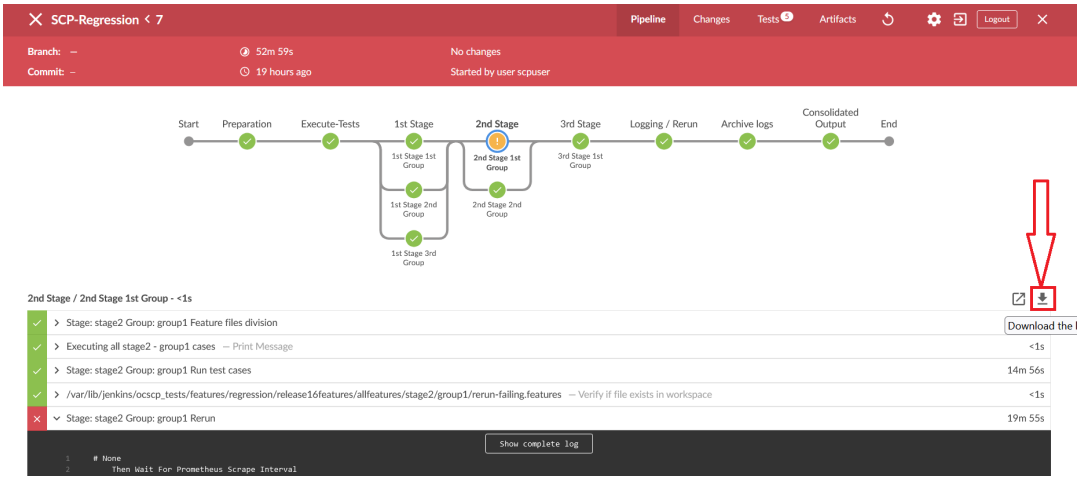
Figure 2-65 Executed Groups



5. Click the **Download** icon on the bottom right of the pipeline. The log for the selected group is downloaded to the local system.

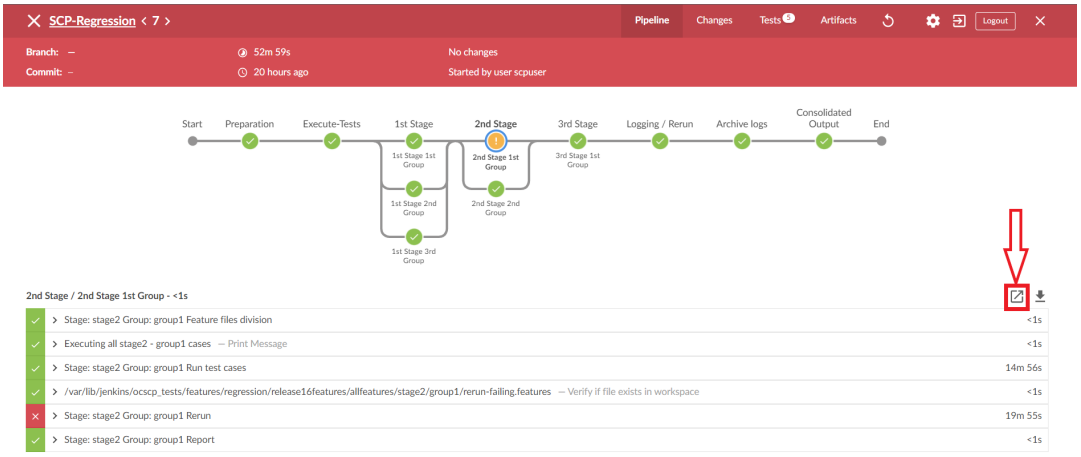
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-66 Download Logs



6. To view the log, click the **Display Log** icon. The logs are displayed in a new window.

Figure 2-67 Display Logs



Viewing Individual Group Logs without using Blue Ocean

There are two alternate ways to view individual group logs:

- Using Stage View
  1. On the Jenkins pipeline page, hover the cursor over the group in stage view to view the logs.
  2. Click the label **Logs**" which appears in a pop-up.
     There will be a new pop-up window. It contains many rows, where each row corresponds to the execution of one Jenkins step.
  3. Click the row labeled **Stage: stage_name>."Group: <group_name>** Run test cases to view the log for this group's execution.
  4. Click the row labeled **Stage: stage_name>." "group_name> Rerun** to display the rerun logs.
- Using Pipeline Steps Page

Copyright © 2019, 2025, Oracle and/or its affiliates.

1. On the Jenkins pipeline page, click the desired build number under the **Build History** drop down.
2. Click the **Pipeline Steps** button on the left pane.
   A table with columns for step, arguments, and status appears.
3. Under the Arguments column, find the label for the desired stage and group.
4. Click the step with the label **Stage: <stage_name>Group: <group_name>**.
5. Run test cases under it, or click the **Console output** icon near the status to view the log for this group execution.
6. To see rerun logs, find the step with the label **Stage: <stage_name> Group: <group_name>Rerun** under it.

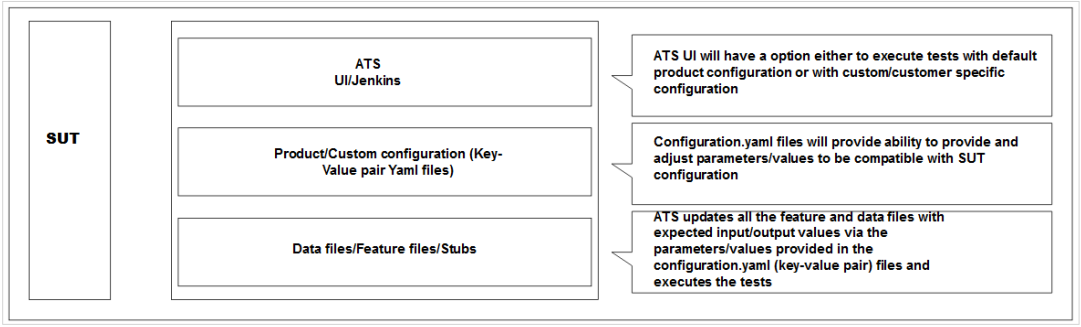
2.19 Parameterization

This feature allows you to provide or adjust values for the input and output parameters needed for the test cases to be compatible with the SUT configuration. You can update or adjust the key-value pair values in the `global.yaml` and `feature.yaml` files for each of the feature files so that they are compatible with SUT configuration. In addition to the existing custom test case folders (Cust New Features, Cust Regression, and Cust Performance), this feature enables folders to accommodate custom data, default product configuration, and custom configuration. You can maintain multiple versions or copies of the custom data folder to suit varied or custom SUT configurations. With this feature, the ATS GUI has the option to either execute test cases with the default product configuration or with a custom configuration.

This feature enables you to perform the following tasks:

- Define parameters and assign or adjust values to make them compatible with SUT configuration.
- Execute test cases either with default product configurations or custom configurations and multiple custom configurations to match varied SUT configurations.
- Assign or adjust values for input or output parameters through custom or default configuration yaml files (key-value pair files).
- Define or adjust the input or output parameters for each feature file with its corresponding configuration.
- Create and maintain multiple configuration files to match multiple SUT configurations.

Figure 2-68 SUT Design Summary

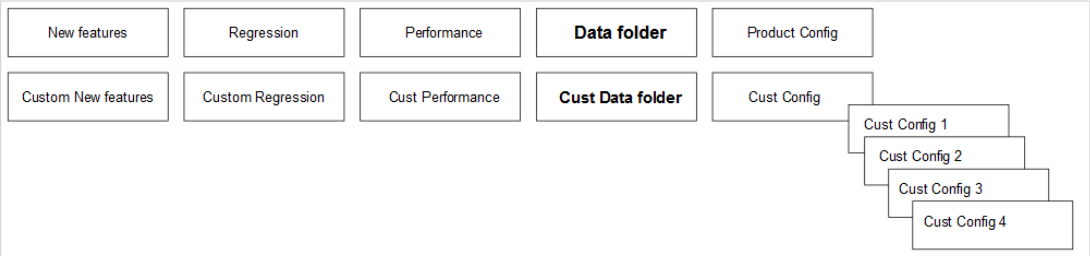


In the folder structure:

Copyright © 2019, 2025, Oracle and/or its affiliates.

- The Product Config folder contains default product configuration files (feature-wise yaml per key-value pair), which are compatible with default product configuration.
- New features, Regression and Performance, Data folder, and Product Config folders are replicated or copied into custom folders and delivered as part of the ATS package in every release.
- You can customize custom folders by:
  - Removing test cases not needed or as appropriate for your use.
  - Adding new test cases as needed or as appropriate for your use.
  - Removing or adding data files in the cust_data folder or as appropriate for your use.
  - Adjusting the parameters or values in the key-value pair per yaml file in the custom config folder for test cases to run or pass with a custom configured SUT.
- The product folders are always intact (unchanged) and you can update the Custom folders
- You can maintain multiple copies of Custom Configurations and bring them to use as needed or as appropriate for the SUT configuration.

Figure 2-69 Folder Structure



2.19.1 Running Test Cases

Enable

Rename or copy the Cust Config [1/2/3/N] folder to the Cust Config folder in order for ATS to run the test cases with a specific custom configuration. When the option to run test cases with custom configuration is chosen, it always points to the Cust Config folder.

To Run ATS Test Cases

ATS has the option to run test cases with default or custom configuration.

- If custom configuration is selected, then test cases from custom folders are populated on the ATS UI, and custom configuration is applied to them through the key-value pair per yaml file defined or present in the "Cust Config" folder.
- If product configuration is selected, then the test cases from product folders are populated on the ATS UI, and product configuration is applied to them through key-value pairs per yaml file defined or present in the Product Config folder.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-70 ATS Execution Flow

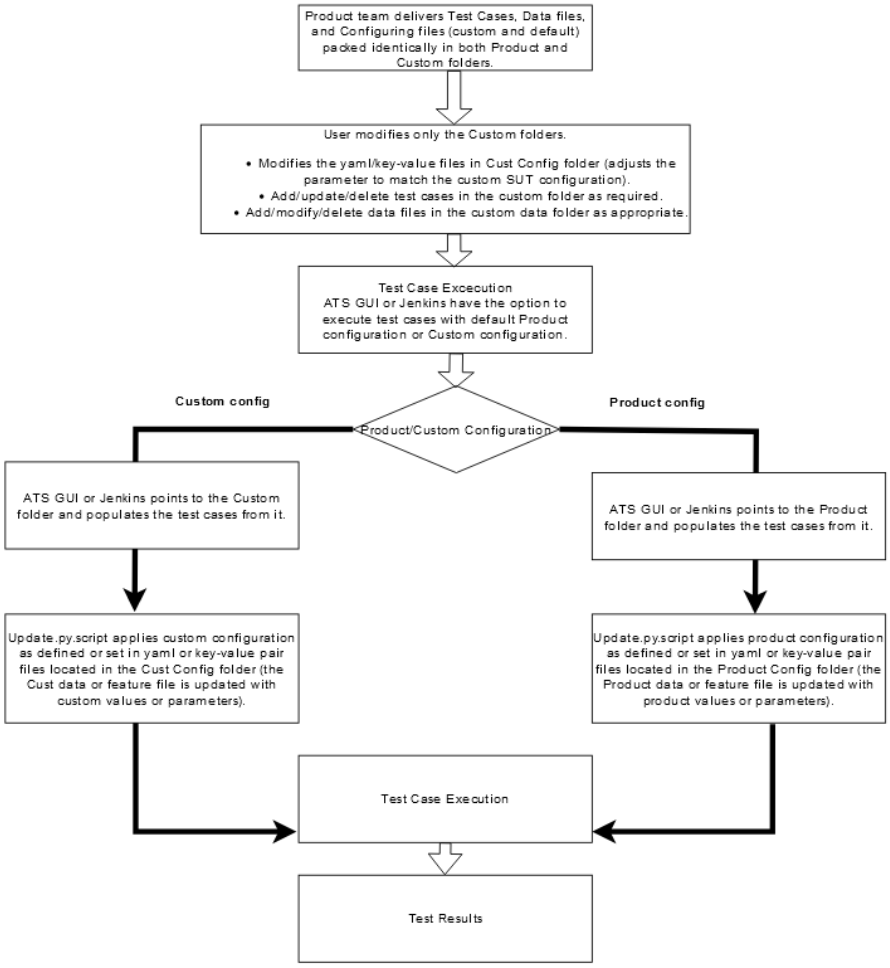
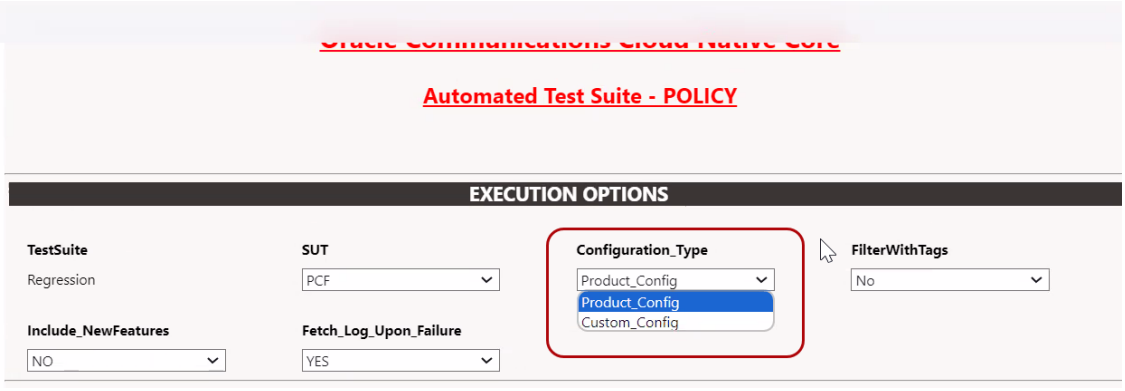


Figure 2-71 Sample: Configuration_Type



Copyright © 2019, 2025, Oracle and/or its affiliates.

2.20 PCAP Log Collection

PCAP Log Collection allows collecting the NF, SUT, or PCAP logs from the debug tool sidecar container. This feature can be integrated and delivered as a standalone or along with the Application Log Collection feature. For information, see Application Log Collection.

PCAP Log Integration

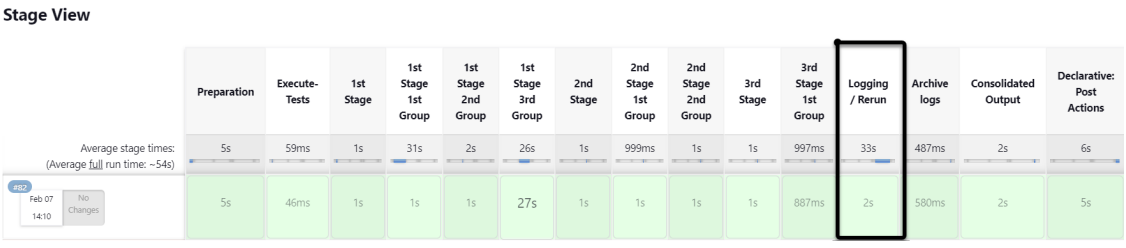
1. The Debug tool should be enabled on SUT Pods while deploying the NF. The name of the Debug container must be "tools".
   For example, in SCP, the debug tool should be enabled for all the SCP microservice pods.
2. Update the following parameters in the values.yaml file, under the resource section, with ATS minimum resource requirements:
   a. CPU: 3
   b. memory: 3Gi
3. On the home page, click any new feature or regression pipeline.
4. In the left navigation pane, click **Build with Parameters**.
5. Select **YES** from the **Fetch_Log_Upon_Failure** drop-down menu.
6. If option **Log_Type** is available, select value **PcapLog [Debug Container Should be Running]** for it.
7. Select **PcapLog [Debug Container Should be Running]** to activate PCAP Log Collection in ATS-NF.
8. After the build execution is complete, go into the ATS pod, then navigate to below path to find the pcaplogs: `.jenkins/jobs/<Pipeline Name>/builds/<build number>/`
   For example, `.jenkins/jobs/SCP-Regression/builds/5/`

   Pcaplogs is present in zip form. Unzip it to get the log files.

2.20.1 Application Log Collection and Parallel Test Execution Integration

A new stage,"Logging/Rerun", has been added at the end of the Execute-Tests stage to collect rerun logs, such as applog and PCAP logs, by running the failed test cases in a sequence.

Figure 2-72 Logging/Rerun new stage



If the `Fetch_Log_Upon_Failure` parameter is set to YES and if any test case fails in the initial run, then:

- The failed test case reruns and log collection start in the Logging/Rerun stage after the initial run is completed for all the test cases.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- The logs from the initial execution are collected, but they might be incorrect.
- Even if the `rerun` parameter is set to 0, the failed test case reruns in the Logging/Rerun stage and the log is collected.

> **Note**
>
> Not applicable for all the NFs.

- If the `Fetch_Log_Upon_Failure` parameter is set to NO and if any test case fails in the initial run, then the failed test case rerun starts in the same stage after the initial execution is over for all the test cases in its group.

2.21 Persistent Volume for 5G ATS

The Persistent Volume (PV) feature allows ATS to retain historical build execution data, test cases, and ATS environment configurations.

ATS Packaging When Using Persistent Volume

- **Without the Persistent Volume option:** ATS package includes an ATS image with test cases.
- **With Persistent Volume option:** ATS package includes the ATS image and test cases separately. The new test cases are provided between the releases.
  To support both with and without Persistent Volume options, test cases and execution job data are packaged in the ATS image as well as a tar file.

2.21.1 Processing Flow

First Time Deployment

Initially, when you deploy ATS, for example, the PI-A ATS pod, you use PVC-A, which is provisioned and mounted to the PI-A ATS pod. By default, PVC-A is empty. So, you have to copy the data (ocslf_tests and jobs folders) from the PI-A tar file to the pod after the pod is up and running. Then restart the PI-A pod. At this point, you can change the number of build logs to maintain in the ATS GUI.

Subsequent Deployments

When you deploy ATS for the subsequent time, for example, in a PI-B ATS pod, you use PVC-B, which is provisioned and mounted to the PI-B ATS pod. By default, the PVC-B is empty, and you have to copy the data (ocslf_tests and jobs folders) from the PI-B tar file to the pod after the pod is up and running. At this point, copy all the necessary changes to the PI-B pod from the PI-A pod and restart the PI-B pod. You can change the number of build logs to maintain in the ATS GUI. After updating the number of builds, you can delete the PI-A pod and continue to retain the PVC-A. If you do not want backward porting, you can delete PVC-A.

2.21.2 Deploying Persistent Volume

Preinstallation Steps for Non-OCI setup

1. Before deploying Persistent Volume, create a PVC in the same namespace where you have deployed ATS. You have to provide values for the following parameters to create a PVC:
   - PVC Name

Copyright © 2019, 2025, Oracle and/or its affiliates.

- Namespace
- Storage Class Name
- Size of the PV

2. Run the following command to create a PVC:

```
kubectl apply -f - <<EOF

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: <Please Provide the PVC Name>
  namespace: <Please Provide the namespace>
  annotations:
spec:
  storageClassName: <Please Provide the Storage Class Name>
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: <Please Provide the size of the PV>
EOF
```

> **Note**
>
> It is recommended to suffix the PVC name with the release version to avoid confusion during subsequent releases. For example: ocats-slf-pvc-1.9.0

The output of the above command with parameters is as follows:

```
[cloud-user@atscne-bastion-1 templates]$ kubectl apply -f - <<EOF
>
> apiVersion: v1
> kind: PersistentVolumeClaim
> metadata:
>     name: ocats-slf-1.9.0-pvc
>     namespace: ocslf
>     annotations:
> spec:
>     storageClassName: standard
>     accessModes:
>         - ReadWriteOnce
>     resources:
>         requests:
>             storage: 1Gi
> EOF
```

The persistentvolumeclaim/ocats-slf-1.9.0-pvc is created.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3. To verify whether PVC is bound to PV and is available for use, run the following command:

   ```
   kubectl get pvc -n <namespace used for pvc creation>
   ```

   The output of the above command is as follows:

   Figure 2-73 Verifying PVC

   ```
   [cloud-user@atscne-bastion-1 templates]$ kubectl get pvc -n ocslf
   NAME                 STATUS   VOLUME                                     CAPACITY   ACCESS MODES STORAGECLASS   AGE
   ocats-slf-1.9.0-pvc  Bound    pvc-1ee50daf-2eb7-4289-b541-51c111270513   1Gi        RWO          standard       8s
   policy-pvc-test      Bound    pvc-d346d5c4-129e-4f13-8bd0-a7b7555e0e2f   1Gi        RWO          standard       31d
   slf-pvc              Bound    pvc-0b0bce8c-b856-4e0e-b4d8-0f981250adbd   1Gi        RWO          standard       12d
   ```

   Check that the STATUS is **Bound** and that the rest of the parameters, such as NAME, CAPACITY, ACCESS MODES, STORAGECLASS, and so on, are the same as specified in the PVC creation command.

   > **Note**
   >
   > Do not proceed further if there is any issue with PVC creation. Contact your administrator to create a PV.

4. After creating persistent volume, change the following parameters in the values.yaml file to deploy persistent volume.
   - Set the **PVEnabled** parameter to "true".
   - Provide the value for the **PVClaimName** parameter. The PVClaimName value should be the same as the value used to create a PVC.

Preinstallation Steps for OCI setup

1. Before deploying persistent volumes, create a PVC in the same namespace where you have deployed ATS. You have to provide values for the following parameters to create a PVC:
   - PVC Name
   - Namespace
   - Storage Class Name
   - Size of the PV
2. Run the following command to create a PVC:

   ```
   kubectl apply -f - <<EOF

   apiVersion: v1
   kind: PersistentVolumeClaim
   metadata:
     name: <Please Provide the PVC Name>
     namespace: <Please Provide the namespace>
     annotations:
   spec:
     storageClassName: <Please Provide the Storage Class Name>
     accessModes:
       - ReadWriteOnce
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  resources:
    requests:
      storage: <Please Provide the size of the PV>
EOF
```

3. When creating Persistent Volume Claim in OCI, add the storageClassName "oci-bv" and maintain the 50 gigabyte storage value.

> **Note**
>
> The minimum allowed value for parameter "resources.requests.storage" is 50Gi. If the storage is set to any value less than 50Gi, it will be ignored.

For example, the output of the above command with parameters is as follows:

```
[cloud-user@atscne-bastion-1 templates]$ kubectl apply -f - <<EOF
>
> apiVersion: v1
> kind: PersistentVolumeClaim
> metadata:
>     name: ocats-slf-1.9.0-pvc
>     namespace: ocslf
>     annotations:
> spec:
>     storageClassName: oci-bv
>     accessModes:
>         - ReadWriteOnce
>     resources:
>         requests:
>             storage: 50Gi
> EOF
```

The persistentvolumeclaim/ocats-slf-1.9.0-pvc is created.

4. To verify whether PVC is bound to PV and is available for use, run the following command:

```
kubectl get pvc -n <namespace used for pvc creation>
```

The output of the above command is as follows:

Figure 2-74 Verifying PVC

```
[opc@operatorinstance Scripts]$ kubectl get pvc -n ocsepp3
NAME                 STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
ocats-sepp-24.1.0-pvc   Bound    csi-48a73509-c08a-4ff1-ba2e-c2a6e1245cd9   50Gi       RWO            oci-bv         5m16s
```

5. After creating persistent volume, the STATUS will be Pending and when the ATS pod is deployed with that PVC, its status will become "BOUND".
6. Change the following parameters in the values.yaml file to deploy persistent volume.
   - Set the **PVEnabled** parameter to "true".
   - Provide the value for the **PVClaimName** parameter. The PVClaimName value should be the same as the value used to create a PVC.

Postinstallation Steps

Copyright © 2019, 2025, Oracle and/or its affiliates.

1. After deploying ATS, copy the <nf_main_folder> and <jenkins jobs> folders from the tar file to their ATS pod, and then restart the pod as a one-time activity.
   a. Run the following command to extract the tar file:

      ```
      ocats-<nf_name>-data-<release-number>.tgz
      ```

      > **Note**
      >
      > The ats_data.tar file is the name of the tar file containing <nf_main_folder> and jobs folders. It can be different for different NFs.

   b. Run the following set of commands to copy the required folders:

      ```
      kubectl cp ats_data/jobs <namespace>/<pod-name>:/var/lib/
      jenkins/.jenkins/
      ```

      ```
      kubectl cp ats_data/<nf_main_folder> <namespace>/<pod-name>:/var/lib/
      jenkins/
      ```

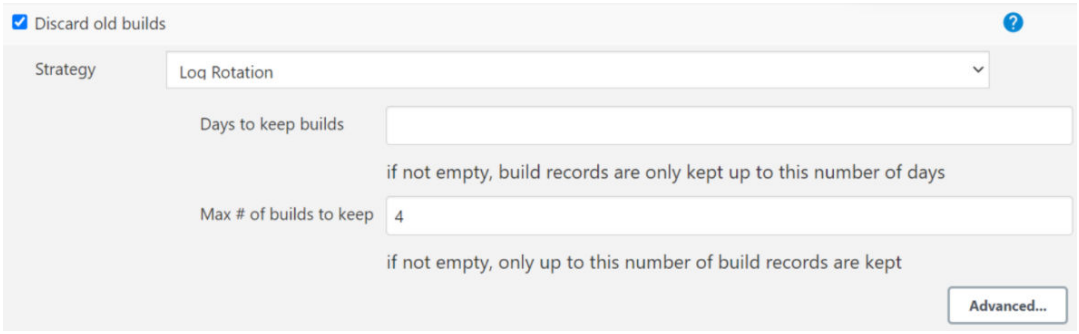
   c. Run the following command to restart the pods as one-time activity:

      ```
      kubectl delete po <pod-name> -n <namespace>
      ```

      > **Note**
      >
      > Before running the following command, copy the changes done on the new release pod from the old release pod using the `kubectl cp` command. **[Applicable in the case of subsequent deployment only]**

2. When the pod is up and running, log in to the ATS GUI and go to your NF specific pipeline. Click **Configure** in the left navigation pane. The **General** tab appears. Configure the **Discard old Builds** option. This option allows you to configure the number of builds you want to retain in the persistent volume.

Figure 2-75 Discard Old Builds



Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> It is recommended to configure this option. If you do not input a value for this option, the application will take into account all builds, which could be a large number, and will completely consume the Persistent Volume.

2.21.3 Backward Porting

The following deployment steps apply to the old release of PVC-supported ATS Pod.

Prerequisite: You should have the old PVC that contains the old release of POD data.

> **Note**
>
> This procedure is for backward porting purposes only and should not be considered a subsequent release of the POD deployment procedure.

The deployment procedure for the old release PVC-supported ATS pod is the same; however, while deploying the ATS pod, you have to update the values.yaml file with the following:

- Change the **PVEnabled** parameter to "true".
- Provide the name of the old PVC as the value for parameter **PVClaimName**.

2.22 Single Click Job Creation

With the help of Single Click Job Creation feature, ATS users can easily create a job to run TestSuite with a single click.

2.22.1 Configuring Single Click Job

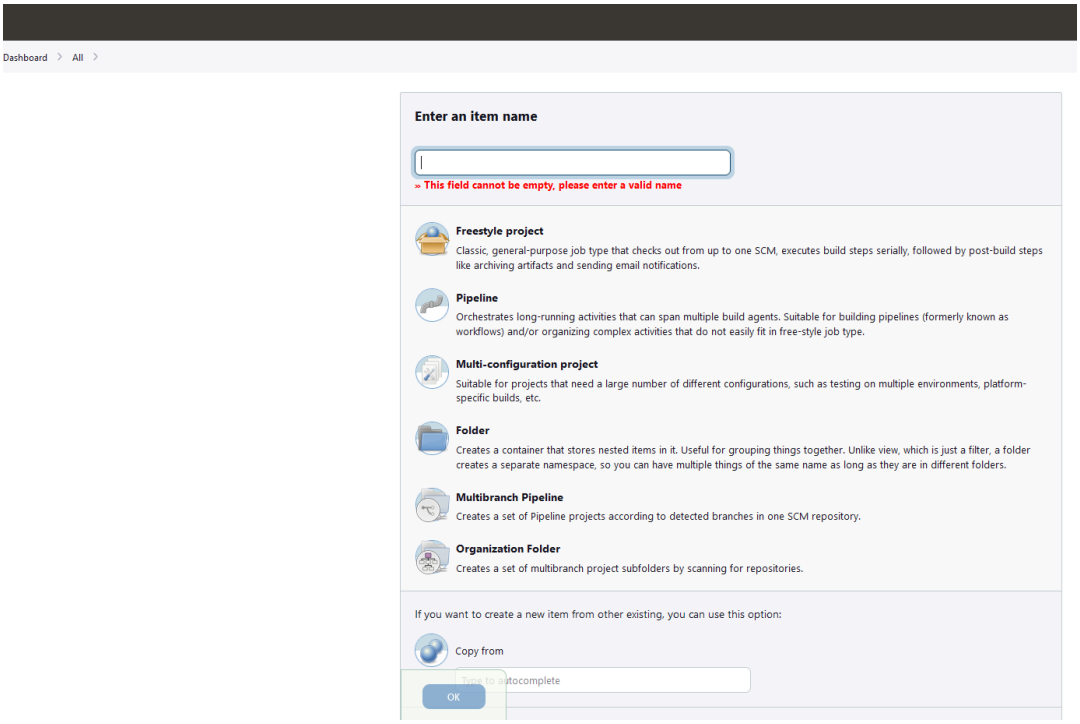
Prerequisite: The network function specific user should have 'Create Job' access.

Perform the following procedure to configure the single-click feature:

1. Log in to ATS using network function specific log-in credentials..
2. Click **New Item** in the left navigation pane of the ATS application. The following page appears:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-76 New Item Window



3. In the **Enter an item name** text box, enter the job name. Example: <NF-Specific-name>-NewFeatures.
4. In the **Copy from** text box, enter the actual job name for which you need single-click execution functionality. Example: <NF-Specific-name>-NewFeatures.
5. Click **OK**. You are automatically redirected to edit the newly created job's configuration.
6. Under the **General** group, deselect the **This Project is Parameterised** option.
7. Under the **Pipeline** group, make the corresponding changes to remove the 'Active Choice Parameters' dependency.
8. Provide the default values for the **TestSuite**, **SUT**, **Select_Option**, **Configuration_Type**, and other parameters, as required, on the **BuildWithParameters** page.
   Example: Pipeline without Active Choice Parameter Dependency

```
node ('built-in'){
    //a = SELECTED_NF    b = PCF_NAMESPACE        c = PROMSVC_NAME       d
= GOSTUB_NAMESPACE
    //e = SECURITY       f = PCF_NFINSTANCE_ID   g = POD_RESTART_TIME   h
= POLICY_TIME
    //i = NF_NOTIF_TIME  j = RERUN_COUNT         k =INITIALIZE_TEST_SUITE
l = STUB_RESPONSE_TO_BE_SET
    //m = POLICY_CONFIGURATION_ADDITION          n = POLICY_ADDITION
o = NEXT_MESSAGE
    //p = PROMSVCIP     q = PROMSVCPORT         r = TIME_INT_POD_DOWN    s
= POD_DOWN_RETRIES
    //t = TIME_INT_POD_UP   u = POD_UP_RETRIES  v = ELK_WAIT_TIME   w =
ELK_HOST
    //x = ELK_PORT  y = STUB_LOG_COLLECTION z = LOG_METHOD A =
enable_snapshot B = svc_cfg_to_be_read C = PCF_API_ROOT
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    //Description of Variables:

    //SELECTED_NF : PCF
    //PCF_NAMESPACE : PCF Namespace
    //PROMSVC_NAME : Prometheus Server Service name
    //GOSTUB_NAMESPACE : Gostub namespace
    //SECURITY : secure or unsecure
    //PCF_NFINSTANCE_ID : nfInstanceId in PCF application-config config map
    //POD_RESTART_TIME : Greater or equal to 60
    //POLICY_TIME : Greater or equal to 120
    //NF_NOTIF_TIME : Greater or equal to 140
    //RERUN_COUNT : Rerun failing scenario count
    //TIME_INT_POD_DOWN : The interval after which we check the POD status
if its down
    //TIME_INT_POD_UP : The interval after which we check the POD status
if its UP
    //POD_DOWN_RETRIES : Number of retry attempt in which will check the
pod down status
    //POD_UP_RETRIES : Number of retry attempt in  which will check the
pod up status
    //ELK_WAIT_TIME : Wait time to connect to Elastic Search
    //ELK_HOST : Elastic Search HostName
    //ELK_PORT : Elastic Search Port
    //STUB_LOG_COLLECTION : To Enable/Disable Stub logs collection
    //LOG_METHOD : To select Log collection method either elasticsearch or
kubernetes
    //enable_snapshot: Enable or disable snapshots that are created at the
start and restored at the end of each test run
    //svc_cfg_to_be_read: Timer to wait for importing service
configurations
    //PCF_API_ROOT: PCF_API_ROOT information to set Ingress gateway
service name and port
    withEnv([
    'TestSuite=NewFeatures',
    'SUT=PCF',
    'Select_Option=All',
    'Configuration_Type=Custom_Config'
    ]){
    sh '''
        sh /var/lib/jenkins/ocpcf_tests/preTestConfig-NewFeatures-PCF.sh \
        -a PCF \
        -b ocpcf \
        -c occne-prometheus-server \
        -d ocpcf \
        -e unsecure \
        -f fe7d992b-0541-4c7d-ab84-c6d70b1b0123 \
        -g 60 \
        -h 120 \
        -i 140 \
        -j 2 \
                -k 0 \
                -l 1 \
                -m 1 \
                -n 15 \
                -o 1 \
                -p occne-prometheus-server.occne-infra\
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
                    -q 80\
                    -r 30\
                    -s 5\
                    -t 30\
                    -u 5\
                    -v 0\
                    -w occne-elastic-elasticsearch-master.occne-infra\
                    -x 9200\
                    -y yes\
                    -z kubernetes\
                    -A no\
                    -B 15\
                    -C ocpcf-occnp-ingress-gateway:80\

    '''
    load "/var/lib/jenkins/ocpcf_tests/jenkinsData/Jenkinsfile-Policy-
NewFeatures"
   }
}
```

9. Click **Save**. The ATS application is ready to run TestSuite with 'SingleClick' using the newly created job.

2.23 Support for ATS Deployment in OCI

Oracle Cloud Infrastructure (OCI) is a set of complementary cloud services that enable you to build and run a range of applications and services in a highly-available, consistently high-performance environment. OCI offers powerful computing capabilities and storage capacity in a flexible virtual network that can be accessed from your on-premises network.

OCI infrastructure consists of Compartments, Network Load balancer, Bastion Host, Dynamic Routing Gateway (DRG), Remote Peering Connection (RPC), Service Gateway, Internet Gateway, and Oracle Kubernetes Engine (OKE) cluster, all in an OKE VCN.

ATS deployment is also supported in OCI.

2.23.1 Accessing ATS GUI in OCI

The following two ways are supported to access the ATS GUI in OCI:

Using Loadbalancer IP

1. Add proper Ingress or Egress security rules for ports (8080/8443) and ATS service Node port in loadbalancer (nf_lb_subnet) and Node subnet (nf_node_subnet). To add ingress and egress security rules, see the Adding Ingress and Egress Rules to Access the OCI Console.
2. Insert the following annotations under the **Metadata** section in ATS service object to assign an external IP (Loadbalancer IP):

```
oci-network-load-balancer.oraclecloud.com/security-list-management-mode:
None
oci.oraclecloud.com/load-balancer-type: nlb
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> This can be done by editing and saving ATS service object after ATS deployment: For example,
>
> ```
> kubectl edit svc ats-service-name -n ats-namespace
> ```

3. Access the GUI using URL: `<http/https>://<Loadbalancer IP>:<8080/8443>`

> **Note**
>
> - To open ATS GUI with https (TLS enabled), Load balancer IP must be known before deployment and listed down in the alt_names section of the ssl.conf file while generating the application or client certificate. Otherwise, the ATS GUI will open, but it will show "Not Secure" in the browser address bar. For more information, see the Support for Transport Layer Security section.
> - The assignment of Loadbalancer IP to the ATS service is subject to availability. If the Loadbalancer IP is not assigned to the ATS service even after applying the required annotations, try to debug on the OCI side.

Using Tunneling

1. Add an ingress security rule for the node subnet (nf_node_subnet) to allow TCP traffic on all ports from the operator subnet. To add ingress and egress security rules, see the Adding Ingress and Egress Rules to Access the OCI Console section.
2. Run the following command from a bash terminal on your local PC:

   ```
   ssh -f -N -i <operator instance private key> -o StrictHostKeyChecking=no -
   o ProxyCommand="ssh -i <bastion private key> -o StrictHostKeyChecking=no -
   W %h:%p <bastion username>@<bastion IP>" <operator instance
   username>@<operator instance IP> -L <local PC port>:<Worker Node IP>:<ATS
   NodePort> -o ServerAliveInterval=60 -o ServerAliveCountMax=300
   ```

3. Access the GUI using URL: `<http/https>://<localhost/127.0.0.1>:<local PC port>` For example,`ssh -f -N -i id_rsa -o StrictHostKeyChecking=no -o ProxyCommand="ssh -i id_rsa -o StrictHostKeyChecking=no -W %h:%p opc@129.287.66.123" opc@10.1.76.7 -L 5009:10.9.60.118:32071 -o ServerAliveInterval=60 -o ServerAliveCountMax=300`

   Here, ATS GUI URLis http://localhost:5009

2.23.2 Adding Ingress and Egress Rules to Access the OCI Console

Perform the following procedure to add ingress and egress rules:

1. Log in to the OCI Console using your login credentials.
2. On the OCI Console Dashboard, click **Navigation menu**.

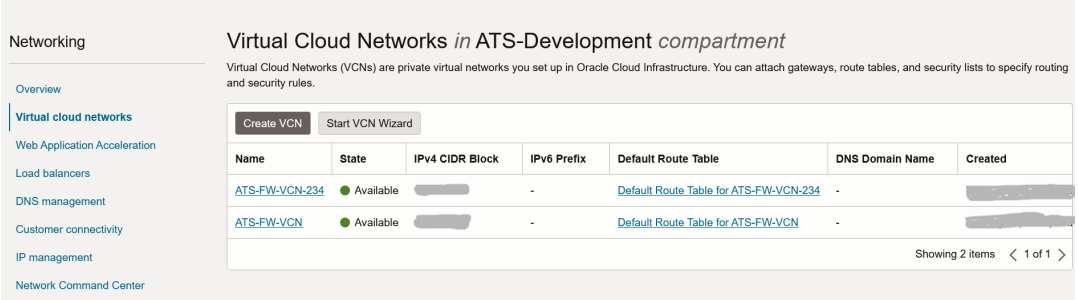
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-77 Navigation menu



3. Click **Networking**, and then click **Virtual cloud networks** on the preview pane.

Figure 2-78 Virtual cloud networks

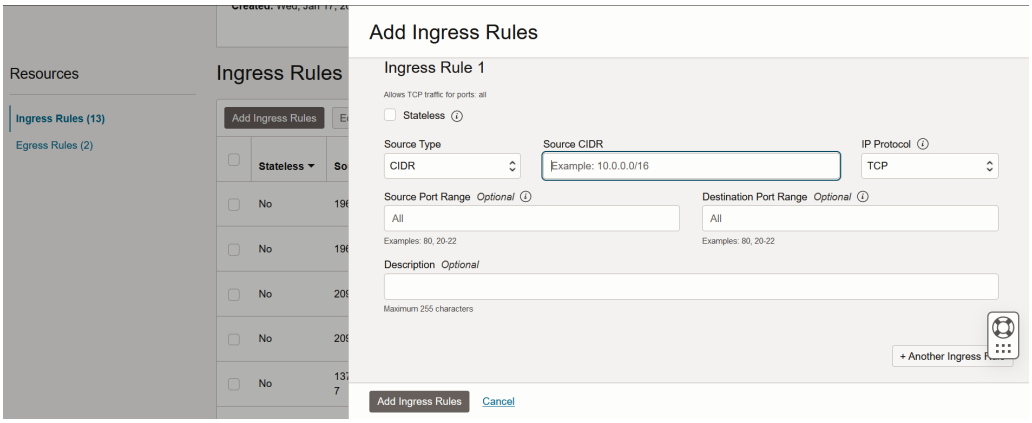


4. Click the required OKE cluster VCN.
   A list of all subnets in that VCN is displayed.
5. Save the value of the "IPv4 CIDR Block" field for **nf_lb_subnet**.
   This information can be used in the subsequent steps as value of `<CIDR for Load balancer subnet>`.
6. Save the value of the "IPv4 CIDR Block" field for **nf_node_subnet**.
   This information can be used in the subsequent steps as value of `<CIDR for Node subnet>`.

Updating Security List for Node Subnet

1. To update security list for Node subnet, do the following:
   a. Navigate to **nf_node_subnet**, and then to **nf_node_security_list**.
   b. Click **Add Ingress Rules**.

Figure 2-79 Add Ingress Rules



   c. Add ingress rule if it is unavailable to access the OCI Console in **nf_node_subnet** as described in the following table:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 2-10 Adding Ingress Rules for Node Subnet

Option Name	Description
Source Type	Indicates the source type. Value: CIDR
Source CIDR	Indicates the source of CIDR. Value: <CIDR for Load balancer subnet> **Note**: The value for the variable <CIDR for Node subnet> was determined in a previous step.
IP Protocol	Indicates the internet protocol for which this rule is applicable. Value: TCP
Source Port Range	Keep this option at its default value.
Destination Port Range	Keep this option at its default value.
Description	Keep this option at its default value.

> **Note**
>
> To access ATS API, you can use the above mentioned ingress rule.

Updating Security List for Load Balancer Subnet

1. To update security list for Load balancer subnet, do the following:
   a. Navigate to **nf_lb_subnet**, and then to **nf_lb_security_list**.
   b. Click **Add Ingress Rules**.
   c. Add ingress rule if it is unavailable to access the OCI Console in **nf_lb_subnet** by configuring the options as described in the following table:

Table 2-11 Adding Ingress Rules Load Balancer Subnet

Option Name	Description
Source Type	Indicates the source type. Value: CIDR
Source CIDR	Indicates the source of CIDR. Value: <CIDR for Network which will be used to access ATS GUI> **Note**: Avoid setting its value to 0.0.0.0/0, as it may result in granting access to ATS GUI from all networks, posing a considerable security risk. If you intend for ATS GUI to only be accessible within the Corporate Proxy (VPN), using a Source CIDR value of 0.0.0.0/0 would allow access from outside the Corporate Proxy (VPN) as well.
IP Protocol	Indicates the internet protocol for which this rule is applicable. Value: TCP
Source Port Range	Keep this option at its default value.
Destination Port Range	Indicates the ATS GUI port number. Port 8080 is used for HTTP and 8443 is used for HTTPS. Value: `<ATS service Port>`
Description	Keep this option at its default value.

   d. To access ATS API, create one more ingress rule if it is unavailable by following the previous step with the following changes:

Copyright © 2019, 2025, Oracle and/or its affiliates.

- In the **Source CIDR** option, provide the `<CIDR for Network which will be used to access ATS API>` value.
- In the **Destination Port Range** option, provide the `<ATS Service API Port>` value, which refers to the ATS API port, that is, 5001.

2. To add egress rule if it is unavailable to access the OCI Console in **nf_lb_subnet**, do the following:
   a. In the left navigation pane, click **Egress Rules**.
   b. Click **Add Egress Rules**.
   c. Configure the following options as described in this table:

Table 2-12 Adding Egress Rules Load Balancer Subnet

Option Name	Description
Destination Type	Indicates the destination type. Value: CIDR
Destination CIDR	Indicates the destination of CIDR. Value: <CIDR for Node subnet> **Note**: The value for the variable <CIDR for Node subnet> was determined in the previous step.
IP Protocol	Indicates the internet protocol for which this rule is applicable. Value: TCP
Source Port Range	Keep this option at its default value.
Destination Port Range	Keep this option at its default value.
Description	Keep this option at its default value.

> **Note**
>
> To access ATS API, you can use the above mentioned egress rule.

Updating Security List for Tunneling

1. Log in to the OCI Console using your login credentials.
2. On the OCI Console Dashboard, click **Navigation menu**.

Figure 2-80 Navigation menu



3. Click **Networking**, and then click **Virtual cloud networks** on the preview pane.
4. Click the required OKE cluster VCN.
   A list of all subnets in that VCN is displayed.
5. Save the value of the "IPv4 CIDR Block" field for **nf_operator_subnet**.
   This information can be used in the subsequent steps as value of `<CIDR for Operator subnet>`.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Updating Security list of Node Subnet

1. To update security list of Node subnet, do the following:
   a. Navigate to **nf_node_subnet**, and then to **nf_node_security_list**.
   b. Click **Add Ingress Rules**.
   c. Add ingress rule if it is unavailable to access the OCI Console in **nf_node_subnet** by configuring the options as described in the following table:

Table 2-13 Adding Ingress Rules for Node Subnet Tunneling

Option Name	Description
Source Type	Indicates the source type. Value: CIDR
Source CIDR	Indicates the source of CIDR. Value: <CIDR for Operator subnet> **Note**: The value for the variable <CIDR for Operator subnet> was determined in the previous step.
IP Protocol	Indicates the internet protocol for which this rule is applicable. Value: TCP
Source Port Range	Keep this option at its default value.
Destination Port Range	Keep this option at its default value.
Description	Keep this option at its default value.

> **Note**
>
> To access ATS API, you can use the above mentioned ingress rule.

2.24 Support for Transport Layer Security

Currently, ATS is accessible through HTTP, which can raise security risks.

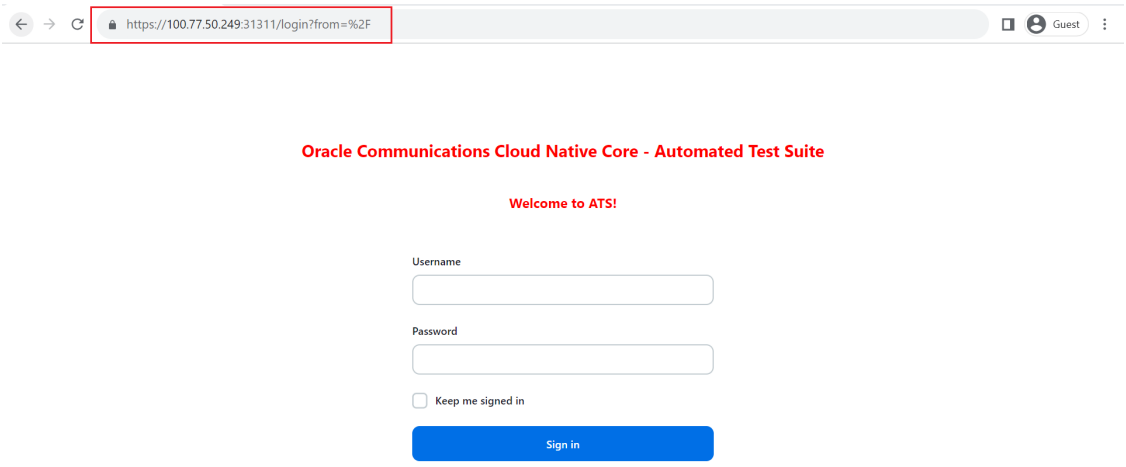
With the support of the TLS feature, Jenkins servers have been upgraded to support HTTPS, ensuring a secure and encrypted connection when accessing the ATS dashboard.

To provide encryption, HTTPS uses an encryption protocol known as Transport Layer Security (TLS), which is a widely accepted standard protocol that provides authentication, privacy, and data integrity between two communicating computer applications.

Now, users can access the ATS GUI with the HTTPS protocol instead of the previously used HTTP protocol.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-81 Access with TLS



> **Note**
>
> If this feature is not enabled before installation, HTTP will continue to operate.

2.24.1 Deploy ATS with TLS Enabled

Follow the steps in this section to create a Java KeyStore (JKS) file and enable the ATS GUI with HTTPS during installation.

2.24.1.1 Generate JKS File for Jenkins Server

A JKS file needs to be created in order for Jenkins to provide ATS GUI access through HTTPS.

Perform the following steps to generate a JKS file:

Generate the Root Certificate

If a root certificate, for example, `caroot.cert`, is not already available, a user can generate one. Users may use their own files if they have a CA-signed root certificate and key or their own root certificates. The root certificate is used to sign the application, or ATS certificate.

Follow the steps to create and use self-signed certificates:

1. Generate a root key with the following command:

   ```
   openssl genrsa 2048 > <path_to_root_key>
   ```

   For example,

   ```
   openssl genrsa 2048 > caroot.key
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

2. Generate a "caroot" certificate with the following command:

```
openssl req -new -x509 -nodes -days 1000 -key
      <path_to_root_key> > <path_to_root_certificate>
```

For example,

```
[cloud-user@star23-bastion-1 cert]$ openssl req -new -x509 -nodes -days
1000 -key caroot.key > caroot.cer
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:IN
State or Province Name (full name) []:KA
Locality Name (eg, city) [Default City]:BLR
Organization Name (eg, company) [Default Company Ltd]:ORACLE
Organizational Unit Name (eg, section) []:CGBU
Common Name (eg, your name or your server's hostname) []:ocats
Email Address []:
[cloud-user@star23-bastion-1 cert]$
```

Generate Application or Client Certificate

Follow the steps to create and edit the `ssl.conf` file:

1. In the **alt_names** section, list the IPs, such as IP.1, IP.2, and so on, that are used to access the ATS GUI:

```
[ req ]
default_bits       = 4096
distinguished_name = req_distinguished_name
req_extensions     = req_ext

[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)
countryName_default         = <Country_Name>
stateOrProvinceName         = State or Province Name (full name)
stateOrProvinceName_default = <State_Name>
localityName                = Locality Name (eg, city)
localityName_default        = <Locality_Name>
organizationName            = Organization Name (eg, company)
organizationName_default    = <Org_Name>
commonName                  = Common Name (e.g. server FQDN or YOUR name)
commonName_max              = 64
commonName_default          = <helm_name>.<namespace>.svc.cluster.local

[ req_ext ]
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth, clientAuth
basicConstraints = critical, CA:FALSE
subjectAltName = critical, @alt_names
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
[alt_names]
IP.1 = 127.0.0.1
IP.2 = <IP1>
IP.3 = <IP2>
DNS.1 = <helm_name>.<namespace>.svc.cluster.local
```

For example,

```
[ req ]
default_bits       = 4096
distinguished_name = req_distinguished_name
req_extensions     = req_ext

[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)
countryName_default         = <Country_Name>
stateOrProvinceName         = State or Province Name (full name)
stateOrProvinceName_default = <State_Name>
localityName                = Locality Name (eg, city)
localityName_default        = <Locality_Name>
organizationName            = Organization Name (eg, company)
organizationName_default    = <Org_Name>
commonName                  = Common Name (e.g. server FQDN or YOUR name)
commonName_max              = 64
commonName_default          = ocats.scpsvc.svc.cluster.local

[ req_ext ]
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth, clientAuth
basicConstraints = critical, CA:FALSE
subjectAltName = critical, @alt_names

[alt_names]
IP.1 = 127.0.0.1
IP.2 = 10.75.217.5
IP.3 = 10.75.217.76
DNS.1 = localhost
DNS.2 = ocats.scpsvc.svc.cluster.local
```

Note

- To access the GUI with DNS, make sure that the commonName_default is the same as the DNS name being used.
  - The `/etc/hosts` file in the local system should also be updated with the assigned IP and the DNS mentioned above to open the ATS with DNS. Otherwise, it is not required.
- Ensure the DNS is in this format: <service_name>.<namespace>.svc.cluster.local
- Multiple DNSs, such as DNS.1, DNS.2, and so on, can be added.
- To support the ATS API, add the IP 127.0.0.1 to the list of IPs.

Copyright © 2019, 2025, Oracle and/or its affiliates.

2. Create a Certificate Signing Request (CSR) with the following command:

```
openssl req -config ssl.conf -newkey rsa:2048 -days 1000 -nodes -keyout
<path_to_application_certificate_key> >
<path_to_certificate_signing_request>
```

For example,

```
[cloud-user@star23-bastion-1 cert]$ openssl req -config ssl.conf -newkey
rsa:2048 -days 1000 -nodes -keyout rsa_private_key_pkcs1.key >
ssl_rsa_certificate.csr
Ignoring -days; not generating a certificate
Generating a RSA private key
...+++++
........+++++
writing new private key to 'rsa_private_key_pkcs1.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IN]:
State or Province Name (full name) [KA]:
Locality Name (eg, city) [BLR]:
Organization Name (eg, company) [ORACLE]:
Common Name (e.g. server FQDN or YOUR name) [ocats]:
[cloud-user@star23-bastion-1 cert]$
```

3. Use the command to display the components of the file and verify the configurations:

```
openssl req -text -noout -verify -in
    ssl_rsa_certificate.csr
```

4. Sign in to this CSR file with the root certificate:

```
openssl x509 -extfile ssl.conf -extensions req_ext -req -in
<path_to_certificate_signing_request> -days 1000 -CA
<path_to_root_certificate> -CAkey <path_to_root_key> -set_serial 04 >
<path_to_application_certificate>
```

For example,

```
[cloud-user@star23-bastion-1 cert]$ openssl x509 -extfile ssl.conf -
extensions req_ext -req -in ssl_rsa_certificate.csr -days 1000 -CA ../
caroot.cer -CAkey ../caroot.key -set_serial 04 > ssl_rsa_certificate.crt
Signature ok
subject=C = IN, ST = KA, L = BLR, O = ORACLE, CN = ocats
Getting CA Private Key
[cloud-user@star23-bastion-1 cert]$
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

5. Use the command to verify that the certificate is properly signed by the root certificate:

   ```
   openssl verify -CAfile <path_to_root_certificate>
         <path_to_application_certificate>
   ```

   For example,

   ```
   [cloud-user@star23-bastion-1 cert]$ openssl verify -CAfile caroot.cer
   ssl_rsa_certificate.crt
   ssl_rsa_certificate.crt: OK
   ```

6. Save the generated application certificates and the root certificates.
7. Add the `caroot.cer` to the browser as a trusted author. For more information, see Enable ATS GUI with HTTPS
8. Generate the `.p12 keystore` file with the following command:

   ```
   openssl pkcs12 -inkey <path_to_application_key> -in
   <path_to_application_certificate> -export -out <path_to_p12_certificate>
   ```

   For example,

   ```
   [cloud-user@star23-bastion-1 ocats]$ openssl pkcs12 -inkey
   rsa_private_key_pkcs1.key -in ssl_rsa_certificate.crt -export -out
   certificate.p12
   Enter Export Password:
   Verifying - Enter Export Password:
   ```

9. In the prompt, create a password and save it for future use.
10. Convert the `.p12 keystore` file into a JKS format file:

    ```
    keytool -importkeystore -srckeystore <path_to_p12_certificate> -
    srcstoretype pkcs12 -destkeystore <path_to_jks_file> -deststoretype JKS
    ```

    For example,

    ```
    [cloud-user@star23-bastion-1 cert]$ keytool -importkeystore -srckeystore ./
    certificate.p12 -srcstoretype pkcs12 -destkeystore jenkinsserver.jks -
    deststoretype JKS
    Importing keystore ./certificate.p12 to jenkinsserver.jks...
    Enter destination keystore password:
    Re-enter new password:
    Enter source keystore password:
    Entry for alias 1 successfully imported.
    Import command completed:  1 entries successfully imported, 0 entries
    failed or cancelled
    ```

11. In the prompt, use the same password used while creating `.p12 keystore` file.

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> Verify that the passwords linked to the `.p12 keystore` and JKS files are the same.

12. The generated file, `jenkinserver.jks`, is provided to the Jenkins server.

2.24.1.2 Enable ATS GUI with HTTPS

Follow the steps to secure or enable TLS on the server.

1. Create a Kubernetes secret by adding the above created files:

```
kubectl create secret generic <secret_name> --from-file=<path_to_jks_file>
--from-file=<path_to_application_certificate> --from-
file=<path_to_application_key> --from-file=<path_to_root_certificate> -n
<namespace>
```

For example,

```
kubectl create secret generic ocats-tls-secret --from-
file=jenkinsserver.jks --from-file=ssl_rsa_certificate.crt --from-
file=rsa_private_key_pkcs1.key --from-file=caroot.cer -n scpsvc
```

Where,

jenkinsserver.jks: This file is needed when `atsGuiTLSEnabled` is set to true. This is necessary to open ATS GUI with secured TLS protocol.

ssl_rsa_certificate.crt: This is client application certificate.

rsa_private_key_pkcs1.key: This is RSA private key.

caroot.cer: This file used during creation of jks file needs to be passed for Jenkins/ATS API communication.

The sample of created secret:

```
[cloud-user@star23-bastion-1 cert]$ kubectl describe secret ocats-tls-
secret -n scpsvc
Name:         ocats-tls-secret
Namespace:    scpsvc
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
caroot.cer:                      1147 bytes
ssl_rsa_certificate.crt:         1424 bytes
jenkinsserver.jks:               2357 bytes
rsa_private_key_pkcs1.key:       1675 bytes
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

2. Apply the following changes in `values.yaml` file.

```
certificates:
  cert_secret_name: "ocats-tls-secret"
  ca_cert: "caroot.cer"
  client_cert: "ssl_rsa_certificate.crt"
  private_key: "rsa_private_key_pkcs1.pem"
  jks_file: "jenkinsserver.jks"
  jks_password: "123456"  #This is the password given to the jks file
while creation.
```

   The user can install the ATS, using the helm install command. Change the `atsGuiTLSEnabled` Helm parameter value to true for ATS to get the certificates and support HTTPS for GUI.

3. A user can now start ATS with HTTPS the protocol.
   The link to open the ATS GUI format is `https://<IP>:<port>`, for example, `https://10.75.217.25:30301`

   The lock symbol in the browser indicates that the server is secured or TLS enabled.

Adding a Certificate in Browser

The created root certificate is added to the browser, either Mozilla Firefox or Chrome.

> **Note**
>
> Future versions of these browsers may involve different menu options. For more information on importing root certificate, see the browser documentation to add a self-signed certificate to the browser as a trusted certificate.

Adding in Windows laptops

1. In the Chrome browser, navigate to the settings and search for security.
2. Click the **security** option that appears next to search.
3. Click the **Manage Device Certificate** option. The Certificate window opens.
4. Click the Trusted root certification authorities bar.
5. Import the `caroot` certificate.
6. Save and restart the browser.

Adding in Mac laptop

1. In the Chrome browser, navigate to the settings and search for security.
2. Click the **security** option that appears next to search.
3. Click the **Manage Device Certificate** option. The Keychain Access window opens.
4. Search the tab certificate and drag and drop the downloaded caroot certificate.
5. Find the uploaded certificate in the list, usually listed by a temporary name.
6. Double click the certificate and expand the Trust option.
7. In **When using this certificate** option, assign it to "always trust".

Copyright © 2019, 2025, Oracle and/or its affiliates.

8. Close the window and validate if it asks for the password.
9. Save and restart the browser.

Adding for Windows and Mac laptop

1. In the Mozilla Firefox browser, navigate to the settings and search for certificates.
2. Click the **View Certificate** that appears next to search. This opens a Certificate Manager window.
3. Navigate to the Authorities section, click the **Import** button, and upload the `caroot` certificate.
4. Click the **Trust** options in the pop-up window and click OK.
5. Save and restart the browser.

2.25 Test Results Analyzer

The Test Results Analyzer is a plugin available in ATS to view pipeline test results based on XML reports. It provides the test results report in a graphical format, which includes consolidated and detailed stack trace results in case of any failures. It allows you to navigate to each and every test.

The test result report shows any one of the following statuses for each test case:

- **PASSED:** If the test case passes.
- **FAILED:** If the test case fails.
- **SKIPPED:** If the test case is skipped.
- **N/A:** If the test cases are not executed in the current build.

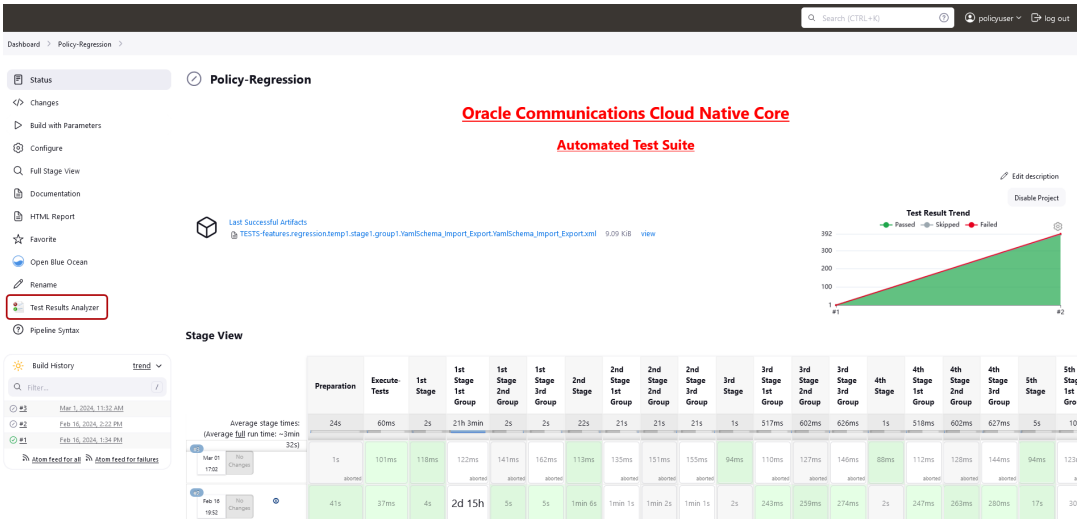
2.25.1 Accessing Test Results Analyzer Feature

To access the test results analyzer feature:

1. From the NF home page, click any new feature pipeline or regression pipeline where you want to run this plugin.
2. In the left navigation pane, click **Test Results Analyzer**.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-82 Test Results Analyzer Option



When the build completes, the test result report appears. A sample test result report is shown below:

Figure 2-83 Sample Test Result Report

Chart	Package/Class/Testmethod	Passed	Transitions	5	4	3	2	1
☐	features.cust_newfeatures.temp1.Binding_Service.Binding_Service	0% (0%)	0	FAILED	N/A	N/A	N/A	N/A
☐	features.cust_newfeatures.temp1.Bulk_Import_Export.Bulk_Import_Export	100% (100%)	0	PASSED	PASSED	SKIPPED	SKIPPED	N/A
☐	features.cust_newfeatures.temp1.Diameter_GW_Error_Handling.Diameter_GW_Error_Handling	0% (0%)	0	FAILED	N/A	N/A	N/A	N/A
☐	features.cust_newfeatures.temp1.Dynamic_Policy_Override.Dynamic_Policy_Override.This feature aims to dynamically override the parameters of pcc and session rules during SM Policy Association	100% (100%)	0	PASSED	PASSED	SKIPPED	SKIPPED	N/A
☐	features.cust_newfeatures.temp1.IMS_Emergency_Call.IMS_Emergency_Call	0% (0%)	0	FAILED	N/A	N/A	N/A	N/A
☐	features.cust_newfeatures.temp1.Operators.Operators	0% (0%)	0	FAILED	N/A	N/A	N/A	N/A
☐	features.cust_newfeatures.temp1.PolicyActionForBulkPccRuleRemoval.Policy_Action_Bulk_Pccrule_Removal	100% (100%)	0	PASSED	PASSED	SKIPPED	SKIPPED	N/A
☐	features.cust_newfeatures.temp1.Policy_Conditions_And_Actions.Policy_Conditions_And_Actions	0% (0%)	0	FAILED	N/A	N/A	N/A	N/A
☐	features.cust_newfeatures.temp1.Policy_Table.Policy_Table	100% (100%)	0	PASSED	PASSED	PASSED	SKIPPED	N/A
☐	features.cust_newfeatures.temp1.Request_AVP.Request_AVP	0% (0%)	0	FAILED	N/A	N/A	N/A	N/A
☐	features.cust_newfeatures.temp1.Sanity.Sanity	0% (0%)	0	FAILED	N/A	N/A	N/A	N/A
☐	features.cust_newfeatures.temp1.SessionRetry.SessionRetry.To verify that SM or User-service retry to alternate FQDN if the	0%	0	FAILED	FAILED	SKIPPED	FAILED	N/A

3. Click any one of the statuses (PASSED, FAILED, SKIPPED) to view the respective feature detail status report.

> **Note**
>
> For N/A status, a detailed status report is not available.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-84 Test Result

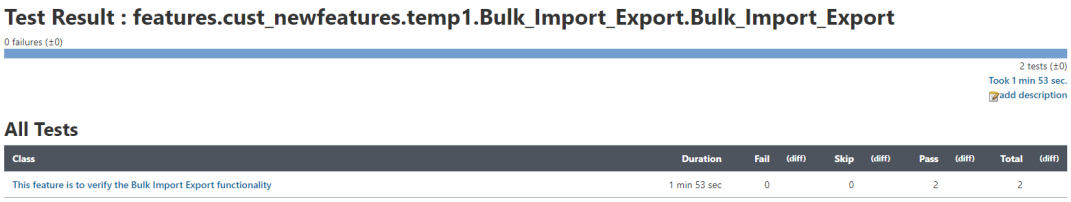
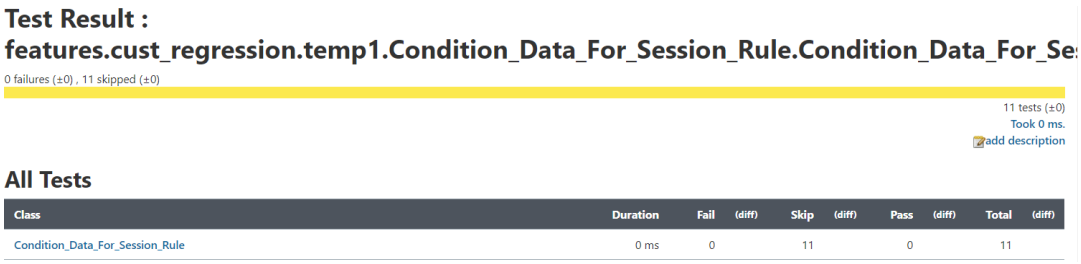


Figure 2-85 Test Result



4. In the case of a rerun, the test cases that passed in the initial run but were skipped in the rerun are considered "passed" in the Test Results Analyzer Report. The following screenshot depicts the scenario: "Variant2_equal_smPolicySnssaiData,Variant2_exist_smPolicyData,Variant2_exist_smPolicyDnnData_dnn" where the test cases passed in the initial run but skipped in the rerun are considered "passed" in general.

Figure 2-86 Test Results



5. Click **PASSED**. The following highlighted message means the test case was passed in the main run but skipped in the rerun.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-87 Test Result Info



2.26 Support for Test Case Mapping and Count

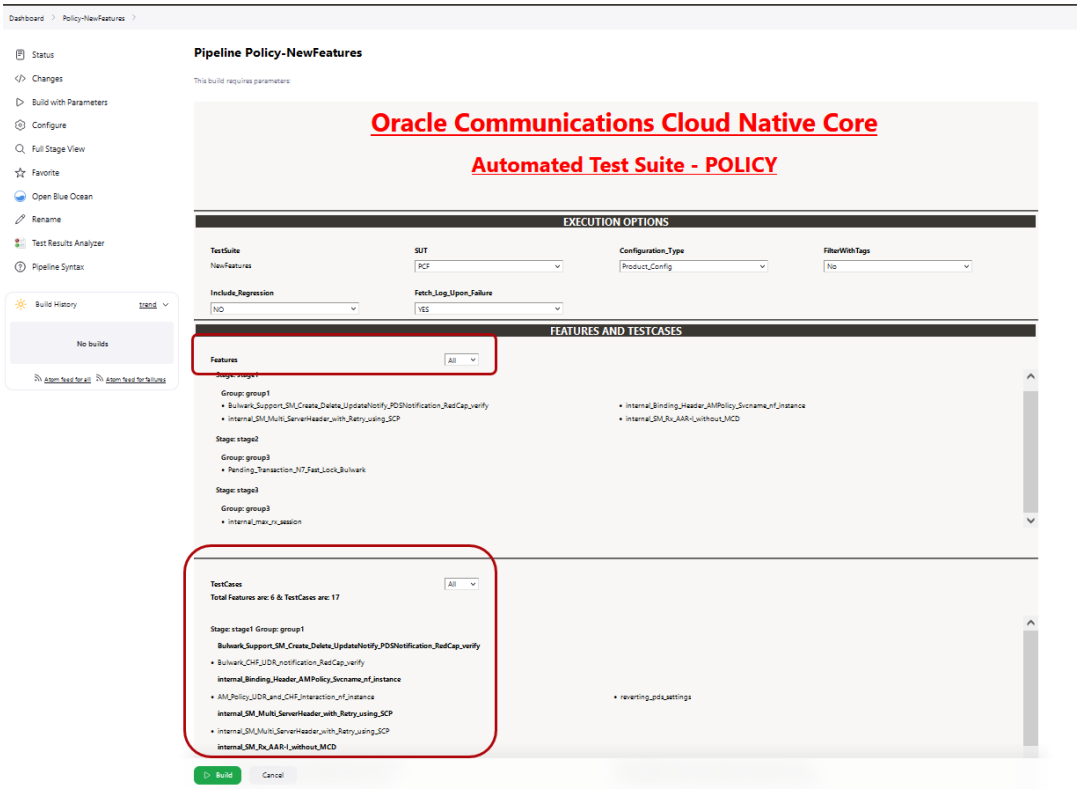
The Test Case Mapping and Count feature displays the total number of features, test cases, or scenarios and their mapping to each feature in the ATS GUI.

2.26.1 Access Test Case Mapping and Count Feature

1. On the NF home page, click any new feature or regression pipeline where you want to use this feature.
2. In the left navigation pane, click **Build with Parameters**.
3. On the FEATURE AND TESTCASES section, click **All** from the Features drop-down menu to view the TestCases details mapped to each feature.

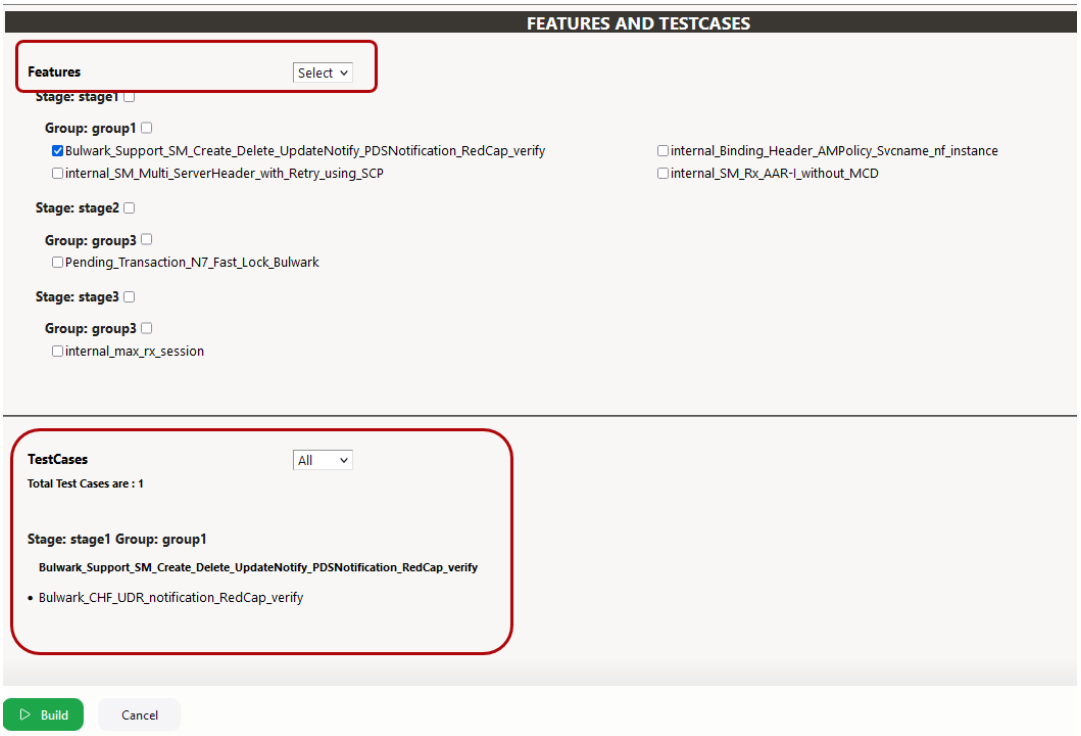
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 2-88 Test Case Mapping



4. Click **Select** from the **Features** drop-down menu to view the the test cases details.

Figure 2-89 Test Cases Details When Select a Single/MultipleFeatures



Copyright © 2019, 2025, Oracle and/or its affiliates.

3
Installing ATS for Different Network Functions

This section describes how to install ATS for different network functions. It includes:

- Installing ATS for BSF
- Installing ATS for NRF
- Installing ATS for NSSF
- Installing ATS for Policy
- Installing ATS for SCP
- Installing ATS for SEPP
- Installing ATS for UDR

3.1 Installing ATS for BSF

The BSF ATS installation procedure covers two steps:

1. Locating and downloading the ATS package for BSF.
2. Deploying ATS and stub pods in Kubernetes cluster.

This includes installation of three stubs (nf1stub, nf11stub, and nf12stub), ocdns-bind stub, and BSF ATS in BSF namespace.

3.1.1 Resource Requirements

This section describes the ATS resource requirements for Binding Support Function.

Overview - Total Number of Resources

The following table describes the overall resource usage in terms of CPUs, memory, and storage:

Table 3-1 BSF - Total Number of Resources

Resource Name	Non-ASM CPU	Non-ASM Memory (GB)	ASM CPU	ASM Memory (GB)
BSF Total	41	36	73	52
ATS Total	11	11	23	17
cnDBTier Total	107.1	175.2	137.1	190.2
Grand Total BSF ATS	159.1	222.2	233.1	259.2

BSF Pods Resource Requirements Details

This section describes the resource requirements, which are needed to deploy BSF ATS successfully.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-2 BSF Pods Resource Requirements Details

BSF Microservices	Max CPU	Memory (GB)	Max Replica	Isito ASM CPU	Isito ASM Memory (GB)	Non-ASM Total CPU	Non-ASM Memory (GB)	ASM Total CPU	ASM Total Memory (GB)
oc-app-info	1	1	1	2	1	1	1	3	2
oc-diam-gateway	4	2	1	2	1	4	2	6	3
alternate-route	2	4	1	2	1	2	4	4	5
oc-config-server	4	2	1	2	1	4	2	6	3
ocegress_gateway	4	6	1	2	1	4	6	6	7
ocingress_gateway	4	6	1	2	1	4	6	6	7
nrf-client-mngt	1	1	2	2	1	2	2	6	4
oc-audit	2	1	1	2	1	2	1	4	2
oc-config-mgmt	4	2	2	2	1	8	4	12	6
oc-query	2	1	2	2	1	4	2	8	4
oc-perf-info	1	1	2	2	1	2	2	6	4
bsf-management-service	4	4	1	2	1	4	4	6	5
BSF Totals						41	36	73	52

ATS Resource Requirements details for BSF

This section describes the ATS resource requirements, which are needed to deploy BSF ATS successfully.

Table 3-3 ATS Resource Requirements Details

ATS Microservices	Max CPU	Max Memory (GB)	Max Replica	Isito ASM CPU	Isito ASM Memory (GB)	Non-ASM Total CPU	Non-ASM Total Memory (GB)	ASM Total CPU	ASM Total Memory (GB)
ocstub1-py	2	2	1	2	1	2	2	4	3
ocstub2-py	2	2	1	2	1	2	2	4	3

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-3 (Cont.) ATS Resource Requirements Details

ATS Microservices	Max CPU	Max Memory (GB)	Max Replica	Isito ASM CPU	Isito ASM Memory (GB)	Non-ASM Total CPU	Non-ASM Total Memory (GB)	ASM Total CPU	ASM Total Memory (GB)
ocstub3-py	2	2	1	2	1	2	2	4	3
ocats-bsf	3	3	1	2	1	3	3	5	4
ocdns-bind	1	1	1	2	1	1	1	3	2
ocdiam-sim	1	1	1	2	1	1	1	3	2
ATS Totals						11	11	23	17

cnDBTier Resource Requirements Details for BSF ATS

This section describes the cnDBTier resource requirements, which are needed to deploy BSF ATS successfully.

> **Note**
>
> For cnDBTier pods, a minimum of 4 worker nodes are required.

Table 3-4 cnDBTier Resource Requirements Details

cnDBTier Microservices	Min CPU	Min Memory (GB)	Min Replica	Isito ASM CPU	Isito ASM Memory (GB)	Total CPU	Total Memory (GB)	ASM Total CPU	ASM Total Memory (GB)
db_monitor_svc	1	1	1	2	1	1	1	3	2
db_replication_svc	2	12	1	2	1	2	12	4	13
db_backup_manager_svc	0.1	0.2	1	2	1	0.1	0.2	2.1	1.2
ndbappmysqld	8	10	4	2	1	32	40	40	44
ndbmgmd	4	10	2	2	1	8	20	12	22
ndbmtd	10	18	4	2	1	40	72	48	76
ndbmysqld	8	10	2	2	1	16	20	20	22
db_infra_moditor_svc	8	10	1	2	1	8	10	8	10
cnDBTier Total						107.1	175.2	137.1	190.2

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.1.2 Downloading the ATS Package

This section provides information on how to locate and download BSF ATS package file from My Oracle Support (MOS).

Locating and Downloading BSF ATS Package

To locate and download the ATS Image from MOS, perform the following steps:

1. Log in to My Oracle Support using the appropriate credentials.
2. Select the **Patches and Updates** tab.
3. In the **Patch Search** window, click **Product or Family (Advanced)**.
4. Enter *Oracle Communications Cloud Native Core - 5G* in the **Product** field.
5. Select *Oracle Communications Cloud Native Binding Support Function <release_number>* from **Release** drop-down.
6. Click **Search**. The **Patch Advanced Search Results** list appears.
7. Select the required patch from the search results. The Patch Details window appears.
8. Click **Download**. The File Download window appears.
9. Click the **<p********_<release_number>_Tekelec>.zip** file to downlaod the BSF ATS package file.
10. Untar the gzip file `ocats-bsf-tools-25.1.200.0.0.tgz` to access the following files:

    ```
    ocats-bsf-pkg-25.1.200.0.0.tgz
    ocdns-pkg-25.1.203.tgz
    ocstub-pkg-25.1.201.tgz
    ocdiam-sim-25.1.203.tgz
    ```

    The contents included in each of these files are as follow:

    ```
    ocats-bsf-tools-25.1.200.0.0.tgz
    |
    |_ _ _ocats-bsf-pkg-25.1.200.tgz
    | |_ _ _ _ _ _ ocats-bsf-25.1.200.tgz (Helm Charts)
    | |_ _ _ _ _ _ ocats-bsf-images-25.1.200.tar (Docker Images)
    | |_ _ _ _ _ _ ocats-bsf-data-25.1.200.tgz (BSF ATS and Jenkins job Data)
    |
    |_ _ _ocstub-pkg-25.1.201.0.0.tgz
    | |_ _ _ _ _ _ ocstub-py-25.1.201.tgz(Helm Charts)
    | |_ _ _ _ _ _ ocstub-py-image-25.1.201.tar (Docker Images)
    |
    |_ _ _ocdns-pkg-25.1.203.0.0.tgz
    | |_ _ _ _ _ _ ocdns-bind-25.1.203.tgz(Helm Charts)
    | |_ _ _ _ _ _ ocdns-bind-image-25.1.203.tar (Docker Images)
    |
    |_ _ _ocdiam-pkg-25.1.203.0.0.tgz
    |      |_ _ _ _ _ _ ocdiam-sim-25.1.203.tgz(Helm Charts)
    |      |_ _ _ _ _ _ ocdiam-sim-image-25.1.203.tar (Docker Images)
    ```

11. Copy the tar file from the downloaded package to CNE, OCI, or Kubernetes cluster where you want to deploy ATS.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.1.3 Deploy ATS with TLS Enabled

> **Note**
>
> - OCATS and Python stubs support both TLS 1.2. and TLS 1.3.
> - DiamSim pod do not support secure calls.

Follow the steps in this section to create a Java KeyStore (JKS) file and enable the BSF ATS GUI with HTTPS during installation.

3.1.3.1 Generate JKS File for Jenkins Server

To access Jenkins ATS GUI access through HTTPS, a JKS file should be created.

Perform the following steps to generate the JKS file:

Generate the Root Certificate

1. If the user has a Certificate Authority (CA) signed root certificate such as `caroot.cert` and key, then the user can use those files.
2. If the root certificate is not already available, the user can generate one self signed root certificate. This root certificate created needs to be added to the `truststore` such as a Browser like Firefox or Chrome. User can follow the Browser specific documentation to upload the root certificate. The root certificate is used to sign the application, or ATS certificate.
3. Generate a root key with the following command:

   ```
   openssl genrsa 2048 > caroot.key
   ```

   This will generate a key called `caroot.key`
4. Generate a `caroot` certificate with the following command:

   ```
   openssl req -new -x509 -nodes -days 1000 -key <root_key> >
   <root_certificate>
   ```

   For example,

   ```
   [cloud-user@platform-bastion-1]$ openssl req -new -x509 -nodes -days 1000 -
   key caroot.key > caroot.cer
   You are about to be asked to enter information that will be incorporated
   into your certificate request.
   What you are about to enter is what is called a Distinguished Name or a DN.
   There are quite a few fields but you can leave some blank
   For some fields there will be a default value,
   If you enter '.', the field will be left blank.
   -----
   Country Name (2 letter code) [XX]:IN
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
State or Province Name (full name) []:KA
Locality Name (eg, city) [Default City]:BLR
Organization Name (eg, company) [Default Company Ltd]:ORACLE
Organizational Unit Name (eg, section) []:CGBU
Common Name (eg, your name or your server's hostname) []:ocats
Email Address []:
[cloud-user@platform-bastion-1]$
```

Generate Application or Client Certificate

1. Create a `ssl.conf` file.
2. Edit the `ssl.conf` file. In the "[alt_names]" section, list the IPs that are used to access ATS GUI as shown in the following sample `ssl.conf` file:

```
[ req ]
default_bits       = 4096
distinguished_name = req_distinguished_name
req_extensions     = req_ext

[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)
countryName_default         = IN
stateOrProvinceName         = State or Province Name (full name)
stateOrProvinceName_default = KN
localityName                = Locality Name (eg, city)
localityName_default        = BLR
organizationName            = Organization Name (eg, company)
organizationName_default    = ORACLE
commonName                  = Common Name (e.g. server FQDN or YOUR name)
commonName_max              = 64
commonName_default          = ocats.ocbsf.svc.cluster.local

[ req_ext ]
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth, clientAuth
basicConstraints = critical, CA:FALSE
subjectAltName = critical, @alt_names

[alt_names]
IP.1 = 127.0.0.1
IP.2 = 10.75.217.5
IP.3 = 10.75.217.76
DNS.1 = localhost
DNS.2 = ocats.ocbsf.svc.cluster.local
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> - To access the GUI with DNS, make sure that the commonName_default is the same as the DNS name being used.
>   - Ensure the DNS is in this format:`<service_name>.<namespace>.svc.cluster.local`
>
>   Multiple DNSs, such as DNS.1, DNS.2, and so on, can be added.
> - To support the ATS API, it is necessary to add the IP 127.0.0.1 to the list of IPs.

3. Create a Certificate Signing Request (CSR) with the following command:

   ```
   openssl req -config ssl.conf -newkey rsa:2048 -days 1000 -nodes -keyout
   rsa_private_key_pkcs1.key > ssl_rsa_certificate.csr
   ```

   Output:

   ```
   [cloud-user@platform-bastion-1 ocbsf]$ openssl req -config ssl.conf -
   newkey rsa:2048 -days 1000 -nodes -keyout rsa_private_key_pkcs1.key >
   ssl_rsa_certificate.csr
   Ignoring -days; not generating a certificate
   Generating a RSA private key
   ...+++++
   ........+++++
   writing new private key to 'rsa_private_key_pkcs1.key'
   -----
   You are about to be asked to enter information that will be incorporated
   into your certificate request.
   What you are about to enter is what is called a Distinguished Name or a DN.
   There are quite a few fields but you can leave some blank
   For some fields there will be a default value,
   If you enter '.', the field will be left blank.
   -----
   Country Name (2 letter code) [IN]:
   State or Province Name (full name) [KA]:
   Locality Name (eg, city) [BLR]:
   Organization Name (eg, company) [ORACLE]:
   Common Name (e.g. server FQDN or YOUR name) [ocbsf]:
   [cloud-user@platform-bastion-1 ocbsf]$
   ```

4. To display all the components of the CSR file and to verify the configurations run the following command:

   ```
   openssl req -text -noout -verify -in ssl_rsa_certificate.csr
   ```

5. Sign the CSR file with root certificate by running the following command:

   ```
   openssl x509 -extfile ssl.conf -extensions req_ext -req -
   inssl_rsa_certificate.csr -days 1000 -CA ../caroot.cer -CAkey ../
   caroot.key -set_serial 04 > ssl_rsa_certificate.crt
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Output:

```
[cloud-user@platform-bastion-1 ocbsf]$ openssl x509 -extfile ssl.conf -
extensions req_ext -req -in ssl_rsa_certificate.csr -days 1000 -CA ../
caroot.cer -CAkey ../caroot.key -set_serial 04 > ssl_rsa_certificate.crt
Signature ok
subject=C = IN, ST = KA, L = BLR, O = ORACLE, CN = ocbsf
Getting CA Private Key
[cloud-user@platform-bastion-1 ocbsf]$
```

6. Verify if the certificate is signed by the root certificate by running the following command:

```
[cloud-user@platform-bastion-1 ocbsf]$ openssl verify -CAfile caroot.cer
ssl_rsa_certificate.crt
```

Output:

```
[cloud-user@platform-bastion-1 ocbsf]$ openssl verify -CAfile caroot.cer
ssl_rsa_certificate.crt
ssl_rsa_certificate.crt: OK
```

7. Save the generated application certificate and root certificate.
8. Add the `caroot.cer` to the browser as a trusted author.
9. The generated application/client certificates cannot be directly given to the Jenkins server. Hence generate the `.p12 keystore` file for the client certificate with the following command:

```
[cloud-user@platform-bastion-1 ocbsf]$ openssl pkcs12 -inkey
rsa_private_key_pkcs1.key -inssl_rsa_certificate.crt -export-out
certificate.p12
Enter Export Password:
Verifying - Enter Export Password:
```

10. In the prompt, create a password and save it for future use.
11. Convert the .p12 keystore file into a JKS format file using the following command:

```
[cloud-user@platform-bastion-1 ocbsf]$ keytool -importkeystore -
srckeystore ./certificate.p12 -srcstoretype pkcs12 -destkeystore
jenkinsserver.jks -deststoretype JKS
```

Output:

```
[cloud-user@platform-bastion-1 ocbsf]$ keytool -importkeystore -
srckeystore ./certificate.p12 -srcstoretype pkcs12 -destkeystore
jenkinsserver.jks -deststoretype JKS
Importing keystore ./certificate.p12 to jenkinsserver.jks...
Enter destination keystore password:
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
Re-enter new password:
Enter source keystore password:
Entry for alias 1 successfully imported.
Import command completed:  1 entries successfully imported, 0 entries
failed or cancelled
```

12. In the prompt, use the same password used while creating `.p12 keystore` file.

> **Note**
>
> Ensure that the .p12 keystore and JKS files has the same passwords.

13. The generated JKS file, `jenkinserver.jks` is added to the Jenkins path, where Jenkins server can access it.

For more details about the ATS TLS feature, refer to Deploy ATS with TLS Enabled section.

3.1.3.2 Enable TLS on Python Stubs

Before starting the creation of certificate, user should have the path details of `caroot.cer` file.

1. Create a `ssl.conf` file.
2. Edit `ssl.conf` file. Ensure that the DNS is in the format of `*.<namespace>.svc`
   A sample **stub_ssl.conf** file:

```
[ req ]
default_bits       = 4096
distinguished_name = req_distinguished_name
req_extensions     = req_ext

[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)
countryName_default         = IN
stateOrProvinceName         = State or Province Name (full name)
stateOrProvinceName_default = KN
localityName                = Locality Name (eg, city)
localityName_default        = BLR
organizationName            = Organization Name (eg, company)
organizationName_default    = ORACLE
commonName                  = svc.cluster.local

[ req_ext ]
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth, clientAuth
basicConstraints = critical, CA:FALSE
subjectAltName = critical, @alt_names

[alt_names]
IP.1 = 127.0.0.1
DNS.1 = *.ocats.svc
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

3. Create a Certificate Signing Request (CSR) for the stubs using the following command:

```
openssl req -config stub_ssl.conf -newkey  rsa:2048 -days 1000 -nodes -
keyout rsa_private_key_stub_pkcs2.key > stub_ssl_rsa_certificate1.csr
```

Output:

```
[cloud-user@platform-bastion-1 stub_certs]$ openssl req -config
stub_ssl.conf -newkey rsa:2048 -days 1000 -nodes -keyout
rsa_private_key_stub_pkcs2.key > stub_ssl_rsa_certificate1.csr
Ignoring -days; not generating a certificate
Generating a RSA private key
.....................+++++
...+++++
writing new private key to 'rsa_private_key_stub_pkcs2.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IN]:
State or Province Name (full name) [KN]:
Locality Name (eg, city) [BLR]:
Organization Name (eg, company) [ORACLE]:
svc.cluster.local []:*.ocbsf.svc
```

4. Sign the certificate with the CA root using the following command:

```
openssl x509 -extfile stub_ssl.conf -extensions req_ext -req -in
stub_ssl_rsa_certificate1.csr -days 1000 -CA  ../ocbsf-caroot.cer -
CAkey ../ocbsf-caroot.key -set_serial 05 >  stub_ssl_rsa_certificate1.crt
```

Output:

```
[cloud-user@platform-bastion-1 stub_certs]$ openssl x509 -extfile
stub_ssl.conf -extensions req_ext -req -in stub_ssl_rsa_certificate1.csr -
days 1000 -CA ../ocbsf-caroot.cer -CAkey ../ocbsf-caroot.key -set_serial
05 > stub_ssl_rsa_certificate1.crt
Signature ok
subject=C = IN, ST = KN, L = BLR, O = ORACLE, CN = *.ocbsf.svc
Getting CA Private Key
```

5. Create a secret for the stub and associate it with the namespace using the following command:

```
kubectl create secret generic ocats-stub-secret1 --from-
file=stub_ssl_rsa_certificate1.crt --from-
file=rsa_private_key_stub_pkcs2.key --from-file=../ocbsf-caroot.cer -n
ocbsf
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Output:

```
[cloud-user@platform-bastion-1 ocbsf]$ kubectl create secret generic ocbsf-
stub-secret1 --from-file=stub_ssl_rsa_certificate1.crt --from-
file=rsa_private_key_stub_pkcs2.key --from-file=../ocbsf-caroot.cer -n
ocbsf
secret/ocats-stub-secret1 created
```

6. Update `values.yaml` file of each Python stub in specific NF namespace with following details:

```
NF: "<NF-Name>"
cert_secret_name: "ocats-stub-secret"
ca_cert: "ocbsf-caroot.cer"
client_cert: "ocbsf-stub_ssl_rsa_certificate.crt"
private_key: "ocbsf-rsa_private_key_stub_pkcs1.key"
expose_tls_service: true
CLIENT_CERT_REQ: true
```

> **Note**
>
> If the Helm, `cert_secret_name` parameter, is null, then `ca_cert`, `client_cert`, and `private_key` values are not considered by the TLS.

7. Ensure to update the deployment of all the Python stubs installed on the setup.

For more details about the ATS TLS feature, refer to Support for Transport Layer Security section.

3.1.3.3 Enable ATS GUI with HTTPS

Follow the steps to secure or enable TLS on the server.

1. Create a Kubernetes secret by adding the above created files:

```
kubectl create secret generic ocats-tls-secret --from-
file=jenkinsserver.jks --from-file=ssl_rsa_certificate.crt --from-
file=rsa_private_key_pkcs1.key --from-file=caroot.cer -n ocbsf
```

Where,

jenkinsserver.jks: This file is needed when `atsGuiTLSEnabled` is set to true. This is necessary to open ATS GUI with secured TLS protocol.

ssl_rsa_certificate.crt: This is client application certificate.

rsa_private_key_pkcs1.key: This is RSA private key.

caroot.cer: This file used during creation of jks file needs to be passed for Jenkins/ATS API communication.

Copyright © 2019, 2025, Oracle and/or its affiliates.

The sample of created secret:

```
[cloud-user@platform-bastion-1 ~]$ kubectl describe secret ocats-tls-
secret -n ocbsf
Name:         ocats-tls-secret
Namespace:    ocats
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
caroot.cer:                        1147 bytes
ssl_rsa_certificate.crt:           1424 bytes
jenkinsserver.jks:                 2357 bytes
rsa_private_key_pkcs1.key:         1675 bytes
```

2. Apply the following changes in `values.yaml` file.

```
certificates:
  cert_secret_name: "ocats-tls-secret"
  ca_cert: "caroot.cer"
  client_cert: "ssl_rsa_certificate.crt"
  private_key: "rsa_private_key_pkcs1.pem"
  jks_file: "jenkinsserver.jks" # This parameter is needed when
atsGuiTLSEnabled
      is set to true. This file is necessary for ATS GUI to  be opend with
secured TLS protocol.
  jks_password: "123456"  #This is the password given to the jks file
while creation.
```

The user can install the ATS, using the helm install command. Change the `atsGuiTLSEnabled` Helm parameter value to true for ATS to get the certificates and support HTTPS for GUI.

3. Upload the `caroot.cer` file to the browser, before accessing it using `https` protocol.
For more details about the uploading the file to the browser, refer *Adding a Certificate in Browser* section in Enable ATS GUI with HTTPS.
4. A user can now start ATS with HTTPS the protocol. The link to open the ATS GUI format is `https://<IP>:<port>`, for example, `https://10.75.217.25:30301`.
The lock symbol in the browser indicates that the server is secured or TLS enabled.

3.1.4 Pushing the Images to Customer Docker Registry

This section describes the pre-deployment steps for deploying ATS and stub pods.

Preparing to deploy ATS and Stub Pods in Kubernetes Cluster

To deploy ATS and Stub pods in a Kubernetes Cluster, perform the following steps:

1. Run the following command to extract the tar file content:
`tar -zxvf ocats-bsf-tools-25.1.200.0.0.tgz`

Copyright © 2019, 2025, Oracle and/or its affiliates.

The output of this command is:

```
ocats-bsf-pkg-25.1.200.tgz
ocstub-pkg-25.1.201.tgz
ocdns-pkg-25.1.203.tgz
ocdiam-pkg-25.1.203.0.0tgz
```

2. Go to the `ocats-bsf-tools-25.1.200.0.0` folder and run the following command to extract the helm charts and docker images of ATS:
   `tar -zxvf ocats-bsf-pkg-25.1.200.0.0.tgz`

   The output of this command is:

   ```
   ocats-bsf-25.1.200.tgz
   ocats-bsf-images-25.1.200.tar
   ocats-bsf-data-25.1.200.tgz
   ```

3. Run the following command in your cluster to load the ATS docker image:
   `docker load --input ocats-bsf-images-25.1.200.tar`
4. Run the following commands to tag and push the ATS images

   ```
   docker tag ocats-bsf:25.1.200 <registry>/ocats-bsf:25.1.200
   docker push <registry>/ocats-bsf:25.1.200
   ```

   Example:

   ```
   docker tag ocats-bsf:25.1.200 localhost:5000/ocats-bsf:25.1.200
   docker push localhost:5000/ocats-bsf:25.1.200
   ```

5. Run the following command to untar the helm charts, in `ocats-bsf-25.1.200.tgz`

   ```
   tar -zxvf ocats-bsf-25.1.200.tgz
   ```

6. Update the registry name, image name and tag in the `ocats-bsf/values.yaml` file as required. For this, you need to update the `image.repository` and `image.tag` parameters in the `ocats-bsf/values.yaml` file.
7. In the `ocats-bsf/values.yaml` file, the `atsFeatures` parameter is configured to control ATS feature deliveries.

   ```
   atsFeatures:  ## DO NOT UPDATE this section without My Oracle Support
   team's support
     testCaseMapping: true              # To display Test cases on GUI along
   with Features
     logging: true                      # To enable feature to collect
   applogs in case of failure
     lightWeightPerformance: false      # The Feature is not implemented yet
     executionWithTagging: true         # To enable Feature/Scenario
   execution with Tag
     scenarioSelection: false           # The Feature is not implemented yet
     parallelTestCaseExecution: true    # To run ATS features parallel
     parallelFrameworkChangesIntegrated: true # To run ATS features parallel
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  mergedExecution: false              # To execute ATS Regression and
NewFeatures pipelines together in merged manner
  individualStageGroupSelection: false  # The Feature is not implemented
yet
  parameterization: true              # When set to false, the
Configuration_Type parameter on the GUI will not be available.
  atsApi: true                        # To trigger ATS using ATS API
  healthcheck: true                   # TO enable/disable ATS Health Check.
  atsGuiTLSEnabled: false             # To run ATS GUI in https mode.
  atsCommunicationTLSEnabled: false  #If set to true, ATS will get
necessary variables to communicate with SUT, Stub or other NFs with TLS
enabled. It is not required in ASM environment.
```

> **Note**
>
> It is recommended to avoid altering `atsFeatures` flags.

3.1.5 Configuring ATS

3.1.5.1 Enabling Static Port

To enable static port, in the `ocats-bsf/values.yaml` file under the service section, set the value of `staticNodePortEnabled` parameter to `true` and enter a valid nodePort value for `staticNodePort` parameter.

```
service:
  customExtension:
    labels: {}
    annotations: {}
  type: LoadBalancer
  ports:
    http:
      port: "8080"
      staticNodePortEnabled: false
      staticNodePort: ""
```

3.1.5.2 Enable Static API Node Port

To enable static API node port, in the ocats-bsf/values.yaml file under the **service** section, set the value of **staticNodePortEnabled** parameter to true and enter a valid nodePort value for **staticNodePort** parameter. The following is a snippet of the service section in the yaml file:

```
service:
  customExtension:
    labels: {}
    annotations: {}
  type: LoadBalancer
  ports:
    api:
      port: "5001"
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    staticNodePortEnabled: false
    staticNodePort: ""
```

3.1.5.3 Service Account Requirements

To run BSF-ATS, use the following rules to create a service account:

```
rules:
- apiGroups: ["extensions"]
  resources: ["deployments", "replicasets"]
  verbs: ["watch", "get", "list", "update"]
- apiGroups: ["apps"]
  resources: ["deployments", "replicasets"]
  verbs: ["watch", "get", "list", "update"]
- apiGroups: [""]
  resources: ["pods", "services", "secrets", "configmaps"]
  verbs: ["watch", "get", "list", "delete", "update", "create"]
- apiGroups: [""]
  resources: ["pods/log"]
  verbs: ["get", "list"]
```

3.1.5.4 Enabling Aspen Service Mesh

This section provides information on how to enable Aspen service mesh while deploying ATS for Binding Support Function. The configurations mentioned in this section are optional and should be performed only if ASM is required.

To enable service mesh for BSF ATS, perform the following steps:

1. In the **service** section of the `values.yaml` file, the **serviceMeshCheck** parameter is set to `false` by default. To enable service mesh, set the value for **serviceMeshCheck** to `true`. The following is a snippet of the service section in the yaml file:

```
service:
  customExtension:
    labels: {}
    annotations: {}
  type: LoadBalancer
  ports:
    https:
      port: "8443"
      staticNodePortEnabled: false
      staticNodePort: ""
    http:
      port: "8080"
      staticNodePortEnabled: false
      staticNodePort: ""
    api:
      port: "5001"
      staticNodePortEnabled: false
      staticNodePort: ""
  serviceMeshCheck: true
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

2. If the ASM is not enabled on the global level for the namespace, run the following command to enable it before deploying the ATS:

   ```
   kubectl label --overwrite namespace <namespace_name> istio-
   injection=enabled
   ```

   For example:

   ```
   kubectl label --overwrite namespace ocbsf istio-injection=enabled
   ```

3. Uncomment and add the following annotation under the **lbDeployments** and **nonlbDeployments** section of the global section in `values.yaml` file as follows:
   `traffic.sidecar.istio.io/excludeInboundPorts: "9000"`

   `traffic.sidecar.istio.io/excludeOutboundPorts: "9000"`

   The following is a snippet from the `values.yaml` of BSF:

   ```
   /home/cloud-user/ocats-bsf/ocats-bsf-tools-25.1.200.0.0/ocats-bsf-
   pkg-25.1.200.0.0/ocats-bsf/
   vim values.yaml

    customExtension:
       allResources:
         labels: {}
         annotations: {
         #Enable this section for service-mesh based installation
            traffic.sidecar.istio.io/excludeInboundPorts: "9000",
            traffic.sidecar.istio.io/excludeOutboundPorts: "9000"
           }


   lbDeployments:
         labels: {}
         annotations: {
           traffic.sidecar.istio.io/excludeInboundPorts: "9000",
           traffic.sidecar.istio.io/excludeOutboundPorts: "9000"}
   ```

4. If service mesh is enabled, then create a destination rule for fetching the metrics from the Prometheus. In most of the deployments, Prometheus is kept outside the service mesh so you need a destination rule to communicate between TLS enabled entity (ATS) and non-TLS entity (Prometheus). You can create a destination rule using the following sample yaml file:

   ```
   kubectl apply -f - <<EOF

   apiVersion: networking.istio.io/v1alpha3
   kind: DestinationRule
   metadata:
     name: prometheus-dr
     namespace: ocats
   spec:
     host: oso-prometheus-server.pcf.svc.cluster.local
     trafficPolicy:
       tls:
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
      mode: DISABLE
EOF
```

In the destination rule:

- **name** indicates the name of destination rule.
- **namespace** indicates where the ATS is deployed.
- **host** indicates the hostname of the prometheus server.

5. Update the `ocbsf_custom_values_servicemesh_config_25.1.200.yaml` with the below additional configuration under virtualService section for Egress Gateway:

```
virtualService:
  - name: nrfvirtual1
    host: ocbsf-ocbsf-egress-gateway
    destinationhost: ocbsf-ocbsf-egress-gateway
    port: 8000
    exportTo: |-
      [ "." ]
    attempts: "0"
```

Where,

host or destination name uses the format - <release_name>-<egress_svc_name>.

You must update the host or destination name as per the deployment.

6. For ServerHeader and SessionRetry features, the user needs to perform the following configurations under the envoyFilters for nf1stub, nf11stub, and nf12stub in the `ocbsf-servicemesh-config-custom-values-25.1.200.yaml`:

> **Note**
>
> occnp_custom_values_servicemesh_config yaml file and helm charts version names would differ based on the deployed BSF NF version. For example, "occnp_custom_values_servicemesh_config_24.3.0.yaml" or "occnp_custom_values_servicemesh_config_24.3.1.yaml".

```
envoyFilters:
  - name: serverheaderfilter-nf1stub
    labelselector: "app: nf1stub-ocstub-py"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionM
anager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
  - name: serverheaderfilter-nf11stub
    labelselector: "app: nf11stub-ocstub-py"
    configpatch:
      - applyTo: NETWORK_FILTER
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionM
anager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
  - name: serverheaderfilter-nf12stub
    labelselector: "app: nf12stub-ocstub-py"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionM
anager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
```

7. Perform helm upgrade on the ocbsf-servicemesh-config release using the modified `ocbsf_custom_values_servicemesh_config_25.1.200.yaml` file.

   ```
   helm upgrade <helm_release_name_for_servicemesh> -n <namespace>
   <servicemesh_charts> -f <servicemesh-custom.yaml>
   ```

   For example,

   ```
   helm upgrade ocbsf-servicemesh-config ocbsf-servicemesh-
   config-25.1.200.tgz -n ocbsf -f
   ocbsf_custom_values_servicemesh_25.1.200.yaml
   ```

8. Configure DNS for Alternate Route service. For more information, see Post-Installation Steps.

3.1.5.5 Enabling Health Check

This section describes how to enable Health Check for ATS.

To enable Health Check, in the `ocats-bsf/values.yaml` file, set the value of `healthcheck` parameter to `true` and enter a valid value to select either **Webscale** or **OCCNE** environment.

To select OCCNE environment, set the `envtype` to **OCCNE** and update the values of the following parameters:

- `Webscale` - Update the value as false
- `envtype` - T0NDTkU= (i.e envtype=$(echo -n '**OCCNE**' | base64))
- `occnehostip` - OCCNE Host IP address
- `occnehostusername` - OCCNE Host Username
- `occnehostpassword` - OCCNE Host Password

To select OCCNE environment, update the values of the following two parameters:

- `Webscale` - Update the value as true

Copyright © 2019, 2025, Oracle and/or its affiliates.

- `envtype` - T0NDTkU= (i.e envtype=$(echo -n 'OCCNE' | base64))

After the configurations are done, encrypt the parameters and provide the values as shown in the following snippet:

```
atsFeatures:  ## DO NOT UPDATE this section without Engineering team's
permission
  healthcheck: true                   # TO enable/disable ATS Health Check.

sshDetails:
  secretname: "healthchecksecret"
  envtype: "T0NDTkU="
  occnehostip: "MTAuMTcuMjE5LjY1"
  occnehostusername: "dXNlcm5hbWU"
  occnehostpassword: "KioqKg=="
```

To select WEBSCALE environment, update the values of the following two parameters:

- `Webscale` - Update the value as true
- `envtype` - V0VCU0NBTEU= (i.e envtype=$(echo -n '**WEBSCALE**' | base64))

After the configurations are done, encrypt the parameters and provide the values as shown in the following snippet:

```
atsFeatures:  ## DO NOT UPDATE this section without Engineering team's
permission
  healthcheck: true                   # TO enable/disable ATS Health Check.

sshDetails:
  secretname: "healthchecksecret"
  envtype: "V0VCU0NBTEU="
  webscalejumpip: "MTAuNzAuMTE3LjQy"
  webscalejumpusername: "dXNlcm5hbWU="
  webscalejumppassword: "KioqKg=="
  webscaleprojectname: "KioqKg=="
  webscalelabserverFQDN: "KioqKg=="
  webscalelabserverport: "KioqKg=="
  webscalelabserverusername: "KioqKg=="
  webscalelabserverpassword: "KioqKg=="
```

> **Note**
>
> Once the ATS is deployed with HealthCheck feature enabled or disabled, then it cannot be changed. To change the configuration, you are required to re-install.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.1.5.6 Enabling Persistent Volume

> **Note**
>
> The steps provided in this section are optional and required only if Persistent Volume needs be to enabled.

ATS supports Persistent storage to retain ATS historical build execution data, test cases and one-time environment variable configurations. With this enhancement, the user can decide whether to use persistent volume based on their resource requirements. By default, the persistent volume feature is not enabled.

To enable persistent storage, perform the following steps:

1. Create a PVC using PersistentVolumeClaim.yaml file and associate the same to the ATS pod.
   Sample PersistentVolumeClaim.yaml file:

   ```
   apiVersion: v1
   kind: PersistentVolumeClaim
   metadata:
     name: <Enter the PVC Name>
     annotations:
   spec:
     storageClassName: <Provide the Storage Class Name>
     accessModes:
       - ReadWriteOnce
     resources:
       requests:
         storage: <Provide the size of the PV>
   ```

   a. Set **PersistentVolumeClaim** to the PVC file name.
   b. Enter the **storageClassName** to the Storage Class Name.
   c. Set **storage** to and size of the persistent volume.
      Sample PVC configuration:

      ```
      apiVersion: v1
      kind: PersistentVolumeClaim
      metadata:
        name: bsf-pvc-25.1.200
      annotations:
      spec:
        storageClassName: standard
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 1Gi
      ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

2. Run the following command to create PVC:

   ```
   kubectl apply -f <filename> -n <namespace>
   ```

   For example:

   ```
   kubectl apply -f PersistentVolumeClaim.yaml -n ocbsf
   ```

   Output:

   ```
   persistentvolumeclaim/bsf-pvc-25.1.200 created
   ```

3. Once the PVC is created, run the following command to verify that it is bound to the namespace and is available.

   ```
   kubectl get pvc -n <namespace used for pvc creation>
   ```

   For example:

   ```
   kubectl get pvc -n ocbsf
   ```

   Sample output:

   ```
   NAME              STATUS        VOLUME
   CAPACITY  ACCESS MODES   STORAGECLASS   AGE
   bsf-pvc-25.1.200   Bound    pvc-65484045-3805-4064-9fc3-f9eeeaccc8b8
   1Gi        RWO            standard      11s
   ```

   Verify that the `STATUS` is `Bound` and rest of the parameters like `NAME`, `CAPACITY`, `ACCESS MODES`, and `STORAGECLASS` are as mentioned in the `PersistentVolumeClaim.yaml` file.

   > **Note**
   >
   > Do not proceed further with the next step if there is an issue with the PV creation and contact your administrator to get the PV Created.

4. Enable PVC:
   a. Set the **PVEnabled** flag to **true**.
   b. Set **PVClaimName** to the PVC created in Step 1.

   ```
   PVEnabled: true
   PVClaimName: "ocbsf-pvc-25.1.200"
   ```

   > **Note**
   >
   > Make sure that ATS is deployed before proceeding to the further steps.

Copyright © 2019, 2025, Oracle and/or its affiliates.

5. Copy the `<nf_main_folder>` and `<jenkins jobs>` folders from the tar file to their ATS pod and restart the pod.
   a. Extract the tar file.

      ```
      tar -xvf ocats-bsf-data-25.1.200.tgz
      ```

   b. Run the following commands to copy the desired folder.

      ```
      kubectl cp ocats-bsf-data-25.1.200/ocbsf_tests <namespace>/<pod-name>:/var/lib/jenkins/

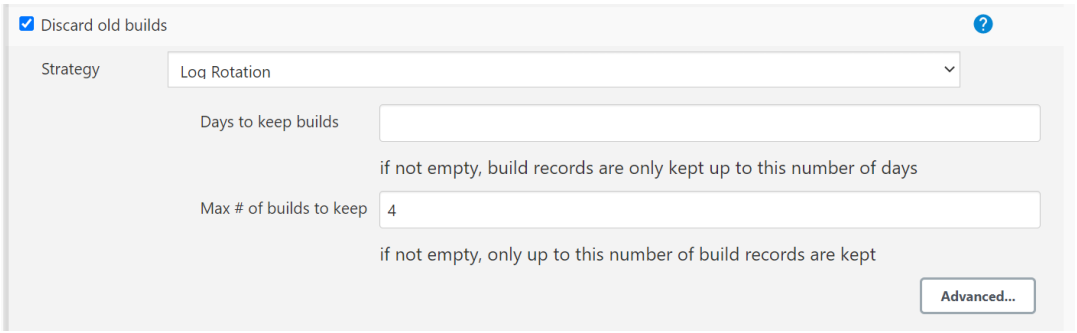

      kubectl cp ocats-bsf-data-25.1.200/jobs <namespace>/<pod-name>:/var/lib/jenkins/.jenkins/
      ```

   c. Restart the pod.

      ```
      kubectl delete po <pod-name> -n <namespace>
      ```

6. Once the Pod is up and running, log in to the Jenkins console and configure the **Discard old Builds** option to configure the number of Jenkins builds, which must be retained in the persistent volume.

Figure 3-1 Discarding Old Builds



> **Note**
>
> If **Discard old Builds** is not configured, Persistent Volume can get filled when there are huge number of builds.

For more details on Persistent Volume Storage, see Persistent Volume for 5G ATS.

3.1.5.7 ATS-BSF API Extended Support

The ATS application programming interface (API) feature provides APIs, to perform routine ATS tasks such as starting the ATS suite, monitoring and stopping the ATS suite etc.

Copyright © 2019, 2025, Oracle and/or its affiliates.

By default, this feature is enabled in `values.yaml` file.

```
atsFeatures:
  atsApi: true
```

For more details about the ATS API feature, refer to ATS API section.

This ATS feature is extended to provide the ability of running single features, or scenarios, or stages, or groups, or based on tags execution using the API. This also allows running of test cases by providing the features, or scenarios, or stages, or groups, or tags in the `curl` request to the server.

For more details about the API interfaces, refer to Use the RESTful Interfaces section.

3.1.6 Deploying ATS and Pods

3.1.6.1 Deploying ATS in Kubernetes Cluster

> **Important**
>
> This Procedure is for Backwards porting purpose only and should not be considered as the Subsequent Release POD Deployment Procedure.

Prerequisite: Make sure that the old PVC, which contains the old release POD data is available.

To deploy ATS, perform the following steps:

1. Run the following command to deploy ATS using the updated helm charts:

   > **Note**
   >
   > Ensure that all the the components, that is, ATS, stub pods and CNC BSF are deployed in the same namespace.

   Using Helm

   ```
   helm install -name <release_name> ocats-bsf-25.1.200.tgz --namespace
   <namespace_name> -f <values-yaml-file>
   ```

   For example:

   ```
   helm install -name ocats ocats-bsf-25.1.200.tgz --namespace ocbsf -f ocats-
   bsf/values.yaml
   ```

2. Run the following command to verify ATS deployment:

   ```
   helm ls -n ocbsf
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

The output of the command is as follows:

```
NAME                    REVISION        UPDATED
STATUS          CHART                   APP VERSION     NAMESPACE
ocats                   1               Mon Nov 14 14:56:11 2020
DEPLOYED        ocats-bsf-25.1.200      25.1.200.0.0       ocbsf
```

If the deployment is successful, the status is **Deployed**.

3.1.6.2 Deploying Stub Pod in Kubernetes Cluster

To deploy Stub Pod in Kubernetes cluster, perform the following steps:

1. Navigate to `ocats-bsf-tools-25.1.200.0.0` folder and run the following command:

   ```
   tar -zxvf ocstub-pkg-25.1.201.0.0.tgz
   ```

   The output of the command shows:

   - `ocstub-py-25.1.201.tgz`
   - `ocstub-py-image-25.1.201.tar`

2. Deploy the additional stubs required to validate the session retry feature.
   You can use `nf11stub` or `nf12stub` as alternte FQDN for `nf1stub`.

   a. Run the following command to load the stub image.

      ```
      docker load --input ocstub-py-image-25.1.201.tar
      ```

   b. Tag and push the image to your docker registry using below commands.

      ```
      docker tag ocstub-py:25.1.201 localhost:5000/ocstub-py:25.1.201
      docker push localhost:5000/ocstub-py:25.1.201
      ```

   c. Untar the helm charts `ocstub-py-25.1.201.tgz` and update the registry name, image name and tag (if required) in `ocstub-py/values.yaml` file.

      ```
      tar -zxvf ocstub-py-25.1.201.0.0.tgz
      ```

> **Note**
>
> From 24.2.0 onwards, service port names are configurable in ocstub-py. But as per Istio standard, it's advisable to keep the default values as it as.
> Example:
>
> ```
> names:
>   http: "http"
>   h2c: "http2-h2c"
>   h2: "http2-h2"
> ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

d. If required, change `apiVersion` to `apps/v1` in `ocstub-py/templates/deployment.yaml` file.

```
apiVersion: apps/v1
```

> **Note**
>
> If the support for Predefined_priming feature is required, perform the following steps to configure Predefined_priming.

e. Copy `ocstub-py/values.yaml` file to a new file with name `pre_priming_values.yaml`.

f. Edit the `ocstub-py/pre_priming_values.yaml` file.

g. Set the value of `preConfig` flag to `true` and replace the default configuration with below configurations under `predefined_prime_configuration` section.
Predefined_priming configuration:

```
preConfig:
  enabled: true
  predefined_prime_configuration: |+
    [
      {
        "method": "GET",
        "statuscode": "200",
        "url": "/nnrf-nfm/v1/nf-instances/fe7d992b-0541-4c7d-ab84-
c6d70b1b0666",
        "data": "{\"nfInstanceId\": \"fe7d992b-0541-4c7d-ab84-
c6d70b1b0666\", \"nfType\": \"BSF\", \"nfStatus\": \"REGISTERED\",
\"heartBeatTimer\": 2, \"fqdn\": \"ocbsf1-2-api-
gateway.bsf1-2.svc.atlantic.morrisville.us.lab.oracle.com\",
\"priority\": 1, \"capacity\": 1, \"load\": 2, \"bsfInfo\":
{\"ipv4AddressRanges\": [{\"start\": \"10.0.0.1\", \"end\":
\"10.113.255.255\"}], \"ipv6PrefixRanges\": [{\"start\":
\"2800:a00:cc03::/64\", \"end\": \"2800:a00:cc04::/64\"}]},
\"nfServices\": [{\"serviceInstanceId\": \"03063893-
cf9e-4f7a-9827-111111111111\", \"serviceName\": \"nbsf-management\",
\"versions\": [{\"apiVersionInUri\": \"v1\", \"apiFullVersion\":
\"1.0.0\", \"expiry\": \"2019-08-03T18:66:08.871+0000\"}], \"scheme\":
\"http\", \"nfServiceStatus\": \"REGISTERED\", \"fqdn\": \"ocbsf1-2-api-
gateway.bsf1-2.svc.atlantic.morrisville.us.lab.oracle.com\",
\"interPlmnFqdn\": null, \"ipEndPoints\": [{\"ipv4Address\":
\"10.233.22.149\", \"transport\": \"TCP\", \"port\": 80}],
\"apiPrefix\": null, \"allowedNfTypes\": [\"PCF\", \"AF\", \"NEF\"],
\"priority\": 1, \"capacity\": 1, \"load\": 2}]}",
        "headers": "{\"Content-Type\": \"application/json\"}"
      },
      {
        "method": "PUT",
        "statuscode": "201",
        "url": "/nnrf-nfm/v1/nf-instances/fe7d992b-0541-4c7d-ab84-
c6d70b1b0666",
        "data": "{\"nfInstanceId\": \"fe7d992b-0541-4c7d-ab84-
c6d70b1b0666\", \"nfType\": \"BSF\", \"nfStatus\": \"REGISTERED\",
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
\"heartBeatTimer\": 30, \"fqdn\": \"ocbsf1-2-api-
gateway.bsf1-2.svc.atlantic.morrisville.us.lab.oracle.com\",
\"priority\": 1, \"capacity\": 1, \"load\": 2, \"bsfInfo\":
{\"ipv4AddressRanges\": [{\"start\": \"10.0.0.1\", \"end\":
\"10.113.255.255\"}], \"ipv6PrefixRanges\": [{\"start\":
\"2800:a00:cc03::/64\", \"end\": \"2800:a00:cc04::/64\"}]},
\"nfServices\": [{\"serviceInstanceId\": \"03063893-
cf9e-4f7a-9827-111111111111\", \"serviceName\": \"nbsf-management\",
\"versions\": [{\"apiVersionInUri\": \"v1\", \"apiFullVersion\":
\"1.0.0\", \"expiry\": \"2019-08-03T18:66:08.871+0000\"}], \"scheme\":
\"http\", \"nfServiceStatus\": \"REGISTERED\", \"fqdn\": \"ocbsf1-2-api-
gateway.bsf1-2.svc.atlantic.morrisville.us.lab.oracle.com\",
\"interPlmnFqdn\": null, \"ipEndPoints\": [{\"ipv4Address\":
\"10.233.22.149\", \"transport\": \"TCP\", \"port\": 80}],
\"apiPrefix\": null, \"allowedNfTypes\": [\"PCF\", \"AF\", \"NEF\"],
\"priority\": 1, \"capacity\": 1, \"load\": 2}]}",
        "headers": "{\"Content-Type\": \"application/json\"}"
      },
      {
        "method": "PATCH",
        "statuscode": "204",
        "url": "/nnrf-nfm/v1/nf-instances/fe7d992b-0541-4c7d-ab84-
c6d70b1b0666",
        "data": "{}",
        "headers": "{\"Content-Type\": \"application/json\"}"
      },
      {
        "method": "POST",
        "statuscode": "201",
        "url": "/nnrf-nfm/v1/subscriptions",
        "data": "{\"nfStatusNotificationUri\": \"http://ocbsf-ocbsf-
ingress-gateway.ocpcf.svc/nnrf-client/v1/notify\", \"reqNfType\":
\"BSF\", \"subscriptionId\": \"2d77e0de-15a9-11ea-8c5b-b2ca002e6839\",
\"validityTime\": \"2050-12-26T09:34:30.816Z\"}",
        "headers": "{\"Content-Type\": \"application/json\"}"
     }
    ]
```

> **Note**
>
> - The `predefined_prime_configuration` contains variables such as `nfInstanceId`, `nfType`, and `fqdn` in the data's content. Make sure to verify and update the variables based on the payload message that must be included in the response from the NRF on a request.
> - The default value of `nfInstanceId` variable is `fe7d992b-0541-4c7d-ab84-c6d70b1b0666`.

h. Deploy the stub:

```
helm install -name <release_name> ocstub-py --set env.NF=<NF> --
setenv.LOG_LEVEL=<DEBUG/INFO> --set service.name=<service_name> --set
service.appendReleaseName=false --namespace=<namespace_name> -f
<valuesyaml-file>
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Install nf1stub and nf11stub with updated ocstub-py/*pre_priming_values.yaml* file.

```
helm install -name nf1stub ocstub-py --set env.NF=BSF --set
env.LOG_LEVEL=DEBUG --set service.name=nf1stub --set
service.appendReleaseName=false  --namespace=ocbsf -f ocstub-py/
pre_priming_values.yaml

helm install -name nf11stub ocstub-py --set env.NF=BSF --set
env.LOG_LEVEL=DEBUG --set service.name=nf11stub --set
service.appendReleaseName=false  --namespace=ocbsf -f ocstub-py/
pre_priming_values.yaml
```

Install nf12stub with default values.yaml.

```
helm install -name nf12stub ocstub-py --set env.NF=BSF --set
env.LOG_LEVEL=DEBUG --set service.name=nf12stub --set
service.appendReleaseName=false  --namespace=ocbsf -f ocstub-py/
values.yaml
```

If the support for Predefined_priming feature is not required, helm installation must be performed using `default values.yaml` file.

```
helm install -name <release_name> ocstub-py --set env.NF=<NF> --set
env.LOG_LEVEL=<DEBUG/INFO> --set service.name=<service_name>--set
service.appendReleaseName=false  --namespace=<namespace_name> -f
<valuesyaml-file>
```

For example,

```
helm install -name nf1stub ocstub-py --set env.NF=BSF --set
env.LOG_LEVEL=DEBUG --set service.name=nf1stub --set
service.appendReleaseName=false  --namespace=ocbsf -f ocstub-py/
values.yaml

helm install -name nf11stub ocstub-py --set env.NF=BSF --set
env.LOG_LEVEL=DEBUG --set service.name=nf11stub --set
service.appendReleaseName=false  --namespace=ocbsf -f ocstub-py/
values.yaml

helm install -name nf12stub ocstub-py --set env.NF=BSF --set
env.LOG_LEVEL=DEBUG --set service.name=nf12stub --set
service.appendReleaseName=false  --namespace=ocbsf -f ocstub-py/
values.yaml
```

i. Run the following command to verify the stub deployment:

```
helm ls -n ocbsf
```

Sample output:

```
NAME                    REVISION            UPDATED
STATUS          CHART                   APP VERSION
NAMESPACE
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
nf11stub            1               Thu Jul  29 05:55:48 2024
DEPLOYED        ocstub-py-25.1.201      25.1.201.0.0            ocbsf
nf12stub            1               Thu Jul  29 05:55:50 2024
DEPLOYED        ocstub-py-25.1.201      25.1.201.0.0            ocbsf
nf1stub             1               Thu Jul  29 05:55:47 2024
DEPLOYED        ocstub-py-25.1.201      25.1.201.0.0            ocbsf
```

j. Run the following command to verify the ATS and Stubs deployment status:

```
helm status -n ocbsf
```

k. Run the following command to verify if all the services are installed.

```
kubectl get po -n ocbsf
```

Sample output:

```
NAME                                                    READY
STATUS    RESTARTS   AGE
nf11stub-ocstub-py-7bffd6dcd7-ftm5f                   1/1     Running
0          3d23h
nf12stub-ocstub-py-547f7cb99f-7mpll                   1/1     Running
0          3d23h
nf1stub-ocstub-py-bdd97cb9-xjrkx                      1/1     Running
0          3d23h
```

3. Verify the changes related to stub predefined prime configuration.
   a. Run the following command to verify the status of all the config-map.

```
kubectl get cm -n ocbsf
```

Notice the change in the number of config-map counts. It includes two extra config-maps of stubs and the number will be the same as of stubs count.

For example:

```
NAME                  DATA   AGE
cm-pystub-nf1stub                1      3h35m
cm-pystub-nf11stub               1      3h35m
```

Updating the Predefined_priming configurations

To update the Predefined_priming configurations such as value of nfInstanceID in the url:

> **Note**
>
> This procedure is applicable only when Predefined_priming configuration is enabled.

1. Run the following command to verify the status of all the config-maps.

```
kubectl get cm -n ocbsf
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

2. Perform the following steps separately for nf1stub and nf11stub pods.
   a. Edit the config-map of the pod.

      To edit the config-map of nf1stub,

      ```
      kubectl edit cm cm-pystub-nf1stub -n ocbsf
      ```

      To edit the config-map of nf11stub,

      ```
      kubectl edit cm cm-pystub-nf11stub -n ocbsf
      ```

   b. Edit the configurations as required, save and close the config-maps.
   c. Restart the nf1stub and nf11stub pods.
   d. Verify the logs of both these pods to confirm the changes.

3.1.6.3 Deploying DNS Stub in Kubernetes Cluster

> **Note**
>
> Ensure there are sufficient resources and limit for DNS Stub. Set the resource request and limit values in the **resources** section of the **values.yaml** file as follows:
>
> ```
> resources: {}
>   # We usually recommend not to specify default resources and to leave
> this as a conscious
>   # choice for the user. This also increases chances charts run on
> environments with little
>   # resources, such as Minikube. If you do want to specify resources,
> uncomment the following
>   # lines, adjust them as necessary, and remove the curly braces after
> 'resources:'.  # limits:
>   #  cpu: 1000m
>   #  memory: 1024Mi
>   # requests:
>   #  cpu: 500m
>   #  memory: 500Mi
> ```

To deploy DNS Stub in Kubernetes cluster, perform the following steps:

1. Go to the `ocats-bsf-tools-25.1.200.0.0` folder and run the following command to extract the ocstub tar file content:
   `tar -zxvf ocdns-pkg-25.1.203.0.0.tgz`

   Sample output:

   ```
   [cloud-user@platform-bastion-1 ocdns-pkg-25.1.203.0.0]$ ls -ltrh
   total 211M
   -rw-------. 1 cloud-user cloud-user 211M Mar 14 14:49 ocdns-bind-
   image-25.1.203.tar
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
-rw-r--r--. 1 cloud-user cloud-user 2.9K Mar 14 14:49 ocdns-
bind-25.1.203.tgz
```

2. Run the following command in your cluster to load the DNS STUB image:
   `docker load --input ocdns-bind-image-25.1.203.tar`
3. Run the following commands to tag and push the DNS STUB image:

   ```
   docker tag ocdns-bind:25.1.203 localhost:5000/ocdns-bind:25.1.203
   docker push localhost:5000/ocdns-bind:25.1.203
   ```

4. Run the following command to untar the helm charts, `ocdns-bind-25.1.203.tgz`.
   `tar -zxvf ocdns-bind-25.1.203.tgz`
5. Update the registry name, image name and tag (if required) in the `ocdns-bind/values.yaml` file as required. For this, open the **values.yaml** file and update the `image.repository` and `image.tag` parameters.
6. Run the following command to deploy the DNS Stub.
   Using Helm:

   ```
   helm install -name ocdns ocdns-bind-25.1.203.tgz --namespace ocbsf -f
   ocdns-bind/values.yaml
   ```

7. Capture the `cluster name` of the deployment, `namespace` where nfstubs are deployed, and the cluster IP of DNS Stub.

   To capture the DNS Stub cluster IP:

   ```
   kubectl get svc -n ocbsf | grep dns
   ```

   Sample output:

   ```
   [cloud-user@platform-bastion-1 ocdns-pkg-25.1.203.0.0]$ kubectl get svc -n
   ocbsf | grep dns
   NAME      TYPE           CLUSTER-IP      EXTERNAL-IP
   PORT(S)               AGE
   ocdns     ClusterIP      10.233.11.45    <none>          53/
   UDP,6236/TCP        19h
   ```

   To capture the cluster name:

   ```
   kubectl -n kube-system get configmap kubeadm-config -o yaml | grep
   clusterName
   ```

   Sample output:

   ```
   clusterName: platform
   ```

3.1.6.4 Deploying ocdiam Simulator in Kubernetes Cluster

To deploy ocdiam Simulator in Kubernetes cluster, perform the following steps:

Copyright © 2019, 2025, Oracle and/or its affiliates.

1. Go to the `ocats-bsf-tools-25.1.200.0.0` folder and run the following command to extract the ocstub tar file content:
   `tar -zxvf ocdiam-pkg-25.1.203.0.0.tgz`

   Sample output:

   ```
   [cloud-user@platform-bastion-1 ocdiam-pkg-25.1.203.0.0]$ ls -ltrh
   total 908M
   -rw-------. 1 cloud-user cloud-user 908M Mar 14 14:49 ocdiam-sim-
   image-25.1.203.tar
   -rw-r--r--. 1 cloud-user cloud-user 3.8K Mar 14 14:49 ocdiam-
   sim-25.1.203.tgz
   ```

2. Run the following command in your cluster to load the Diameter Simulator image:

   ```
   docker load --input ocdiam-sim-image-25.1.203.tar
   ```

3. Run the following commands to tag and push the Diameter Simulator image:

   ```
   docker tag ocdiam-sim:25.1.203 localhost:5000/ocdiam-sim:25.1.203
   docker push localhost:5000/ocdiam-sim:25.1.203
   ```

4. Run the following command to untar the helm charts, `ocdiam-sim-25.1.203.tgz`.

   ```
   tar -zxvf ocdiam-sim-25.1.203.tgz
   ```

5. Update the registry name, image name and tag (if required) in the `ocdiam-sim/values.yaml` file as required. For this, open the **values.yaml** file and update the `image.repository` and `image.tag` parameters.
6. Run the following command to deploy the Diameter Simulator.
   Using Helm:

   ```
   helm install -name ocdiam-sim ocdiam-sim --namespace ocbcf -f ocdiam-sim/
   values.yaml
   ```

   Output:

   ```
   ocdiam-sim-69968444b6-fg6ks                1/1     Running   0
   5h47m
   ```

Sample of BSF namespace with BSF and ATS after installation:

```
[cloud-user@platform-bastion-1 ocstub-pkg-25.1.201.0.0]$ kubectl get po -n
ocbsf
NAME                                                    READY   STATUS
RESTARTS   AGE
ocbsf-appinfo-6fc99ffb85-f96j2                         1/1     Running
1          3d23h
ocbsf-bsf-management-service-df6b68d75-m77dv           1/1     Running
0          3d23h
ocbsf-oc-config-79b5444f49-7pwzx                       1/1     Running
0          3d23h
ocbsf-oc-diam-connector-77f7b855f4-z2p88               1/1     Running
0          3d23h
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
ocbsf-oc-diam-gateway-0                                1/1     Running
0          3d23h
ocbsf-ocats-bsf-5d8689bc77-cxdvx                       1/1     Running
0          3d23h
ocbsf-ocbsf-egress-gateway-644555b965-pkxsb            1/1     Running
0          3d23h
ocbsf-ocbsf-ingress-gateway-7558b7d5d4-lfs5s           1/1     Running
4          3d23h
ocbsf-ocbsf-nrf-client-nfmanagement-d6b955b48-4pptk    1/1     Running
0          3d23h
ocbsf-ocdns-ocdns-bind-75c964648-j5fsd                 1/1     Running
0          3d23h
ocbsf-ocpm-cm-service-7775c76c45-xgztj                 1/1     Running
0          3d23h
ocbsf-ocpm-queryservice-646cb48c8c-d72x4               1/1     Running
0          3d23h
ocbsf-performance-69fc459ff6-frrvs                     1/1     Running
4          3d23h
ocbsfnf11stub-7bffd6dcd7-ftm5f                         1/1     Running
0          3d23h
ocbsfnf12stub-547f7cb99f-7mpll                         1/1     Running
0          3d23h
ocbsfnf1stub-bdd97cb9-xjrkx                            1/1     Running
0          3d23h
ocdiam-sim-69968444b6                                  1/1     Running
0          3d23h
```

3.1.7 Post-Installation Steps

The section describes post-installation steps that users should perform after deploying ATS and stub pods.

Alternate Route Service Configurations

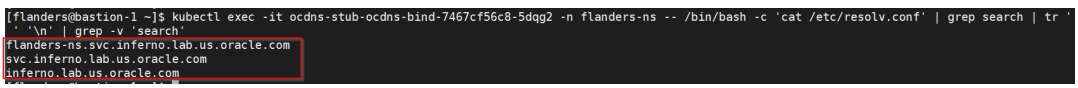
To edit the Alternate Route Service deployment file (ocbcf-ocbsf-alternate-route) that points to DNS Stub, perform the following steps:

1. Run the following command to get searches information from dns-bind pod to enable communication between Alternate Route and dns-bind service:

   ```
   kubectl exec -it <dns-bind pod> -n <NAMESPACE> -- /bin/bash -c 'cat /etc/
   resolv.conf' | grep search | tr ' ' '\n' | grep -v 'search'
   ```

   The following output is displayed after running the command:

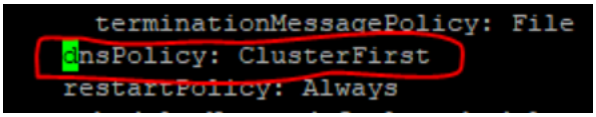
Figure 3-2 Sample Output



By default alternate service will point to CoreDNS and you will see following settings in deployment file:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 3-3 Alternate Route Service Deployment File



2. Run the following command to edit the deployment file and add the following content in alternate service to query DNS stub:

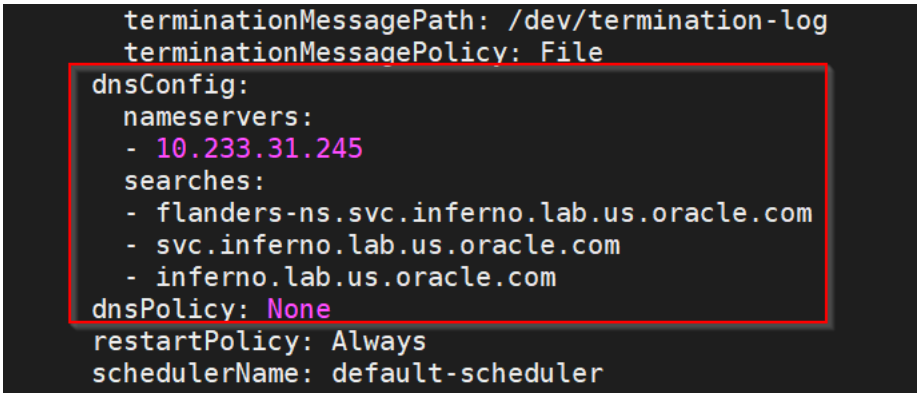
```
$kubectl edit deployment ocpcf-occnp-alternate-route -n ocpcf
```

   a. Add the IP Address of the nameserver that you have recorded after installing the DNS stub (cluster IP Address of DNS Stub).
   b. Add the search information one by one which you recorded earlier.
   c. Set dnsPolicy to "None".

```
dnsConfig:
  nameservers:
  - 10.233.33.169      // cluster IP of DNS Stub
  searches:
  - ocpcf.svc.occne15-ocpcf-ats
  - svc.occne15-ocpcf-ats
  - occne15-ocpcf-ats
dnsPolicy: None
```

For example:

Figure 3-4 Example



NRF client configmap

1. In the **application-config** configmap, configure the following parameters with the respective values:
   - `primaryNrfApiRoot=nf1stub.<namespace_gostubs_are_deployed_in>.svc:8080`
     Example: `primaryNrfApiRoot=nf1stub.ocats.svc:8080`
   - `secondaryNrfApiRoot=nf11stub.<namespace_gostubs_are_deployed_in>.svc:8080`
     Example:`secondaryNrfApiRoot=nf11stub.ocats.svc:8080`
   - `virtualNrfFqdn = nf1stub.<namespace_gostubs_are_deployed_in>.svc`
     Example:`virtualNrfFqdn=nf1stub.ocats.svc`

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> To get all configmaps in your namespace, run the following command:
> `kubectl get configmaps -n <BSF_namespace>`

2. (Optional) If persistent volume is used, follow the post-installation steps provided in the Persistent Volume for 5G ATS section.

3.2 Installing ATS for NRF

3.2.1 Resource Requirements

This section describes the ATS resource requirements for NRF.

Overview - Total Number of Resources

The following table describes the overall resource usage in terms of CPUs, memory, and storage for the following:

- NRF SUT
- cnDBTier
- ATS

Table 3-5 NRF - Total Number of Resources

Resource Name	CPU	Memory (Gi)	Storage (Mi)
NRF SUT Totals	61	69	0
DBTier Totals	40.5	50.5	720
ATS Totals	7	6	0
Grand Total NRF ATS	108.5	125.5	720

NRF Pods Resource Requirements Details

For NRF Pods resource requirements, see *Oracle Communications Cloud Native Core, Network Repository Function Installation, Upgrade, and Fault Recovery Guide*.

ATS Resource Requirements details for NRF

This section describes the ATS resource requirements, which are needed to deploy NRF ATS successfully.

Table 3-6 ATS Resource Requirements Details

Microservice	CPUs Required per Pod	Memory Required per Pod (GB)	Storage PVC Required per Pod (GB)	# Replicas (regular deployment)	# Replicas (ATS deployment)	CPUs Required - Total	Memory Required - Total (GB)	Storage PVC Required - Total (GB)
ATS Behave	2	1	1	1	1	2	1	0
ATS Stub (Python)	1	1	1	1	5	5	5	0

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-6 (Cont.) ATS Resource Requirements Details

Microservice	CPUs Required per Pod	Memory Required per Pod (GB)	Storage PVC Required per Pod (GB)	# Replicas (regular deployment)	# Replicas (ATS deployment)	CPUs Required - Total	Memory Required - Total (GB)	Storage PVC Required - Total (GB)
ATS Totals						7	6	0

cnDBTier Resource Requirements Details for NRF

This section describes the cnDBTier resource requirements, which are needed to deploy NRF ATS successfully.

> **Note**
>
> For cnDBTier pods, a minimum of 4 worker nodes are required.

Table 3-7 cnDBTier Services Resource Requirements

Service Name	Min Pod Replica #	Min CPU/Pod	Min Memory/Pod (in Gi)	PVC Size (in Gi)	Min Ephemeral Storage (Mi)
MGMT (ndbmgmd)	2	4	6	15	90
DB (ndbmtd)	4	5	5	4	90
SQL (ndbmysqld)	2	4	5	8	90
SQL (ndbappmysqld)	2	2	3	1	90
Monitor Service (db-monitor-svc)	1	0.4	490 (Mi)	NA	90
Backup Manager Service (db-backup-manager-svc)	1	0.1	130(Mi)	NA	90
Replication Service - Leader	1	2	2	2	90
Replication Service - Other	0	1	2	0	90

3.2.2 Downloading the ATS Package

Locating and Downloading ATS Images

To locate and download the ATS image from MOS:

1. Log in to My Oracle Support using the appropriate credentials.
2. Select the **Patches & Updates** tab.
3. In the **Patch Search** window, click **Product or Family (Advanced)**.
4. Enter *Oracle Communications Cloud Native Core - 5G* in the **Product** field.
5. Select *Oracle Communications Cloud Native Core Network Repository Function <release_number>* from the **Release** drop-down.
6. Click **Search**. The **Patch Advanced Search Results** list appears.

Copyright © 2019, 2025, Oracle and/or its affiliates.

7. Select the required ATS patch from the list. The Patch Details window appears.
8. Click **Download**. The File Download window appears.
9. Click the **<p********_<release_number>_Tekelec>.zip** file to download the NRF ATS package file.
10. Untar the zip file to access all the ATS images. The **<p********_<release_number>_Tekelec>.zip** directory has the following files:

```
ocats_ocnrf_csar_25_1_203_0_0.zip
ocats_ocnrf_csar_25_1_203_0_0.zip.sha256
ocats_ocnrf_csar_mcafee-25.1.203.0.0.log
```

11. **Note**

    The above zip file contains all the images and custom values required for 25.1.203 release of OCATS-NRF.

    The `ocats_ocnrf_csar_25_1_203_0_0.zip` file has the following files and folders:

```
├─── Definitions
│    ├─── ocats_ocnrf_ats_tests.yaml
│    └─── ocats_ocnrf.yaml
├─── Files
│    ├─── ChangeLog.txt
│    ├─── Helm
│    │    └─── ocats-ocnrf-25.1.203.tgz
│    ├─── Licenses
│    ├─── ocats-nrf-25.1.203.tar
│    ├─── Oracle.cert
│    ├─── ocstub-py-25.1.203.tar
│    └─── Tests
├─── ocats_ocnrf.mf
├─── Scripts
│    ├─── ocats_ocnrf_custom_serviceaccount_25.1.203.yaml
│    ├─── ocats_ocnrf_custom_values_25.1.203.yaml
│    └─── ocats_ocnrf_tests_jenkinsjobs_25.1.203.tgz
└─── TOSCA-Metadata
     └─── TOSCA.meta
```

12. Copy the zip file to Kubernetes cluster where you want to deploy ATS.

3.2.3 Pushing the Images to Customer Docker Registry

Preparing to Deploy ATS and Stub Pod in Kubernetes Cluster

To deploy ATS and Stub Pod in Kubernetes Cluster:

1. Run the following command to extract tar file content:

```
unzip ocats_ocnrf_csar_25_1_203_0_0.zip
```

   The following docker image tar files are located at **Files** folder:

Copyright © 2019, 2025, Oracle and/or its affiliates.

- ocats-nrf-25.1.203.tar
- ocstub-py-25.1.203.tar

2. Run the following commands in your cluster to load the ATS docker image, '`ocats-nrf-25.1.203.tar`' and Stub docker image '`ocstub-py-25.1.203.tar`', and push it to your registry.

```
$ docker load -i ocats-nrf-25.1.203.tar
$ docker load -i ocstub-py-25.1.203.tar

$ docker tag ocats/ocats-nrf:25.1.203 <local_registry>/ocats/ocats-nrf:25.1.203

$ docker tag ocats/ocstub-py:25.1.203 <local_registry>/ocats/ocstub-py:25.1.203

$ docker push <local_registry>/ocats/ocats-nrf:25.1.203

$ docker push <local_registry>/ocats/ocstub-py:25.1.203
```

3. Create a copy of the custom values located at **Scripts/ocats_ocnrf_custom_values_25.1.203.yaml** and update it for image name, tag and other parameters as per the requirement.

3.2.4 Configuring ATS

3.2.4.1 Enabling Static Port

1. To enable static port:

> **Note**
>
> ATS supports static port. By default, this feature is not available.

- In the **ocats-ocnrf-custom-values.yaml** file under service section, set the **staticNodePortEnabled** parameter value to 'true' and **staticNodePort** parameter value with valid nodePort.

```
service:
  customExtension:
    labels: {}
    annotations: {}
  type: LoadBalancer
  port: "8080"
  staticNodePortEnabled: true
  staticNodePort: "32385"
```

3.2.4.2 Enabling Aspen Service Mesh

To enable service mesh for ATS:

Copyright © 2019, 2025, Oracle and/or its affiliates.

1. To enable service mesh, set the value for **serviceMeshCheck** to `true`. The following is a snippet of the service section in the yaml file:

   ```
   ocats-nrf:
      serviceMeshCheck: true
   ```

2. If the ASM is not enabled on the global level for the namespace, run the following command to enable it before deploying the ATS:

   ```
   kubectl label --overwrite namespace <namespace_name> istio-
   injection=enabled
   ```

   For example:

   ```
   kubectl label --overwrite namespace ocnrf istio-injection=enabled
   ```

3. Add the following annotations under the **lbDeployments** and **nonlbDeployments** section of the global section in **ocats-nrf-custom-values.yaml** file for ATS deployment as follows:
   ```
   traffic.sidecar.istio.io/excludeInboundPorts: "8080"
   ```
   ```
   traffic.sidecar.istio.io/excludeOutboundPorts: "9090"
   ```
   For example:

   ```
   lbDeployments:
      labels: {}
      annotations:
        traffic.sidecar.istio.io/excludeInboundPorts: "8080"
        traffic.sidecar.istio.io/excludeOutboundPorts: "9090"
   ```

   ```
   lbDeployments:
     labels: {}
     annotations:
       traffic.sidecar.istio.io/excludeInboundPorts: "8090"
       traffic.sidecar.istio.io/excludeOutboundPorts: "9090"

   nonlbServices:
     labels: {}
     annotations: {}

   nonlbDeployments:
     labels: {}
     annotations:
       traffic.sidecar.istio.io/excludeInboundPorts: "8090"
       traffic.sidecar.istio.io/excludeOutboundPorts: "9090"
   ```

4. Add the following annotations in NRF deployment to work with ATS in service mesh environment:
   For example:

   ```
   oracle.com/cnc: "true"
   traffic.sidecar.istio.io/excludeInboundPorts: "9090,8095,8096,7,53"
   traffic.sidecar.istio.io/excludeOutboundPorts: "9090,8095,8096,7,53"
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    lbDeployments:
      labels: {}
      annotations:
        oracle.com/cnc: "true"
        traffic.sidecar.istio.io/excludeInboundPorts: "9090,8095,8096,7,53"
        traffic.sidecar.istio.io/excludeOutboundPorts:
"9090,8095,8096,7,53"


    nonlbServices:
      labels: {}
      annotations: {}

    nonlbDeployments:
      labels: {}
      annotations:
        oracle.com/cnc: "true"
        traffic.sidecar.istio.io/excludeInboundPorts: "9090,8095,8096,7,53"
        traffic.sidecar.istio.io/excludeOutboundPorts:
"9090,8095,8096,7,53"
```

> **Note**
>
> If the above annotations are not provided in NRF deployment under lbDeployments and nonlbDeployments, all the metrics and alerts related test cases will fail.

3.2.4.3 Enabling Persistent Volume

ATS supports Persistent storage to retain ATS historical build execution data, test cases, and one-time environment variable configurations.

To enable persistent storage:

1. Create a PVC and associate the same to the ATS pod.
2. Set the **PVEnabled** flag to **true**.
3. Set **PVClaimName** to PVC that is created for ATS.

```
deployment:
  customExtension:
    labels: {}
    annotations: {}
  PVEnabled: true
  PVClaimName: "ocats-nrf-25.1.203-pvc"
```

For more details on Persistent Volume Storage, you can refer to Persistent Volume for 5G ATS.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.2.4.4 Enabling NF FQDN Authentication

> **Note**
>
> This procedure is applicable only if the NF FQDN Authentication feature is being tested else, proceed to the "Deploying ATS and Stub in Kubernetes Cluster" section.

You must enable this feature while deploying Service Mesh. For more information on how to enable NF FQDN Authentication feature, see *Oracle Communications Cloud Native Core, Network Repository Function User Guide*.

However, there is some change in the ATS deployment process, which is as follows:

1. Use previously unzipped file **"ocats-nrf-custom-serviceaccount-25.1.203.yaml"** to create a service account. Add the following annotation in the **"ocats-nrf-custom-serviceaccount-25.1.203.yaml"** file where the kind is `ServiceAccount`.
   `"certificate.aspenmesh.io/customFields": '{ "SAN": { "DNS": [ "<NF-FQDN>" ] } }'`

   Sample format:

   ```
   apiVersion: v1
   kind: ServiceAccount
   metadata:
     name: ocats-custom-serviceaccount
     namespace: ocnrf
     annotations:
       "certificate.aspenmesh.io/customFields": '{ "SAN": { "DNS":
   [ "AMF.d5g.oracle.com" ] } }'
   ```

   > **Note**
   >
   > "AMF.d5g.oracle.com" is the NF FQDN that you must provide in the serviceaccount DNS field.

2. Run the following command to create a service account:
   `kubectl apply -f ocats-nrf-custom-serviceaccount-25.1.203.yaml`
3. Update the service account name in the **ocats-ocnrf-custom-values-25.1.203.yaml** file as follows:

   ```
   ocats-nrf:
     serviceAccountName: "ocats-custom-serviceaccount"
   ```

3.2.5 Deploying ATS and Stub in Kubernetes Cluster

> **Note**
>
> It is important to ensure that all the three components; ATS, Stub and NRF are in the same namespace.

Copyright © 2019, 2025, Oracle and/or its affiliates.

ATS and Stub supports Helm3 for deployment.

The NRF release name has to be provided in the `ocats-ocnrf-custom-values.yaml` file under `ocats-nrf.nrfReleaseName`.

```
ocats-nrf:
   nrfReleaseName: "ocnrf"
```

If the namespace does not exists, run the following command to create a namespace:

```
kubectl create namespace ocnrf
```

Using Helm for Deploying ATS:

```
helm install <release_name> ocats-ocnrf-25.1.203.tgz --namespace
<namespace_name> -f <values-yaml-file>
```

Example:

```
helm install ocats ocats-ocnrf-25.1.203.tgz --namespace ocnrf -f ocats-ocnrf-
custom-values.yaml
```

> **Note**
>
> The above `helm install` command will deploy ATS along with the stub servers required for ATS executions, which include 1 ATS pod and 5 stub server pods.

3.2.6 Verifying ATS Deployment

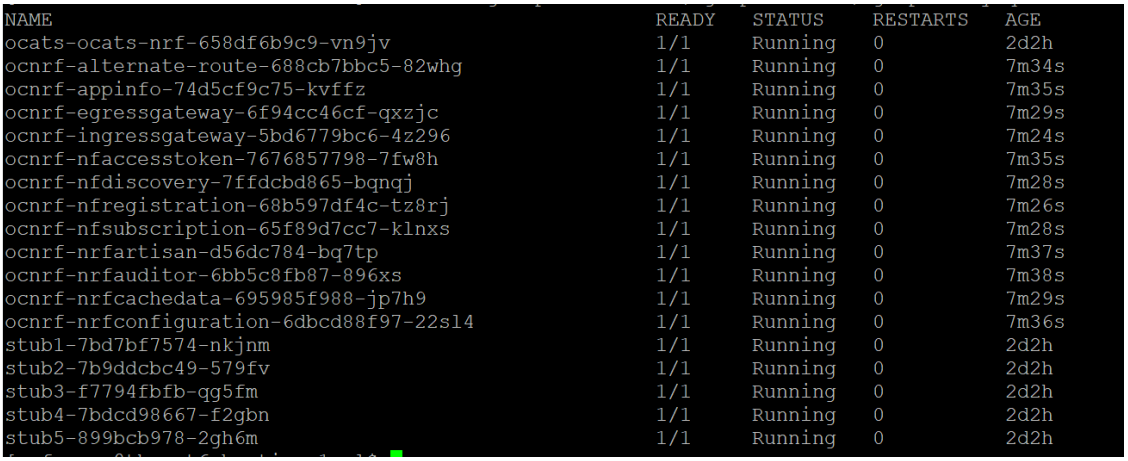
Run the following command to verify ATS deployment.
`helm status <release_name>`

Once ATS and Stub are deployed, run the following command to check the pod and service deployment.

Checking Pod Deployment:
`kubectl get pod -n ocnrf`
Checking Service Deployment:
`kubectl get service -n ocnrf`

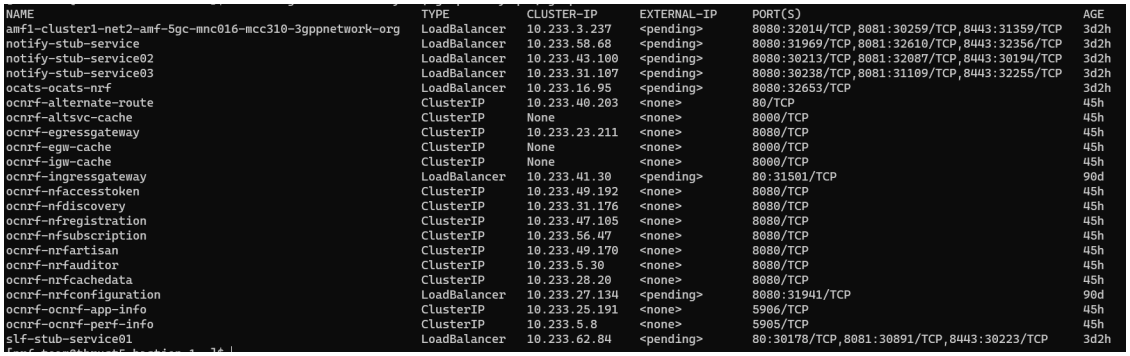
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 3-5 Checking Pod Deployment without Service Mesh



```
NAME                                                  READY   STATUS    RESTARTS   AGE
ocats-ocats-nrf-658df6b9c9-vn9jv                      1/1     Running   0          2d2h
ocnrf-alternate-route-688cb7bbc5-82whg                1/1     Running   0          7m34s
ocnrf-appinfo-74d5cf9c75-kvffz                        1/1     Running   0          7m35s
ocnrf-egressgateway-6f94cc46cf-qxzjc                  1/1     Running   0          7m29s
ocnrf-ingressgateway-5bd6779bc6-4z296                 1/1     Running   0          7m24s
ocnrf-nfaccesstoken-7676857798-7fw8h                  1/1     Running   0          7m35s
ocnrf-nfdiscovery-7ffdcbd865-bqnqj                    1/1     Running   0          7m28s
ocnrf-nfregistration-68b597df4c-tz8rj                 1/1     Running   0          7m26s
ocnrf-nfsubscription-65f89d7cc7-klnxs                 1/1     Running   0          7m28s
ocnrf-nrfartisan-d56dc784-bq7tp                       1/1     Running   0          7m37s
ocnrf-nrfauditor-6bb5c8fb87-896xs                     1/1     Running   0          7m38s
ocnrf-nrfcachedata-695985f988-jp7h9                   1/1     Running   0          7m29s
ocnrf-nrfconfiguration-6dbcd88f97-22sl4               1/1     Running   0          7m36s
stub1-7bd7bf7574-nkjnm                                1/1     Running   0          2d2h
stub2-7b9ddcbc49-579fv                                1/1     Running   0          2d2h
stub3-f7794fbfb-qg5fm                                 1/1     Running   0          2d2h
stub4-7bdcd98667-f2gbn                                1/1     Running   0          2d2h
stub5-899bcb978-2gh6m                                 1/1     Running   0          2d2h
```

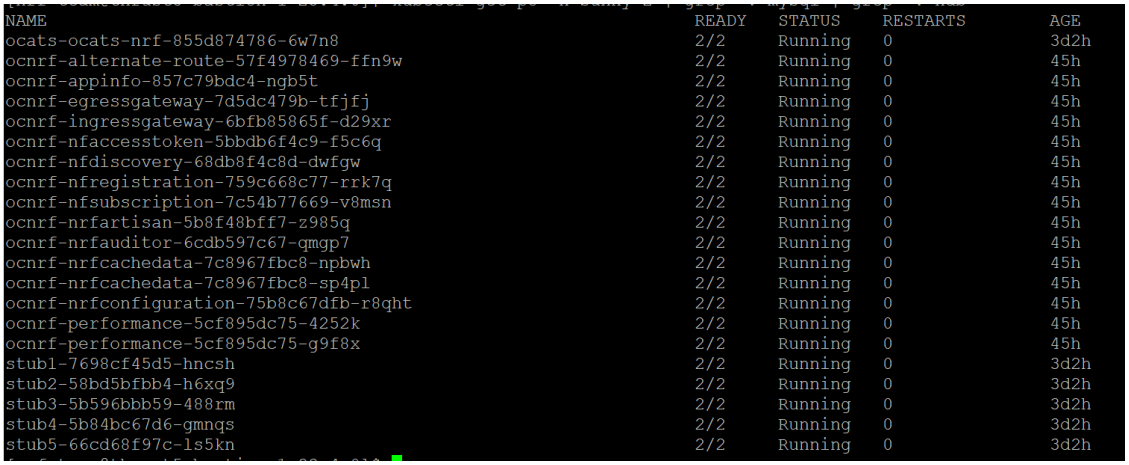
Figure 3-6 Checking Service Deployment without Service Mesh



```
NAME                                                  TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)                                        AGE
amf1-cluster1-net2-amf-5gc-mnc016-mcc310-3gppnetwork-org   LoadBalancer   10.233.3.237    <pending>     8080:32014/TCP,8081:30259/TCP,8443:31359/TCP   3d2h
notify-stub-service                                   LoadBalancer   10.233.58.68    <pending>     8080:31969/TCP,8081:32610/TCP,8443:32356/TCP   3d2h
notify-stub-service02                                 LoadBalancer   10.233.43.100   <pending>     8080:30213/TCP,8081:32087/TCP,8443:30194/TCP   3d2h
notify-stub-service03                                 LoadBalancer   10.233.31.107   <pending>     8080:30238/TCP,8081:31109/TCP,8443:32255/TCP   3d2h
ocats-ocats-nrf                                       LoadBalancer   10.233.16.95    <pending>     8080:32653/TCP                                 3d2h
ocnrf-alternate-route                                 ClusterIP      10.233.40.203   <none>        80/TCP                                         45h
ocnrf-altsvc-cache                                    ClusterIP      None            <none>        8000/TCP                                       45h
ocnrf-egressgateway                                   ClusterIP      10.233.23.211   <none>        8080/TCP                                       45h
ocnrf-egw-cache                                       ClusterIP      None            <none>        8000/TCP                                       45h
ocnrf-igw-cache                                       ClusterIP      None            <none>        8000/TCP                                       45h
ocnrf-ingressgateway                                  LoadBalancer   10.233.41.30    <pending>     80:31501/TCP                                   90d
ocnrf-nfaccesstoken                                   ClusterIP      10.233.49.192   <none>        8080/TCP                                       45h
ocnrf-nfdiscovery                                     ClusterIP      10.233.31.176   <none>        8080/TCP                                       45h
ocnrf-nfregistration                                  ClusterIP      10.233.47.105   <none>        8080/TCP                                       45h
ocnrf-nfsubscription                                  ClusterIP      10.233.56.47    <none>        8080/TCP                                       45h
ocnrf-nrfartisan                                      ClusterIP      10.233.49.170   <none>        8080/TCP                                       45h
ocnrf-nrfauditor                                      ClusterIP      10.233.5.30     <none>        8080/TCP                                       45h
ocnrf-nrfcachedata                                    ClusterIP      10.233.28.20    <none>        8080/TCP                                       45h
ocnrf-nrfconfiguration                                LoadBalancer   10.233.27.134   <pending>     8080:31941/TCP                                 90d
ocnrf-ocnrf-app-info                                  ClusterIP      10.233.25.191   <none>        5906/TCP                                       45h
ocnrf-ocnrf-perf-info                                 ClusterIP      10.233.5.8      <none>        5905/TCP                                       45h
slf-stub-service01                                    LoadBalancer   10.233.62.84    <pending>     80:30178/TCP,8081:30891/TCP,8443:30223/TCP     3d2h
```

If ATS is deployed with side car of service mesh, ensure that both ATS and Stub pods have two containers in ready state and shows **"2/2"** as follows:

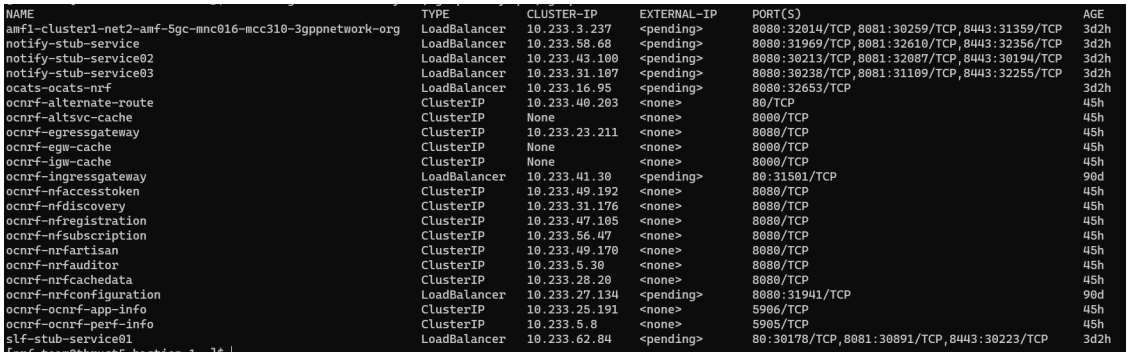
Figure 3-7 ATS and Stub Deployed with Service Mesh



```
NAME                                                  READY   STATUS    RESTARTS   AGE
ocats-ocats-nrf-855d874786-6w7n8                      2/2     Running   0          3d2h
ocnrf-alternate-route-57f4978469-ffn9w                2/2     Running   0          45h
ocnrf-appinfo-857c79bdc4-ngb5t                        2/2     Running   0          45h
ocnrf-egressgateway-7d5dc479b-tfjfj                   2/2     Running   0          45h
ocnrf-ingressgateway-6bfb85865f-d29xr                 2/2     Running   0          45h
ocnrf-nfaccesstoken-5bbdb6f4c9-f5c6q                  2/2     Running   0          45h
ocnrf-nfdiscovery-68db8f4c8d-dwfgw                    2/2     Running   0          45h
ocnrf-nfregistration-759c668c77-rrk7q                 2/2     Running   0          45h
ocnrf-nfsubscription-7c54b77669-v8msn                 2/2     Running   0          45h
ocnrf-nrfartisan-5b8f48bff7-z985q                     2/2     Running   0          45h
ocnrf-nrfauditor-6cdb597c67-qmgp7                     2/2     Running   0          45h
ocnrf-nrfcachedata-7c8967fbc8-npbwh                   2/2     Running   0          45h
ocnrf-nrfcachedata-7c8967fbc8-sp4pl                   2/2     Running   0          45h
ocnrf-nrfconfiguration-75b8c67dfb-r8qht               2/2     Running   0          45h
ocnrf-performance-5cf895dc75-4252k                    2/2     Running   0          45h
ocnrf-performance-5cf895dc75-g9f8x                    2/2     Running   0          45h
stub1-7698cf45d5-hncsh                                2/2     Running   0          3d2h
stub2-58bd5bfbb4-h6xq9                                2/2     Running   0          3d2h
stub3-5b596bbb59-488rm                                2/2     Running   0          3d2h
stub4-5b84bc67d6-gmnqs                                2/2     Running   0          3d2h
stub5-66cd68f97c-ls5kn                                2/2     Running   0          3d2h
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 3-8 ATS and Stub Deployed with Service Mesh



NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
amf1-cluster1-net2-amf-5gc-mnc016-mcc310-3gppnetwork-org	LoadBalancer	10.233.3.237	<pending>	8080:32014/TCP,8081:30259/TCP,8443:31359/TCP	3d2h
notify-stub-service	LoadBalancer	10.233.58.68	<pending>	8080:31969/TCP,8081:32610/TCP,8443:32356/TCP	3d2h
notify-stub-service02	LoadBalancer	10.233.43.100	<pending>	8080:30213/TCP,8081:32087/TCP,8443:30194/TCP	3d2h
notify-stub-service03	LoadBalancer	10.233.31.107	<pending>	8080:30238/TCP,8081:31109/TCP,8443:32255/TCP	3d2h
ocats-ocats-nrf	LoadBalancer	10.233.16.95	<pending>	8080:32653/TCP	3d2h
ocnrf-alternate-route	ClusterIP	10.233.40.203	<none>	80/TCP	45h
ocnrf-altsvc-cache	ClusterIP	None	<none>	8000/TCP	45h
ocnrf-egressgateway	ClusterIP	10.233.23.211	<none>	8080/TCP	45h
ocnrf-egw-cache	ClusterIP	None	<none>	8000/TCP	45h
ocnrf-igw-cache	ClusterIP	None	<none>	8000/TCP	45h
ocnrf-ingressgateway	LoadBalancer	10.233.41.30	<pending>	80:31501/TCP	90d
ocnrf-nfaccesstoken	ClusterIP	10.233.49.192	<none>	8080/TCP	45h
ocnrf-nfdiscovery	ClusterIP	10.233.31.176	<none>	8080/TCP	45h
ocnrf-nfregistration	ClusterIP	10.233.47.105	<none>	8080/TCP	45h
ocnrf-nfsubscription	ClusterIP	10.233.56.47	<none>	8080/TCP	45h
ocnrf-nrfartisan	ClusterIP	10.233.49.170	<none>	8080/TCP	45h
ocnrf-nrfauditor	ClusterIP	10.233.5.30	<none>	8080/TCP	45h
ocnrf-nrfcachedata	ClusterIP	10.233.28.20	<none>	8080/TCP	45h
ocnrf-nrfconfiguration	LoadBalancer	10.233.27.134	<pending>	8080:31941/TCP	90d
ocnrf-ocnrf-app-info	ClusterIP	10.233.25.191	<none>	5906/TCP	45h
ocnrf-ocnrf-perf-info	ClusterIP	10.233.5.8	<none>	5905/TCP	45h
slf-stub-service01	LoadBalancer	10.233.62.84	<pending>	80:30178/TCP,8081:30891/TCP,8443:30223/TCP	3d2h

3.2.7 Post-Installation Steps (if Persistent Volume is Used)

If persistent volume is used, follow the post-installation steps mentioned in the Persistent Volume for 5G ATS section.

3.3 Installing ATS for NSSF

This section describes Automated Testing Suite (ATS) installation procedures for Network Slice Selection Function (NSSF) in a cloud native environment. You must perform ATS installation procedures for NSSF in the same sequence as outlined in the following sections.

3.3.1 Resource Requirements

Total Number of Resources

The total number of resource requirements are as follows:

Table 3-8 Total Number of Resources

Resource	CPUs	Memory(GB)	Storage(GB)
NSSF SUT Total	30.2	22	4
cnDBTier Total	40	40	20
ATS Total	14	14	0
Grand Total NSSF ATS	79.2	72	24

Resource Details

The details of resources required to install NSSF-ATS are as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-9 Resource Details

Microservice	CPUs Required per Pod	Memory Required per Pod (GB)	Storage PVC Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	CPUs Required - Total	Memory Required - Total (GB)	Storage PVC Required - Total (GB)
NSSF Pods								
ingressgateway	4	4	0	2	1	4	4	0
egressgateway	4	4	0	2	1	4	4	0
nsselection	4	2	0	2	1	4	2	0
nsavailability	4	2	0	2	1	4	2	0
nsconfig	2	2	0	1	1	2	2	0
nssubscription	2	2	0	1	1	2	2	0
nrf-client-discovery	1	1	0	2	1	1	1	0
nrf-client-management	1	1	0	1	1	1	1	0
appinfo	0.2	1	0	2	1	0.2	1	0
perfinfo	0.2	0.5	0	1	1	0.2	0.5	0
config-server	0.2	0.5	0	1	1	0.2	0.5	0
NSSF SUT Totals						**22.6 CPU**	**20 GB**	0
ATS								
ATS Behave	4	4	0	0	1	4	4	0
ATS AMF Stub (Python)	3	3	0	0	1	3	3	0
ATS NRF Stub (Python)	2	2	0	0	1	2	2	0
ATS NRF Stub1 (Python)	2	2	0	0	1	2	2	0
ATS NRF Stub2 (Python)	2	2	0	0	1	2	2	0
OCDNS-BIND	1	1	0	0	1	1	1	0
ATS Totals						**14**	**14**	0
cnDBTier Pods (minimum of 4 worker nodes required)								
vrt-launcher-dt-1.cluster.local	4	4	2	2	1	4	4	2

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-9 (Cont.) Resource Details

Microservice	CPUs Required per Pod	Memory Required per Pod (GB)	Storage PVC Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	CPUs Required - Total	Memory Required - Total (GB)	Storage PVC Required - Total (GB)
vrt-launcher-dt-2.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-dt-3.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-dt-4.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-mt-1.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-mt-2.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-mt-3.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-sq-1.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-sq-2.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-db-installer.cluster.local	4	4	2	2	1	4	4	2
cnDBTier Totals						**40**	**40**	**20**

3.3.2 Locating and Downloading ATS and Simulator Images

To locate and download the ATS Image from MOS:

1. Log in to My Oracle Support using the appropriate credentials.
2. Select the **Patches and Updates** tab.
3. In the **Patch Search** window, click **Product or Family (Advanced)**.
4. Enter *Oracle Communications Cloud Native Core - 5G* in the **Product** field.

Copyright © 2019, 2025, Oracle and/or its affiliates.

5. Select *Oracle Communications Cloud Native Core Network Slice Selection Function <release_number>* from **Release** drop-down.
6. Click **Search**. The **Patch Advanced Search Results** list appears.
7. Select the required ATS patch from the search results. The Patch Details window appears.
8. Click **Download**. The File Download window appears.
9. Click the **<p********_<release_number>_Tekelec>.zip** file to download the NSSF ATS package file.
10. Untar the zip file to get `ocats-nssf` directory, which consists of all the ATS Images. The `ocats-nssf` directory has the following files:

> **Note**
>
> **Prerequisites:**
>
> - To run oauth test cases for NSSF, oauth secrets needs to be generated. For more information, see "Configuring Secrets to Enable Access Token " section in *Oracle communications Cloud Native Core, Network Slice Selection Function Installation, Upgrade, Fault Recovery Guide*.

a. To ensure the functionality of the `Virtual_Host_NRF_Resolution_By_NSSF_Using_DNSSRV` ATS feature, the following configuration must be enabled as per the engineering team's guidance:

```
nrf-client:
  # This config map is for providing inputs to NRF-Client
  configmapApplicationConfig:
      enableVirtualNrfResolution=true
      virtualNrfFqdn=nrfstub.changeme-ocats.svc
      virtualNrfScheme=http
```

b. The necessary changes to the NSSF custom-values.yaml file are outlined below. For instance, if the NSSF ATS is deployed in the "`ocnssfats`" namespace, the `virtualNrfFqdn` configuration must be updated as follows:

```
nrf-client:
  # This config map is for providing inputs to NRF-Client
  configmapApplicationConfig:
      enableVirtualNrfResolution=true
      virtualNrfFqdn=nrfstub.ocnssfats.svc
      virtualNrfScheme=http
```

c. If the NSSF is installed with a HELM release name different from "`ocnssf`" (for example, the HELM release name is "`ocnssfats`"), the following parameter must be updated accordingly.
If the HELM release name is "`ocnssf`", no changes are required.

```
# Alternate Route Service Host Value
# Replace ocnssf with Release Name
alternateRouteServiceHost: ocnssf-alternate-route
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

d. If the HELM release name for NSSF is "ocnssfats", the following parameter must be updated accordingly.

```
# Alternate Route Service Host Value
# Replace ocnssf with Release Name
alternateRouteServiceHost: ocnssfats-alternate-route
```

```
ocats-nssf
├─── ocats-nssf-custom-configtemplates-25.1.201-README.txt   - file
contains all the information required for the package.
├─── ocats-nssf-custom-configtemplates-25.1.201.zip   - contains
serviceaccount,PVC File and Custom values file
├─── ocats-nssf-tools-pkg-25.1.201-README.txt   - file contains all
the information required for the package.
└─── ocats-nssf-tools-pkg-25.1.201.tgz   - file has the following
images and charts packaged as tar files
```

11. Untar the `ocats-nssf-tools-pkg-25.1.201.tgz` tar file
The structure of the file looks as given below:

```
ocats-nssf-tools-pkg-25.1.201
├─── amfstub-25.1.201.tar      - AMF Stub Server Docker image
├─── amfstub-25.1.201.tar.sha256
├─── ats_data-25.1.201.tar     - ATS data, After untar "ocnssf_tests"
folder will gets created in which ATS feature files present
├─── ats_data-25.1.201.tar.sha256
├─── ocats-nssf-25.1.201.tar   - NSSF ATS Docker Image
├─── ocats-nssf-25.1.201.tar.sha256
├─── ocats-nssf-25.1.201.tgz   - ATS Helm Charts, after untar "ocats-
nssf" ats charts folder gets created.
├─── ocats-nssf-25.1.201.tgz.sha256
├─── ocdns-bind-25.1.201.tar.  - NSSF DNS Stub Server Docker Image
├─── ocdns-bind-25.1.201.tar.sha256
└─── README.md
```

12. Copy the `ocats-nssf-tools-pkg-25.1.201.tgz` tar file to the CNE or Kubernetes cluster where you want to deploy ATS.

13. Along with the above packages, there is `ocats-nssf-custom-configtemplates-25.1.201.zip` at the same location.
The readme file `ocats-nssf-custom-configtemplates-25.1.201-README.txt` contains information about the content of this zip file.

    Content of `ocats-nssf-custom-configtemplates-25.1.201.zip` is as below:

```
 Archive:  ocats-nssf-custom-configtemplates-25.1.201.zip
  inflating: nssf_ats_pvc_25.1.201.yaml
  inflating: ocats_nssf_custom_values_25.1.201.yaml
  inflating: ocats_ocnssf_custom_serviceaccount_25.1.201.yaml
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Copy these files to CNE or Kubernetes cluster where you want to deploy ATS.

```
ocats-nssf-tools-pkg-25.1.201
├─── amfstub-25.1.201.tar       - AMF Stub Server Docker image
├─── ats_data-25.1.201.tar      - ATS data, After untar "ocnssf_tests"
folder will gets created in which ATS feature files present
├─── ocats-nssf-25.1.201.tar    - NSSF ATS Docker Image
├─── ocats-nssf-25.1.201.tgz    - ATS Helm Charts, after untar "ocats-
nssf" ats charts folder gets created.
├─── ocdns-bind-25.1.201.tar.   - NSSF DNS Stub Server Docker Image
```

3.3.3 Deploying ATS in Kubernetes Cluster

To deploy ATS in Kubernetes Cluster:

1. Verify checksums of the tarballs mentioned in the file `Readme.txt`.
2. Run the following commands to extract tar file content, Helm charts, and Docker images of ATS:
   `tar -xvzf ocats-nssf-tools-pkg-25.1.201.tgz`

   The output of this command will return the following files:

```
ocats-nssf-tools-pkg-25.1.201
├─── amfstub-25.1.201.tar      - AMF Stub Server Docker image
├─── amfstub-25.1.201.tar.sha256
├─── ats_data-25.1.201.tar     - ATS data, After untar "ocnssf_tests"
folder will gets created in which ATS feature files present
├─── ats_data-25.1.201.tar.sha256
├─── ocats-nssf-25.1.201.tar   - NSSF ATS Docker Image
├─── ocats-nssf-25.1.201.tar.sha256
├─── ocats-nssf-25.1.201.tgz   - ATS Helm Charts, after untar "ocats-
nssf" ats charts folder gets created.
├─── ocats-nssf-25.1.201.tgz.sha256
├─── ocdns-bind-25.1.201.tar.  - NSSF DNS Stub Server Docker Image
├─── ocdns-bind-25.1.201.tar.sha256
└─── README.md
```

3. **NSSF-ATS and Stub Images Load and Push**: Run the following commands in your cluster to load the `ocats` image and `amf stubserver` image:
   Docker Commands:

```
docker load -i ocats-nssf-<version>.tar
```

```
docker load -i amfstub-<version>.tar
```

```
docker load -i ocdns-bind-<version>.tar
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Examples:

```
docker load -i ocats-nssf-25.1.201.tar

docker load -i amfstub-25.1.201.tar

docker load -i ocdns-bind-25.1.201.tar
```

Podman Commands:

```
podman load -i ocats-nssf-<version>.tar

podman load -i amfstub-<version>.tar

podman load -i ocdns-bind-<version>.tar
```

Examples:

```
podman load -i ocats-nssf-25.1.201.tar

podman load -i amfstub-25.1.201.tar

podman load -i ocdns-bind-25.1.201.tar
```

4. Run the following commands to tag and push the ATS image registry.
   a. Run the following commands to grep the image:

      ```
      docker images | grep ocats-nssf

      docker images | grep amfstub

      docker images | grep ocdns
      ```

   b. Copy the Image ID from the output of the grep command and change the tag (version number) to your registry.
      Docker Commands:

      ```
      docker tag <Image_ID> <your-registry-name/ocats-nssf:<tag>>
      docker push <your-registry-name/ocats-nssf:<tag>>
      docker tag <Image_ID> <your-registry-name/amfstub:<tag>>
      docker push <your-registry-name/amfstub:<tag>>
      docker tag <Image_ID> <your-registry-name/ocdns-bind:<tag>>
      docker push <your-registry-name/ocdns-bind:<tag>>
      ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Podman Commands:

```
podman tag <Image_ID> <your-registry-name/ats/ocats-nssf:<tag>>
```

```
podman push <your-registry-name/ocats-nssf:<tag>>
```

```
podman tag <Image_ID> <your-registry-name/amfstub:<tag>>
```

```
podman push <your-registry-name/amfstub:<tag>>
```

```
docker tag <Image_ID> <your-registry-name/ocdns-bind:<tag>>
```

```
docker push <your-registry-name/ocdns-bind:<tag>>
```

5. **ATS Helm Charts** : Run the following command to get "ATS" Helm charts as shown below:

   ```
   tar -xvzf ocats-nssf-25.1.201.tgz
   ```

   The above command creates "ocats-nssf" helm charts of ATS.

6. **ATS Data**: Run the following command to get ATS data, which contains feature files and data:

   ```
   tar -xvf ats_data-25.1.201.tar
   ```

   The above command creates "ocnssf_tests" ATS feature files "dat" and "jobs" folder, which are copied inside after the ATS installation is complete.

   a. Copy the `ocnssf_tests` folder under NSSF ATS pod as shown below:

      ```
      kubectl cp ocnssf_tests <namespace>/<nssf ats podname>:/var/lib/jenkins/
      ```

      Example:

      ```
      kubectl cp ocnssf_tests cicdnssf-241204053638/ocats-nssf-8566d64cfb-
      cgmvc:/var/lib/jenkins/
      ```

   b. Copy the jobs folder under NSSF ATS pod as shown below:

      ```
      kubectl cp jobs <namespace>/<nssf ats podname>:/var/lib/
      jenkins/.jenkins/
      ```

      Example:

      ```
      kubectl cp jobs cicdnssf-241204053638/ocats-nssf-8566d64cfb-
      cgmvc:/var/lib/jenkins/.jenkins/
      ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

7. <Optional> Go to certificate folder inside ocats-nssf and run the following command:

   ```
   kubectl create secret generic ocnssf-secret --from-file=certificates/
   rsa_private_key_pkcs1.pem --from-file=certificates/trust.txt --from-
   file=certificates/key.txt --from-file=certificates/ocnssf.cer --from-
   file=certificates/caroot.cer -n ocnssf
   ```

8. **ATS Custom Values File Changes**: Update the image name and tag in the `ocats_nssf_custom_values_25.1.201.yaml` file as required.
   a. For this, open the `ocats_nssf_custom_values_25.1.201.yaml` file.
   b. Update the `image.repository` and `image.tag` parameters for `ocats-nssf`, `ocats-amf-stubserver`, `ocats-nrf-stubserver`, `ocats-nrf-stubserver1`, `ocats-nrf-stubserver2`, and `ocdns-bind`.
   c. Save and close the file after making the updates.
9. <Optional>**To enable static port**: ATS supports static port. By default, this feature is not available.
   In the `ocats-nssf/values.yaml` file under service section, set the value of `staticNodePortEnabled` parameter as '`true`' and provide a valid `nodePort` value for `staticNodePort`.
10. **ATS Service Account Creation**: In `ocats-nssf-custom-serviceaccount_.yaml`, change namespace as below:

    ```
    sed -i "s/changeme-ocats/${namespace}/g"
    ocats_ocnssf_custom_serviceaccount_.yaml
    ```

11. Run the following command to apply `ocats-nssf-custom-serviceaccount_.yaml` file:
    `kubectl apply -f <serviceaccount.yaml file> -n <namespace_name>`

    For example:

    ```
    kubectl apply -f ocats-nssf-custom-serviceaccount_.yaml -n ocnssf
    ```

12. **Pointing NSSF to Stub Servers**: Follow this step to point out NSSF to NRF-Stubserver and AMF-Stubserver in NSSF custom values file:

    ```
    sed -i "s/changeme-ocats/${namespace}/g" $NSSF_CUSTOM_DEPLOY_FILE
    ```

    For example:

    ```
    sed -i "s/changeme-ocats/${namespace}/g" ocnssf_custom_values_25.1.201.yaml
    ```

    The NSSF custom values snippet is as follows:

    ```
    nrf-client:
      # This config map is for providing inputs to NRF-Client
      configmapApplicationConfig:
        &configRef
        profile: |-
          [appcfg]
          primaryNrfApiRoot=nrf-stubserver.changeme-ocats:8080
          secondaryNrfApiRoot=nrf-stubserver.changeme-ocats:8080
    ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
# ARS Helm Configuration
staticVirtualFqdns:
    - name: https://abc.test.com
      alternateFqdns:
        - target: amf-stubserver.changeme-ocats
          port: 8080
          priority: 10
        - target: nrf-stubserver.changeme-ocats
          port: 8080
          priority: 20
    - name: http://xyz.test.com
      alternateFqdns:
        - target: amf-stubserver.changeme-ocats
          port: 8080
          priority: 10
        - target: nrf-stubserver.changeme-ocats
          port: 8080
          priority: 20
```

13. Deploy ATS as shown below. NSSF ATS Helm release name should be "ocats".
    `helm install <release_name> <charts> -n <namespace_name> -f <custom_values file> –-version <helm-chart-version>`

    For example:

    ```
    helm install ocats ocats-nssf -n ocnssf -f
    ocats_nssf_custom_values_25.1.201.yaml --version 25.1.201
    ```

    Running the above command creates the following pods:

    ```
    ocats-nssf

    ocats-amf-stubserver

    ocats-nrf-stubserver

    ocats-nrf-stubserver1

    ocats-nrf-stubserver2

    ocats-ocdns-bind
    ```

    For example:

    ```
    $ kubectl get pods -n cicdnssf-241204053638
    NAME                                                    READY
    STATUS      RESTARTS       AGE
    ocats-amf-stubserver-5c5775dcb5-n9hlw                   2/2
    Running     0              3h22m
    ocats-nrf-stubserver-6bc6ff5ccc-6wkv7                   2/2
    Running     0              3h22m
    ocats-nrf-stubserver1-567f55488c-7bdxk                  2/2
    Running     0              3h22m
    ocats-nrf-stubserver2-697d4cffd9-c7dvr                  2/2
    Running     0              3h22m
    ocats-nssf-8566d64cfb-cgmvc                             2/2
    Running     0              3h22m
    ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
ocats-ocdns-bind-967bd4bc8-hpcjn                          2/2
Running     0              3h22m
```

```
$ kubectl get svc -n cicdnssf-241204053638
NAME                                    TYPE           CLUSTER-IP
EXTERNAL-IP   PORT(S)             AGE
amf-stubserver                          LoadBalancer   10.96.27.125
<pending>     8080:31064/TCP      5h34m
dnssim                                  ClusterIP      10.96.99.119
<none>        53/UDP,6236/TCP     5h34m
nrf-stubserver                          LoadBalancer   10.96.192.123
<pending>     8080:30409/TCP      5h34m
nrf-stubserver1                         LoadBalancer   10.96.235.187
<pending>     8080:31157/TCP      5h34m
nrf-stubserver2                         LoadBalancer   10.96.77.69
<pending>     8080:31056/TCP      5h34m
ocats-nssf                              LoadBalancer   10.96.14.30
<pending>     8080:31995/TCP      5h34m
```

14. Run the following command to verify the ATS deployment:
    `helm status <release_name>`

    The following screenshot is an example of a successful ATS deployment, where `STATUS:DEPLOYED` is an indicator of the same.

```
$ helm status ocats -n cicdnssf-241204053638
NAME: ocats
LAST DEPLOYED: Wed Dec 4 05:37:54 2024
NAMESPACE: cicdnssf-hrithik-241204053638
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

Virtual_Host_NRF_Resolution_By_NSSF_Using_DNSSRV ATS Feature

To ensure the functionality of the **Virtual_Host_NRF_Resolution_By_NSSF_Using_DNSSRV** ATS feature, the following configuration must be enabled:

```
nrf-client:
  # Configuration map for providing inputs to the NRF-Client
  configmapApplicationConfig:
    enableVirtualNrfResolution: true
    virtualNrfFqdn: nrfstub.changeme-ocats.svc
    virtualNrfScheme: http
```

The necessary changes to the **NSSF** `custom-values.yaml` file are outlined below:

Example Configuration for Namespace "ocnssfats"

Copyright © 2019, 2025, Oracle and/or its affiliates.

If the NSSF ATS is deployed in the `ocnssfats` namespace, update the `virtualNrfFqdn` configuration as follows:

```
nrf-client:
  # Configuration map for providing inputs to the NRF-Client
  configmapApplicationConfig:
    enableVirtualNrfResolution: true
    virtualNrfFqdn: nrfstub.ocnssfats.svc
    virtualNrfScheme: http
```

Configuration for Different HELM Release Names

- **Default Release Name** (`ocnssf`): No changes are required.
- **Alternate Release Name**: If the HELM release name is different (e.g., `ocnssfats`), update the following parameter:

  ```
  # Alternate Route Service Host Value
  # Replace 'ocnssf' with the HELM release name
  alternateRouteServiceHost: ocnssfats-alternate-route
  ```

Steps to Integrate DNS Stub with NSSF

1. Enable virtual FQDN resolution in the nrf-client management app profile configuration. Example:

   ```
   nrf-client:
       profile: |-
         [appcfg]
         primaryNrfApiRoot=nrf-stubserver.changeme-ocats:8080
         secondaryNrfApiRoot=nrf-stubserver.changeme-ocats:8080
         nrfScheme=http
         retryAfterTime=PT120S
         nrfClientType=NSSF
         nrfClientSubscribeTypes=
         appProfiles=[{"nfInstanceId":"9faf1bbc-6e4a-4454-a507-
   aef01a101a01","nfType":"NSSF","nfStatus":"REGISTERED","heartBeatTimer":30,"
   fqdn":"ocnssf-
   nsgateway.ocnssf.svc","priority":1,"capacity":1,"load":2,"plmnList":
   [{"mcc":"311","mnc":"480"}],"nfSetIdList":
   ["setEast.nssfset.5gc.mnc480.mcc311"],"locality":"rcnltxekloc1","nfServices
   ":[{"serviceInstanceId":"92d59bfc-e5d6-47f5-
   a26b-3a03facdebcc","serviceName":"nnssf-nsselection","versions":
   [{"expiry":null,"apiFullVersion":"1.0.0","apiVersionInUri":"v1"}],"scheme":
   "http","nfServiceStatus":"REGISTERED","fqdn":"ocnssf1-ingress-
   gateway.ocnssf.svc","interPlmnFqdn":null,"ipEndPoints":
   [{"ipv4Address":"10.224.45.178","transport":"TCP","port":80}],"allowedNfTyp
   es":["AMF","NSSF"],"priority":1,"capacity":1,"load":2},
   {"serviceInstanceId":"d33728cd-6e21-434b-bc5a-
   ed69bc612377","serviceName":"nnssf-nssaiavailability","versions":
   [{"expiry":null,"apiFullVersion":"1.0.0","apiVersionInUri":"v1"}],"scheme":
   "http","nfServiceStatus":"REGISTERED","fqdn":"ocnssf2-ingress-
   gateway.ocnssf.svc","interPlmnFqdn":null,"ipEndPoints":
   [{"ipv4Address":"10.224.45.179","transport":"TCP","port":80}],"allowedNfTyp
   es":["AMF","NSSF"],"priority":1,"capacity":1,"load":2}]}]
         enableF3=true
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
enableF5=true
renewalTimeBeforeExpiry=3600
validityTime=30
enableSubscriptionAutoRenewal=true
nfHeartbeatRate=80
acceptAdditionalAttributes=false
retryForCongestion=5
enableVirtualNrfResolution=true
virtualNrfFqdn=nrfstub.changeme-ocats.svc
virtualNrfScheme=http
```

2. In the above configuration, since "`enableVirtualNrfResolution=true`" and "`virtualNrfFqdn=nrfstub.changeme-ocats.svc`" have been set, the NRF client management attempts to contact the Egress Gateway, and Egress Gateway attempts to connect Alternate Route Service for virtual FQDN Resolution. In this service, we configure static settings related to the FQDN, as shown below:

   ```
   #Static virtual FQDN Config
   staticVirtualFqdns:
     - name: https://abc.test.com
       alternateFqdns:
         - target: amf-stubserver.changeme-ocats
           port: 8080
           priority: 10
         - target: nrf-stubserver.changeme-ocats
           port: 8080
           priority: 20
     - name: http://xyz.test.com
       alternateFqdns:
         - target: amf-stubserver.changeme-ocats
           port: 8080
           priority: 10
         - target: nrf-stubserver.changeme-ocats
           port: 8080
           priority: 20
   ```

   According to the static virtual FQDN configuration under ARS, there is no record for the FQDN "`virtualNrfFqdn=nrfstub.changeme-ocats.svc`". As a result, ARS will attempt to resolve this FQDN by contacting the default CoreDNS server (the default Kubernetes cluster DNS server). To ensure proper resolution, we need to point ARS to the DNS stub we installed by editing the ARS deployment file after deploying NSSF.

3. Before running ATS Test suite, nrf-client-nfdiscovery and nrf-client-nfmanagement pods has to be restarted. Here are the steps to edit the "alternate route service" deployment pointing towards DNS Stub:

   a. Run the follwing command to get the list of pods:

      ```
      $ kubectl get pods -n <namespace>
      ```

      For example:

      ```
      $ kubectl get pods -n ocnssf
      ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Expected sample response:

```
NAME                                                    READY
STATUS      RESTARTS      AGE
ocats-amf-stubserver-6f57f7f57f-bk75w                   2/2
Running     0             64m
ocats-nrf-stubserver-5f89cdb74c-649kx                   2/2
Running     0             64m
ocats-nssf-76cf4fc678-vrl24                             2/2
Running     0             64m
ocdns-bind-5975fc59c8-6vllx                             2/2
Running     0             41m
ocnssf-alternate-route-5f857d94bb-dm5vl                 2/2
Running     0
40m                                             0/1     Completed
0             30m
```

b. Run following command to get searches information from dns-bind pod to enable communication between Alternate Route and dns-bind service:

```
kubectl exec -it <dnsbind-pod> -n <NAMESPACE> -- /bin/bash -c 'cat /etc/
resolv.conf' | grep search | tr ' ' '\n' | grep -v 'search'
```

For example:

```
$ kubectl exec -it ocdns-bind-5975fc59c8-6vllx -n ocnssf -- /bin/bash -
c 'cat /etc/resolv.conf' | grep search | tr ' ' '\n' | grep -v 'search'
```

Expected sample response:

```
ocnssf.svc.cluster.local
svc.cluster.local
cluster.local
gbucdsint02phx.oraclevcn.com
snphxprshared1.gbucdsint02phx.oraclevcn.com
```

c. By default alternate service will point to CoreDNS and you will see following settings in deployment file. Now, edit alternate route service deployment and add configuration as below:

```
$ kubectl edit deployment ocnssf-alternate-route -n <NAMESPACE>
```

For example:

```
$ kubectl edit deployment ocnssf-alternate-route -n ocnssf
```

```
terminationMessagePath: /dev/termination-log
  terminationMessagePolicy: File
dnsPolicy: ClusterFirst
restartPolicy: Always
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

d. Edit the deployment file to add following content in alternate service to query DNS stub:

- Add the nameservers IPaddress which you recorded after installing the DNS stub (cluster IP of DNS Stub)
- Add all the search information one by one which you recorded earlier.
- Set dnsPolicy to "None"

```
dnsConfig:
  nameservers:
  - <dns_stub_cluster_ip_address>
  searches:
  - dns-bind search
  - dns-bind search
  - dns-bind search
  dnsPolicy: None
```

```
$ kubectl get svc -n ocnssf
```

Expected sample response:

```
NAME                                     TYPE           CLUSTER-IP
EXTERNAL-IP   PORT(S)             AGE
dnssim                                   ClusterIP      10.96.65.56
<none>        53/UDP,6236/TCP     48m
```

For example:

```
terminationMessagePath: /dev/termination-log
  terminationMessagePolicy: File
dnsConfig:
  nameservers:
  - 10.96.65.56
  searches:
  - ocnssf.svc.cluster.local
  - svc.cluster.local
  - cluster.local
  - gbucdsint02phx.oraclevcn.com
  - snphxprshared1.gbucdsint02phx.oraclevcn.com
dnsPolicy: None
restartPolicy: Always
```

Verification

Use the following `curl` command to verify the setup:

```
curl -v --http2-prior-knowledge -X GET "http://ocnssf-alternate-route:80/
lookup?fqdn=nf1stub.ocnssf.svc&scheme=h
ttp"
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Expected sample response:

```
Note: Unnecessary use of -X or --request, GET is already inferred.
*   Trying 10.96.194.219...
* TCP_NODELAY set
* Connected to ocnssf-alternate-route (10.96.194.219) port 80 (#0)
* Using HTTP2, server supports multi-use
* Connection state changed (HTTP/2 confirmed)
* Copying HTTP/2 data in stream buffer to connection buffer after upgrade:
len=0
* Using Stream ID: 1 (easy handle 0x55b19b19c6f0)
> GET /lookup?fqdn=nf1stub.ocnssf.svc&scheme=http HTTP/2
> Host: ocnssf-alternate-route
> User-Agent: curl/7.61.1
> Accept: */*
>
* Connection state changed (MAX_CONCURRENT_STREAMS == 2147483647)!
< HTTP/2 200
< content-type: application/json
< date: Tue, 08 Oct 2024 05:48:02 GMT
< x-envoy-upstream-service-time: 5
<
* Connection #0 to host ocnssf-alternate-route left intact
[{"target":"ocats-amf-
stubserver.ocnssf.svc","port":8080,"ttl":86400,"type":"SRV","dclass":"IN","pri
ority":1,"weight":60},{"target":"ocats-nrf-
stubserver.ocnssf.svc","port":8080,"ttl":86400,"type":"SRV","dclass":"IN","pri
ority":20,"weight":20},{"target":"ocats-nrf-
stubserver1.ocnssf.svc","port":8080,"ttl":86400,"type":"SRV","dclass":"IN","pr
iority":30,"weight":20}][ocnrfusr@ocnssf-ocnssf-nrf-client-
nfmanagement-744ff58956-tfr5g app]$
```

3.4 Installing ATS for Policy

Installing ATS for Policy procedure consists of the following two steps:

1. Locating and downloading the ATS package
2. Deploying ATS and stub pods in Kubernetes cluster

This includes installation of nine stubs (nf1stub, nf11stub, nf12stub, nf2stub, nf21stub, nf22stub, nf3stub, nf31stub, nf32stub), ocamf stub, ocdns-bind stub, ocldap-stub, and Policy ATS in the namespace where CNC Policy is deployed.

3.4.1 Resource Requirements

This section describes the ATS resource requirements for CNC Policy.

Overview - Total Number of Resources

The following table describes the overall resource usage in terms of CPUs, and memory for the following:

- PCF SUT
- cnDBTier

Copyright © 2019, 2025, Oracle and/or its affiliates.

- ATS

Table 3-10 PCF - Total Number of Resources

Resource Name	Non-ASM CPU	Non-ASM Memory (GB)	ASM CPU	ASM Memory (GB)
PCF SUT Total	219	197	293	244
ATS Total	26	28	54	42
CnDBTier Total	107.1	175.2	137.1	190.2
Grand Total PCF ATS	352.1	400.2	484.1	476.2

PCF Pods Resource Requirements Details

This section describes the resource requirements, which are needed to deploy Policy ATS successfully.

Table 3-11 PCF Pods Resource Requirements Details

Policy Microservices	Max CPU	Memory (GB)	Max Replica	Non-ASM Total CPU	Non-ASM Memory (GB)	ASM Total CPU	ASM Total Memory (GB)	Isito ASM CPU	Isito ASM Memory (GB)
oc-app-info	2	2	1	2	2	4	3	2	1
oc-bulwark	8	6	2	16	12	20	14	2	1
oc-diam-connector	4	2	2	8	4	12	6	2	1
oc-diam-gateway	4	2	1	4	2	6	3	2	1
alternate-route	2	4	1	2	4	4	5	2	1
oc-config-server	4	2	1	4	2	6	3	2	1
ocegress_gateway	4	6	1	4	6	6	7	2	1
ocingress_gateway	5	6	1	5	6	7	7	2	1
nrf-client-disc	4	2	2	8	4	12	6	2	1
nrf-client-mngt	1	1	2	2	2	6	4	2	1
oc-audit	2	4	1	2	4	4	5	2	1
oc-config-mgmt	4	2	2	8	4	12	6	2	1

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-11 (Cont.) PCF Pods Resource Requirements Details

Policy Microservices	Max CPU	Memory (GB)	Max Replica	Non-ASM Total CPU	Non-ASM Memory (GB)	ASM Total CPU	ASM Total Memory (GB)	Isito ASM CPU	Isito ASM Memory (GB)
oc-ldap-gateway	4	2	2	8	8	12	10	2	1
oc-policy-ds	7	8	2	14	16	18	18	2	1
oc-pre	4	4	2	8	8	12	10	2	1
oc-query	2	1	2	4	2	8	4	2	1
oc-soap-connector	4	4	2	8	8	12	10	2	1
oc-pcf-am	8	8	2	16	16	20	18	2	1
oc-pcf-sm	7	10	2	14	20	18	22	2	1
oc-pcf-ue	8	6	2	16	12	20	24	2	1
oc-pcrf-core	8	8	2	16	16	0	18	2	1
oc-perf-info	2	2	2	4	4	8	6	2	1
oc-binding	6	8	1	6	8	8	9	2	1
oc-udr-connector	6	4	2	12	8	16	10	2	1
oc-chf-connector	6	4	2	12	8	16	10	2	1
usage-mon	5	4	2	10	8	14	10	2	1
nwdaf-agent	2	1	1	2	1	4	2	2	1
notifier	2	1	2	4	2	8	4	2	1
Policy Totals				219	197	293	244		

ATS Resource Requirements details for Policy

This section describes the ATS resource requirements, which are needed to deploy Policy ATS successfully.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-12 ATS Resource Requirements Details

ATS Microservices	Max CPU	Max Memory (GB)	Max Replica	Non-ASM Total CPU	Non-ASM Total Memory (GB)	ASM Total CPU	ASM Total Memory (GB)	Isito ASM CPU	Isito ASM Memory (GB)
ocstub1-py	2	2	1	2	2	4	3	2	1
ocstub2-py	2	2	1	2	2	4	3	2	1
ocstub3-py	2	2	1	2	2	4	3	2	1
ocstub11-py	2	2	1	2	2	4	3	2	1
ocstub12-py	2	2	1	2	2	4	3	2	1
ocstub21-py	2	2	1	2	2	4	3	2	1
ocstub22-py	2	2	1	2	2	4	3	2	1
ocstub31-py	2	2	1	2	2	4	3	2	1
ocstub32-py	2	2	1	2	2	4	3	2	1
ocamf-stub	1	1	1	1	1	3	2	2	1
ocats-policy	4	6	1	4	6	6	7	2	1
ocdns-bind	1	1	1	1	1	3	2	2	1
oc-ldap-org1	1	1	1	1	1	3	2	2	1
ocdiam-sim	1	1	1	1	1	3	2	2	1
ATS Totals				26	28	54	42		

cnDBTier Resource Requirements Details for Policy ATS

This section describes the cnDBTier resource requirements, which are needed to deploy Policy ATS successfully.

> **Note**
>
> For cnDBTier pods, a minimum of 4 worker nodes are required.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-13 cnDBTier Resource Requirements Details

cnDBTier Microservices	Min CPU	Min Memory (GB)	Min Replica	Total CPU	Total Memory (GB)	ASM Total CPU	ASM Total Memory (GB)	Isito ASM CPU	Isito ASM Memory (GB)
db_monitor_svc	1	1	1	1	1	3	2	2	1
db_replication_svc	2	12	1	2	12	4	13	2	1
db_backup_manager_svc	0.1	0.2	1	0.1	0.2	2.1	1.2	2	1
ndbappmysqld	8	10	4	32	40	40	44	2	1
ndbmgmd	4	10	2	8	20	12	22	2	1
ndbmtd	10	18	4	40	72	48	76	2	1
ndbmysqld	8	10	2	16	20	20	22	2	1
db_infra_moditor_svc	8	10	1	8	10	8	10		
DB Tier Total				107.1	175.2	137.1	190.2		

> **Note**
>
> The requirements shown in the above table for CnDBTier are the default numbers and must be changed as per the deployment requirements.

3.4.2 Downloading the ATS Package

This section provides information on how to locate and download the Policy ATS package file from My Oracle Support (MOS).

Locating and Downloading Policy ATS Package

To locate and download the ATS package from MOS, perform the following steps:

1. Log in to My Oracle Support using the valid credentials.
2. Select the **Patches & Updates** tab.
3. In the **Patch Search** window, click **Product or Family (Advanced)**.
4. Enter `Oracle Communications Cloud Native Core - 5G` in the **Product** field.
5. Select `Oracle Communications Cloud Native Core Policy <release_number>` using the drop-down menu of the **Release** field.
6. Click **Search**. The list of **Patch Advanced Search Results** appears.
7. Select the required ATS patch from the list. The Patch Details window appears.

Copyright © 2019, 2025, Oracle and/or its affiliates.

8. Click **Download**. The File Download window appears.
9. Click the **<p********_<release_number>_Tekelec>.zip** file to downlaod the CNC Policy ATS package file.
10. Untar the gzip file `ocats-policy-tools-25.1.200.0.0.tgz` to access the following files:

```
ocats-policy-pkg-25.1.200.0.0.tgz
ocdns-pkg-25.1.204.0.0.tgz
ocamf-pkg-25.1.204.0.0.tgz
oc-ldap-org1-pkg-25.1.204.0.0.tgz
ocstub-pkg-25.1.201.0.0.tgz
ocdiam-pkg-25.1.204.0.0tgz
```

The contents included in each of these files are as follow:

```
|

|_ _ _ocats-policy-pkg-25.1.200.0.0.tgz

|      |_ _ _ _ _ _ ocats-policy-25.1.200.tgz (Helm Charts)

|      |_ _ _ _ _ _ ocats-policy-image-25.1.200.tar (Docker Images)

|      |_ _ _ _ _ _ ocats-policy-data-25.1.200.tgz (Policy ATS and Jenkins
job Data)

|

|_ _ _ocstub-pkg-25.1.201.0.0.tgz

|      |_ _ _ _ _ _ ocstub-py-25.1.201.tgz(Helm Charts)

|      |_ _ _ _ _ _ ocstub-py-image-25.1.201.tar (Docker Images)

|

|_ _ _ocdns-pkg-25.1.204.0.0.tgz

|      |_ _ _ _ _ _ ocdns-bind-25.1.204.tgz(Helm Charts)

|      |_ _ _ _ _ _ ocdns-bind-image-25.1.204.tar (Docker Images)

|

|_ _ _ocamf-pkg-25.1.204.0.0.tgz

|      |_ _ _ _ _ _ ocamf-stub-25.1.204.tgz(Helm Charts)

|      |_ _ _ _ _ _ ocamf-stub-image-25.1.204.tar (Docker Images)

|

|_ _ _oc-ldap-org1-pkg-25.1.204.0.0.tgz
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
|      |_ _ _ _ _ _ oc-ldap-org1-25.1.204.tgz(Helm Charts)

|      |_ _ _ _ _ _ oc-ldap-org1-image-25.1.204.tar (Docker Images)

|

|_ _ __ocdiam-pkg-25.1.204.0.0.tgz

|      |_ _ _ _ _ _ ocdiam-sim-25.1.204.tgz(Helm Charts)

|      |_ _ _ _ _ _ ocdiam-sim-image-25.1.204.tar (Docker Images)
```

11. Copy the tar file from the downloaded package to CNE, OCI, or Kubernetes cluster where you want to deploy ATS.

3.4.3 Deploy Policy ATS with TLS enabled

3.4.3.1 Generate JKS File for Jenkins Server

To access Jenkins ATS GUI access through HTTPS, a JKS file should be created.

Perform the following steps to generate the JKS file:

Generate the Root Certificate

1. If the user has a Certificate Authority (CA) signed root certificate such as `caroot.cert` and key, then the user can use those files.
2. If the root certificate is not already available, the user can generate one self signed root certificate. This root certificate created needs to be added to the `truststore` such as a Browser like Firefox or Chrome. User can follow the Browser specific documentation to upload the root certificate. The root certificate is used to sign the application, or ATS certificate.
3. Generate a root key with the following command:

   ```
   openssl genrsa 2048 > caroot.key
   ```

   This will generate a key called `caroot.key`
4. Generate a `caroot` certificate with the following command:

   ```
   openssl req -new -x509 -nodes -days 1000 -key <root_key> >
   <root_certificate>
   ```

   For example,

   ```
   [cloud-user@platform-bastion-1]$ openssl req -new -x509 -nodes -days 1000 -
   key caroot.key > caroot.cer
   You are about to be asked to enter information that will be incorporated
   into your certificate request.
   What you are about to enter is what is called a Distinguished Name or a DN.
   There are quite a few fields but you can leave some blank
   For some fields there will be a default value,
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:IN
State or Province Name (full name) []:KA
Locality Name (eg, city) [Default City]:BLR
Organization Name (eg, company) [Default Company Ltd]:ORACLE
Organizational Unit Name (eg, section) []:CGBU
Common Name (eg, your name or your server's hostname) []:ocats
Email Address []:
[cloud-user@platform-bastion-1]$
```

Generate Application or Client Certificate

1. Create a `ssl.conf` file.
2. Edit the `ssl.conf` file. In the "[alt_names]" section, list the IPs that are used to access ATS GUI as shown in the following sample `ssl.conf` file:

```
[ req ]
default_bits       = 4096
distinguished_name = req_distinguished_name
req_extensions     = req_ext

[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)
countryName_default         = IN
stateOrProvinceName         = State or Province Name (full name)
stateOrProvinceName_default = KN
localityName                = Locality Name (eg, city)
localityName_default        = BLR
organizationName            = Organization Name (eg, company)
organizationName_default    = ORACLE
commonName                  = Common Name (e.g. server FQDN or YOUR name)
commonName_max              = 64
commonName_default          = ocats.ocpcf.svc.cluster.local

[ req_ext ]
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth, clientAuth
basicConstraints = critical, CA:FALSE
subjectAltName = critical, @alt_names

[alt_names]
IP.1 = 127.0.0.1
IP.2 = 10.75.217.5
IP.3 = 10.75.217.76
DNS.1 = localhost
DNS.2 = ocats.ocpcf.svc.cluster.local
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> - To access the GUI with DNS, make sure that the commonName_default is the same as the DNS name being used.
>   - Ensure the DNS is in this format:`<service_name>.<namespace>.svc.cluster.local`
>
>   Multiple DNSs, such as DNS.1, DNS.2, and so on, can be added.
> - To support the ATS API, it is necessary to add the IP 127.0.0.1 to the list of IPs.

3. Create a Certificate Signing Request (CSR) with the following command:

   ```
   openssl req -config ssl.conf -newkey rsa:2048 -days 1000 -nodes -keyout
   rsa_private_key_pkcs1.key > ssl_rsa_certificate.csr
   ```

   Output:

   ```
   [cloud-user@platform-bastion-1 ocpcf]$ openssl req -config ssl.conf -
   newkey rsa:2048 -days 1000 -nodes -keyout rsa_private_key_pkcs1.key >
   ssl_rsa_certificate.csr
   Ignoring -days; not generating a certificate
   Generating a RSA private key
   ...+++++
   ........+++++
   writing new private key to 'rsa_private_key_pkcs1.key'
   -----
   You are about to be asked to enter information that will be incorporated
   into your certificate request.
   What you are about to enter is what is called a Distinguished Name or a DN.
   There are quite a few fields but you can leave some blank
   For some fields there will be a default value,
   If you enter '.', the field will be left blank.
   -----
   Country Name (2 letter code) [IN]:
   State or Province Name (full name) [KA]:
   Locality Name (eg, city) [BLR]:
   Organization Name (eg, company) [ORACLE]:
   Common Name (e.g. server FQDN or YOUR name) [ocpcf]:
   [cloud-user@platform-bastion-1 ocpcf]$
   ```

4. To display all the components of the CSR file and to verify the configurations run the following command:

   ```
   openssl req -text -noout -verify -in ssl_rsa_certificate.csr
   ```

5. Sign the CSR file with root certificate by running the following command:

   ```
   openssl x509 -extfile ssl.conf -extensions req_ext -req -
   inssl_rsa_certificate.csr -days 1000 -CA ../caroot.cer -CAkey ../
   caroot.key -set_serial 04 > ssl_rsa_certificate.crt
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Output:

```
[cloud-user@platform-bastion-1 ocpcf]$ openssl x509 -extfile ssl.conf -
extensions req_ext -req -in ssl_rsa_certificate.csr -days 1000 -CA ../
caroot.cer -CAkey ../caroot.key -set_serial 04 > ssl_rsa_certificate.crt
Signature ok
subject=C = IN, ST = KA, L = BLR, O = ORACLE, CN = ocpcf
Getting CA Private Key
[cloud-user@platform-bastion-1 ocpcf]$
```

6. Verify if the certificate is signed by the root certificate by running the following command:

```
[cloud-user@platform-bastion-1 ocpcf]$ openssl verify -CAfile caroot.cer
ssl_rsa_certificate.crt
```

Output:

```
[cloud-user@platform-bastion-1 ocpcf]$ openssl verify -CAfile caroot.cer
ssl_rsa_certificate.crt
ssl_rsa_certificate.crt: OK
```

7. Save the generated application certificate and root certificate.
8. Add the `caroot.cer` to the browser as a trusted author.
9. The generated application/client certificates cannot be directly given to the Jenkins server. Hence generate the `.p12 keystore` file for the client certificate with the following command:

```
[cloud-user@platform-bastion-1 ococf]$ openssl pkcs12 -inkey
rsa_private_key_pkcs1.key -inssl_rsa_certificate.crt -export-out
certificate.p12
Enter Export Password:
Verifying - Enter Export Password:
```

10. In the prompt, create a password and save it for future use.
11. Convert the .p12 keystore file into a JKS format file using the following command:

```
[cloud-user@platform-bastion-1 ocpcf]$ keytool -importkeystore -
srckeystore ./certificate.p12 -srcstoretype pkcs12 -destkeystore
jenkinsserver.jks -deststoretype JKS
```

Output:

```
[cloud-user@platform-bastion-1 ocpcf]$ keytool -importkeystore -
srckeystore ./certificate.p12 -srcstoretype pkcs12 -destkeystore
jenkinsserver.jks -deststoretype JKS
Importing keystore ./certificate.p12 to jenkinsserver.jks...
Enter destination keystore password:
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
Re-enter new password:
Enter source keystore password:
Entry for alias 1 successfully imported.
Import command completed:  1 entries successfully imported, 0 entries
failed or cancelled
```

12. In the prompt, use the same password used while creating `.p12 keystore` file.

> **Note**
>
> Ensure that the .p12 keystore and JKS files has the same passwords.

13. The generated JKS file, `jenkinserver.jks` is added to the Jenkins path, where Jenkins server can access it.

For more details about the ATS TLS feature, refer to Deploy ATS with TLS Enabled section.

3.4.3.2 Enable TLS on Python Stubs

Before starting the creation of certificate, user should have the path details of `caroot.cer` file.

1. Create a `ssl.conf` file.
2. Edit `ssl.conf` file. Ensure that the DNS is in the format of `*.<namespace>.svc`
   A Sample **stub_ssl.conf** file:

```
[ req ]
default_bits       = 4096
distinguished_name = req_distinguished_name
req_extensions     = req_ext

[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)
countryName_default         = IN
stateOrProvinceName         = State or Province Name (full name)
stateOrProvinceName_default = KN
localityName                = Locality Name (eg, city)
localityName_default        = BLR
organizationName            = Organization Name (eg, company)
organizationName_default    = ORACLE
commonName                  = svc.cluster.local

[ req_ext ]
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth, clientAuth
basicConstraints = critical, CA:FALSE
subjectAltName = critical, @alt_names

[alt_names]
IP.1 = 127.0.0.1
DNS.1 = *.ocats.svc
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

3. Create a Certificate Signing Request (CSR) for the stubs using the following command:

```
$ openssl req -config stub_ssl.conf -newkey  rsa:2048 -days 1000 -nodes -
keyout rsa_private_key_stub_pkcs2.key > stub_ssl_rsa_certificate1.csr
```

Output:

```
$ openssl req -config stub_ssl.conf -newkey rsa:2048 -days 1000 -nodes -
keyout rsa_private_key_stub_pkcs2.key > stub_ssl_rsa_certificate1.csr
Ignoring -days; not generating a certificate
Generating a RSA private key
.....................+++++
...+++++
writing new private key to 'rsa_private_key_stub_pkcs2.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IN]:
State or Province Name (full name) [KN]:
Locality Name (eg, city) [BLR]:
Organization Name (eg, company) [ORACLE]:
svc.cluster.local []:*.ocats.svc
```

4. Sign the certificate with the CA root using the following command:

```
openssl x509 -extfile stub_ssl.conf -extensions req_ext -req -in
stub_ssl_rsa_certificate1.csr -days 1000 -CA  ../ocats-caroot.cer -
CAkey ../ocats-caroot.key -set_serial 05 >  stub_ssl_rsa_certificate1.crt
```

Output:

```
$ openssl x509 -extfile stub_ssl.conf -extensions req_ext -req -in
stub_ssl_rsa_certificate1.csr -days 1000 -CA ../ocats-caroot.cer -CAkey ../
ocats-caroot.key -set_serial 05 > stub_ssl_rsa_certificate1.crt
Signature ok
subject=C = IN, ST = KN, L = BLR, O = ORACLE, CN = *.ocats.svc
Getting CA Private Key
```

5. Create a secret for the stub and associate it with the namespace using the following command:

```
$ kubectl create secret generic ocats-stub-secret1 --from-
file=stub_ssl_rsa_certificate1.crt --from-
file=rsa_private_key_stub_pkcs2.key --from-file=../ocats-caroot.cer -n
ocpcf
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Output:

```
kubectl create secret generic ocats-stub-secret1 --from-
file=stub_ssl_rsa_certificate1.crt --from-
file=rsa_private_key_stub_pkcs2.key --from-file=../ocats-caroot.cer -n
ocats
secret/ocats-stub-secret1 created
```

6. Update `values.yaml` file of each Python stub in specific NF namespace with following details:

```
NF: "<NF-Name>"
cert_secret_name: "ocats-stub-secret"
ca_cert: "ocats-caroot.cer"
client_cert: "ocats-stub_ssl_rsa_certificate.crt"
private_key: "ocats-rsa_private_key_stub_pkcs1.key"
expose_tls_service: true
CLIENT_CERT_REQ: true
```

> **Note**
>
> If the Helm, `cert_secret_name` parameter, is null, then `ca_cert`, `client_cert`, and `private_key` values are not considered by the TLS.

7. Ensure to update the deployment of all the Python stubs installed on the setup.

For more details about the ATS TLS feature, refer to Support for Transport Layer Security section.

3.4.3.3 Enable TLS on ocams Stubs

Before starting the creation of certificate, user should have the path details of `ocats-caroot.cer` and `ocats-caroot.key`file.

1. In the `values.yaml` file the following parameters are provided to configure the TLS for ocamf stub:

```
tls:
  enabled: false
  version: "TLSv1.2,TLSv1.3"
  appPort: 8443
  initialAlgorithm: 'RS256'
  secretName: 'app-tls-secret'
  rsaPrivateKeyFileName: 'app-key.pem'
  ecdsaPrivateKeyFileName: 'app-ecdsa-private-key.pem'
  rsaCertificateFileName: 'app-cert.crt'
  ecdsaCertifacateFileName: 'app-ecdsa-certificate.crt'
  caBundleFileName: 'ocats-caroot.cer'
  certReloadEnabled: true
  certReloaderDelay: 15000
  cipherSuites:
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- TLS_AES_128_GCM_SHA256
- TLS_CHACHA20_POLY1305_SHA256
```

a. Configure the following parameters to enable TLS for ocamf stub **tls.enabled** to `true`, **tls.version** to (either TLSv1.2 or TLSv1.3), **tls.appPort**, **tls.initialAlgorithm**, and **tls.secretName** parameters.

b. It is not mandatory to provide both **tls.rsaPrivateKey** and **tls.ecdsaPrivateKeyFile** files. You can either provide both or any one of them.

c. Configure the **tls.cipherSuites** parameter to test compatibility of different ciphers based on TLS versions.

d. A Sample `values.yaml` file configured for enabling TLS on ocamf stub:

```
tls:
  enabled: true
  version: "TLSv1.3"
  appPort: 8443
  initialAlgorithm: 'RS256'
  secretName: '${NAMESPACE}-ocamf-app-tls-secret'
  rsaPrivateKeyFileName: '${NAMESPACE}-ocamf-app-key.pem'
  ecdsaPrivateKeyFileName: 'app-ecdsa-private-key.pem'
  rsaCertificateFileName: '${NAMESPACE}-ocamf-app-cert.crt'
  ecdsaCertifacateFileName: 'app-ecdsa-certificate.crt'
  caBundleFileName: 'ocats-caroot.cer'
  certReloadEnabled: true
  certReloaderDelay: 15000
  cipherSuites:
    - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
    - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
    - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
    - TLS_AES_256_GCM_SHA384
    - TLS_AES_128_GCM_SHA256
    - TLS_CHACHA20_POLY1305_SHA256
```

2. Create certificate and private key generation for ocamf application by using the following steps.

   a. Run the following command to create private key:

   ```
   openssl genrsa -out ${NAMESPACE}-ocamf-app-key.pem 4096
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

b. Run the following command to generate a certificate request for the generated private key:

```
openssl req -new -sha256 -key ${NAMESPACE}-ocamf-app-key.pem -out $
{NAMESPACE}-ocamf-app-csr.csr -subj "/CN=ocamf2.$
{NAMESPACE}.svc.cluster.local"
```

c. Run the following command to generate a self signed root certificate:

```
openssl x509 -req -sha256 -days 1000 \    -in ${NAMESPACE}-ocamf-app-
csr.csr \    -CA ocats-caroot.cer \    -CAkey ocats-caroot.key \    -
out ${NAMESPACE}-ocamf-app-cert.pem\    -CAcreateserial
```

d. Run the following command to convert pem file to crt file:

```
mv ${NAMESPACE}-ocamf-app-cert.pem ${NAMESPACE}-ocamf-app-cert.crt
```

3. Create a secret for the stub and associate it with the namespace using the following command:

```
kubectl create secret generic ${NAMESPACE}-ocamf-tls-secret --from-file=$
{NAMESPACE}-ocamf-app-key.pem --from-file=ocats-caroot.cer --from-file=$
{NAMESPACE}-ocamf-app-cert.crt -n ${NAMESPACE}
```

Example:

```
kubectl create secret generic altair-ns11-ocamf-tls-secret --from-
file=altair-ns11-ocamf-app-key.pem --from-file=ocats-caroot.cer --from-
file=altiar-ns11-ocamf-app-cert.crt -n altair-ns11
secret/altair-ns11-ocamf-tls-secret created
```

3.4.3.4 Enable TLS on Diam-Sim

Before starting the creation of certificate, user should have the path details of `ocats-caroot.cer` and `ocats-caroot.key` file.

1. In the `values.yaml` file the following parameters are provided to configure the TLS for diam-sim stub:

```
tls:
  enabled: false
  secretName: 'dcli-tls-secret'
  rsaPrivateKeyFileName: 'dcli-key.pem'
  ecdsaPrivateKeyFileName: 'dcli-ecdsa-private-key.pem'
  rsaCertificateFileName: 'dcli-cert.crt'
  ecdsaCertifacateFileName: 'dcli-ecdsa-certificate.crt'
  caBundleFileName: 'ca-cert.cer'
```

a. By default, **tls.enabled** parameter is **false**. Set this to true to enable TLS on diam-sim.

b. By default **secretName** will be **dcli-tls-secret**. User can change this name, but ensure that name of the secret created should match with the name provided for **secretName**.

Copyright © 2019, 2025, Oracle and/or its affiliates.

c. A Sample `values.yaml` file configured for enabling TLS on diam-sim stub:

```
tls:
  enabled: true
  secretName: '${NAMESPACE}-dcli-tls-secret'
  rsaPrivateKeyFileName: '${NAMESPACE}-dcli-key.pem'
  ecdsaPrivateKeyFileName: 'dcli-ecdsa-private-key.pem'
  rsaCertificateFileName: '${NAMESPACE}-dcli-cert.crt'
  ecdsaCertifacateFileName: 'dcli-ecdsa-certificate.crt'
  caBundleFileName: 'ocats-caroot.cer'
```

2. Create certificate and private key generation for Diameter Simulaltor by using the following steps.

   a. Run the following command to create private key:

   ```
   openssl genrsa -out ${NAMESPACE}-dcli-key.pem 4096
   ```

   b. Run the following command to generate a certificate request for the generated private key:

   ```
   openssl req -new -sha256 -key ${NAMESPACE}-dcli-key.pem -out $
   {NAMESPACE}-dcli-csr.csr -subj "/CN=occnp-occnp-diam-gateway.$
   {namespace}.svc"
   ```

   c. Run the following command to generate a self signed root certificate:

   ```
   openssl x509 -req -sha256 -days 1000 \    -in ${NAMESPACE}-dcli-csr.csr
   \    -CA ocats-caroot.cer \    -CAkey ocats-caroot.key \    -out $
   {NAMESPACE}-dcli-cert.pem \    -CAcreateserial
   ```

   d. Run the following command to convert pem file to crt file:

   ```
   mv ${NAMESPACE}-dcli-cert.pem ${NAMESPACE}-dcli-cert.crt
   ```

3. Create a secret for the stub and associate it with the namespace using the following command:

   ```
   kubectl create secret generic ${NAMESPACE}-dcli-tls-secret --from-file=$
   {NAMESPACE}-dcli-key.pem --from-file=ocats-caroot.cer --from-file=$
   {NAMESPACE}-dcli-cert.crt -n ${NAMESPACE}
   ```

   Example:

   ```
   kubectl create secret generic u-ns1-dcli-tls-secret --from-file=u-ns1-dcli-
   key.pem --from-file=ocats-caroot.cer --from-file=u-ns1-dcli-cert.crt -n u-
   ns1
   secret/dcli-tls-secretcreated
   ```

3.4.3.5 Enable ATS GUI with HTTPS

Follow the steps to secure or enable TLS on the server.

Copyright © 2019, 2025, Oracle and/or its affiliates.

1. Create a Kubernetes secret by adding the above created files:

```
kubectl create secret generic ocats-tls-secret --from-
file=jenkinsserver.jks --from-file=ssl_rsa_certificate.crt --from-
file=rsa_private_key_pkcs1.key --from-file=caroot.cer -n ocpcf
```

Where,

jenkinsserver.jks: This file is needed when `atsGuiTLSEnabled` is set to true. This is necessary to open ATS GUI with secured TLS protocol.

ssl_rsa_certificate.crt: This is client application certificate.

rsa_private_key_pkcs1.key: This is RSA private key.

caroot.cer: This file used during creation of jks file needs to be passed for Jenkins/ATS API communication.

The sample of created secret:

```
[cloud-user@platform-bastion-1 ~]$ kubectl describe secret ocats-tls-
secret -n ocpcf
Name:         ocats-tls-secret
Namespace:    ocats
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
caroot.cer:                      1147 bytes
ssl_rsa_certificate.crt:         1424 bytes
jenkinsserver.jks:               2357 bytes
rsa_private_key_pkcs1.key:       1675 bytes
```

2. Apply the following changes in `values.yaml` file.

```
certificates:
  cert_secret_name: "ocats-tls-secret"
  ca_cert: "caroot.cer"
  client_cert: "ssl_rsa_certificate.crt"
  private_key: "rsa_private_key_pkcs1.pem"
  jks_file: "jenkinsserver.jks" # This parameter is needed when
atsGuiTLSEnabled
      is set to true. This file is necessary for ATS GUI to  be opend with
secured TLS protocol.
  jks_password: "123456"  #This is the password given to the jks file
while creation.
```

The user can install the ATS, using the helm install command. Change the `atsGuiTLSEnabled` Helm parameter value to true for ATS to get the certificates and support HTTPS for GUI.

3. Upload the `caroot.cer` file to the browser, before accessing it using `https` protocol.

Copyright © 2019, 2025, Oracle and/or its affiliates.

For more details about the uploading the file to the browser, refer to *Adding a Certificate in Browser* section in Enable ATS GUI with HTTPS.

4. A user can now start ATS with HTTPS the protocol. The link to open the ATS GUI format is `https://<IP>:<port>`, for example, `https://10.75.217.25:30301`.
   The lock symbol in the browser indicates that the server is secured or TLS enabled.

3.4.4 Pushing the Images to Customer Docker Registry

Preparing to deploy ATS and Stub Pods in Kubernetes Cluster

To deploy ATS and stub pods in Kubernetes Cluster, perform the following steps:

1. Run the following command to extract tar file content.
   ```
   tar -zxvf ocats-policy-tools-25.1.200.0.0.tgz
   ```
   The following is the output of this command:

   ```
   ocats-policy-pkg-25.1.200.0.0.tgz
   ocstub-pkg-25.1.201.0.0.tgz
   ocdns-pkg-25.1.204.0.0.tgz
   ocamf-stub-25.1.204.0.0.tgz
   oc-ldap-org1-25.1.204.tgz
   ocdiam-pkg-25.1.204.0.0tgz
   ```

2. Run the following command to extract the helm charts and docker images of Policy ATS:
   ```
   tar -zxvf ocats-policy-pkg-25.1.200.0.0.tgz
   ```
   The following is the output:

   ```
   ocats-policy-25.1.200.tgz
   ocats-policy-images-25.1.200.tar
   ocats-policy-data-25.1.200.tgz
   ```

3. Run the following command to load the ATS docker image:
   ```
   docker load --input ocats-policy-images-25.1.200.tar
   ```
4. Run the following commands to tag and push the ATS images:

   ```
   docker tag ocats-policy:25.1.200 <registry>/ocats-policy:25.1.200
   docker push <registry>/ocats-policy:25.1.200
   ```

   Example:

   ```
   docker tag ocats-policy:25.1.200 localhost:5000/ocats-policy:25.1.200
   docker push localhost:5000/ocats-policy:25.1.200
   ```

   > **Note**
   >
   > If you are using Podman instead of Docker, replace docker with podman in all the docker commands given in this document.

5. Run the following command to untar the helm charts – 25.1.200.

Copyright © 2019, 2025, Oracle and/or its affiliates.

```
tar -zxvf ocats-policy-25.1.200.tgz
```

> **Note**
>
> ```
> atsFeatures section is newly introduced in values.yaml which helps
> Engineering team to control feature deliveries over the releases.
> It is not advisable to update any of the following flags without Engineering
> team's permission.
>
> atsFeatures:  ## DO NOT UPDATE this section without Engineering
> team's permission
>   testCaseMapping: true               # To display Test cases on
> GUI along with Features
>   logging: true                       # To enable feature to
> collect applogs in case of failure
>   lightWeightPerformance: false       # The Feature is not
> implemented yet
>   executionWithTagging: true          # To enable Feature/Scenario
> execution with Tag
>   scenarioSelection: false            # The Feature is not
> implemented yet
>   parallelTestCaseExecution: true     # To run ATS features parallel
>   parallelFrameworkChangesIntegrated: true # To run ATS features
> parallel
>   mergedExecution: false              # To execute ATS Regression
> and NewFeatures pipelines together in merged manner
>   individualStageGroupSelection: false  # The Feature is not
> implemented yet
>   parameterization: true              # When set to false, the
> Configuration_Type parameter on the GUI will not be available.
>   atsApi: true                        # To trigger ATS using ATS API
>   healthcheck: true                   # TO enable/disable ATS
> Health Check.
>   atsGuiTLSEnabled: false             # To run ATS GUI in https
> mode.
>   atsCommunicationTLSEnabled: false  #If set to true, ATS will get
> necessary variables to communicate with SUT, Stub or other NFs with
> TLS enabled. It is not required in ASM environment.
> ```

6. Update the registry name, image name and tag in the `ocats-policy/values.yaml` file as required. For this, you need to update the `image.repository` and `image.tag` parameters in the `ocats-policy/values.yaml` file.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.4.5 Configuring ATS

3.4.5.1 Enabling Static Port

To enable static port, in the `ocats-policy/values.yaml` file under the service section, set the value of `staticNodePortEnabled` parameter to `true` and enter a valid nodePort value for `staticNodePort` parameter.

```
service:
  customExtension:
    labels: {}
    annotations: {}
  type: LoadBalancer
  ports:
    http:
      port: "8080"
      staticNodePortEnabled: false
      staticNodePort: ""
```

3.4.5.2 Enabling Static API Node Port

To enable static API node port, in the ocats-policy/values.yaml file under the service section, set the value of staticAPINodePortEnabled parameter to `true` and enter a valid nodePort value for staticAPINodePortEnabled parameter.

```
service:
  customExtension:
    labels: {}
    annotations: {}
  type: LoadBalancer
  ports:
    api:
      port: "5001"
      staticNodePortEnabled: false
      staticNodePort: ""
```

3.4.5.3 Service Account Requirements

To run Policy-ATS, use the following rules to create a service account:

```
rules:
- apiGroups: ["extensions"]
  resources: ["deployments", "replicasets", "statefulsets"]
  verbs: ["watch", "get", "list", "update"]
- apiGroups: ["apps"]
  resources: ["deployments", "replicasets", "statefulsets"]
  verbs: ["watch", "get", "list", "update"]
- apiGroups: [""]
  resources: ["pods", "services", "secrets", "configmaps"]
  verbs: ["watch", "get", "list", "delete", "update", "create"]
- apiGroups: [""]
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  resources: ["pods/log"]
  verbs: ["get", "list"]
```

To run oc-ldap ATS, use the following rules to create a service account:

```
PolicyRule:
  Resources                   Non-Resource URLs  Resource Names      Verbs
  ---------                   -----------------  --------------      -----
  deployments.apps            []                 [rc1-oc-ldap-org1]  [get
list watch create update patch delete]
  deployments.extensions      []                 [rc1-oc-ldap-org1]  [get
list watch create update patch delete]
  podsecuritypolicies.policy  []                 [1org1-rc1]         [use]
```

> **Note**
>
> For information about creating service account, see the *Oracle Communications Cloud Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide* available on MOS.

3.4.5.4 Enabling Aspen Service Mesh

This section provides information on how to enable Aspen service mesh while deploying ATS for CNC Policy. The configurations mentioned in this section are optional and should be performed only if ASM is required.

To enable service mesh for CNC Policy ATS, perform the following steps:

1. If the ASM is not enabled on the global level for the namespace, run the following command before deploying the ATS:

   ```
   kubectl label --overwrite namespace <namespace_name> istio-
   injection=enabled
   ```

   For example:

   ```
   kubectl label --overwrite namespace ocpcf istio-injection=enabled
   ```

2. In the **service** section of the `values.yaml` file, the **ASMEnabled** parameter is set to `false` (default configuration). To enable service mesh, set the value for **ASMEnabled** to `true`. The following is a snippet of the service section in the yaml file:

   ```
   service:
     customExtension:
       labels: {}
       annotations: {}
     type: LoadBalancer
     ports:
       https:
         port: "8443"
         staticNodePortEnabled: false
         staticNodePort: ""
       http:
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
      port: "8080"
      staticNodePortEnabled: false
      staticNodePort: ""
    api:
      port: "5001"
      staticNodePortEnabled: false
      staticNodePort: ""

  #####set ASMEnabled value to true for service-mesh based installation
  ASMEnabled: false
```

3. Uncomment and add the following annotation under the **customExtension** section of the global section in values.yaml file and deploy the ATS Pods:

```
customExtension:
    allResources:
      labels: {}
      annotations: {
      #Enable this section for service-mesh based installation
         traffic.sidecar.istio.io/excludeInboundPorts: "9000",
         traffic.sidecar.istio.io/excludeOutboundPorts: "9000"
        }
```

After making this update in the values.yaml file, make sure that all the ATS and stub pods come up with istio container.

4. For ServerHeader and SessionRetry features, the user needs to perform the following configurations under the envoyFilters for nf1stub, nf11stub, nf12stub, nf2stub, nf21stub, nf22stub, nf3stub, nf31stub, and nf32stub in the `occnp-servicemesh-config-custom-values-25.1.200.yaml`:

> **Note**
>
> occnp_custom_values_servicemesh_config yaml file and helm charts version names would differ based on the deployed Policy NF version. For example, "occnp_custom_values_servicemesh_config_24.2.0.yaml" or "occnp_custom_values_servicemesh_config_24.2.1.yaml".

```
envoyFilters:
  - name: serverheaderfilter-nf1stub
    labelselector: "app: nf1stub-ocstub-py"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionM
anager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
  - name: serverheaderfilter-nf11stub
    labelselector: "app: nf11stub-ocstub-py"
    configpatch:
      - applyTo: NETWORK_FILTER
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionM
anager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
  - name: serverheaderfilter-nf12stub
    labelselector: "app: nf12stub-ocstub-py"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionM
anager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
  - name: serverheaderfilter-nf2stub
    labelselector: "app: nf2stub-ocstub-py"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionM
anager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
  - name: serverheaderfilter-nf21stub
    labelselector: "app: nf21stub-ocstub-py"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionM
anager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
  - name: serverheaderfilter-nf22stub
    labelselector: "app: nf22stub-ocstub-py"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionM
anager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
  - name: serverheaderfilter-nf3stub
    labelselector: "app: nf3stub-ocstub-py"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionM
anager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
  - name: serverheaderfilter-nf31stub
    labelselector: "app: nf31stub-ocstub-py"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionM
anager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
  - name: serverheaderfilter-nf32stub
    labelselector: "app: nf32stub-ocstub-py"
    configpatch:
      - applyTo: NETWORK_FILTER
        filtername: envoy.filters.network.http_connection_manager
        operation: MERGE
        typeconfig: type.googleapis.com/
envoy.extensions.filters.network.http_connection_manager.v3.HttpConnectionM
anager
        configkey: server_header_transformation
        configvalue: PASS_THROUGH
```

5. Perform helm upgrade on the occnp-servicemesh-config release using the modified `occnp-servicemesh-config-custom-values-25.1.200.yaml`.

   ```
   helm upgrade <helm_release_name_for_servicemesh> -n <namespace>
   <servicemesh_charts> -f <servicemesh-custom.yaml>
   ```

   Example

   ```
   helm upgrade occnp-servicemesh-config occnp-servicemesh-
   config-25.1.200.tgz -n <namespace> -f occnp-servicemesh-config-custom-
   values-25.1.200.yaml
   ```

6. Configure DNS for Alternate Route service. For more information, see Post-Installation Steps.

3.4.5.5 Enabling Persistent Volume

ATS supports Persistent storage to retain ATS historical build execution data, test cases and one-time environment variable configurations. With this enhancement, the user can decide whether to use persistent volume based on their resource requirements. By default, the persistent volume feature is not enabled.

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> The steps provided in this section are optional and required only if Persistent Volume needs be to enabled.

To enable persistent storage, perform the following steps:

1. PVC creation is part of helm charts. To create the PVC, configure the following set of parameters in `values.yaml` file.
   `PVEnabled`: This indicates whether to enable PVC or not.

   `PVClaimName`: This indicates the name of persistent volume.

   `PVStorageClassName`: This indicates the storage classname.

   `PVStorage`: This indicates the size of the persistent volume.

   `retainPVC`: This indicates whether to delete or retain created PVC during uninstallation of ATS.

   A sample edited `values.yaml` file for enabling PVC:

   ```
   PVEnabled: true
   truePVClaimName: "policy-pvc-25.1.200"
   PVStorageClassName: "standard"
   PVStorage:"5Gi"
   retainPVC: true
   ```

   > **Note**
   >
   > - The value of `truePVClaimName` parameter to be suffixed with the release version to avoid confusion during the subsequent releases.
   > - To use already existing PVC, update the `PVClaimName` with existing name, ATS will use it. If not, a new PVC will be created with given properties.
   > - The `retainPVC` parameter if set to `true`, then deletes the PVC after uninstallation of ATS. Set it false for default behavior.

2. Verify the created PVC is bound to the Persistent Volume and available for use, by running:

   ```
   kubectl get pvc -n <namespace used for pvc creation>
   ```

   ```
   kubectl get pvc -n ocpcf
   ```

   Sample Output:

   ```
   kubectl get pvc -n ocpcf
   NAME                  STATUS   VOLUME
   CAPACITY       ACCESS MODES   STORAGECLASS   AGE
   policy-pvc-25.1.200   Bound    pvc-65484045-3805-4064-9fc3-f9eeeaccc8b8
   1Gi        RWO            standard       11s
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> - Do not proceed further with the next step if there is in issue with the PV creation and contact your administrator to get the PV Created
> - Make sure that ATS is deployed before proceeding to the further steps.

3. Copy the `<nf_main_folder>` and `<jenkins jobs>` folders from the tar file to their ATS pod and restart the pod.
   a. Extract the tar file.

      ```
      tar -xvf ocats-policy-data-25.1.200.tgz
      ```

   b. Run the following commands to copy the desired folder.

      ```
      kubectl cp ocats-policy-data-25.1.200/ocpcf_tests <namespace>/<pod-name>:/var/lib/jenkins/

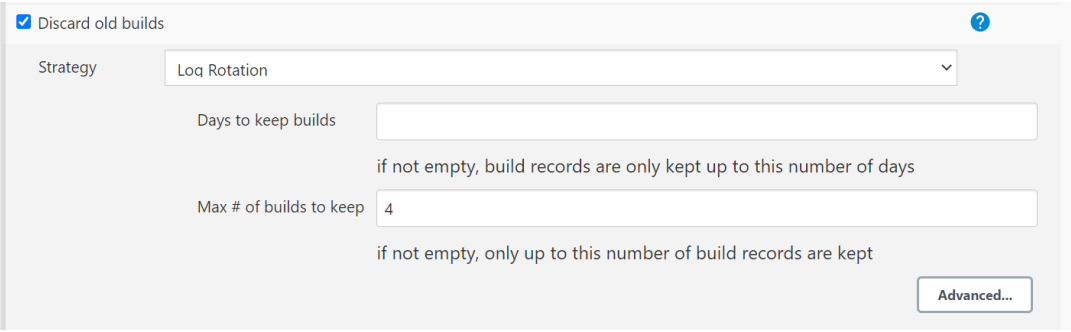

       kubectl cp ocats-policy-data-25.1.200/jobs <namespace>/<pod-name>:/var/lib/jenkins/.jenkins/
      ```

   c. Restart the pod.

      ```
      kubectl delete po <pod-name> -n <namespace>
      ```

4. Once the Pod is up and running, log in to the Jenkins console and configure the **Discard old Builds** option to configure the number of Jenkins builds, which must be retained in the persistent volume.

Figure 3-9 Discarding Old Builds



> **Note**
>
> If **Discard old Builds** is not configured, Persistent Volume can get filled when there are huge number of builds.

For more details on Persistent Volume Storage, see Persistent Volume for 5G ATS.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.4.5.6 Enabling Health Check

This section describes how to enable Health Check for ATS.

To enable Health Check, in the `ocats-policy/values.yaml` file, set the value of `healthcheck` parameter to `true` and enter a valid value to select the environment.

```
sshDetails:
  secretname: "healthchecksecret"
  envtype: ""
  occnehostip: ""
  occnehostusername: ""
  occnehostpassword: ""
  webscalejumpserverip: ""
  webscalejumpserverusername: ""
  webscalejumpserverpassword: ""
  webscaleprojectname: ""
  webscalelabserverFQDN: ""
  webscalelabserverport: ""
  webscalelabserverusername: ""
  webscalelabserverpassword: ""
  ociHealthCheck:
    passwordAuthenticationEnabled: false
    bastion:
      ip: ""
      username: ""
      password: ""
    operatorInstance:
      ip: ""
      username: ""
      password: ""
```

To select OCCNE environment, update the values of the following two parameters:

- `Webscale` - Update the value as false
- `envtype` - `T0NDTkU=` (i.e envtype=$(echo -n '**OCCNE**' | base64))
- `occnehostip` - OCCNE Host IP address
- `occnehostusername` - OCCNE Host Username
- `occnehostpassword` - OCCNE Host Password

The following is the sample configuration for OCCNE environment:

```
atsFeatures:  ## DO NOT UPDATE this section without Engineering team's
permission
  healthcheck: true                   # TO enable/disable ATS Health Check.

sshDetails:
  secretname: "healthchecksecret"
  envtype: "T0NDTkU="
  occnehostip: "MTAuMTcuMjE5LjY1"
  occnehostusername: "dXNlcm5hbWU"
  occnehostpassword: "KioqKg=="
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

To select WEBSCALE environment, update the values of the following two parameters:

- `Webscale` - Update the value as true
- `envtype` - V0VCU0NBTEU= (i.e envtype=$(echo -n 'WEBSCALE' | base64))

After the configurations are done, encrypt below parameters and provide the values as shown in the following snippet:

The following is the sample configuration for WEBSCALE environment:

```
atsFeatures:  ## DO NOT UPDATE this section without Engineering team's
permission
  healthcheck: true                   # TO enable/disable ATS Health Check.

 sshDetails:
  secretname: "healthchecksecret"
  envtype: "V0VCU0NBTEU="
  webscalejumpip: "MTAuNzAuMTE3LjQy"
  webscalejumpusername: "dXNlcm5hbWU="
  webscalejumppassword: "KioqKg=="
  webscaleprojectname: "KioqKg=="
  webscalelabserverFQDN: "KioqKg=="
  webscalelabserverport: "KioqKg=="
  webscalelabserverusername: "KioqKg=="
  webscalelabserverpassword: "KioqKg=="
```

> **Note**
>
> Once the ATS is deployed with HealthCheck feature enabled or disabled, then it cannot be changed. To change the configuration, you are required to re-install.

3.4.5.7 ATS-Policy API Extended Support

The ATS application programming interface (API) feature provides APIs, to perform routine ATS tasks such as starting the ATS suite, monitoring and stopping the ATS suite etc.

By default, this feature is enabled in `values.yaml` file.

```
  atsFeatures:
    atsApi: true
```

For more details about the ATS API feature, refer to ATS API section.

This ATS feature is extended to provide the ability of running single features, or scenarios, or stages, or groups, or based on tags execution using the API. This also allows running of test cases by providing the features, or scenarios, or stages, or groups, or tags in the `curl` request to the server.

For more details about the API interfaces, refer to Use the RESTful Interfaces section.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.4.6 Deploying ATS and Pods

3.4.6.1 Deploying ATS in Kubernetes Cluster

To deploy ATS, perform the following steps:

1. Run the following command using the updated helm charts.

> **Note**
>
> Ensure that all the components, ATS, go-Stub, dns-bind, ocamf, and CNC Policy are deployed in the same namespace.

Using Helm

```
helm install -name <release_name> ocats-policy-25.1.200.tgz --namespace
<namespace_name> -f <values-yaml-file>
```

Example:

```
helm install -name ocats ocats-policy-25.1.200.tgz --namespace ocpcf -f
ocats-policy/values.yaml
```

2. Run the following command to verify ATS deployment:
   `helm ls -n ocpcf`

   The sample output is as follows:

```
NAME    REVISION        UPDATED                     STATUS
CHART              APP VERSION     NAMESPACE
ocats      1         Mon November 6 14:56:11 2023      DEPLOYED    ocats-
policy-25.1.200      1.0             ocpcf
```

   The status appears as **DEPLOYED** after the deployment is successful.

3.4.6.2 Deploying Stub Pod in Kubernetes Cluster

To deploy Stub Pod in Kubernetes cluster, perform the following steps:

1. Navigate to the `ocats-policy-tools-25.1.200.0.0` folder and run the following command to extract the ocstub tar file content.

```
tar -zxvf ocstub-pkg-25.1.201.0.0.tgz
```

   The output of this command is:

   - ocstub-py-25.1.201.tgz
   - ocstub-py-image-25.1.201.tar

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> To deploy additional stubs required for session retry feature validation:
>
> - nf11stub, nf12stub → Alternate FQDN for nf1stub
> - nf21stub, nf22stub → Alternate FQDN for nf2stub
> - nf31stub, nf32stub → Alternate FQDN for nf3stub

2. Run the following command in your cluster to load the STUB image:

   ```
   docker load --input ocstub-py-image-25.1.201.tar
   ```

3. Run the following commands to tag and push the STUB image:

   ```
   docker tag ocstub-py:25.1.201 <registry>/ocstub-py:25.1.201
   docker push <registry>/ocstub-py:25.1.201
   ```

4. Run the following command to untar the helm charts, `ocstub-py-25.1.201.tgz`:

   ```
   tar -zxvf ocstub-py-25.1.201.tgz
   ```

5. Update the registry name, image name and tag (if required) in the `ocstub-py/values.yaml` file as required. For this, open the **values.yaml** file and update the `image.repository` and `image.tag` parameters.

   > **Note**
   >
   > From 24.2.0 onwards, service port names are configurable in ocstub-py. But as per Istio standard, it's advisable to keep the default values as it as.
   > Example:
   >
   > ```
   > names:
   >   http: "http"
   >   h2c: "http2-h2c"
   >   h2: "http2-h2"
   > ```

6. If required, change the `apiVersion` to `apps/v1` in the **ocstub-py/templates/deployment.yaml** file as follows:
   `apiVersion: apps/v1`
7. If the support for Predefined priming feature is required, perform the following steps to configure Predefined_priming
   a. Copy `ocstub-py/values.yaml` file to a new file with name `pre_priming_values.yaml`.
   b. Edit the `ocstub-py/pre_priming_values.yaml` file.
   c. Set the value of `preConfig` flag to `true` and replace the default configuration with below configurations under `predefined_prime_configuration` section.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Sample Predefined_priming configuration:

```
enabled: true
predefined_prime_configuration: |+
  [
    {
      "method": "GET",
      "statuscode": "200",
      "url": "/nnrf-nfm/v1/nf-instances/${nfInstanceId}",
      "data": "{\"nfInstanceId\": \"${nfInstanceId}\"}",
      "headers": "{\"Content-Type\":\"application/json\"}"
    },
    {
      "method": "PUT",
      "statuscode": "201",
      "url": "/nnrf-nfm/v1/nf-instances/${nfInstanceId}",
      "data": "{\"nfInstanceId\": \"${nfInstanceId}\", \"nfType\":
\"PCF\", \"nfStatus\": \"REGISTERED\", \"heartBeatTimer\": 300,
\"capacity\": 100, \"load\": 0, \"pcfInfo\": {\"dnnList\": [\"dnn1\",
\"dnn2\"], \"supiRanges\": [{\"start\": \"12123444444\", \"end\":
\"232332323323232\"}]}, \"nfServices\": [{\"serviceInstanceId\":
\"03063893-cf9e-4f7a-9827-067f6fa9dd01\", \"serviceName\": \"npcf-am-
policy-control\", \"versions\": [{\"apiVersionInUri\": \"v1\",
\"apiFullVersion\": \"1.R15.1.0\", \"expiry\":
\"2020-03-03T18:55:08.871+0000\"}], \"scheme\": \"http\",
\"nfServiceStatus\": \"REGISTERED\", \"fqdn\": \"ocpcf-pcf-ingress-
gateway.ocpcf.svc\", \"allowedNfTypes\": [\"AMF\", \"NEF\"],
\"capacity\": 100, \"load\": 0}, {\"serviceInstanceId\": \"03063893-
cf9e-4f7a-9827-067f6fa9dd02\", \"serviceName\": \"npcf-
smpolicycontrol\", \"versions\": [{\"apiVersionInUri\": \"v1\",
\"apiFullVersion\": \"1.R15.1.0\", \"expiry\":
\"2019-08-03T18:55:08.871+0000\"}], \"scheme\": \"http\",
\"nfServiceStatus\": \"REGISTERED\", \"fqdn\": \"ocpcf-api-gateway.occ-
demo-pcf.svc\", \"allowedNfTypes\": [\"SMF\", \"NEF\", \"AF\"],
\"capacity\": 100, \"load\": 0}]}",
      "headers": "{\"Content-Type\":\"application/json\"}"
    },
    {
      "method": "PATCH",
      "statuscode": "204",
      "url": "/nnrf-nfm/v1/nf-instances/${nfInstanceId}",
      "data": "{}",
      "headers": "{\"Content-Type\":\"application/json\"}"
    },
    {
      "method": "POST",
      "statuscode": "201",
      "url": "/nnrf-nfm/v1/subscriptions",
      "data": "{\"nfStatusNotificationUri\": \"http://ocpcf-pcf-ingress-
gateway.ocpcf.svc/nnrf-client/v1/notify\", \"reqNfType\": \"PCF\",
\"subscriptionId\": \"2d77e0de-15a9-11ea-8c5b-b2ca002e6839\",
\"validityTime\": \"2050-12-26T09:34:30.816Z\"}",
      "headers": "{\"Content-Type\": \"application/json\"}"
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  }
]
```

> **Note**
>
> - The `predefined_prime_configuration` contains variables such as `nfInstanceId`, `nfType`, and `fqdn` in the data's content. Ensure to verify and update the variables based on the payload message that must be included in the response from the NRF on a request.
> - The default value of `nfInstanceId` variable is `fe7d992b-0541-4c7d-ab84-c6d70b1b0123`.

8. Deploy Stub.

```
helm install -name <release_name> ocstub-py --set env.NF=<NF> --set
env.LOG_LEVEL=<DEBUG/INFO> --set service.name=<service_name>--set
service.appendReleaseName=false --namespace=<namespace_name> -f
<valuesyaml-file>
```

Install nf1stub and nf11stub with updated `ocstub-py/pre_priming_values.yaml` file.

> **Note**
>
> If the support for Predefined_priming feature is not required, helm installation must be performed using default `values.yaml` file.

```
helm install -name nf1stub ocstub-py --set env.NF=PCF --set
env.LOG_LEVEL=DEBUG --set service.name=nf1stub --set
service.appendReleaseName=false  --namespace=ocpcf -f ocstub-py/
pre_priming_values.yaml
```

Install other stubs such as nf12stub, nf11stub and others with default `values.yaml` file.

Example:

```
helm install -name nf1stub ocstub-py --set env.NF=PCF --set
env.LOG_LEVEL=DEBUG --set service.name=nf1stub --set
service.appendReleaseName=false  --namespace=ocpcf -f ocstub-py/values.yaml
helm install -name nf2stub ocstub-py --set env.NF=PCF --set
env.LOG_LEVEL=DEBUG --set service.name=nf2stub --set
service.appendReleaseName=false  --namespace=ocpcf -f ocstub-py/values.yaml
helm install -name nf3stub ocstub-py --set env.NF=PCF --set
env.LOG_LEVEL=DEBUG --set service.name=nf3stub --set
service.appendReleaseName=false  --namespace=ocpcf -f ocstub-py/values.yaml
helm install -name nf11stub ocstub-py --set env.NF=PCF --set
env.LOG_LEVEL=DEBUG --set service.name=nf11stub --set
service.appendReleaseName=false  --namespace=ocpcf -f ocstub-py/values.yaml
helm install -name nf12stub ocstub-py --set env.NF=PCF --set
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
env.LOG_LEVEL=DEBUG --set service.name=nf12stub --set
service.appendReleaseName=false  --namespace=ocpcf -f ocstub-py/values.yaml
helm install -name nf21stub ocstub-py --set env.NF=PCF --set
env.LOG_LEVEL=DEBUG --set service.name=nf21stub --set
service.appendReleaseName=false  --namespace=ocpcf -f ocstub-py/values.yaml
helm install -name nf22stub ocstub-py --set env.NF=PCF --set
env.LOG_LEVEL=DEBUG --set service.name=nf22stub --set
service.appendReleaseName=false  --namespace=ocpcf -f ocstub-py/values.yaml
helm install -name nf31stub ocstub-py --set env.NF=PCF --set
env.LOG_LEVEL=DEBUG --set service.name=nf31stub --set
service.appendReleaseName=false  --namespace=ocpcf -f ocstub-py/values.yaml
helm install -name nf32stub ocstub-py --set env.NF=PCF --set
env.LOG_LEVEL=DEBUG --set service.name=nf32stub --set
service.appendReleaseName=false  --namespace=ocpcf -f ocstub-py/values.yaml
```

9. Run the following command to verify stub deployment.
   `helm ls -n ocpcf`

   The sample output is as follows:

```
NAME         REVISION            UPDATED
STATUS          CHART                        APP VERSION     NAMESPACE
nf11stub               1              Tue March 14 10:05:59 2024
DEPLOYED        ocstub-py-25.1.201        1.0             ocpcf
nf12stub               1              Tue March 14 10:06:00 2024
DEPLOYED        ocstub-py-25.1.201        1.0             ocpcf
nf1stub                1              Tue March 14 10:05:57 2024
DEPLOYED        ocstub-py-25.1.201        1.0             ocpcf
nf21stub               1              Tue March 14 10:06:01 2024
DEPLOYED        ocstub-py-25.1.201        1.0             ocpcf
nf22stub               1              Tue March 14 10:06:02 2024
DEPLOYED        ocstub-py-25.1.201        1.0             ocpcf
nf2stub                1              Tue March 14 10:05:58 2024
DEPLOYED        ocstub-py-25.1.201        1.0             ocpcf
nf31stub               1              Tue March 14 10:06:03 2024
DEPLOYED        ocstub-py-25.1.201        1.0             ocpcf
nf32stub               1              Tue March 14 10:06:11 2024
DEPLOYED        ocstub-py-25.1.201        1.0             ocpcf
nf3stub                1              Tue March 14 10:05:59 2024
DEPLOYED        ocstub-py-25.1.201        1.0             ocpcf
```

   The status changes to **DEPLOYED** after the deployment is successful. Similarly, install all other stubs.

10. Run the following command to check the status of Stub deployment.
    `helm status <release_name> -n ocpcf`

    The sample output is as follows:

```
NAME                                        READY   STATUS    RESTARTS
AGE
nf11stub-ocstub-py-66449ddb94-qg2j9                   1/1     Running
0          19h
nf12stub-ocstub-py-6b8575487-l8pxv                    1/1     Running
0          19h
nf1stub-ocstub-py-5ff485954c-prc2x                    1/1     Running
0          19h
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
nf21stub-ocstub-py-56cf5b77fc-x8wkr                    1/1     Running
0          19h
nf22stub-ocstub-py-547dfdf476-4j2sn                    1/1     Running
0          19h
nf2stub-ocstub-py-6fb6f786d6-bc9fr                     1/1     Running
0          19h
nf31stub-ocstub-py-c6c6d5584-5m48z                     1/1     Running
0          19h
nf32stub-ocstub-py-848dfc7757-q797z                    1/1     Running
0          19h
nf3stub-ocstub-py-6cb769ccd9-4fv9b                     1/1     Running
0          19h
```

11. Run the following command to verify if all the services are installed.

    ```
    kubectl get po -n ocpcf
    ```

    A sample output of Policy namespace with Policy and ATS after installation is as follows:

    ```
    NAME                                              READY   STATUS    RESTARTS
    AGE
    ocpcf-appinfo-6c74cccd47-zsbb2                      1/1     Running
    0          155m
    ocpcf-oc-binding-77fbb9b79c-jv7kd                   1/1     Running
    0          155m
    ocpcf-oc-diam-connector-6c6fd868bd-4zfrn            1/1     Running
    0          155m
    ocpcf-oc-diam-gateway-0                             1/1     Running
    0          147m
    ocpcf-oc-oc-stub-595bb858d4-smzj8                   1/1     Running
    0          147m
    ocpcf-ocats-ocats-policy-667d8cf78-b8bc8            1/1     Running
    0          147m
    ocpcf-occnp-alternate-route-75455c858d-f6qs8        1/1     Running
    0          146m
    ocpcf-occnp-alternate-route-75455c858d-sqvlg        1/1     Running
    0          147m
    ocpcf-occnp-chf-connector-6b8b8bfcd6-jjch6          1/1     Running
    0          155m
    ocpcf-occnp-config-server-77bd99f96-mpscn           1/1     Running
    0          155m
    ocpcf-occnp-egress-gateway-59c4b784cc-6dx4w         1/1     Running
    0          16m
    ocpcf-occnp-ingress-gateway-75c47c57bc-pljtc        1/1     Running
    0          39m
    ocpcf-occnp-nrf-client-nfdiscovery-74b854956b-s6blq 1/1     Running
    0          155m
    ocpcf-occnp-nrf-client-nfmanagement-76cb55b8b8-tdjkj 1/1    Running
    0          49m
    ocpcf-occnp-udr-connector-75ffb9db9b-7xz9v          1/1     Running
    0          155m
    ocpcf-ocdns-ocdns-bind-57fbcd95dc-h4dtl             1/1     Running
    0          147m
    ocpcf-ocpm-audit-service-5cc46665c4-j6vhh           1/1     Running
    0          155m
    ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
ocpcf-ocpm-cm-service-7795bb4c6c-446rb                1/1     Running
0          155m
ocpcf-ocpm-policyds-75cbc9fc9d-7lbl5                  1/1     Running
0          155m
ocpcf-ocpm-pre-59b94d979-jzkv4                        1/1     Running
0          155m
ocpcf-ocpm-pre-test-84d9c89dd8-fqlpg                  1/1     Running
0          155m
ocpcf-ocpm-queryservice-94895bf88-bhwcc               1/1     Running
0          155m
ocpcf-pcf-amservice-56cdbb75c9-ph7tt                  1/1     Running
0          155m
ocpcf-pcf-smservice-64b899d766-jfhjm                  1/1     Running
0          155m
ocpcf-pcf-ueservice-7c6bd7ccc9-mrnxn                  1/1     Running
0          155m
ocpcf-pcrf-core-7594dbb7f8-z95vt                      1/1     Running
0          155m
ocpcf-performance-689dd556b-7vblc                     1/1     Running
0          155m
ocpcfnf11stub-5bb6b4f95d-v6fbb                        1/1     Running
0          147m
ocpcfnf12stub-59fb974f5d-2qr42                        1/1     Running
0          147m
ocpcfnf1stub-5bdf545fcb-zgbjb                         1/1     Running
0          147m
ocpcfnf21stub-ff6db9d86-5hvj6                         1/1     Running
0          147m
ocpcfnf22stub-794456fd66-sxq8q                        1/1     Running
0          147m
ocpcfnf2stub-656755dc46-hnr8m                         1/1     Running
0          147m
ocpcfnf31stub-68c6596b6-jdsgj                         1/1     Running
0          147m
ocpcfnf32stub-f49b57d86-rklc8                         1/1     Running
0          147m
ocpcfnf3stub-6c4c697648-lj6q7                         1/1     Running
0          147m
ocpcf-ocpm-ldap-gateway-5fd489b8fd-52dqn              1/1     Running
0          147m
```

12. Verify the changes related to stub pre-defined prime configurations. Run the following command to verify the status of all the config-map.

    ```
    kubectl get cm -n ocpcf
    ```

    Notice the change in the number of config-map counts. It includes two extra config-maps of stubs and the number will be the same as of stubs count.

    For example:

    ```
    NAME                                 DATA   AGE
    cm-pystub-nf1stub                     1      3h35m
    cm-pystub-nf11stub                    1      3h35m
    ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Updating the Predefined priming configurations

To update the Predefined priming configurations such as value of nfInstanceID in the url:

> **Note**
>
> This procedure is applicable only when Predefined priming configuration is enabled.

1. Run the following command to verify the status of all the config-maps.

   ```
   kubectl get cm -n ocpcf
   ```

2. Perform the following steps separately for nf1stub and nf11stub pods.
   a. Edit the config-map of the pod. To edit the config-map of nf1stub, run the command:

      ```
      kubectl edit cm cm-pystub-nf1stub -n ocpcf
      ```

      To edit the config-map of nf11stub, run the command:

      ```
      kubectl edit cm cm-pystub-nf11stub -n ocpcf
      ```

3. Edit the configurations as required, save and close the config-maps.
4. Restart the nf1stub and nf11stub pods.
5. Verify the logs of both these pods to confirm the changes.

3.4.6.3 Deploying DNS Stub in Kubernetes Cluster

> **Note**
>
> Ensure there is sufficient resource requests and limit is configured for DNS Stub. Set the resource request and limit values in the **resources** section in the **values.yaml** file as follows:

```
resources: {}
  # We usually recommend not to specify default resources and to leave
this as a conscious
  # choice for the user. This also increases chances charts run on
environments with little
  # resources, such as Minikube. If you do want to specify resources,
uncomment the following
  # lines, adjust them as necessary, and remove the curly braces after
'resources:'.  # limits:
  #  cpu: 1000m
  #  memory: 1024Mi
  # requests:
  #  cpu: 500m
  #  memory: 500Mi
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

To deploy DNS stub in Kubernetes cluster, perform the following steps:

1. Go to the **ocats-policy-tools-25.1.200.0.0** folder and run the following command:
   `tar -zxvf ocdns-pkg-25.1.204.0.0.tgz`

   The output of this command is:

   ```
   [cloud-user@platform-bastion-1 ocdns-pkg-25.1.204.0.0]$ ls -ltrh
   total 211M
   -rw-------. 1 cloud-user cloud-user 211M Mar 14 14:49 ocdns-bind-
   image-25.1.204.tar
   -rw-r--r--. 1 cloud-user cloud-user 2.9K Mar 14 14:49 ocdns-
   bind-25.1.204.tgz
   ```

2. Run the following command in your cluster to load the DNS Stub image:
   `docker load --input ocdns-bind-image-25.1.204.tar`
3. Run the following commands to tag and push the DNS stub to the registry:

   ```
   docker tag ocdns-bind:25.1.204 localhost:5000/ocdns-bind:25.1.204
   docker push localhost:5000/ocdns-bind:25.1.204
   ```

4. Run the following command to untar the helm charts (ocdns-bind-25.1.204.tgz):
   `tar -zxvf ocdns-bind-25.1.204.tgz`
5. Update the registry name, image name and tag (if required) in the `ocdns-bind/values.yaml` file as required. Open the **values.yaml** file and update the `image.repository` and `image.tag` parameters.
6. Run the following command to install DNS Stub:

   ```
   helm :
   [cloud-user@platform-bastion-1 ocdns-bind]$ helm install -name ocdns
   ocdns-bind-25.1.204.tgz --namespace ocpcf -f ocdns-bind/values.yaml
   ```

7. Run the following command to capture the cluster name of the pcf deployment, namespace where nfstubs are deployed and cluster IP of DNS Stub.
   `kubectl get svc -n ocpcf | grep dns`

   ```
   NAME      TYPE           CLUSTER-IP      EXTERNAL-IP
   PORT(S)               AGE
   ocdns     ClusterIP      10.233.11.45    <none>          53/
   UDP,6236/TCP        19h
   ```

> **Note**
>
> This information is required to configure DNS stub.

Figure 3-10 Cluster Name

```
kubectl -n kube-system get configmap kubeadm-config -o yaml | grep
clusterName
    clusterName: platform
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

3.4.6.4 Deploying AMF Stub in Kubernetes Cluster

To deploy OCAMF stub in Kubernetes cluster:

1. Go to the **ocats-policy-tools-25.1.200.0.0** folder and run the following command:
   `tar -zxvf ocamf-pkg-25.1.204.0.0.tgz`

   The output of this command is:

   ```
   [cloud-user@platform-bastion-1 ocamf-pkg-25.1.204.0.0]$ ls -ltrh
   total 211M
   -rw-------. 1 cloud-user cloud-user 211M Mar 14 14:49 ocamf-stub-
   image-25.1.204.tar
   -rw-r--r--. 1 cloud-user cloud-user 2.9K Mar 14 14:49 ocamf-
   stub-25.1.204.tgz
   ```

2. Run the following command in your cluster to load the AMF Stub image:
   `docker load --input ocamf-stub-image-25.1.204.tar`
3. Run the following command to tag and push the DNS stub to the registry:

   ```
   docker tag ocamf-stub:25.1.204 localhost:5000/ocamf-stub:25.1.204
   docker push localhost:5000/ocamf-stub:25.1.204
   ```

4. Run the following command to untar the helm charts (ocamf-stub-25.1.204.tgz):
   `tar -zxvf ocamf-stub-25.1.204.tgz`
5. Update the registry name, image name and tag (if required) in the `ocamf-stub/values.yaml` file as required. Open the **values.yaml** file and update the registry name, image name, and tag (if required) in the file.
6. Run the following command to install AMF Stub:
   Using Helm:

   ```
   [cloud-user@platform-bastion-1 ocamf-stub]$ helm install -name ocamf2
   ocamf-stub-25.1.204.tgz --set service.name=ocamf2 --namespace ocpcf -f
   ocamf-stub/values.yaml
   ```

The status changes to **RUNNING** after the deployment is successful.

The following is a sample output for a successful deployment:

```
ocamf2-ocamf-ocamf-stub-79c8fbd6f7-qp5cl               1/1     Running
0          5h47m
```

> **Note**
>
> Do not deploy ocamf-stub when TLS communication is enabled. Temporarily, ATS features for AM are disabled until ocamf-stub supports TLS communication. Even if the AM related ATS features are populated on the ATS UI, it will not be executed

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.4.6.5 Deploying LDAP Stub in Kubernetes Cluster

To deploy oc-ldap stub in the Kubernetes cluster, perform the following steps:

1. Go to the ocats-policy-tools-25.1.200.0.0 folder and run the following command:
   `tar -zxvf oc-ldap-org1-pkg-25.1.204.0.0`

   The following is the output:

   ```
   [cloud-user@platform-bastion-1 oc-ldap-org1-pkg-25.1.204.0.0]$ ls -ltrh
   total 211M
   -rw-------. 1 cloud-user cloud-user 211M Mar 14 14:49 oc-ldap-org1-
   image-25.1.204.tar
   -rw-r--r--. 1 cloud-user cloud-user 2.9K Mar 14 14:49 oc-ldap-
   org1-25.1.204.tgz
   ```

2. Run the following command in your cluster to load the LDAP Stub image:
   `docker load --input oc-ldap-org1-image-25.1.204.tar`
3. Run the following command to tag and push the DNS stub to the registry:

   ```
   docker tag oc-ldap-org1:25.1.204 localhost:5000/oc-ldap-org1:25.1.204
   docker push localhost:5000/oc-ldap-org1:25.1.204
   ```

4. Run the following command to untar the helm charts (oc-ldap-org1-25.1.204.tgz):
   `tar -zxvf oc-ldap-org1-25.1.204.tgz`
5. Update the registry name, image name and tag (if required) in the `oc-ldap-org1/values.yaml` file as required. Open the **values.yaml** file and update the registry name, image name, and tag (if required) in the file.
6. Run the following command to install AMF Stub:
   Using Helm:

   ```
   [cloud-user@platform-bastion-1 oc-ldap-org1]$ helm upgrade --install --
   namespace ocpcf --set image.repository=localhost:5000/occnp/oc-ldap-org1
   oc-ldap-org1 oc-ldap-org1-25.1.204.tgz
   ```

The status changes to **RUNNING** after the deployment is successful.

The following is a sample output for a successful deployment:

```
ocpcf-oc-ldap-org1-7b9d957bc6-ngtrl               1/1     Running
0          5h47m
```

> **Note**
>
> oc-ldap-org1-secret of the OC-LDAP Stub is being created by Helm chart that comes with ATS package.

3.4.6.6 Deploying ocdiam Simulator in Kubernetes Cluster

To deploy ocdiam Simulator in the Kubernetes cluster, perform the following steps:

Copyright © 2019, 2025, Oracle and/or its affiliates.

1. Go to the ocats-policy-tools-25.1.200.0.0 folder and run the following command:

   ```
   tar -zxvf ocdiam-pkg-25.1.204.0.0
   ```

   The following is the output:

   ```
   [cloud-user@platform-bastion-1 ocdiam-pkg-25.1.204.0.0]$ ls -ltrh
   total 908M
   -rw-------. 1 cloud-user cloud-user 908M Mar 14 14:49 ocdiam-sim-
   image-25.1.204.tar
   -rw-r--r--. 1 cloud-user cloud-user 3.8K Mar 14 14:49 ocdiam-
   sim-25.1.204.tgz
   ```

2. Run the following command in your cluster to load the Diameter Simulator Image using the command :

   ```
   docker load --input ocdiam-sim-image-25.1.204.tar
   ```

3. Run the following command to tag and push the DNS stub to the registry:

   ```
   docker tag ocdiam-sim:25.1.204 localhost:5000/ocdiam-sim:25.1.204
   docker push localhost:5000/ocdiam-sim:25.1.204
   ```

4. Run the following command to untar the helm charts (ocdiam-sim-25.1.204.tgz):

   ```
   tar -zxvf ocdiam-sim-25.1.204.tgz
   ```

5. Update the registry name, image name and tag (if required) in the `ocdiam-sim/values.yaml` file as required. Open the **values.yaml** file and update the registry name, image name, and tag (if required) in the file.
6. Run the following command to install Diameter Simulator:
   Using Helm:

   ```
   [cloud-user@platform-bastion-1 ocdiam-sim]$ helm install -name ocdiam-sim
   ocdiam-sim --namespace ocpcf -f ocdiam-sim/values.yaml
   ```

The status changes to **RUNNING** after the deployment is successful.

The following is a sample output for a successful deployment:

```
ocdiam-sim-69968444b6-fg6ks    1/1     Running   0   5h47m
```

Sample of Policy namespace with Policy and ATS after installation:

```
[cloud-user@platform-bastion-1 ocstub-pkg-25.1.201.0.0]$ kubectl get po -n
ocpcf
NAME                                                READY   STATUS
RESTARTS   AGE
ocpcf-appinfo-6c74cccd47-zsbb2                      1/1     Running
0          155m
ocpcf-oc-binding-77fbb9b79c-jv7kd                   1/1     Running
0          155m
ocpcf-oc-diam-connector-6c6fd868bd-4zfrn            1/1     Running
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
0          155m
ocpcf-oc-diam-gateway-0                                  1/1     Running
0          147m
ocamf2-ocamf-stub-595bb858d4-smzj8                       1/1     Running
0          147m
ocpcf-ocats-ocats-policy-667d8cf78-b8bc8                 1/1     Running
0          147m
ocpcf-occnp-alternate-route-75455c858d-f6qs8             1/1     Running
0          146m
ocpcf-occnp-chf-connector-6b8b8bfcd6-jjch6               1/1     Running
0          155m
ocpcf-occnp-config-server-77bd99f96-mpscn                1/1     Running
0          155m
ocpcf-occnp-egress-gateway-59c4b784cc-6dx4w              1/1     Running
0          16m
ocpcf-occnp-ingress-gateway-75c47c57bc-pljtc             1/1     Running
0          39m
ocpcf-occnp-nrf-client-nfdiscovery-74b854956b-s6blq      1/1     Running
0          155m
ocpcf-occnp-nrf-client-nfmanagement-76cb55b8b8-tdjkj     1/1     Running
0          49m
ocpcf-occnp-udr-connector-75ffb9db9b-7xz9v               1/1     Running
0          155m
ocpcf-ocdns-ocdns-bind-57fbcd95dc-h4dtl                  1/1     Running
0          147m
ocpcf-ocpm-audit-service-5cc46665c4-j6vhh                1/1     Running
0          155m
ocpcf-ocpm-cm-service-7795bb4c6c-446rb                   1/1     Running
0          155m
ocpcf-ocpm-policyds-75cbc9fc9d-7lbl5                     1/1     Running
0          155m
ocpcf-ocpm-pre-59b94d979-jzkv4                           1/1     Running
0          155m
ocpcf-ocpm-pre-test-84d9c89dd8-fqlpg                     1/1     Running
0          155m
ocpcf-ocpm-queryservice-94895bf88-bhwcc                  1/1     Running
0          155m
ocpcf-pcf-amservice-56cdbb75c9-ph7tt
ocpcf-pcf-smservice-64b899d766-jfhjm                     1/1     Running
0          155m
ocpcf-pcf-ueservice-7c6bd7ccc9-mrnxn                     1/1     Running
0          155m
ocpcf-pcrf-core-7594dbb7f8-z95vt                         1/1     Running
0          155m
ocpcf-performance-689dd556b-7vblc                        1/1     Running
0          155m
ocpcfnf11stub-5bb6b4f95d-v6fbb                           1/1     Running
0          147m
ocpcfnf12stub-59fb974f5d-2qr42                           1/1     Running
0          147m
ocpcfnf1stub-5bdf545fcb-zgbjb                            1/1     Running
0          147m
ocpcfnf21stub-ff6db9d86-5hvj6                            1/1     Running
0          147m
ocpcfnf22stub-794456fd66-sxq8q                           1/1     Running
0          147m
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
ocpcfnf2stub-656755dc46-hnr8m                          1/1     Running
0          147m
ocpcfnf31stub-68c6596b6-jdsgj                          1/1     Running
0          147m
ocpcfnf32stub-f49b57d86-rklc8                          1/1     Running
0          147m
ocpcfnf3stub-6c4c697648-lj6q7                          1/1     Running
0          147m
ocpcf-ocpm-ldap-gateway-5fd489b8fd-52dqn               1/1     Running
0          147m
ocdiam-sim-69968444b6                                  1/1     Running
0          147m
```

3.4.7 Post-Installation Steps

This section describes the post-installation steps for Policy.

Alternate Route Service Configurations

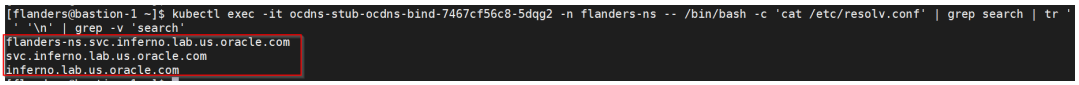
To edit the Alternate Route Service deployment file (ocpcf-occnp-alternate-route) that points to DNS Stub, perform the following steps:

1. Run the following command to get searches information from dns-bind pod to enable communication between Alternate Route and dns-bind service:

   ```
   kubectl exec -it <dns-bind pod> -n <NAMESPACE> -- /bin/bash -c 'cat /etc/
   resolv.conf' | grep search | tr ' ' '\n' | grep -v 'search'
   ```

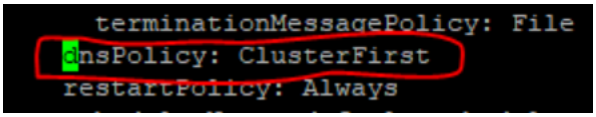
   The following output is displayed after running the command:

   Figure 3-11 Sample Output

   By default alternate service will point to CoreDNS and you will see following settings in deployment file:

   Figure 3-12 Alternate Route Service Deployment File

2. Run the following command to edit the deployment file and add the following content in alternate service to query DNS stub:

   ```
   $kubectl edit deployment ocpcf-occnp-alternate-route -n ocpcf
   ```

   a. Add the IP Address of the nameserver that you have recorded after installing the DNS stub (cluster IP Address of DNS Stub).

Copyright © 2019, 2025, Oracle and/or its affiliates.

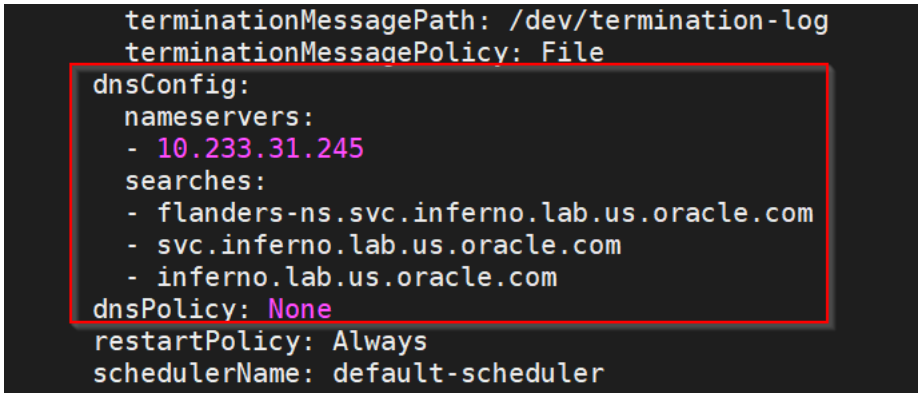
b. Add the search information one by one which you recorded earlier.

c. Set dnsPolicy to "None".

```
dnsConfig:
  nameservers:
  - 10.233.33.169      // cluster IP of DNS Stub
  searches:
  - ocpcf.svc.occne15-ocpcf-ats
  - svc.occne15-ocpcf-ats
  - occne15-ocpcf-ats
dnsPolicy: None
```

For example:

Figure 3-13 Example



NRF client configmap

In the `-application-config` configmap, configure the following parameters with the respective values:

- `primaryNrfApiRoot=nf1stub.<namespace_gostubs_are_deployed_in>.svc:8080`
  Example: `primaryNrfApiRoot=nf1stub.ocats.svc:8080`
- `secondaryNrfApiRoot=nf1stub.ocats.svc:8080 (Remove the secondaryNrfApiRoot)`
- nrfClientSubscribeTypes=UDR, CHF, NWDAF
- supportedDataSetId=POLICY (Remove the supportedDataSetId)

> **Note**
>
> Configure these values at the time of Policy deployment.

> **Note**
>
> To get all configmaps in your namespace, execute the following command:
> `kubectl get configmaps -n <Policy_namespace>`

Copyright © 2019, 2025, Oracle and/or its affiliates.

Persistent Volume (Optional)

If persistent volume is used, follow the post-installation steps provided in the Persistent Volume for 5G ATS section.

3.5 Installing ATS for SCP

This section describes Automated Testing Suite (ATS) installation procedures for Service Communication Proxy (SCP) in a cloud native environment.

You must perform ATS installation procedures for SCP in the same sequence as outlined in the following sections.

3.5.1 Prerequisites

To run SCP test cases, the following prerequisites are required.

3.5.1.1 Software Requirements

This section lists the software that must be installed before installing ATS.

Table 3-14 Preinstalled Software

Software	Version
Kubernetes	1.32.x, 1.31.x, 1.30.x
Helm	3.17.1
Podman	4.9.4

To check the versions of the preinstalled software in the cloud native environment, run the following commands:

```
kubectl version
```

```
helm version
```

```
podman version
```

3.5.1.2 Environment Setup Requirements

This section describes the requirements for the client machine, that is, the machine used by the user to run deployment commands.

The client machine should have:

- Helm repository configured.
- Network access to the Helm repository and Docker image repository.
- Network access to the Kubernetes cluster.
- Required environment settings to run `kubectl`, `docker`, and `podman` commands. The environment should have privileges to create a namespace in the Kubernetes cluster.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- Helm client is installed with the push plugin. Configure the environment in such a manner that the `helm install` command deploys the software in the Kubernetes cluster.

3.5.1.3 Resource Requirements

This section describes ATS resource requirements for SCP.

Overview - Total Number of Resources

The following table describes the total resource usage by different resource types:

- SCP SUT
- cnDB Tier
- ATS

Table 3-15 SCP - Total Number of Resources

Resource Name	CPU	Memory (GB)	Storage (GB)
SCP SUT Totals	61	66	0
cnDBTier Totals	40	40	20
ATS Totals	100	106	4
Grand Total SCP ATS	201	212	24

SCP Pods Resource Requirements

This section describes the resources required to deploy SCP ATS.

Table 3-16 SCP Pods Resource Requirements for a Non ASM Setup

Microservice	CPUs Required per Pod	Memory Required per Pod (GB)	Storage PVC Required per Pod (GB)	# Replicas (ATS deployment)	CPUs Required (ATS) - Total	Memory Required (ATS) - Total (GB)	Storage PVC Required - Total (GB)
SCP Pods							
scpc-subscription	2	2	0	1	2	2	0
scpc-notification	4	4	0	1	4	4	0
scpc-audit	3	4	0	1	3	4	0
scpc-configuration	2	2	0	1	2	2	0
scp-worker	8	12	0	1	8	12	0
scpc-alternate-resolution	2	2	0	1	2	2	0
scp-nrfproxy	8	8	0	1	8	8	0
scp-cache	8	8	0	1	8	8	0
scp-mediation	8	8	0	1	8	8	0
scp-load-manager	8	8	0	1	8	8	0
scp-oauth-nrfproxy	8	8	0	1	8	8	0
SCP SUT Totals					**61**	**66**	0

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-17 SCP Pods Resource Requirements for an ASM Setup

Microservice	CPUs Required per Pod	vCPUs(Sidecar)	Memory Required per Pod (GB)	Memory(required by sidecar)	Storage PVC Required per Pod (GB)	# Replicas (ATS deployment)	CPUs Required (ATS) - Total	Memory Required (ATS) - Total (GB)	Storage PVC Required - Total (GB)
SCP Pods									
scpc-subscription	2	2	2	1	0	1	4	3	0
scpc-notification	4	2	4	1	0	1	6	5	0
scpc-audit	3	2	4	1	0	1	5	5	0
scpc-configuration	2	2	2	1	0	1	4	3	0
scp-worker	8	2	12	1	0	1	10	13	0
scpc-alternate-resolution	2	2	2	1	0	1	4	3	0
scp-nrfproxy	8	2	8	1	0	1	10	9	0
scp-cache	8	2	8	1	0	1	10	9	0
scp-mediation	8	2	8	1	0	1	10	9	0
scp-load-manager	8	2	8	1	0	1	10	9	0
scp-oauth-nrfproxy	8	2	8	1	0	1	10	9	0
SCP SUT Totals							**83**	**77**	0

ATS Resource Requirements for SCP

This section describes the ATS resources required to deploy SCP-ATS.

Table 3-18 ATS Resource Requirements for a Non ASM Setup

Microservice	CPUs Required per Pod	Memory Required per Pod (GB)	Storage PVC Required per Pod (GB)	# Replicas (ATS deployment)	CPUs Required - Total	Memory Required - Total (GB)	Storage PVC Required - Total (GB)
ATS Behave	6	12	4	1	6	12	4
ATS pystub	1	1	-	91	91	91	0
DNS Stub	1	1	-	1	1	1	0
Global Egress Rate Limiting Stub	1	1	-	1	1	1	0
ATS DD Client stub	1	1	-	1	1	1	0
ATS Totals					**100**	**106**	4

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-19 ATS Resource Requirements for an ASM Setup

Microservice	CPUs Required per Pod	vCPUs(required by sidecar)	Memory Required per Pod (GB)	Memory(required by sidecar)	Storage PVC Required per Pod (GB)	# Replicas (ATS deployment)	CPUs Required - Total	Memory Required - Total (GB)	Storage PVC Required - Total (GB)
ATS Behave	6	2	12	1	4	1	8	13	4
ATS pystub	1	2	1	1	-	91	273	182	0
DNS Stub	1	2	1	1	-	1	3	2	0
Global Egress Rate Limiting Stub	1	2	1	1	-	1	3	2	0
ATS DD Client stub	1	2	1	1	-	1	3	2	0
ATS Totals							**290**	**201**	4

3.5.1.4 Downloading the ATS Package

This section provides information about how to download the ATS package.

To locate and download the ATS Image from MOS:

1. Log in to My Oracle Support using the appropriate login credentials.
2. Click the **Patches & Updates** tab.
3. In the Patch Search section, click **Product or Family (Advanced)**.
4. Enter `Oracle Communications Cloud Native Core - 5G` in the **Product** field.
5. Select `Oracle Communications Cloud Native Core Service Communication Proxy <release_number>` from the **Release** drop-down list.
6. Click **Search**.
   The Patch Advanced Search Results list appears.
7. Select the required ATS patch from the list.
   The Patch Details window appears.
8. Click **Download**.
   The File Download window appears.
9. Click the `ocats_ocscp_csar_25_1_202_0_0.pkg` file to download the CNC SCP ATS package file.
   The `ocats_ocscp_csar_25_1_202_0_0.pkg` package contains the following files:

   ```
   ocats_ocscp_csar_25_1_202_0_0.zip
   mcafee-gen-ats-csar-25.1.202.log
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> The above zip file contains all the images and custom values required for 25.1.202 release of OCATS-OCSCP.

Unzip the `ocats_ocscp_csar_25_1_202_0_0.zip` file to get the following files and folders:

```
.
|-- Definitions
|  |-- ocats_ocscp_ats_tests.yaml
|  |-- ocats_ocscp_cne_compatibility.yaml
|  `-- ocats_ocscp.yaml
|-- Files
|  |-- ChangeLog.txt
|  |-- Helm
|  |  `-- ocats-ocscp-25.1.202.tgz
|  |-- Licenses
|  |-- ocats-ddclientstub-25.1.202.tar
|  |-- ocats-dnsstub-25.1.202.tar (Docker Image)
|  |-- ocats-pystub-25.1.202.tar (Docker Image)
|  |-- ocats-scp-25.1.202.tar (Docker Image)
|  |-- ocats-scpglbratelimitstub-25.1.202.tar (Docker Image)
|  |-- Oracle.cert
|  `-- Tests
|-- ocats_ocscp_csar_25_1_202_0_0.zip
|-- ocats_ocscp.mf
|-- Scripts
|  |-- ocats_ocscp_custom_serviceaccount_25.1.202.yaml (Template to create
custom service account)
|  |-- ocats_ocscp_tests_jenkinsjobs_25.1.202.tgz. ( ocscp_tests and
jenkins jobs folder to be copied if persistent volume is deployed)
|  `-- ocats_ocscp_values_25.1.202.yaml (Custom values file for
installation)
`-- TOSCA-Metadata
  `-- TOSCA.meta
```

10. Copy the umbrella Helm chart `ocats-ocscp-25.1.202.tgz` file from the **Files** folder to Kubernetes cluster where you want to deploy ATS.
    The following table describes ATS parameters in the `ocats_ocscp_values_25.1.202.yaml` file:

Table 3-20 ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
ocatsdnsstubService	true	true, false	Set it to true or false depending on the need of user. Setting these values to true to deploy ocats-dnsstub, ocats-scp stub, ocats-scpglbratelimitstub, ocats-ddclientStub, ocats-pystubs (ausf1, udm3 etc.)

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-20 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
ocatsscpService	true	true, false	Set it to true or false depending on the requirement. Setting these values to true to deploy ocats-dnsstub, ocats-scp stub, ocats-scpglbratelimitstub, ocats-ddclientStub, ocats-pystubs (ausf1, udm3, and so on)
ocatsscpglbratelimitstubService	true	true, false	Set it to true or false depending on the requirement. Setting these values to true to deploy ocats-dnsstub, ocats-scp stub, ocats-scpglbratelimitstub, ocats-ddclientStub, ocats-pystubs (ausf1, udm3, and so on.)
ocatsddclientstubService	true	true, false	Set it to true or false depending on the requirement. Setting these values to true to deploy ocats-dnsstub, ocats-scp stub, ocats-scpglbratelimitstub, ocats-ddclientStub, ocats-pystubs (ausf1, udm3,and so on.)
ausf1Stubs	true	true, false	Set it to true or false depending on the requirement. Setting these values to true to deploy all the ocats-pystubs (ausf1, udm3, chf1, scp3, and so on.)
sutScpIPFamiliesforATSRun	[IPv4]	[IPv4], [IPv6], [IPv4,IPv6], [IPv6,IPv4]	The parameter is used to specify the IP families that ATS will consider when running test cases. **Note**: If any value other than those specified is provided, ATS will proceed with the assumption that the deployment supports IPv4 only.
traffic.sidecar.istio.io/ excludeOutboundPorts	8091	-	This annotation under lbDeployment is required for fetching metrics from soothsayer pods, which are required for a few FT's of ATS. When ATS runs in an ASPEN MESH environment, do not change this port.
traffic.sidecar.istio.io/ excludeInboundPorts	8080	-	This annotation under lbDeployment is required for fetching metrics from soothsayer pods, which are required for a few FT's of ATS. When ATS runs in an ASPEN MESH environment, do not change this port.
tokenConsumptionIntervalInMillis	50	Count in millliseconds	Token Consumption Simulation Parameters.
scpStubNfInstanceId	2faf1bbc-6e4a-4454-a507-a14ef8e1bc22	nfInstanceID of SCP	NF Instance ID of SCP.
rateDataReporterStartingOffset	35	Value in milliseconds	-

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-20 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
coherence.clusterName	scpstub-coherence-cluster	Local Coherence Cluster name, size not more than 66 characters	Local Coherence Cluster name size must not be more than 66 characters.
coherence.clusterName.federation.remoteScpOne.fqdnOrIp	ocscp-scp-cache.scpsvc.svc.cluster.local	-	FQDN or IP of federation configuration.
coherence.clusterName.federation.remoteScpOne.port	30001	-	Port number of federation configuration.
coherence.clusterName.federation.remoteScpOne.clusterName	scp-coherence-cluster	Ensure that the cluster name is unique among all participants. Size not more than 66 characters.	remoteScpOne Coherence Cluster name.
serviceIpFamilyPolicy.ocatsdnsstubService serviceIpFamilyPolicy.ocatsscpService serviceIpFamilyPolicy.ocatsscpglbratelimitstubService serviceIpFamilyPolicy.ocatsddclientstubService serviceIpFamilyPolicy.stubService	SingleStack	SingleStack, PreferDualStack, RequireDualStack	IpFamilyPolicy of ocatsdnsstubService, ocatsscpService, ocatsscpglbratelimitstubService, ocatsddclientstubService and pyStubs. **Note**: PreferDualStack and RequireDualStack values can only be used if the setup is dual stack.
serviceIpFamilies.ocatsdnsstubService serviceIpFamilies.ocatsscpService serviceIpFamilies.ocatsscpglbratelimitstubService serviceIpFamilies.ocatsddclientstubService serviceIpFamilies.stubService	[IPv4]	[IPv4], [IPv6], [IPv4,IPv6], [IPv6,IPv4]	IpFamilies of ocatsdnsstubService, ocatsscpService, ocatsscpglbratelimitstubService, ocatsddclientstubService and pyStubs. **Note**: If serviceIpFamilyPolicy is SingleStack, then serviceIpFamilies can be [IPv4] or [IPv6]. If serviceIpFamilyPolicy is PreferDualStack or RequireDualStack, then serviceIpFamilies can be [IPv4,IPv6] or [IPv6, IPv4].
PVEnabled	false	true, false	Enabling persistent volume.
PVClaimName	false	Name	Persistent volume claim name.
atsGuiTLSEnabled	false	true, false	Enabling Https in Jenkins GUI.
atsCommunicationTLSEnabled	false	true, false	Enabling Https communication for ATS and stubs.
ocats-scp.image.repository	<docker-registryIP:docker-registryport>/ocats/ocats-scp	<Image repository name:port>/ocats/ocats-scp	Image repository and port of ocats scp image.
ocats-scp.image.tag	helm-tag	Value of tag to be deployed	Tag of ocats-scp image.
ocats-scp.image.pullPolicy	Always	Always, IfNotPresent, Never	Tag of ocats-scp image.
ocats-scp.replicaCount	1	Positive integers	Replica count of ocats-scp stub.
ocats-scp.resources.limits.cpu	6	CPU value that is allocated	Limit to CPU allocated to ocats-scp pod.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-20 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
ocats-scp.resources.limits.memory	12Gi	Memory values that is allocated (in Gi or mi)	Limit to memory allocated to ocats-scp pod .
ocats-scp.resources.requests.cpu	6	CPU value that is allocated (must be less than or equal to limits)	Request of CPU allocation for ocats-scp.
ocats-scp.resources.requests.memory	12Gi	Memory value that is allocated (in Gi or mi) (must be less than or equal to limits)	Request of memory allocation for ocats-scp.
ocats-scp.service.customExtension.labels	{}	Label of node	Node labels for node allocation during deployment.
ocats-scp.service.customExtension.type	LoadBalancer	ClusterIP, NodePort, LoadBalancer	Service type of ocats-scp pod
ocats-scp.service.customExtension.port	8080	Port number	Port number of ocats-scp service.
ocats-scp.service.customExtension.staticNodePortEnabled	false	true, false	Enabling of static node port.
ocats-scp.service.customExtension.staticNodePort	false	Port number	Port number of static node port.
ocats-scp.service.ports.http.port	8080	Port number	Port number of the ocats-scp service if https is not enabled.
ocats-scp.service.ports.http.staticNodePortEnabled	false	true, false	Enabling of static node port.
ocats-scp.service.ports.http.staticNodePort	false	Port number	Port number of static node port.
ocats-scp.service.ipFamilyPolicy	SingleStack	SingleStack, PreferDualStack, RequireDualStack	ipFamilyPolicy to be allocated to the ocats-scp service. This value will be referred to as whatever is defined at serviceIpFamilyPolicy.ocatsscpService under global parameters.
ocats-scp.service.ipFamilies	[IPv4]	[IPv4], [IPv6], [IPv4,IPv6], [IPv6,IPv4]	ipFamilies to be allocated to the ocats-scp service. This value will be referred to as the value defined at serviceIpFamilies.ocatsscpService under global parameters.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-20 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
SELECTED_NF	SCP	NF name	ATS parameters are set with default values in the ocats_ocscp_values_23.3.0.yaml file. Update ATS parameters with the actual value based on the environment, and then deploy the OCATS_OCSCP chart using this ocats_ocscp_values_23.3.0.yaml file. The updated ATS parameters are automatically applied during the deployment process, and the ATS will come up with the configuration as mentioned in this file.
NFNAMESPACE	scpsvc	Update the SCP namespace	-
CLUSTERDOMAIN	cluster.local	Cluster Domain where SCP is Deployed	-
DESTNAMESPACE	scpsvc	Test Stubs NameSpace same as SCP Namespace	-
ocats-dnsstub.image.repository	<docker-registryIP:docker-registryport>/ocats/ocats-dnsstub	<Image repository name:port>/ocats/ocats-dnsstub	Image repository and port of ocats-dnsstub.
ocats-dnsstub.image.tag	helm-tag	Value of tag to be deployed	Tag of ocats-dnsstub image.
ocats-dnsstub.image.pullPolicy	Always	Always, IfNotPresent, Never	Image pull policy of ocats-dnsstub image.
ocats-dnsstub.replicaCount	1	Positive integers	Replica count of ocats-dnsstub.
ocats-dnsstub.service.customExtension.type	ClusterIP	ClusterIP, NodePort, LoadBalancer	Service type of ocats-dnsstub pod.
ocats-dnsstub.service.customExtension.port	53	Port	Port of ocats-dnsstub.
ocats-dnsstub.service.ipFamilyPolicy	SingleStack	SingleStack, PreferDualStack, RequireDualStack	ipFamilyPolicy to be allocated to the ocats-dnsstub service. This value will be referred to as whatever is defined at serviceIpFamilyPolicy.ocatsdnsstubService under global parameters.
ocats-dnsstub.service.ipFamilies	[IPv4]	[IPv4], [IPv6], [IPv4,IPv6], [IPv6,IPv4]	ipFamilies to be allocated to the ocats-dnsstub service. This value will be referred to as whatever is defined at serviceIpFamilies.ocatsdnsstubService under global parameters.
ocats-dnsstub.resources.limits.cpu	1	CPU value that is allocated	Limit to CPU allocated to ocats-dnsstub pod.
ocats-dnsstub.resources.limits.memory	1Gi	Memory value that is allocated (in Gi or mi)	Limit to memory allocated to ocats-dnsstub pod .

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-20 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
ocats-dnsstub.resources.requests.cpu	1	CPU value that is allocated (must be less than or equal to limits)	Request of CPU allocation for ocats-dnsstub.
ocats-dnsstub.resources.requests.memory	1Gi	Memory values that is allocated (in Gi or mi) (must be less than or equal to limits)	Request of memory allocation for ocats-dnsstub.
ocats-ddclientstub.image.repository	<docker-registryIP:docker-registryport>/ocats/ocats-ddclientstub	<Image repository name:port>/ocats/ocats-ddclientstub	Image repository and port of ocats-ddclientstub.
ocats-ddclientstub.image.tag	helm-tag	Value of tag to be deployed	Tag of ocats-ddclientstub image.
ocats-ddclientstub.image.pullPolicy	Always	Always, IfNotPresent, Never	Image pull policy of ocats-ddclientstub image.
ocats-ddclientstub.replicaCount	1	Positive integers	Replica count of ocats-ddclientstub.
ocats-ddclientstub.service.type	LoadBalancer	ClusterIP, NodePort, LoadBalancer	Service type of ocats-ddclientstub.
ocats-ddclientstub.ipFamilyPolicy	SingleStack	SingleStack, PreferDualStack, RequireDualStack	ipFamilyPolicy to be allocated to the ocatsddclientstubService service. This value will be referred to as whatever is defined at serviceIpFamilyPolicy.ocatsddclientstubService under global parameters.
ocats-ddclientstub.ipFamilies	[IPv4]	[IPv4], [IPv6], [IPv4,IPv6], [IPv6,IPv4]	ipFamilies to be allocated to the ocatsddclientstubService service. This value will be referred to as whatever is defined at serviceIpFamilies.ocatsddclientstubService under global parameters.
ocats-ddclientstub.resources.limits.cpu	1	CPU value that is allocated	Limit to CPU allocated to ocats-ddclientstub pod.
ocats-ddclientstub.resources.limits.memory	1Gi	Memory value that is allocated (in Gi or mi)	Limit to memory allocated to ocats-ddclientstub pod.
ocats-ddclientstub.resources.requests.cpu	1	CPU value that is allocated (should be less than or equal to limits)	Request of CPU allocation for ocats-ddclientstub.
ocats-ddclientstub.resources.requests.memory	1Gi	Memory value that is allocated (in Gi or mi) (should be less than or equal to limits)	Request of memory allocation for ocats-ddclientstub.
ocats-ddclientstub.log.level	INFO	INFO, WARN, DEBUG	Log level of ddclientstub pod.
ocats-ddclientstub.kafka_broker	"kafka-broker-0.kafka-broker.ddkafkanamespace.svc.cluster.local:9092"	Kafka broker fqdn and port	Kafka broker fqdn or port for ddClientStub.
ocats-ddclientstub.string_topic_name	"string_topic"	String topic	ddClientStub string topic name.
ocats-ddclientstub.json_topic_name	"json_topic"	Json topic	ddClientStub json topic name.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-20 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
ocats-scpglbratelimitstub.image.repository	<docker-registry IP:docker-registry port>/ocats/ocats-scpglbratelimitstub	<Image repository name:port>/ocats/ocats-scpglbratelimitstub	Image repository and port of ocats-scpglbratelimitstub.
ocats-scpglbratelimitstub.image.tag	helm-tag	Value of tag to be deployed	Tag of ocats-scpglbratelimitstub.
ocats-scpglbratelimitstub.image.pullPolicy	Always	Always, IfNotPresent, Never	Image pull policy of ocats-scpglbratelimitstub.
ocats-scpglbratelimitstub.replicaCount	1	Positive integers	Replica count of ocats-scpglbratelimitstub.
ocats-scpglbratelimitstub.service.type	ClusterIP	ClusterIP, NodePort, LoadBalancer	Service type of ocats-scpglbratelimitstub.
ocats-scpglbratelimitstub.service.ipFamilyPolicy	SingleStack	SingleStack, PreferDualStack, RequireDualStack	ipFamilyPolicy to be allocated to the ocatsscpglbratelimitstubService service. This value will be referred to as whatever is defined at serviceIpFamilyPolicy.ocatsscpglbratelimitstubService under global parameters.
ocats-scpglbratelimitstub.service.ipFamilies	[IPv4]	[IPv4], [IPv6], [IPv4,IPv6], [IPv6,IPv4]	ipFamilies to be allocated to the ocatsscpglbratelimitstubService service. This value will be referred to as whatever is defined at serviceIpFamilies.ocatsscpglbratelimitstubService under global parameters.
ocats-scpglbratelimitstub.deployment.customExtension.labels	{}	Label of node	Node labels for node allocation during deployment.
ocats-scpglbratelimitstub.resources.limits.cpu	1	CPU value that is allocated	Limit to CPU allocated to ocats-scpglbratelimitstub.
ocats-scpglbratelimitstub.resources.limits.memory	1Gi	Memory values that is allocated (in Gi or mi)	Limit to memory allocated to ocats-scpglbratelimitstub.
ocats-scpglbratelimitstub.resources.requests.cpu	1	CPU value that is allocated (should be less than or equal to limits)	Request of CPU allocation for ocats-scpglbratelimitstub.
ocats-scpglbratelimitstub.resources.requests.memory:	1Gi	Memory value that is allocated (in Gi or mi) (should be less than or equal to limits)	Request of memory allocation for ocats-scpglbratelimitstub.
ocats-scpglbratelimitstub.minreplicas	1	Positive integer	Minimum replicas of ocats-scpglbratelimitstub.
ocats-scpglbratelimitstub.maxreplicas	1	Positive integer	Maximum replicas of ocats-scpglbratelimitstub.
ocats-scpglbratelimitstub.maxPdbUnavailable	1	Positive integer	-

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-20 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
ocats-scpglbratelimitstub.log.level	INFO	INFO, WARN, DEBUG	Log level of ocats-scpglbratelimitstub.
ocats-pystub.image.repository	<docker-registry IP:docker-registry port>/ocats/ocats-pystub	<Image repository name:port>/ocats/ocats-pystub	Image repository and port of ocats-pystub.
ocats-pystub.image.tag	helm-tag	Value of tag to be deployed	Tag of ocats-pystub.
ocats-pystub.image.pullPolicy	Always	Always, IfNotPresent, Never	Image pull policy of ocats-pystub.
ocats-pystub.replicaCount	1	Positive integers	Replica count of ocats-pystub.
ocats-pystub.service.type	ClusterIP	ClusterIP, NodePort, LoadBalancer	Service type of ocats-pystub
ocats-pystub.service.ports.https.port	8443	port number	Indicates the port number to allocate for HTTPS to ocats-pystub.
ocats-pystub.service.ports.http.port	8080	port number	Indicates the port number to allocate for HTTP to ocats-pystub.
ocats-pystub.service.ipFamilyPolicy	SingleStack	SingleStack, PreferDualStack, RequireDualStack	Indicates the allocation of "ipFamilyPolicy" to pystub services. This value will be referenced from whatever is defined under " serviceIpFamilyPolicy.stubService" within global parameters.
ocats-pystub.service.ipFamilies	[IPv4]	[IPv4], [IPv6], [IPv4,IPv6], [IPv6,IPv4]	Indicates the allocation of "ipFamilies" to pystub services. This value will be referenced from whatever is defined under "serviceIpFamilies.stubService" within global parameters.
RESPONSE_FROM_HEADER	true	true, false	When true pystub returns podname.
ocats-pystub.resources.limits.cpu	1	CPU value that is allocated	Indicates the CPU allocation limit for ocats-pystub.
ocats-pystub.resources.limits.memory	1Gi	Memory values allocated in Gi or Mi.	Indicates the memory allocation limit for ocats-pystub.
ocats-pystub.resources.requests.cpu	1	CPU value allocated (should be less than or equal to limits).	Indicates the CPU allocation request for ocats-pystub.
ocats-pystub.resources.limits.memory	1Gi	Memory value allocated (in Gi or Mi) should be less than or equal to limits.	Indicates the memory allocation request for ocats-pystub.
ausf1.service.name:*	ausf1svc	Service name of ausf1	Indicates the service name allocated to ausf1.
ausf1.deploymentName:*	ausf1	Deployment name of ausf1	Indicates the deployment name allocated to ausf1.

Note

"*" indicates that all NF stubs will follow the same pattern.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.5.1.5 Pushing the Images to Customer Docker Registry

Preparing to Deploy ATS and Stub Pod in Kubernetes Cluster

To deploy ATS and Stub Pods in the Kubernetes Cluster:

1. Click the file to download the CNC SCP ATS package file:

   ```
   ocats_ocscp_csar_25_1_202_0_0.pkg
   ```

   The package contains the following files:

   ```
   ocats_ocscp_csar_25_1_202_0_0.zip
   mcafee-gen-ats-csar-25.1.202.log
   ```

   Unzip the `ocats_ocscp_csar_25_1_202_0_0.zip` file to get the following files and folders:

   The output of this command is:

   ```
   .
   |-- Definitions
   |  |-- ocats_ocscp_ats_tests.yaml
   |  |-- ocats_ocscp_cne_compatibility.yaml
   |  `-- ocats_ocscp.yaml
   |-- Files
   |  |-- ChangeLog.txt
   |  |-- Helm
   |  |  `-- ocats-ocscp-25.1.202.tgz
   |  |-- Licenses
   |  |-- ocats-ddclientstub-25.1.202.tar
   |  |-- ocats-dnsstub-25.1.202.tar (Docker Image)
   |  |-- ocats-pystub-25.1.202.tar (Docker Image)
   |  |-- ocats-scp-25.1.202.tar (Docker Image)
   |  |-- ocats-scpglbratelimitstub-25.1.202.tar (Docker Image)
   |  |-- Oracle.cert
   |  `-- Tests
   |-- ocats_ocscp_csar_25_1_202_0_0.zip
   |-- ocats_ocscp.mf
   |-- Scripts
   |  |-- ocats_ocscp_custom_serviceaccount_25.1.202.yaml (Template to create
   custom service account)
   |  |-- ocats_ocscp_tests_jenkinsjobs_25.1.202.tgz. ( ocscp_tests and
   jenkins jobs folder to be copied if persistent volume is deployed)
   |  `-- ocats_ocscp_values_25.1.202.yaml (Custom values file for
   installation)
   `-- TOSCA-Metadata
     `-- TOSCA.meta
   ```

2. Run the following commands in your cluster to load the ATS and stubs docker images, `ocats-scp-image-25.1.203.tar`, and push it to your registry:

   ```
   docker load --input ocats-scp-25.1.202.tar
   docker load --input ocats-dnsstub-25.1.202.tar
   docker load --input ocats-pystub-25.1.202.tar
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
docker load --input ocats-scpglbratelimitstub.tar
docker load --input ocats-ddclientstub-25.1.202.tar
```

3. Run the following command in your cluster to load the ATS image:

   ```
   docker load --input ocats-scp-images-25.1.202.tar
   ```

4. Run the following commands to push the ATS image to the registry:

   ```
   docker tag ocats/ocats-scp:25.1.202 <local_registry>/ocats/
   ocatsscp:25.1.202
   docker push <local_registry>/ocats/ocats-scp:25.1.202
   ```

   Where, `<local_registry>` indicates the registry where you can push the downloaded images.

5. Run the following commands to push the Stub image to the registry:

   ```
   docker tag ocats/ocats-pystub:25.1.202 <local_registry>/ocats/ocats-
   pystub:25.1.202
   docker push <local_registry>/ocats/ocats-pystub:25.1.202
   ```

6. Run the following command to push the DNS Stub Image to the registry:

   ```
   docker tag ocats/ocats-dnsstub:25.1.202 <local_registry>/ocats/ocats-
   dnsstub:25.1.202
   docker push <local_registry>/ocats/ocats-dnsstub:25.1.202
   ```

7. Run the following command to push the Global Rate Limiting Stub Image to the registry:

   ```
   docker tag ocats/ocats-scpglbratelimitstub:25.1.202 <local_registry>/ocats/
   ocats-scpglbratelimitstub:25.1.202
   docker push <local_registry>/ocats/ocats-scpglbratelimitstub:25.1.202
   ```

8. Run the following command to push the Data Director Stub Image to the registry:

   ```
   docker tag ocats/ocats-ddclientstub:25.1.202 <local_registry>/ocats/ocats-
   ddclientstub:25.1.202
   docker push <local_registry>/ocats/ocats-ddclientstub:25.1.202
   ```

9. In the **Scripts** folder, extract the following content:

   ```
   ocats-ocscp-values-25.1.202.yaml
   ocats-ocscp-custom-serviceaccount-25.1.202.yaml
   ocats-ocscp-tests-jenkinsjobs-25.1.202.tgz
   ```

10. Update the image name and tag in the `ocats-ocscp-values-25.1.202.yaml` file as required.

3.5.1.6 Preinstall Preparation of SCP for SCP-ATS

Complete the following steps before performing an installation:

Copyright © 2019, 2025, Oracle and/or its affiliates.

- When deploying default ATS with role binding, deploy ATS and test stubs in the same namespace as SCP.
- The SCP stub for the Global Egress Rate Limiting feature must be deployed by setting the required Helm parameters as described in *Oracle Communications Cloud Native Core, Service Communication Proxy Installation, Upgrade, and Fault Recovery Guide* to support the Global Egress Rate Limiting test cases.
- In the `ocats_ocscp_values_25.1.202.yaml`, add the following for Prometheus that is required for alert test case:

  ```
  traffic.sidecar.istio.io/excludeInboundPorts: "9090"
  ```

- If ASM is adding additional XFCC headers, the `certExtractIndex` and `extractIndex` of the `xfccHeaderDecode` value must be -1, otherwise `certExtractIndex` and `extractIndex` of `xfccHeaderDecode` value must be 0.
- If ASM is enabled, for fetching the metrics from Prometheus, a destination rule must be created. In most deployments, Prometheus is kept outside of the service mesh, so a destination rule is required to communicate between a TLS enabled entity (ATS) and a non-TLS entity (Prometheus). The rule can be created as follows:

  ```
  kubectl apply -f - <<EOF

  apiVersion:networking.istio.io/v1alpha3
  kind:DestinationRule
  metadata:
    name:prometheus-dr
    namespace:ocscp
  spec:
    host:oso-prometheus-server.ocscp.svc.cluster.local
    trafficPolicy:
      tls:
        mode:DISABLE
  EOF
  ```

  Where,

  - `name` indicates the name of the destination rule.
  - `namespace` indicates where ATS is deployed.
  - `host` indicates the hostname of the Prometheus server.
- FQDN and interPlmnFqdn must be the same for both nfServices for NRF profile NRF1.
  Sample

  ```
  # NRF profiles for primary(Priority=0) and secondry(Priority=1) NRF. Note
  that these NRFs needs to be backend DB Synced.
  # For Secondary NRF profile always make it priority lesser than First
  priority NRF, currently we set secondary NRF priority to 1.
  # In case of no secondry NRF user can comment the secondary NRF Profile.
  # Service level FQDN's of NRF are from the same namespace as that of SCP,
  this is put for SCP ATS cases. Otherwise, NRF's can be part of other
  namespaces or even other k8s clusters.
    nrfProfiles:
    - capacity: 10000
  ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    locality: USEast
    nfInstanceId: 6faf1bbc-6e4a-2828-a507-a14ef8e1bc5a
    nfStatus: REGISTERED
    nfType: NRF
    priority: '0'
    # with rel15 flag is enabled, below mentioned NRF region to be
specified.
#    nfSetIdList: ["Reg1"]
    # with rel16 flag is enabled, below mentioned NRF set Id to be
specified.
    nfSetIdList: ["setnrfl1.nrfset.5gc.mnc012.mcc345"]
    #Uncomment below section to configure interPlmnFqdn, plmnList or
snpnList
    #NRF-Change-3
    interPlmnFqdn: nrf1.5gc.mnc213.mcc410.3gppnetwork.org
    plmnList:
    - mcc: 410
      mnc: 213
    - mcc: 410
      mnc: 214
    #snpnList:
    #- mcc: 345
    #  mnc: 445
    #  nid: 000007ed9d5
    customInfo:
      preferredNrfForOnDemandDiscovery: true


    nfServices:
    - capacity: 5000
      #apiPrefix: USEast
      #NRF-Change-4
      fqdn: nrf1svc.scpsvc.svc.cluster.local
      interPlmnFqdn: nrf1svc.scpsvc.svc.cluster.local
      # Sample IPEndpoint with all the fields and it is commented out.
      #NRF-Change-5
      #ipEndPoints: [{"ipv4Address": "NRF-IP", "port": "NRF-PORT"}]
      #ipEndPoints: [{"ipv4Address": "10.75.213.56", "port": "31014"}]
      # ATS test cases need 8080 port with FQDN. Hence, in order to run
ATS cases, below "ipEndPoints" field is left uncommented.
      #NRF-Change-6
      ipEndPoints: [{"port": "8080"}]
      load: 0
      nfServiceStatus: REGISTERED
      scheme: http
      serviceInstanceId: fe137ab7-740a-46ee-aa5c-951806d77b01
      serviceName: nnrf-nfm
      priority: 0
      versions:
      - apiFullVersion: 1.0.0
        apiVersionInUri: v1

    - capacity: 5000
      #apiPrefix: USEast
      #NRF-Change-4
      fqdn: nrf1svc.scpsvc.svc.cluster.local
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
      interPlmnFqdn: nrf1svc.scpsvc.svc.cluster.local
      # Sample IPEndpoint with all the fields and it is commented out.
      #NRF-Change-5
      #ipEndPoints: [{"ipv4Address": "NRF-IP", "port": "NRF-PORT"}]
      #ipEndPoints: [{"ipv4Address": "10.75.213.56", "port": "31014"}]
      # ATS test cases need 8080 port with FQDN. Hence, in order to run
ATS cases, below "ipEndPoints" field is left uncommented.
      #NRF-Change-6
      ipEndPoints: [{"port": "8080"}]
      load: 0
      nfServiceStatus: REGISTERED
      scheme: http
      serviceInstanceId: fe137ab7-740a-46ee-aa5c-951806d77b02
      serviceName: nnrf-disc
      priority: 0
      versions:
      - apiFullVersion: 1.0.0
        apiVersionInUri: v1

    - capacity: 5000
      #apiPrefix: USEast
      #NRF-Change-4
      fqdn: nrf1svc.scpsvc.svc.cluster.local
      interPlmnFqdn: nrf1.5gc.mnc213.mcc410.3gppnetwork.org
      # Sample IPEndpoint with all the fields and it is commented out.
      #NRF-Change-5
      #ipEndPoints: [{"ipv4Address": "NRF-IP", "port": "NRF-PORT"}]
      #ipEndPoints: [{"ipv4Address": "10.75.213.56", "port": "31014"}]
      # ATS test cases need 8080 port with FQDN. Hence, in order to run
ATS cases, below "ipEndPoints" field is left uncommented.
      #NRF-Change-6
      ipEndPoints: [{"port": "8080"}]
      load: 0
      nfServiceStatus: REGISTERED
      scheme: http
      serviceInstanceId: fe137ab7-740a-46ee-aa5c-951806d77b03
      serviceName: nnrf-oauth2
      priority: 0
      versions:
      - apiFullVersion: 1.0.0
        apiVersionInUri: v1
```

- To execute SCP ATS under the `scp values.yaml`, ensure that `dnsSrvSchemeConfig.defaultScheme` is set to ‘https’.
- Ensure SCP is deployed with the following parameters:
  - Make sure that while providing NRF information at the time of SCP deployment, stub NRF details like **nrf1svc** and **nrf2svc** should also be provided at the time of ATS deployment before running these test cases. For example, if the teststub namespace is scpsvc, then SCP should have been deployed with primary NRF as `nrf1svc.scpsvc.svc.<clusterDomain>` and secondary NRF as `nrf2svc.scpsvc.svc.<clusterDomain>` for NRF test cases to work.
  - Ensure the `defaultTopologySource` parameter is set to `NRF` in the `ocscp_values.yaml` file.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- Ensure the `preventiveAuditOnLastNFInstanceDeletion` parameter is set to `false` in the `ocscp_values.yaml` file.
- The number of replicas of all SCP microservices pods must be set to 1 during SCP deployment as ATS is enabled to perform metric validations for metrics obtained from a single pod.
- When you deploy, make sure to define the additional NRF stubs needed for InterSCP cases as `nrfr2l1svc` (preferred NRF of Reg2), `nrfr2l2svc` (non-preferred NRF of Reg2), `nrfr3l1svc` (non-preferred NRF of Reg3), and `nrfr3l2svc` (preferred NRF of Reg3), which are provided in the default custom value file. Also, in the SCP deployment file, ensure that the namespace of all these NRFs is the same as the deployed SCP namespace. Reg1, Reg2, and Reg3 are replaced with `setnrfl1.nrfset.5gc.mnc012.mcc345`, `setnrfr1.nrfset.5gc.mnc012.mcc345`, and `setnrfr2.nrfset.5gc.mnc012.mcc345` for Release 16 SCP deployment.
- Ensure the `supportedNRFRegionOrSetIdList` must have Reg1, Reg2, and Reg3 for Release 15 SCP deployment or `setnrfl1.nrfset.5gc.mnc012.mcc345`, `setnrfr1.nrfset.5gc.mnc012.mcc345`, and `setnrfr2.nrfset.5gc.mnc012.mcc345` for Release 16 SCP deployment.
- Ensure only Loc7, Loc8, Loc9, and USEast should be part of the `servingLocalities` for Release 15 SCP deployment and the `servingScope` for Release 16 SCP deployment.
- Recommended `auditInterval` is 60 seconds and `guardTime` is 10 seconds in the SCP deployment file.
- The regions such as `Reg2` and `Reg3` are the corresponding values for Release 15 SCP deployment, while NRF setIDS such as `servingLocalities` and `setnrfr2.nrfset.5gc.mnc012.mcc345` are the corresponding values for Release 16 SCP deployment. The NRF belonging to either regions or the NRF set IDS's localities must not match with the SCP `servingLocalities` or SCP serving scope.
- SCP deployment file should have the attribute `scpToRegisterWithNrfRegions` set to `Reg1` for Release 15 SCP deployment and `setnrfl1.nrfset.5gc.mnc012.mcc345` for Release 16 SCP deployment. For information about Release 15 and Release 16, see 3GPP TS 23.501.
- To run CCA Validation feature tests, refer to Configuring ATS for CCA Test Cases section.
- To enable OAuth support while deploying SCP, refer Configuring SCP to Run OAuth Test Cases in ATS section.
- To enable alternate resolution service support while deploying SCP, refer to Configuring SCP to Run DNS SRV Test Cases in ATS section.
- To enable mediation support while deploying SCP, refer to Configuring SCP to Run Mediation Test Cases in ATS section.
- To enable nrfproxy support, refer to Configuring SCP to Run Model D Test Cases in ATS section.
- To enable load manager support, refer to Configuring SCP to Run LCI Test Cases in ATS section.
- To enable the Global Egress Rate Limiting feature for ATS environment, refer to Updating the Global Egress Rate Limiting Changes in the SCP Deployment File for ATS section.

Copyright © 2019, 2025, Oracle and/or its affiliates.

– By default, the ATS suite runs HTTPS test cases if the "ALL" option is selected, and SCP must be deployed with HTTPS support enabled to support the same. To enable HTTPs for ATS, refer to Enabling HTTPs for ATS, pystubs and Jenkins

```
# If ingress gateway is available then set ingressGWAvailable flag to
true
  # and provide ingress gateway IP and Port in publicSignalingIP and
publicSignalingPort respectively.

      publicSignalingPort: &publicSignalingPort 8000   #Signaling PORT
      publicSignalingPortHttps: &publicSignalingPortHttps 9443
#Signaling PORT for HTTPS
    # uncomment below lines when deployed with release 16. Also, take a
note that http port for SCP should be same as "publicSignalingPort" of
SCP mentioned above.
    scpInfo:
      scpPrefix: scpPrefix
      scpPorts:
       http: *publicSignalingPort
    #Uncomment below key-value(https) to enable https for ingress
connections for rel16 deployment. this port should be same as
"publicSignalingPortHttps" of SCP mentioned above.
    #   https: *publicSignalingPortHttps
     # Note: If this flag is false, then by default all connections to
PNF will be made using http protocol.
     nativeEgressHttpsSupport: false
```

If SCP is deployed with HTTP support, either the single or multiple feature execution option must be selected, excluding all https test cases. Otherwise, make sure in an ASM environment where HTTPS is not enabled; in that case, manually remove the HTTPS-related test cases from the features directory on the ATS pod.

3.5.1.7 Preinstallation Preparation for SCP-ATS

Complete the following steps before performing an installation of SCP-ATS.

3.5.1.7.1 Enabling Aspen Service Mesh

To enable Aspen Service Mesh (ASM) for ATS, complete the following procedure:

> **Note**
>
> By default, this feature is disabled.

1. If ASM is not enabled on the global level for the namespace, run the following command before deploying ATS:

```
kubectl label --overwrite namespace <namespace_name> istio-
injection=enabled
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Example:

```
kubectl label --overwrite namespace scpsvc istio-injection=enabled
```

2. Add the following annotations in the lbDeployments section of the global section in the `ocats_scp_values_25.1.202.yaml` file:

```
traffic.sidecar.istio.io/excludeOutboundPorts: "8091"
traffic.sidecar.istio.io/excludeInboundPorts: "8080"
```

Sample file with annotations:

```
lbDeployments:
labels: {}
annotations:
traffic.sidecar.istio.io/excludeOutboundPorts: "8091"
traffic.sidecar.istio.io/excludeInboundPorts: "8080"
```

3. Add Envoy Filter to enable the XFCC header forwarding by ASM sidecar.
Envoy Filter for ATS:

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  name: <name>
  namespace: <namespace>
spec:
  workloadSelector:
    labels:
      app: ocats-scp
  configPatches:
  - applyTo: NETWORK_FILTER
    match:
      listener:
        filterChain:
          filter:
            name: "envoy.http_connection_manager"
    patch:
      operation: MERGE
      value:
        typed_config:
          '@type': type.googleapis.com/
envoy.config.filter.network.http_connection_manager.v2.HttpConnectionManage
r
          forward_client_cert_details: ALWAYS_FORWARD_ONLY
          use_remote_address: true
          xff_num_trusted_hops: 1
```

Envoy filter to enable the XFCC header forwarding on the application sidecar:

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  name: <name>
  namespace: <namespace>
spec:
  workloadSelector:
    labels:
      app.kubernetes.io/instance: ocscp
  configPatches:
  - applyTo: NETWORK_FILTER
    match:
      listener:
        filterChain:
          filter:
            name: "envoy.http_connection_manager"
    patch:
      operation: MERGE
      value:
        typed_config:
          '@type': type.googleapis.com/
envoy.config.filter.network.http_connection_manage
r
          forward_client_cert_details: ALWAYS_FORWARD_ONLY
          use_remote_address: true
          xff_num_trusted_hops: 1
```

Envoy filter to enable server header pass through on sidecar:

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  name: <name>
  namespace: <namespace>
spec:
  configPatches:
  - applyTo: NETWORK_FILTER
    match:
      listener:
        filterChain:
          filter:
            name: envoy.filters.network.http_connection_manager
    patch:
      operation: MERGE
      value:
        typed_config:
          '@type': type.googleapis.com/
envoy.config.filter.network.http_connection_manager.v2.HttpConnectionManage
r
          server_header_transformation: PASS_THROUGH
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Updating Virtual Services

> **Note**
>
> - The sidecar configuration for response timeout or stream timeout should not be applied for any of the SCP microservices.
> - For virtual service CRD, :
>   - when the destinationhost is any SCP microservice, do not configure the timeout value.
>   - SCP producer NFs as destinationhosts can be configured with a timeout value of more than 50 seconds, or it should not be configured.

Disabling retry attempts in virtual services:

For all SCP and ATS pods in the virtual services, the retry attempt should be set to 0.

```
retries:
  attempts: 0
```

3.5.1.7.2 Enabling Persistent Volume

ATS supports persistent storage to retain ATS historical build execution data, test cases, and one-time environment variable configurations.

To enable persistent storage:

1. Create a PVC and associate it with the ATS pod.
2. Set the **PVEnabled** flag to **true** in the `ocats_ocscp_values_25.1.202.yaml` file.
3. Set **PVClaimName** to PVC that is created for ATS.

```
ocats-scp:
  PVEnabled: false
  PVClaimName: "ocats-scp-25.1.202-pvc"
```

> **Note**
>
> In the event that Persistent Volume (PV) is enabled, ATS starts up with the parameter values specified in the `ocats_ocscp_values_25.1.202.yaml` file. If the ATS pod is restarted, the PV restores the configuration, ensuring that the new ATS pod will have the same configuration settings as the previous pod.

For more details on Persistent Volume Storage, you can refer to Persistent Volume for 5G ATS.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.5.2 Configuring SCP-ATS and SCP

This section provides information about updating ATS deployment configuration, enabling and disabling stubs, configuring SCP to run test cases, and so on.

> **Note**
>
> Make sure to follow the steps mentioned in the Preinstall Preparation of SCP for SCP-ATS to deploy SCP. For more information on SCP deployment, refer to the *Oracle Communications Cloud Native Core, Service Communication Proxy Installation, Upgrade, and Fault Recovery Guide*.

3.5.2.1 Updating ATS Configuration

The following section covers updating the ATS deployment configuration and ATS input parameter configuration.

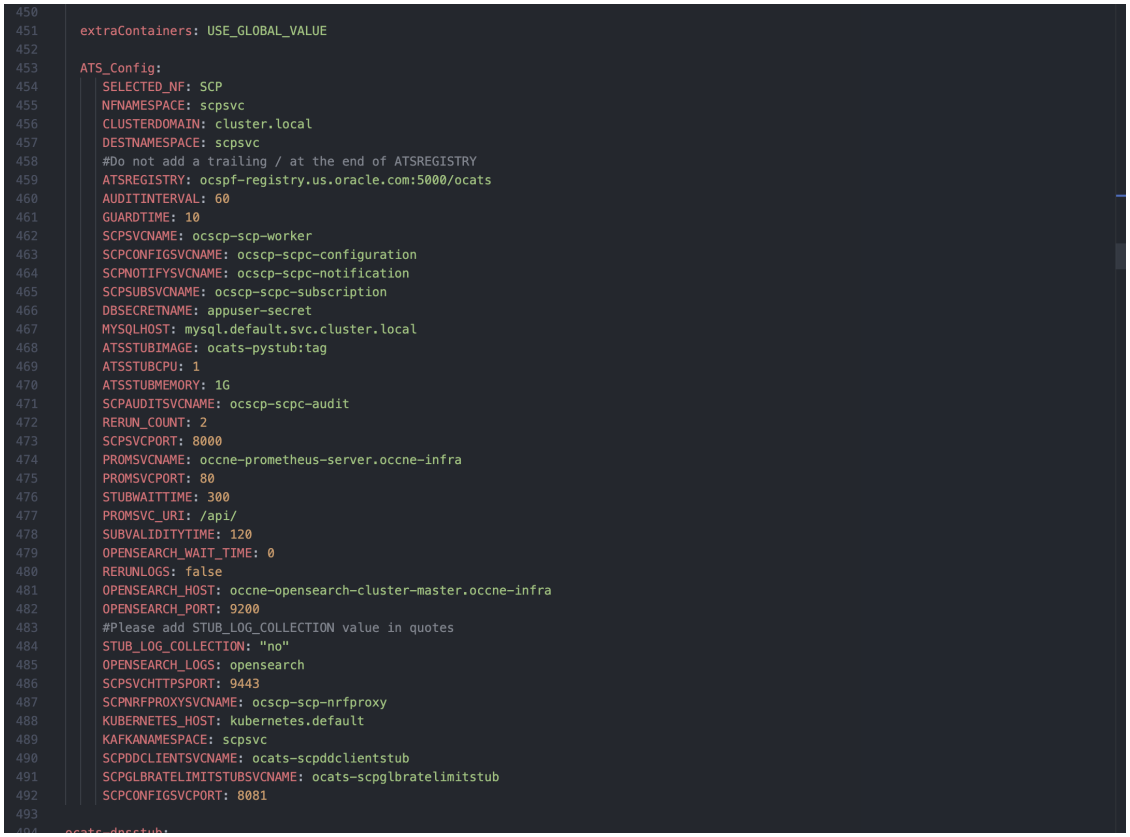
3.5.2.1.1 Updating ATS Deployment Configuration

Previously, manual modification of the ATS configuration parameters for New feature and Regression jobs was required in the ATS graphical user interface (GUI). Now, with the introduction of a section "ATS_Config" in the `ocats_ocscp_values_25.1.202.yaml` file, you can update the values for ATS parameters and then deploy the OCATS_OCSCP charts using this modified `ocats_ocscp_values_25.1.202.yaml` file. The updated ATS parameters are automatically applied during the deployment process, and the ATS will come up with the configuration as mentioned in the `ocats_ocscp_values_25.1.202.yaml` file.

The `ocats_ocscp_values_25.1.202.yaml` file can be modified to update the ATS parameters according to their environment. To illustrate this, here's an example of how you can update the ATS parameters in the `ocats_ocscp_values_25.1.202.yaml` file:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 3-14 ATS Deployment Configuration



```
extraContainers: USE_GLOBAL_VALUE

ATS_Config:
   SELECTED_NF: SCP
   NFNAMESPACE: scpsvc
   CLUSTERDOMAIN: cluster.local
   DESTNAMESPACE: scpsvc
   #Do not add a trailing / at the end of ATSREGISTRY
   ATSREGISTRY: ocspf-registry.us.oracle.com:5000/ocats
   AUDITINTERVAL: 60
   GUARDTIME: 10
   SCPSVCNAME: ocscp-scp-worker
   SCPCONFIGSVCNAME: ocscp-scpc-configuration
   SCPNOTIFYSVCNAME: ocscp-scpc-notification
   SCPSUBSVCNAME: ocscp-scpc-subscription
   DBSECRETNAME: appuser-secret
   MYSQLHOST: mysql.default.svc.cluster.local
   ATSSTUBIMAGE: ocats-pystub:tag
   ATSSTUBCPU: 1
   ATSSTUBMEMORY: 1G
   SCPAUDITSVCNAME: ocscp-scpc-audit
   RERUN_COUNT: 2
   SCPSVCPORT: 8000
   PROMSVCNAME: occne-prometheus-server.occne-infra
   PROMSVCPORT: 80
   STUBWAITTIME: 300
   PROMSVC_URI: /api/
   SUBVALIDITYTIME: 120
   OPENSEARCH_WAIT_TIME: 0
   RERUNLOGS: false
   OPENSEARCH_HOST: occne-opensearch-cluster-master.occne-infra
   OPENSEARCH_PORT: 9200
   #Please add STUB_LOG_COLLECTION value in quotes
   STUB_LOG_COLLECTION: "no"
   OPENSEARCH_LOGS: opensearch
   SCPSVCHTTPSPORT: 9443
   SCPNRFPROXYSVCNAME: ocscp-scp-nrfproxy
   KUBERNETES_HOST: kubernetes.default
   KAFKANAMESPACE: scpsvc
   SCPDDCLIENTSVCNAME: ocats-scpddclientstub
   SCPGLBRATELIMITSTUBSVCNAME: ocats-scpglbratelimitstub
   SCPCONFIGSVCPORT: 8081

ocats-dnssstub:
```

> **Note**
>
> Initially, at the time of deployment, you can configure or modify the parameter in the `ocats_ocscp_values_25.1.202.yaml`file or you can also update or modify the parameters post deployment by following the process described in the Configuring New Feature Pipelines section.

3.5.2.1.2 Configuring SCP-ATS for OCI Setup

To leverage the existing infrastructure of Oracle Cloud, SCP which was only deployed on CNE, can now be integrated on the Oracle Cloud Infrastructure (OCI).

Creating Secret for Alarm(Alerts)

When deploying ATS within the OCI environment, users are required to provide the following inputs in the form of a Kubernetes secret named ocats-oci-secret:

```
user="<your user ocid>"
tenancy="<your tenancy ocid>"
region="<your oci region>"
fingerprint="<fingerprint of your public key>"
key_file="<full path to your private key>"
metric_namespace="<metric_namespace under which all the metrics and alarms of
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
SUT NF will be captured>"
nf_compartment_id="<Compartment Id of SUT NF>"
```

Run the command to create the secret for alarm or alerts related to test cases:

```
kubectl
          create secret generic ocats-oci-secret --from-
literal=user='<your_user_ocid>'--from-literal=tenancy='<your_tenancy_ocid>'--
from-literal=region='<your_oci_region>'--from-
literal=fingerprint='<fingerprint_of_your_oci_api_public_key>'--from-
literal=metric_namespace='<metric_namespace under which all the metrics and
alarms of SUT NF will be
          captured>'--from-literal=nf_compartment_id='Compartment Id of SUT
NF'--from-file=key_file='<full_path_to_your_oci_api_private_key
          on_the_host_machine_where_you_are_running_this_command>'-n
<namespace>
```

For example,

```
kubectl
          create secret generic ocats-oci-secret --from-
literal=user='ocid1.user.oc1..aaaaaaaajjxlzewn3e76aufhdjfhdkfjkl6ea3aaazgzx7cx
g74ljs5an3a'--from-
literal=tenancy='ocid1.tenancy.oc1..aaaaaaaa5oqwziy4bngiebry6letze4hdjskjksdkd
lksurhc6pojwe4wxe34a'--from-literal=region='us-ashburn-1'--from-
literal=fingerprint='79:17:f2:89:76:d6:82:b2:13:b9:1d:9f:ff:92:28:3b'--from-
literal=metric_namespace=scpdemons
          --from-
literal=nf_compartment_id='ocid1.compartment.oc1..aaaaaaaa6crdjhjkkdjkldlxbi7e
rwtmo3wa7jy6q6ldjjskkdnnmitot4smcczgq'--from-file=key_file='/tmp/
oci_api_key.pem'-n scpsvc
```

ATS Deployment Configuration in OCI Setup

Update the values for ATS parameters and then deploy the OCATS_OCSCP charts using this modified `ocats_ocscp_values_25.1.202.yaml` file. To update the ATS parameters in the `ocats_ocscp_values_25.1.202.yaml` file, see the Updating ATS Deployment Configuration section.

Modify the `scpMetricVersion` parameter to "v2" in the `ocats_ocscp_values_25.1.202.yaml` file. For more information, see *Oracle Communications Cloud Native Core, Service Communication Proxy Installation, Upgrade, and Fault Recovery Guide*.

3.5.2.1.3 Updating ATS Input Parameter Configuration

This section provides information about how to modify different services in SCP and configure SCP test cases in ATS.

3.5.2.1.3.1 Enabling or Disabling Stubs

By default, all the stubs are enabled.

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> Deploy NRF stubs with port 8080. NRF details of SCP should specify the **ipEndPoints** port as 8080 without any **ipv4Address** field. For example, ipEndPoints: [{"port": "8080"}]).

To enable or disable the stubs or pods, set the variable to `true` or `false`, respectively. You can install the required stubs or pods during `ocats-ocscp` deployments.

The following sample parameters show how the stub is enabled by setting different variables to `true` in the `ocats_ocscp_values_25.1.202.yaml` file:

```
global:
  # ********  Sub-Section Start: Custom Extension Global Parameters ********
  #************************************************************************
  ocatsdnsstubService: true
  ocatsscpService: true
  ocatsscpglbratelimitstubService: true
  ocatsddclientstubService: true
  ausf1Stubs: true
  ausf2Stubs: true
  ausf3Stubs: true
  ausf4Stubs: true
  ausf5Stubs: true
  ausf6Stubs: true
  ausf7Stubs: true
  ausf11Stubs: true
  ausf12Stubs: true
  ausf13Stubs: true
  ausf14Stubs: true
  ausf15Stubs: true
  ausf16Stubs: true
  ausf21Stubs: true
  ausf22Stubs: true
  ausf23Stubs: true
  chf1Stubs: true
  chf2Stubs: true
  nrf1Stubs: true
  nrf2Stubs: true
  nrfr2l1Stubs: true
  nrfr2l2Stubs: true
  nrfr3l1Stubs: true
  nrfr3l2Stubs: true
  pcf1Stubs: true
  pcf1cStubs: true
  pcf2Stubs: true
  pcf3Stubs: true
  pcf4Stubs: true
  pcf5Stubs: true
  pcf6Stubs: true
  pcf7Stubs: true
  pcf8Stubs: true
  pcf10Stubs: true
  pcf11Stubs: true
  pcf12Stubs: true
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
pcf13Stubs: true
pcf14Stubs: true
pcf15Stubs: true
pcf16Stubs: true
pcf21Stubs: true
pcf22Stubs: true
pcf23Stubs: true
pcf24Stubs: true
pcf25Stubs: true
pcf26Stubs: true
pcf27Stubs: true
pcf28Stubs: true
scp1Stubs: true
scp2Stubs: true
scp3Stubs: true
scp11Stubs : true
scp12Stubs: true
scp51Stubs: true
scp52Stubs: true
scp61Stubs: true
smf1Stubs: true
smf2Stubs: true
smf3Stubs: true
smf4Stubs: true
smf5Stubs: true
smf11Stubs: true
udm1Stubs: true
udm2Stubs: true
udm3Stubs: true
udm4Stubs: true
udm5Stubs: true
udm22Stubs: true
udm23Stubs: true
udm33Stubs: true
udm21Stubs: true
udm31Stubs: true
udm32Stubs: true
udr1Stubs: true
udr2Stubs: true
scp51svcxxxxStubs: true
scp52svcxxxxStubs: true
scp61svcxxxxStubs: true
sepp1Stubs: true
sepp2Stubs: true
sepp3Stubs: true
nef1Stubs: true
nef2Stubs: true
nef3Stubs: true
nef4Stubs: true
nef5Stubs: true
nef6Stubs: true
nef7Stubs: true
nef8Stubs: true
gen1Stubs: true
gen2Stubs: true
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> Replica count of 'scp51svcxxxx' 'scp52svcxxxx' 'scp61svcxxxx' stubs must be set to Zero.

3.5.2.1.3.1.1 Modifying IpFamilyPolicy or IpFamilies of Stubs

The deployment of all stubs can adhere to the following:

- If IpFamilyPolicy is set to "SingleStack," then the value of IpFamilies can either be [IPv4] or [IPv6] only.
- If IpFamilyPolicy is set as "PreferDualStack" or "RequireDualStack", then the values of IpFamilies can either be [IPv4,IPv6] or [IPv6,IPv4] only.

> **Note**
>
> All the pyStubs should be deployed with the same combination of IpFamilyPolicy and IpFamilies.

3.5.2.1.3.2 Configuring ATS YAML File for Deployment

Umbrella Chart

Helm charts contain different charts that are referred to as subcharts through their dependencies section in the `requirements.yaml` file. When a chart is created for the purpose of grouping related subcharts or services, such as composing a whole application or deployment, it is known as umbrella chart.

Perform the following procedure to create umbrella charts and add stubs to the umbrella charts:

1. To add stubs to the umbrella chart, do the following:
   a. Below parameters can be updated to `ocats_ocscp_values_25.1.202.yaml` file: **ocats-scp**

```
ocats-scp:
  image:
    repository: cgbu-ocscp-dev-docker.dockerhub-phx.oci.oraclecorp.com/
ocats/ocats-scp
    tag: 25.1.202
    pullPolicy: Always
  replicaCount: 1
  resources:
    limits:
      cpu: 3
      memory:23Gi
      #ephemeral-storage: 4Gi
    requests:
      cpu: 3
      memory: 3Gi
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

ocats-dnsstub

```
ocats-dnsstub:
  image:
    repository: cgbu-ocscp-dev-docker.dockerhub-phx.oci.oraclecorp.com/
ocats/ocats-dnsstub
    tag: helm-tag
    pullPolicy: Always
  replicaCount: 1
  service:
    customExtension:
      labels: {}
      annotations: {}
    type: ClusterIP
    port: 53

  deployment:
    customExtension:
      labels: {}
      annotations: {}

  resources:
    limits:
      cpu: 1
      memory: 1Gi
    requests:
      cpu: 1
      memory: 1Gi
```

ocats-pystub

```
ocats-pystub:
  image:
    repository: cgbu-ocscp-dev-docker.dockerhub-phx.oci.oraclecorp.com/
ocats/ocats-pystub
    tag: helm-tag
    pullPolicy: Always
  replicaCount: 1
  resources:
    limits:
      cpu: 1
      memory: 1Gi
    requests:
      cpu: 1
      memory: 1Gi
  ausf1:
    service:
      name: ausf1svc
      type: ClusterIP
      ports:
        port: 8080
    deploymentName: ausf1
  ausf2:
    service:
      name: ausf2svc
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
      type: ClusterIP
      ports:
        port: 8080
    deploymentName: ausf2
```

ocats-scpglbratelimitstub

```
ocats-scpglbratelimitstub:
  image:
    repository: <docker-registry IP:docker-registry port>/ocats/ocats-
scpglbratelimitstub
    tag: helm-tag
    pullPolicy: Always
  replicaCount: 1
  service:
    type: ClusterIP
  deployment:
    customExtension:
      labels: {}
      annotations: {}
  resources:
   limits:
    cpu: 1
    memory: 1Gi
   requests:
    cpu: 1
    memory: 1Gi
  minreplicas: 1
  maxreplicas: 1
  maxPdbUnavailable: 11:27
ocats-ddclientstub:
  image:
    repository: <docker-registry IP:docker-registry port>/ocats/ocats-
ddclientstub
    tag: helm-tag
    pullPolicy: Always
  replicaCount: 1
  service:
    type: LoadBalancer
  resources:
    limits:
      cpu: 1
      memory: 1Gi
      #ephemeral-storage: 55Mi
    requests:
      cpu: 1
      memory: 1Gi
      #ephemeral-storage: 55Mi
  log:
    level: INFO
  extraContainers: USE_GLOBAL_VALUE

  kafka_broker: "kafka-broker1-0.kafka-
broker1.ddkafkanamespace.svc.cluster.local:9092"
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  string_topic_name: "string_topic"
  json_topic_name: "json_topic"
```

ocats-dnsstub

```
ocats-dnsstub:
  image:
    repository: <docker-registry IP:docker-registry port>/ocats/ocats-
dnsstub
    tag: helm-tag
    pullPolicy: Always
  replicaCount: 1
  service:
    customExtension:
      labels: {}
      annotations: {}
    type: ClusterIP
    port: 53

  deployment:
    customExtension:
      labels: {}
      annotations: {}

  resources:
    limits:
      cpu: 1
      memory: 1Gi
    requests:
      cpu: 1
      memory: 1Gi
  extraContainers: USE_GLOBAL_VALUE
```

3.5.2.1.3.3 Dual Stack Support

Using the dual stack mechanism, applications or NFs can establish connections with pods and services in a Kubernetes cluster using IPv4 or IPv6 or both simultaneously.

With the introduction of the Dual Stack IPv6 support feature, there will be two categories of features performing the same tests. However, they are categorized differently based on the type of endpoint the stack supports (single endpoint or multiple endpoints). It is important to consider the IP endpoint support of the stack while running these tests.

For example:

- **SCP_22.3.0_BugFixes_MultipleIpEndpoint_P0.feature**: Shall run on stacks supporting multiple endpoints (IPv4 and IPv6)
- **SCP_22.3.0_BugFixes_SingleIpEndpoint_P0.feature**: Shall run on stacks supporting single endpoints (IPv4 or IPv6)

Feature files with "SingleIpEndpoint" in the name:

These test cases run on a single IP stack setup or a dual IP stack setup, where ipFamilies should be [IPv4] or [IPv6] in the ATS deployment file.

Feature files with "MultipleIpEndpoint" in name:

Copyright © 2019, 2025, Oracle and/or its affiliates.

These test cases run only on a dual IP stack setup, where ipFamilies should be [IPv4, IPv6] or [IPv6, IPv4] in the ATS deployment file.

All other feature files can run on both setups.

3.5.2.1.3.4 Enabling HTTPs for ATS, pystubs and Jenkins

Perform the following procedure to enable HTTPs for ATS. These steps can be avoided if your environment does not require HTTPS by adhering to the Preinstall Preparation of SCP for SCP-ATS.

Ensure that you have the below file generated before you proceed with deployment:

- Private key
- Client certificate
- Certificate Authority Root certificate

1. To enable HTTPS for ATS, run the following command to create the Kubernetes secret:

```
kubectl create secret generic ocats-scp-secret
      --from-file=rsa_private_key_pkcs1_client.pem --from-file=client.pem
--from-file=caroot.pem
      --from-file=jenkinsserver.jks -n scpsvc
```

> **Note**
>
> Name of the secret, private key, client certificate, and CA root certificate must be the same as mentioned in the above mentioned command.

   a. Set the atsGuiTLSEnaled and atsCommunicationTLSEnabled parameters to true in ocats_ocscp_values.yaml, as shown below:

      Generate Certificate

```
cat >>$SERVER_EXT<<EOF
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment,
dataEncipherment
extendedKeyUsage = serverAuth, clientAuth
subjectAltName = @alt_names

[alt_names]
IP.1 = 127.0.0.1
# replace IP.2 with local setup IP when enabling https Gui for jenkins
IP.2 = 10.75.226.134
DNS.1 = *.${NAMESPACE}.${COMMON_NAME}
DNS.2 = localhost
EOF
```

   b. Make changes to the SERVER_EXT. Set the IP.2 value as the local setup IP as mentioned below:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Updated Value

```
atsGuiTLSEnabled: true
atsCommunicationTLSEnabled: true
```

c. Deploy ATS pod with above mentioned changes.

2. To enable HTTPS for pystubs, run the following command to create the Kubernetes secret:

```
kubectl create secret generic  ocats-pystub-secret
      --from-file=rsa_private_key_pkcs1_server.pem  --from-file=server.pem
--from-file=caroot.pem -n
      scpsvc
```

> **Note**
>
> Name of the secret, private key, client certificate, and CA root certificate must be the same as mentioned in the above mentioned command.

a. Run the following command to create role and role binding for GET access to the pystub pod:

```
kubectl create role scpsvc-pystub-scp-role --verb=get --
resource=secrets -n scpsvc
kubectl create rolebinding --role=scpsvc-pystub-scp-role scpsvc-pystub-
scp-rolebinding --serviceaccount=scpsvc:default -n scpsvc
```

b. Deploy pystub pods with above mentioned changes.

In the `ocats_ocscp_values_25.1.202.yaml`file, replace the value of `atsCommunicationTLSEnabled` with `true`.

3.5.2.1.3.5 Configuring SCP to Run OAuth Test Cases in ATS

By default, the ATS suite runs oauth test cases if the "ALL" option is selected, and SCP must be deployed with oauth support enabled to support the same.

```
# Enable nrfproxy-oauth service (only for Rel16)
nrfProxyOauthService: true
```

If SCP is deployed without oauth support, either the single or multiple feature execution option must be selected, excluding all oauth test cases, or these options must be manually removed from the feature directory on the ATS pod.

3.5.2.1.3.6 Configuring ATS for CCA Test Cases

Run CCA Validation feature tests with ATS.

- Enable HTTPs to run these feature tests.
- Generate 9 client certificates (using the gen_certificates.sh script) before running the feature tests and write these certificates inside the ocats-scp-secret kubernetes secret. This script is to be executed when SCP-ATS deployments have IP families such as [IPv4] or [IPv4, IPv6].
  The following lists the sample SANs that must be used while creating certificates:

Copyright © 2019, 2025, Oracle and/or its affiliates.

- **client2.pem**

```
[alt_names]

IP = 10.75.213.1

DNS.1 = chf1svc.${NAMESPACE}.${COMMON_NAME}

URI.1 = urn:uuid:11111111-aaaa-aaaa-aaaa-111111111177

URI.2 = https://chf1svc.${NAMESPACE}.${COMMON_NAME}:443

URI.3 = https://10.75.213.1:443
```

- **client3.pem**

```
[alt_names]
URI.1 = urn:uuid:11111111-aaaa-aaaa-aaaa-111111111177
```

- **client4.pem**

```
[
alt_names]
DNS.1 = chf1svc.${NAMESPACE}.${COMMON_NAME}
```

- **client5.pem**

```
[alt_names]
IP = 10.75.213.1
```

- **client6.pem**

```
[alt_names]
URI.1 = https://chf1svc.${NAMESPACE}.${COMMON_NAME}:443
```

- **client7.pem**

```
[alt_names]
URI.1 = https://10.75.213.1:443
```

- **client8.pem**

```
[alt_names]
IP = 10.75.213.1
DNS.1 = chf1svc.${NAMESPACE}.${COMMON_NAME}
URI.1 = https://chf1svc.${NAMESPACE}.${COMMON_NAME}:443
URI.2 = https://10.75.213.1:443
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

- **client9.pem**

```
[alt_names]
IP.1 = 10.75.213.2
IP.2 = 10.75.213.3
IP.3 = 10.75.213.4
IP.4 = 10.75.213.5
IP.5 = 10.75.213.6
IP.6 = 10.75.213.7
IP.7 = 10.75.213.8
IP.8 = 10.75.213.9
IP.9 = 10.75.213.10
IP.10 = 10.75.213.11
IP.11 = 10.75.213.1
DNS.1 = chf1svc.${NAMESPACE}.${COMMON_NAME}
URI.1 = urn:uuid:11111111-aaaa-aaaa-aaaa-111111111177
URI.2 = https://chf1svc.${NAMESPACE}.${COMMON_NAME}:443
URI.3 = https://10.75.213.10:443
```

- **client10.pem**

```
[alt_names]
IP = 10.75.213.1
DNS.1 = chf1svc.${NAMESPACE}.${COMMON_NAME}
URI.1 = urn:uuid:11111111-aaaa-aaaa-aaaa-111111111177
URI.2 = https://chf1svc.${NAMESPACE}.${COMMON_NAME}:443
URI.3 = https://10.75.213.10:443
URI.4 = https://10.75.213.2:443
URI.5 = https://10.75.213.3:443
URI.6 = https://10.75.213.4:443
URI.7 = https://10.75.213.5:443
URI.8 = https://10.75.213.6:443
URI.9 = https://10.75.213.7:443
URI.10 = https://10.75.213.8:443
URI.11 = https://10.75.213.1:443
```

- Generate 9 client certificates before running the feature tests, and write these certificates inside the ocats-scp-secret Kubernetes secret. Run the gen_certificates_ipv6.sh script when SCP-ATS deployment is either [IPv6] or [IPv6,IPv4].
  The following lists the sample SANs that must be used while creating certificates:
  - **client2.pem**

```
[alt_names]
IP = 2001:db8:85a3:0:0:8a2e:370:7334

DNS.1 = chf1svc.${NAMESPACE}.${COMMON_NAME}

URI.1 = urn:uuid:11111111-aaaa-aaaa-aaaa-111111111177

URI.2 = https://chf1svc.${NAMESPACE}.${COMMON_NAME}:443

URI.3 = https://[2001:db8:85a3:0:0:8a2e:370:7334]:443
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

- **client3.pem**

```
[alt_names]
URI.1 = urn:uuid:11111111-aaaa-aaaa-aaaa-111111111177
```

- **client4.pem**

```
[
alt_names]
DNS.1 = chf1svc.${NAMESPACE}.${COMMON_NAME}
```

- **client5.pem**

```
[alt_names]
IP = 2001:db8:85a3:0:0:8a2e:370:7334
```

- **client6.pem**

```
[alt_names]
URI.1 = https://chf1svc.${NAMESPACE}.${COMMON_NAME}:443
```

- **client7.pem**

```
[alt_names]
URI.1 =https://[2001:db8:85a3:0:0:8a2e:370:7334]:443
```

- **client8.pem**

```
[alt_names]
IP = 2001:db8:85a3:0:0:8a2e:370:7334
DNS.1 = chf1svc.${NAMESPACE}.${COMMON_NAME}
URI.1 = https://chf1svc.${NAMESPACE}.${COMMON_NAME}:443
URI.2 = https://[2001:db8:85a3:0:0:8a2e:370:7334]:443
```

- **client9.pem**

```
[alt_names]
IP.1 = 2002:db8:85a3:0:0:8a2e:370:7334
IP.2 = 2003:db8:85a3:0:0:8a2e:370:7334
IP.3 = 2004:db8:85a3:0:0:8a2e:370:7334
IP.4 = 2005:db8:85a3:0:0:8a2e:370:7334
IP.5 = 2006:db8:85a3:0:0:8a2e:370:7334
IP.6 = 2007:db8:85a3:0:0:8a2e:370:7334
IP.7 = 2008:db8:85a3:0:0:8a2e:370:7334
IP.8 = 2009:db8:85a3:0:0:8a2e:370:7334
IP.9 = 2010:db8:85a3:0:0:8a2e:370:7334
IP.10 = 2011:db8:85a3:0:0:8a2e:370:7334
IP.11 = 2001:db8:85a3:0:0:8a2e:370:7334
DNS.1 = chf1svc.${NAMESPACE}.${COMMON_NAME}
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
URI.1 = urn:uuid:11111111-aaaa-aaaa-aaaa-111111111177
URI.2 = https://chf1svc.${NAMESPACE}.${COMMON_NAME}:443
URI.3 = https://[2010:db8:85a3:0:0:8a2e:370:7334]:443
```

- **client10.pem**

```
[alt_names]
IP = 2001:db8:85a3:0:0:8a2e:370:7334
DNS.1 = chf1svc.${NAMESPACE}.${COMMON_NAME}
URI.1 = urn:uuid:11111111-aaaa-aaaa-aaaa-111111111177
URI.2 = https://chf1svc.${NAMESPACE}.${COMMON_NAME}:443
URI.3 = https://[2010:db8:85a3:0:0:8a2e:370:7334]:443
URI.4 = https://[2002:db8:85a3:0:0:8a2e:370:7334]:443
URI.5 = https://[2003:db8:85a3:0:0:8a2e:370:7334]:443
URI.6 = https://[2004:db8:85a3:0:0:8a2e:370:7334]:443
URI.7 = https://[2005:db8:85a3:0:0:8a2e:370:7334]:443
URI.8 = https://[2006:db8:85a3:0:0:8a2e:370:7334]:443
URI.9 = https://[2007:db8:85a3:0:0:8a2e:370:7334]:443
URI.10 = https://[2008:db8:85a3:0:0:8a2e:370:7334]:443
URI.11 = https://[2001:db8:85a3:0:0:8a2e:370:7334]:443
```

3.5.2.1.3.7 Updating the Global Egress Rate Limiting Changes in the SCP Deployment File for ATS

Perform the following procedure to enable the Global Egress Rate Limiting feature for ATS environment.

1. In the SCP `custom-values.yaml` file, update the following:
   - `federation.remoteScpOne.fqdnOrIp`: FQDN of the `scpglbratelimitstub` pod.
   - `federation.remoteScpOne.clusterName`: Coherence Cluster Name of global rate limit Stub [Example: `scpstub-coherence-cluster`].
   - `federation.remoteScpOne.nfInstanceId`: NFInstanceID of global rate limit Stub [Example: `2faf1bbc-6e4a-4454-a507-a14ef8e1bc22`].

3.5.2.1.3.8 Configuring SCP to Run DNS SRV Test Cases in ATS

By default, the ATS suite runs alternate resolution (DNS SRV) if the "ALL" option is selected, and SCP must be deployed with alternate resolution service support enabled in order to support the same.

```
# Enable DNS SRV Alternate Routing Feature
  dnsSRVAlternateRouting: false
```

If SCP is deployed without alternate resolution support, either single or multiple feature execution options must be selected, excluding all DNS SRV test cases, or these options must be manually removed from the feature directory on the ATS pod.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.5.2.1.3.9 Configuring SCP to Run Mediation Test Cases in ATS

By default, the ATS suite runs Mediation test cases if the "ALL" option is selected, and SCP must be deployed with Mediation support enabled to support the same.

```
# Enable mediation service
  mediationService: false
```

If SCP is deployed without the Mediation support, then either a single or multiple feature execution option can be selected, excluding all Mediation test cases, or these options must be manually removed from the feature directory on the ATS pod.

3.5.2.1.3.10 Configuring SCP to Run Model D Test Cases in ATS

By default, the ATS suite runs delegated discovery (Model D) if the "ALL" option is selected, and SCP must be deployed with nrfproxy support enabled to support the same.

```
# Enable Nrf Proxy service (only for Rel16)
  nrfProxyService: false
```

If SCP is deployed without nrfproxy support, either single or multiple feature execution options must be selected, excluding all Model D test cases, or these options must be manually removed from the feature directory on the ATS pod.

3.5.2.1.3.11 Configuring ATS for Traffic Feed Test Cases

1. To run Traffic Feed ATS testcases, install the kafka broker.
2. In the ATS deployment file, update the following parameters:
   - kafka_broker: <kafka broker host>:<kafka broker host> (Example, “kafka-broker-0.kafka-broker.scpsvc.svc.cluster.local:9092”)
   - string_topic_name: <topic name for string serialization> (Example, “string_topic”)
   - json_topic_name: <topic name for json serialization> (Example, “json_topic”)
3. In the `global.yaml` file, update the following parameters:
   - global_traffic_feed_key_Serializer: <key serialization> (Example, string)
   - global_traffic_feed_value_Serializer: <value serialization> (Example, string)
   - global_traffic_feed_topic_name: <topic name for selected serialization> (Example, string_topic)
   - global_traffic_feed_bootstrap_server_host: <kafka broker host> (Example, kafka-broker1-0.kafka-broker1.scpsvc.svc.cluster.local)
   - global_traffic_feed_bootstrap_server_port: <kafka broker port> (Example, 9092)
     For more information on `global.yaml` file, see ATS Testcase Parametrization on User Input.

   For installation of the Data Director Kafka broker, perform the following:
   - Update the `ocnadd-custom-values.yaml` as documented in the Data Director Installation and Upgrade Guide.
   - Disable all services apart from `ocnaddkafka` by marking them as false.
   - Keep `ocnaddkafka` as true.

Copyright © 2019, 2025, Oracle and/or its affiliates.

For more information, see *Oracle Communications Network Analytics Data Director Installation and Upgrade Guide*.

> **Note**
>
> Kafka broker should be deployed with 3 partitions under string_topic.

3.5.2.1.3.12 Configuring SCP to Run LCI Test Cases in ATS

By default, the ATS suite runs load manager test cases if the "ALL" option is selected, and SCP must be deployed with load manager support enabled to support the same.

```
# Enable load-manager service (only for Rel16)
 loadManagerService: true
```

If SCP is deployed without load manager support, either the single or multiple feature execution option must be selected, excluding all load manager test cases, or these options must be manually removed from the feature directory on the ATS pod.

3.5.3 Deploying ATS and Stub in the Kubernetes Cluster

This section provides information about how to deploy ATS and stubs.

To deploy ATS in the Kubernetes Cluster:

> **Note**
>
> Deploy ATS, SCP, and stubs in the same namespace.

1. Ensure the `ocats_ocscp_values_25.1.202.yaml` file has been updated with correct repository, image tag, and parameters as per requirements.
2. In the Files or Helm folder, find the ocats-ocscp charts for version 23.3.0, which have to be used for installation.
3. Run the following command to deploy ATS:

   ```
   helm install ocats-ocscp-25.1.202.tgz --name <release_name> --namespace
   <namespace_name> -f ocats-ocscp-values-25.1.202.yaml
   ```

   Example:

   ```
   helm install ocats-ocscp-25.1.202.tgz --name ocats-ocscp --namespace scpsvc-f
   ocats-ocscp-values-25.1.202.yaml
   ```

   > **Note**
   >
   > Update image name, tag, service name, and deployment name in ocats-pystub of the `ocats_pystub_values_25.1.202.yaml` file before deploying.

4. Verify whether all stub pods are up and running in the deployed namespace as updated in the `ocats_ocscp_values_25.1.202.yaml`.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.5.4 Post Installation and Deployment Steps

This section describes the post-installation steps for SCP.

3.5.4.1 Verifying ATS Deployment

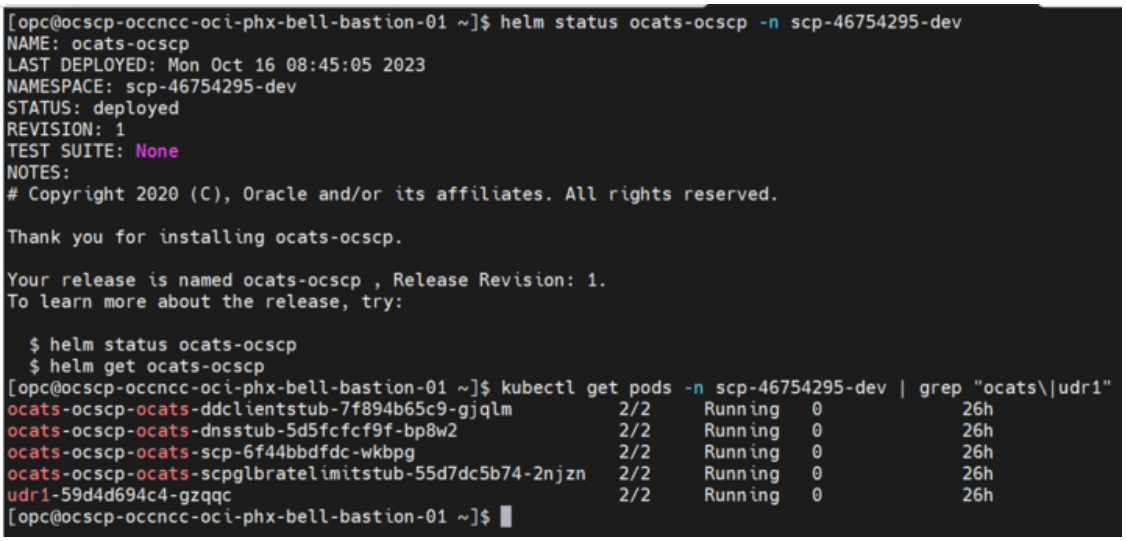
Run the following command to verify the ATS deployment status:

```
helm status <release_name>
```

> **Note**
>
> If ATS is deployed in the service mesh environment, the **Ready** field for pods displays 2/2.

The following image displays that the deployment is complete because the **STATUS** field has changed to `deployed`.

Figure 3-15 Checking ATS Helm Release and ATS related Pod Status



3.5.4.2 Modifying the scpc-alternate-resolution Microservice

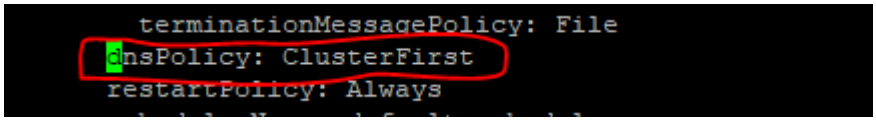
Perform the following procedure to modify the scpc-alternate-resolution microservice to point to the DNS Stub for the Alternate Routing based on the Domain Name System (DNS) Service Record (SRV) Records feature.

1. Capture the cluster IP of DNS Stub service.

   By default, scpc-alternate-resolution points to CoreDNS and displays the following settings in the `ocscp_values.yaml` deployment file.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 3-16 CoreDNS



2. Run the following command to change the deployment file to add content in scpc-alternate-resolution to query the DNS Stub.

   Uncomment ocscp-alternate-resolution's dnsConfig and dnsPolicy before deployment to allow editing.

   ```
   $kubectl edit deployment ocscp-scpc-alternate-resolution -n
       scpsvc
   ```

   Sample deployment file:

   ```
   dnsConfig:
           nameservers:
           - 10.96.77.54
           searches:
           - cicdscpsvc-230228133808.svc.cluster.local
           - svc.cluster.local
           - cluster.local
    dnsPolicy: None
   ```

   Add the following content:

   - nameservers: Add the IP address that you recorded after installing the DNS Stub (cluster IP of the DNS Stub).
   - searches: Add all the search based on the cluster name.
     - scpsvc
       This is the namespace.
     - cluster.local
       This is the cluster name.
   - dnsPolicy: Set it to "None" if it is not already set by default.

3.5.4.3 ATS Testcase Parametrization on User Input

Parameterization is an approach to decoupling the data from the feature files in ATS so that ATS test cases can be run against some predefined configuration that contains customer-specific data and service configurations. Parametrization allows you to provide or adjust values for the input and output parameters needed for the test cases to be compatible with the SUT configuration. You can update or adjust the key-value pair values in the `global.yaml` and `feature.yaml` files for each of the feature files so that they are compatible with SUT configuration. For more information, see the Parameterization section.

Three new folders "cust_data", "product_config" and "custom_config" are added to the base path inside ATS pod `/var/lib/jenkins/ocscp_tests`. Where the cust_data folder is a replica of the already existing data folder, the product_config folder contains configuration files that are compatible with the default product configurations, and the cust_config folder is a replica of the product_config folder. You can update the custom folders, such as cust_data and custom_config.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Product Config folder
The product config folder contains two types of YAML files: `global.yaml` and `feature_name>. yaml` (feature file-specific yaml).

Global File

The `global.yaml` contains global variable names and their corresponding default values that can be used across all the feature files. The variable name declared in `global.yaml` should start with `global_Var_` as prefix.

For example, <key_name>: &<variable_name> <default_value_of_variable>

```
       #START_GLOBAL
                 global:
                   File_Parameters:
                     global_Var_SUT_apiPrefix1: &global_Var_SUT_apiPrefix1
USEast
                    global_Var_SUT_ServingLocality1:
&global_Var_SUT_ServingLocality1 USEast
                      global_Var_localNrfSetId1: &global_Var_localNrfSetId1
setnrfl1.nrfset.5gc.mnc012.mcc345
                      global_Var_stubPort: &global_Var_stubPort 8080
                  #END_GLOBAL
```

Sample:

```
#START_GLOBAL
global:
  File_Parameters:
    global_Var_SUT_apiPrefix1: &global_Var_SUT_apiPrefix1 USEast
    # Following Serving Localities if needs to be changed are also to be
changed while deploying SCP before running ATS
    global_Var_SUT_ServingLocality1: &global_Var_SUT_ServingLocality1 USEast
    global_Var_SUT_ServingLocality2: &global_Var_SUT_ServingLocality2 Loc7
    global_Var_SUT_ServingLocality3: &global_Var_SUT_ServingLocality3 Loc8
    global_Var_SUT_ServingLocality4: &global_Var_SUT_ServingLocality4 Loc9

    # Following SetId's if needs to be changed are also to be changed while
deploying SCP before running ATS
    global_Var_localNrfSetId1: &global_Var_localNrfSetId1
setnrfl1.nrfset.5gc.mnc012.mcc345
    global_Var_remoteNrfSetId1: &global_Var_remoteNrfSetId1
setnrfr1.nrfset.5gc.mnc012.mcc345
    global_Var_remoteNrfSetId2: &global_Var_remoteNrfSetId2
setnrfr2.nrfset.5gc.mnc012.mcc345

    # If stubPort has to be changed then stub has to be deployed with the
same port number before running ATS
    global_Var_stubPort: &global_Var_stubPort 8080
    global_Var_stubErrorCode: &global_Var_stubErrorCode 404

    global_Var_udm_nfSetIdList1: &global_Var_udm_nfSetIdList1
set1.udmset.5gc.mnc012.mcc345
    global_Var_udm_nfSetIdList2: &global_Var_udm_nfSetIdList2
set2.udmset.5gc.mnc012.mcc345
    global_Var_udm_nfSetIdList3: &global_Var_udm_nfSetIdList3
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
set3.udmset.5gc.mnc012.mcc345

    global_Var_smf_nfSetIdList1: &global_Var_smf_nfSetIdList1
set1.smfset.5gc.mnc012.mcc345
    global_Var_smf_nfSetIdList2: &global_Var_smf_nfSetIdList2
set2.smfset.5gc.mnc012.mcc345
    global_Var_smf_nfSetIdList3: &global_Var_smf_nfSetIdList3
set3.smfset.5gc.mnc012.mcc345

    global_Var_pcf_nfSetIdList1: &global_Var_pcf_nfSetIdList1
set1.pcfset.5gc.mnc012.mcc345
    global_Var_pcf_nfSetIdList2: &global_Var_pcf_nfSetIdList2
set2.pcfset.5gc.mnc012.mcc345
    global_Var_pcf_nfSetIdList3: &global_Var_pcf_nfSetIdList3
set3.pcfset.5gc.mnc012.mcc345

    global_Var_udm1_nfInstanceId: &global_Var_udm1_nfInstanceId 11111111-aaaa-
aaaa-aaaa-111111111111
    global_Var_udm2_nfInstanceId: &global_Var_udm2_nfInstanceId 21111111-aaaa-
aaaa-aaaa-111111111111
    global_Var_udm3_nfInstanceId: &global_Var_udm3_nfInstanceId 11111111-aaaa-
aaaa-aaaa-111111111122
    global_Var_smf1_nfInstanceId: &global_Var_smf1_nfInstanceId 11111111-aaaa-
aaaa-aaaa-111111111111
    global_Var_smf2_nfInstanceId: &global_Var_smf2_nfInstanceId 11111111-aaaa-
aaaa-aaaa-111111111122
    global_Var_smf3_nfInstanceId: &global_Var_smf3_nfInstanceId 11111111-aaaa-
aaaa-aaaa-111111111133
    global_Var_smf4_nfInstanceId: &global_Var_smf4_nfInstanceId 11111111-aaaa-
aaaa-aaaa-111111111144
    global_Var_smf5_nfInstanceId: &global_Var_smf5_nfInstanceId 11111111-aaaa-
aaaa-aaaa-111111111155
    global_Var_pcf1_nfInstanceId: &global_Var_pcf1_nfInstanceId
1faf1bbc-6e4a-3994-a507-a14ef8e1bc5a
    global_Var_pcf2_nfInstanceId: &global_Var_pcf2_nfInstanceId
1faf1bbc-6e4a-3994-a507-a14ef8e1bc6b
    global_Var_scp51_nfInstanceId: &global_Var_scp51_nfInstanceId
2fbf1bbc-6e4b-3994-b507-b14ef8e1bc51
    global_Var_scp61_nfInstanceId: &global_Var_scp61_nfInstanceId
2fbf1bbc-6e4b-3994-b507-b14ef8e1bc61

    # If svc name has to be changed then stub has to be deployed with the
same svc name before running ATS
    global_Var_udm1_svc_name: &global_Var_udm1_svc_name udm1svc
    global_Var_udm2_svc_name: &global_Var_udm2_svc_name udm2svc
    global_Var_udm3_svc_name: &global_Var_udm3_svc_name udm3svc
    global_Var_smf1_svc_name: &global_Var_smf1_svc_name smf1svc
    global_Var_smf2_svc_name: &global_Var_smf2_svc_name smf2svc
    global_Var_smf3_svc_name: &global_Var_smf3_svc_name smf3svc
    global_Var_smf4_svc_name: &global_Var_smf4_svc_name smf4svc
    global_Var_smf5_svc_name: &global_Var_smf5_svc_name smf5svc
    global_Var_pcf1_svc_name: &global_Var_pcf1_svc_name pcf1svc
    global_Var_pcf2_svc_name: &global_Var_pcf2_svc_name pcf2svc
    global_Var_scp51_svc_name: &global_Var_scp51_svc_name scp51svc
    global_Var_scp61_svc_name: &global_Var_scp61_svc_name scp61svc
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    global_Var_nrf1_svc_name: &global_Var_nrf1_svc_name nrf1svc
#END_GLOBAL
```

Feature File

The <Feature_name>.yaml file contains feature-specific variables that can be parameterized. For example, if the name of the feature file that needs to be parameterized is "ModelC_NF_Set.feature", then the YAML file corresponding to it will be named`ModelC_NF_Set.yaml`. This yaml file contains the `#START_GLOBAL` and `#END_GLOBAL` tags without any data in between, as the data is copied over from the `global.yaml` file to this section during test execution. The variable name in `feature.yaml` should have `feature_Var_` as prefix.

For example:- <key_name>: &<variable_name> <default_value_of_variable>

```
        #START_GLOBAL
                #END_GLOBAL

                            ModelC_NF_Set.feature:

                               File_Parameters:

                                 feature_Var_udm1_Priority:
&feature_Var_udm1_Priority 0
                                 feature_Var_smf1_Priority:
&feature_Var_smf1_Priority 0
                                 feature_Var_traffic_rate:
&feature_Var_traffic_rate 100
```

Scenario

The variables are referenced under the scenario tag to use in the feature file; for this, the scenario tag has to be concatenated with `Scenario_` and enclosed within double commas.

The variables defined under the scenario tag should have `sc_` as its prefix.

For example:- "Scenario_<Scenario_tag>":

```
         "Scenario_Scenario-1- <Scenario_tag>":
                    Input:
                      File_Parameters:
                          sc_http_requests_total_udm1: 100
```

Sample:

```
#START_GLOBAL
#END_GLOBAL

ModelC_NFSet.feature:
  File_Parameters:
    # The priorities can be changed without having the impact on the order of
the priorities
    feature_Var_udm1_Priority: &feature_Var_udm1_Priority 0
    feature_Var_udm2_Priority: &feature_Var_udm2_Priority 1
    feature_Var_udm3_Priority: &feature_Var_udm3_Priority 1
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    feature_Var_smf1_Priority: &feature_Var_smf1_Priority 0
    feature_Var_smf2_Priority: &feature_Var_smf2_Priority 0
    feature_Var_smf3_Priority: &feature_Var_smf3_Priority 1
    feature_Var_smf4_Priority: &feature_Var_smf4_Priority 3
    feature_Var_smf5_Priority: &feature_Var_smf5_Priority 4

    feature_Var_pcf1_Priority: &feature_Var_pcf1_Priority 0
    feature_Var_pcf2_Priority: &feature_Var_pcf2_Priority 0

    feature_Var_supiOfPathURI: &feature_Var_supiOfPathURI imsi-100000001

    feature_Var_traffic_rate: &feature_Var_traffic_rate 100

    # The traffic for the below metrics to be configured in the same
proportionate as the traffice sent
    feature_Var_scp_http_rx_req_total_cnt: *feature_Var_traffic_rate
    feature_Var_scp_http_tx_req_total_cnt: *feature_Var_traffic_rate
    feature_Var_scp_http_tx_req_total_cnt_alternate_route:
&feature_Var_scp_http_tx_req_total_cnt_alternate_route 200
    feature_Var_http_requests_total_udm1:
&feature_Var_http_requests_total_udm1 0
    feature_Var_http_requests_total_udm2:
&feature_Var_http_requests_total_udm2 0
    feature_Var_http_requests_total_udm3:
&feature_Var_http_requests_total_udm3 0
    feature_Var_http_requests_total_smf1:
&feature_Var_http_requests_total_smf1 0
    feature_Var_http_requests_total_smf2:
&feature_Var_http_requests_total_smf2 0
    feature_Var_http_requests_total_smf3:
&feature_Var_http_requests_total_smf3 0
    feature_Var_http_requests_total_smf4:
&feature_Var_http_requests_total_smf4 0
    feature_Var_http_requests_total_pcf1:
&feature_Var_http_requests_total_pcf1 0
    feature_Var_scp_http_rx_res_total_cnt: *feature_Var_traffic_rate
    feature_Var_scp_http_tx_res_total_cnt: *feature_Var_traffic_rate

  "Scenario_Scenario-1- Forward route initial UECM AMF Registration messages
with 3gpp-Sbi-Target-apiRoot and 3GPP-Sbi-Disocvery-target-NfSetid Header":
      Input:
        File_Parameters:
          sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-2- Alternate route initial UECM AMF Registration
messages with 3gpp-Sbi-Target-apiRoot and 3GPP-Sbi-Discovery-target-NfSetid
Header":
      Input:
        File_Parameters:
          sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent
          sc_http_requests_total_udm2: 100     #To be configured in the same
proportionate as the traffice sent
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  "Scenario_Scenario-3- Load Balance initial UECM AMF Registration messages
with 3gpp-Sbi-Target-apiRoot and 3GPP-Sbi-Disocvery-target-NfSetid Header":
      Input:
        File_Parameters:
          sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent
          sc_http_requests_total_udm2: 50      #To be configured in the same
proportionate as the traffice sent
          sc_http_requests_total_udm3: 50      #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-4- Alternate route initial UECM AMF Registration
messages with 3gpp-Sbi-Target-apiRoot missing and 3GPP-Sbi-Disocvery-target-
NfSetid Header present":
      Input:
        File_Parameters:
          sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-5- Forward route subsequent UECM AMF Registration
messages with 3gpp-Sbi-Target-apiRoot and 3gpp-sbi-routing-binding Header
with bl=nfset":
      Input:
        File_Parameters:
          sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-6- Alternate route subsequent UECM AMF Registration
messages with 3gpp-Sbi-Target-apiRoot and 3gpp-sbi-routing-binding Header
with bl=nfset":
      Input:
        File_Parameters:
          sc_udm3_Priority: 30     #Priority can be changed without having
the impact on the order of the priorities
          sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent
          sc_http_requests_total_udm2: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-7- Load Balance subsequent UECM AMF Registration
messages with 3gpp-Sbi-Target-apiRoot and 3gpp-sbi-routing-binding Header
with bl=nfset":
      Input:
        File_Parameters:
          sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent
          sc_http_requests_total_udm2: 50     #To be configured in the same
proportionate as the traffice sent
          sc_http_requests_total_udm3: 50     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-8- Alternate route subsequent UECM AMF Registration
messages with 3gpp-Sbi-Target-apiRoot missing and 3gpp-sbi-routing-binding
Header with bl=nfset is present":
      Input:
        File_Parameters:
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
          sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent


  "Scenario_Scenario-9- To test when Forward route for notification request
UECM AMF Registration messages with 3gpp-Sbi-Target-apiRoot and 3gpp-sbi-
routing-binding header bl=nfset":
      Input:
        File_Parameters:
          sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-10- To test when Forward route fails for notification
request UECM AMF Registration messages with 3gpp-Sbi-Target-apiRoot and 3gpp-
sbi-routing-binding header,alternate route should happen on NfSet":
      Input:
        File_Parameters:
          sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent
          sc_http_requests_total_udm2: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-11- To test Forward route for notification request UECM
AMF Registration messages with 3gpp-Sbi-Discovery-target-nf-set-id,3gpp-Sbi-
Target-apiRoot":
    Input:
      File_Parameters:
        sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-12- To test when Forward route fails for notification
request UECM AMF Registration messages with 3gpp-Sbi-Discovery-target-nf-set-
id and 3gpp-Sbi-Target-apiRoot,alternate route should happen on NfSet":
      Input:
        File_Parameters:
          sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent
          sc_http_requests_total_udm2: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-13- To test when Forward route fails for notification
request UECM AMF Registration messages with 3gpp-Sbi-Discovery-target-nf-set-
id and 3gpp-Sbi-Target-apiRoot,load balancing should happen on NfSet on NFs
with similar priority":
      Input:
        File_Parameters:
          sc_http_requests_total_udm1: 100    #To be configured in the same
proportionate as the traffice sent
          sc_http_requests_total_udm2: 50     #To be configured in the same
proportionate as the traffice sent
          sc_http_requests_total_udm3: 50     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-14- Forward route initial SMF PduSession sm-contexts
create messages with 3gpp-Sbi-Target-apiRoot and 3GPP-Sbi-Disocvery-target-
NfSetid Header":
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
      Input:
        File_Parameters:
          sc_http_requests_total_smf1: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-15- Alternate route SMF PduSession sm-contexts create
messages with 3gpp-Sbi-Target-apiRoot and 3GPP-Sbi-Disocvery-target-NfSetid
Header":
      Input:
        File_Parameters:
          sc_http_requests_total_smf1: 100     #To be configured in the same
proportionate as the traffice sent
          sc_http_requests_total_smf2: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-16- Error response code received when SMF PduSession sm-
contexts create message is sent with missing information in 3gpp-sbi-routing-
binding header and 3gpp-Sbi-Target-apiRoot header is missing":
      Input:
        File_Parameters:
          sc_scp_http_tx_req_total_cnt: 0     #To be configured in the same
proportionate as the traffice sent
          sc_ocscp_metric_scp_generated_response_total: 100     #To be
configured in the same proportionate as the traffice sent

  "Scenario_Scenario-17-Error response code received when SMF PduSession sm-
contexts create message is sent with missing information in 3gpp-Sbi-
Discovery-target-nf-set-id header and 3gpp-Sbi-Target-apiRoot header":
      Input:
        File_Parameters:
          sc_targetNfSetId_Send: 1
          sc_http_requests_total_smf1: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-18- Alternate route SMF PduSession sm-contexts create
messages with 3gpp-Sbi-Target-apiRoot and 3GPP-Sbi-Disocvery-target-NfSetid
Header":
      Input:
        File_Parameters:
          sc_smf1_Priority: 2     #Priority can be changed without having the
impact on the order of the priorities
          sc_scp_http_tx_req_total_cnt_alternate_route: 400     #To be
configured in the same proportionate as the traffice sent
          sc_http_requests_total_smf1: 100     #To be configured in the same
proportionate as the traffice sent
          sc_http_requests_total_smf4: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-19- No Alternate route for initial UECM AMF Registration
messages with 3gpp-Sbi-Target-apiRoot and 3GPP-Sbi-Discovery-target-NfSetid
Header as reroute Policy is disabled":
      Input:
        File_Parameters:
          sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent
          sc_scp_http_rx_res_total_cnt: 0     #To be configured in the same
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
proportionate as the traffice sent

  "Scenario_Scenario-20- Forward route PCF SMPolicyControl Create
SMPolicyAssociation with 3gpp-Sbi-Target-apiRoot and 3GPP-Sbi-Disocvery-
target-NfSetid Header and verify that 3gpp-Sbi-Target-apiRoot and 3gpp-sbi-
producer-id headers are not present in response":
      Input:
        File_Parameters:
          sc_pcf1_Service_Priority: 80        #Priority can be changed
without having the impact on the order of the priorities
          sc_http_requests_total_pcf1: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-21- Alternate route PCF SMPolicyControl Create
SMPolicyAssociation messages with 3gpp-Sbi-Target-apiRoot and 3GPP-Sbi-
Disocvery-target-NfSetid Header and verify that 3gpp-Sbi-Target-apiRoot and
3gpp-sbi-producer-id headers are present in response":
      Input:
        File_Parameters:
          sc_pcf1_Priority: 2     #Priority can be changed without having the
impact on the order of the priorities
          sc_http_requests_total_pcf2: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-22- Alternate route initial UECM AMF Registration
messages with 3gpp-Sbi-Target-apiRoot missing and 3GPP-Sbi-Disocvery-target-
NfSetid Header present and verify that only 3gpp-sbi-producer-id header is
present in response since location header is present":
      Input:
        File_Parameters:
          sc_http_requests_total_udm1: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-23- Alternate route PCF SMPolicyControl Create messages
with 3gpp-Sbi-Target-apiRoot and 3GPP-Sbi-Disocvery-target-NfSetid Header and
verify that 3gpp-Sbi-Target-apiRoot and 3gpp-sbi-producer-id headers are
present in response":
      Input:
        File_Parameters:
          sc_pcf1_Priority: 2     #Priority can be changed without having the
impact on the order of the priorities
          sc_http_requests_total_pcf2: 100     #To be configured in the same
proportionate as the traffice sent

  "Scenario_Scenario-24- Alternate route PCF SMPolicyControl Create messages
with 3gpp-Sbi-Target-apiRoot and 3gpp-sbi-routing-binding Header and verify
that 3gpp-Sbi-Target-apiRoot and 3gpp-sbi-producer-id headers are present in
response":
      Input:
        File_Parameters:
          sc_Var_pcf1_Priority: 2     #Priority can be changed without having
the impact on the order of the priorities
          sc_http_requests_total_pcf2: 100     #To be configured in the same
proportionate as the traffice sent
```

Updates in Feature Files

Copyright © 2019, 2025, Oracle and/or its affiliates.

The variables are defined in either global.yaml or <feature_name>.yaml is used in the feature files by enclosing the variable name in curly brackets. The steps in a feature file without parameterization are as follows:

Figure 3-17 Feature File without Parameterization

```
   Then Update File /var/lib/jenkins/ocscp_tests/data/udm1_nfset_reg.json For Key nfProfile.nfSetIdList.0 With Value set1.udmset.5
gc.mnc012.mcc345
   Then Update File /var/lib/jenkins/ocscp_tests/data/udm1_nfset_reg.json For Key nfProfile.nfSetIdList.1 With Value set3.udmset.5
gc.mnc012.mcc345
   Then Update Json /var/lib/jenkins/ocscp_tests/data/udm1_nfset_reg.json With Key nfProfile.locality:USEast,nfProfile.nfInstanceI
d:11111111-aaaa-aaaa-aaaa-111111111111, nfProfile.nfServices.ipEndPoints.port:null, nfProfile.nfServices.fqdn:udm1svc.default.svc.
cluster.local,nfProfile.priority:0 For Service udm1svc
   Then Soothsayer NF With Serving Scope Register ss2 setnrfl1.nrfset.5gc.mnc012.mcc345 /var/lib/jenkins/ocscp_tests/data/udm1_nfs
et_reg.json
```

The steps in a feature file when it is parameterized:

Figure 3-18 Feature File when it is Parameterized

```
   Then Update File /var/lib/jenkins/ocscp_tests/data/udm1_nfset_reg.json For Key nfProfile.nfSetIdList.0 With Value {global_Var_u
dm_nfSetIdList1}
   Then Update File /var/lib/jenkins/ocscp_tests/data/udm1_nfset_reg.json For Key nfProfile.nfSetIdList.1 With Value {global_Var_u
dm_nfSetIdList3}
   Then Update Json /var/lib/jenkins/ocscp_tests/data/udm1_nfset_reg.json With Key nfProfile.nfServices.apiPrefix:{global_Var_SUT_
apiPrefix1}, nfProfile.locality:{global_Var_SUT_ServingLocality1},nfProfile.nfInstanceId:{global_Var_udm1_nfInstanceId}, nfProfile
.nfServices.ipEndPoints.port:null, nfProfile.nfServices.fqdn:{global_Var_udm1_svc_name}.default.svc.cluster.local,nfProfile.priori
ty:{feature_Var_udm1_Priority} For Service {global_Var_udm1_svc_name}
   Then Soothsayer NF With Serving Scope Register ss2 {global_Var_localNrfSetId1} /var/lib/jenkins/ocscp_tests/data/udm1_nfset_reg
.json
```

Running Feature Files

The changes are made in the product config folder, so the same configuration type can be chosen from the Jenkins UI and the test case can be run, and in the logs, it can be seen that the values of variables are being replaced with the ones that are provided in `global.yaml` or `<feature_name>.yaml`.

> **Note**
>
> Only variables that bring in some value addition require parameterization, and a change in the value of those variables does not affect the test case. A variable that can have an effect across all the feature files should be kept under `global.yaml`, a variable that is specific to a feature file has to be kept under `<feature_name>.yaml`, and a variable that is related to a specific scenario must be kept under scenario level.

3.5.4.4 ATS Tagging Support for SCP

The ATS Tagging Support feature allows you to select features and scenarios based on specific tags. There are two types of tags, Feature tag and Scenario tag. For more information about this feature, see ATS Tagging Support. The following tags are supported by SCP:

- @non_asm: This tag is for all test cases that should not run on an ASM setup.
- @https: This tag is for all https test cases.
- @multiple_ip_endpoint: This tag is for all test cases that should run only on multiple IP endpoint setups, such as IPv4 or IPv6 preferred dual stack setups.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- @single_ip_endpoint: This tag is for all test cases that should run only on single IP endpoint setups, such as IPv4 or IPv6.
- @alert: This tag is for all test cases with alert scenarios.
- @api: This tag is for all test cases with API scenarios.
- @sanity: This tag is for all the SCP sanity test cases.

Tags are available at Feature and Scenario levels. You can create tags by appending "@" as a prefix to the feature or scenario name.

Examples:

- @SCP_InterSCP_PCF_HTTPS_P0 @https
- @S1_To_test_alert_for_SCP_Observability @alert

3.5.5 Appendix

This section provides supplementary information that may be helpful for a more comprehensive understanding of installing and running SCP test cases in ATS.

3.5.5.1 Creating Custom Service Account

By default, ATS will create a service account with the below rules. If the user does not want to use the default service account, then the service account needs to be manually created with the below permission, and the created custom service account name needs to be specified in the `ocats_ocscp_values_25.1.202.yaml` file.

To run SCP-ATS, use the following rules to create a custom service account:

```
rules:
- apiGroups: ["apps"]
  resources: ["deployments", "replicasets"]
  verbs: ["watch", "get", "list", "create", "delete", "update" ,"patch"]
- apiGroups: [""]
  resources: ["pods", "services", "pod/logs"]
  verbs: ["watch", "get", "list", "create", "delete"]
- apiGroups: [""]
  resources: ["secrets"]
  verbs: ["get", "list"]
- apiGroups: [""]
  resources: ["configmaps"]
  verbs: ["watch", "get", "list", "create", "delete", "update", "patch"]
```

3.5.5.2 Adding New Stubs to SCP-ATS

For adding new stubs related to pystub, add deployment, and service yaml files inside the template folder of `ocats-pystub` charts and update the deployment name in the stub deployment file as follows:

```
name- .Values.ausf11.deploymentName

app-    .Values.ausf11.deploymentName
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Update service name, label and selectors, and ports in the `service.yaml` file.

```
name: {{ .Values.ausf11.service.name }}

app: {{ .Values.ausf11.deploymentName }}

port: {{ .Values.ausf11.service.ports.port }}
```

Update the `ocats_ocscp_values_25.1.202.yaml` file with new information:

```
ausf1:
service:
name: ausf1svc
type: ClusterIP
ports:
port: 8080
```

The images represent pystub service and deployment files for each stubs:

Figure 3-19 Sample Deployment File

```
-rw-r--r--. 1 root root   782 Oct 25 14:36 ausf12-service.yaml
-rw-r--r--. 1 root root  2597 Oct 25 14:36 ausf12-deployment.yaml
-rw-r--r--. 1 root root   782 Oct 25 14:36 ausf11-service.yaml
-rw-r--r--. 1 root root  2597 Oct 25 14:36 ausf11-deployment.yaml
```

3.6 Installing ATS for SEPP

This section describes Automated Testing Suite (ATS) installation procedures for Security Edge Protection Proxy (SEPP) in a cloud native environment.

3.6.1 Resource Requirements

Total Number of Resources

The resources required to install SEPP-ATS are as follows:

Table 3-21 Total Number of Resources

Resource	CPUs	Memory(GB)	Storage(GB)
SEPP SUT Total	15.1	24.128	0
cnDBTier Total	40	40	20
ATS Total	4.5	4.5	1
Grand Total SEPP ATS	59.6	68.628	21

Resource Details

The details of resources required to install SEPP-ATS are as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-22 Resource Details

Microservice	CPUs Required per Pod	Memory Required per Pod (GB)	Storage PVC Required per Pod (GB)	# Replicas (regular deployment)	# Replicas (ATS deployment)	CPUs Required - Total	Memory Required - Total (GB)	Storage PVC Required - Total (GB)
SEPP Pods								
n32-ingress-gateway	1.5	2	0	1	1	1.5	2	0
n32-egress-gateway	1.5	2	0	1	1	1.5	2	0
plmn-ingress-gateway	1.5	2	0	1	1	1.5	2	0
plmn-egress-gateway	1.5	2	0	1	1	1.5	2	0
pn32f-svc	1	2	0	1	1	1	2	0
cn32f-svc	1	2	0	1	1	1	2	0
cn32c-svc	0.5	1	0	1	1	0.5	1	0
pn32c-svc	0.5	1	0	1	1	0.5	1	0
config-mgr-svc	1	2	0	1	1	1	2	0
nrf-client-nfdiscovery	1	2	0	1	1	1	2	0
nrf-client-nfmanagement	1	1	0	1	1	1	1	0
ocpm-config-server	0.5	1	0	1	1	0.5	1	0
appinfo	0.5	2	0	1	1	0.5	2	0
perfinfo	0.1	0.128	0	1	1	0.1	0.128	0
nf-mediation	1	1	0	1	1	1	1	0
alternate-route	1	1	0	1	1	1	1	0
coherence-svc	1	2	0	1	1	1	2	0
SEPP SUT Totals						15.1 CPU	25.128 GB	0
ATS								
ATS Behave	3	3	1 (Optional)	1	1	3	3	1
ATS Stub (Python)	.5	.5	0	1	1	.5	.5	0

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-22 (Cont.) Resource Details

Microservice	CPUs Required per Pod	Memory Required per Pod (GB)	Storage PVC Required per Pod (GB)	# Replicas (regular deployment)	# Replicas (ATS deployment)	CPUs Required - Total	Memory Required - Total (GB)	Storage PVC Required - Total (GB)
ATS Stub-2 (Python)	.5	.5	0	1	1	.5	.5	0
ATS Stub-3 (Python)	.5	.5	0	1	1	.5	.5	0
ATS Totals						**4.5**	**4.5**	**1**
DB Tier Pods (minimum of 4 worker nodes required)								
vrt-launcher-dt-1.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-dt-2.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-dt-3.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-dt-4.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-mt-1.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-mt-2.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-mt-3.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-sq-1.cluster.local	4	4	2	2	1	4	4	2
vrt-launcher-sq-2.cluster.local	4	4	2	2	1	4	4	2

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-22 (Cont.) Resource Details

Microservice	CPUs Required per Pod	Memory Required per Pod (GB)	Storage PVC Required per Pod (GB)	# Replicas (regular deployment)	# Replicas (ATS deployment)	CPUs Required - Total	Memory Required - Total (GB)	Storage PVC Required - Total (GB)
vrt-launcher-db-installer.cluster.local	4	4	2	2	1	4	4	2
DB Tier Totals						40	40	20

3.6.1.1 SEPP ATS Compatibility Matrix

The following table lists the versions SEPP ATS and the comapability with the SEPP and ATS framework:

Table 3-23 SEPP ATS Compatibility Matrix

SEPP ATS Release	SEPP Release	ATS Framework version
25.1.203	25.1.202	25.1.203
25.1.202	25.1.201	25.1.202
25.1.201	25.1.200	25.1.202
25.1.200	25.1.200	25.1.202
25.1.101	25.1.101	25.1.101
25.1.100	25.1.100	25.1.100
24.3.0	24.3.0	24.3.0

3.6.2 Downloading the ATS Pakage

Locating and Downloading ATS and Simulator Images

To locate and download the ATS Image from MOS:

1. Log in to My Oracle Support with your credentials.
2. Select the **Patches and Updates** tab to locate the patch.
3. In the **Patch Search** window, click **Product or Family (Advanced)**.
4. Enter *Oracle Communications Cloud Native Core - 5G* in the **Product** field.
5. Select *Oracle Communications Cloud Native Core Security Edge Protection Proxy <release_number>* from **Release** drop-down.
6. Click **Search**. The **Patch Advanced Search Results** list appears.
7. Select the required ATS patch from the search results. The Patch Details window appears.
8. Click **Download**. The File Download window appears.
9. Click the **<p********_<release_number>_Tekelec>.zip** file to downlaod the CNC SEPP ATS package file.

Copyright © 2019, 2025, Oracle and/or its affiliates.

10. Untar the zip file to access all the ATS Images. The ocsepp-ats directory has the following files:
    The `csar` directory has following files:

```
ocats_ocsepp_csar_25_1_203_0_0.zip
ocats_ocsepp_csar_25_1_203_0_0.zip.sha256
```

> **Note**
>
> The above zip file contains all the images and custom values required for 25.1.203 release of OCATS-OCSEPP.

11. The `ocats_ocsepp_csar_25_1_203_0_0.zip` file has following definitions:

```
├─── Definitions
│    ├─── ocats_ocsepp_cne_compatibility.yaml
│    └─── ocats_ocsepp.yaml
├─── Files
│    ├─── ChangeLog.txt
│    ├─── Helm
│    │    └─── ocats-sepp-25.1.203.tgz (Helm Charts)
│    ├─── Licenses
│    ├─── ocats-sepp-25.1.203.tar (BDD client image)
│    ├─── Oracle.cert
│    ├─── seppstub-25.1.203.tar (Stub server image)
│    └─── Tests
├─── Scripts/
│    ├─── ocats_ocsepp_tests_jenkinsjobs_25.1.203.tgz
│    ├─────       ├─────jobs (For Persistent Volume)
│    └─────       ├─────ocsepp_tests (For Persistent Volume)
│    └─────  ocats_ocsepp_values_25.1.203.yaml (Custom values file for
installation)
├─── ocats_ocsepp.mf
└─── TOSCA-Metadata
     └─── TOSCA.meta
```

12. Copy the zip file to Kubernetes cluster where you want to deploy ATS

3.6.3 Pushing the Images to Customer Docker Registry

Preparing to Deploy ATS and Stub Pod in Kubernetes Cluster

To deploy ATS and Stub Pod in Kubernetes Cluster:

1. Run the following command to extract tar file content:

```
unzip ocats_ocsepp_csar_25_1_203_0_0.zip
```

   The following docker image tar files are located at **Files** folder:

   - ocats-sepp-25.1.203.tar
   - seppstub-25.1.203.tar

Copyright © 2019, 2025, Oracle and/or its affiliates.

2. Run the following commands in your cluster to load the ATS docker image, 'ocats-sepp-25.1.203.tar'' and Stub docker image seppstub-25.1.203.tar, and push it to your registry.

```
$ docker load -i ocats-sepp-25.1.203.tar
$ docker load -i seppstub-25.1.203.tar

$ docker tag ocats/ocats-sepp:25.1.203 <local_registry>/ocats/ocats-
sepp:25.1.203

$ docker tag ocats/seppstub:25.1.203 <local_registry>/ocats/
seppstub:25.1.203

$ docker push <local_registry>/ocats/ocats-sepp:25.1.203

$ docker push <local_registry>/ocats/seppstub:25.1.203
```

3. Run the following command to get the helm charts are located in Helm directory of files folder:

```
tar -xvf ocats-sepp-25.1.203.tgz
```

The output of this command is:

```
ocats-sepp/
ocats-sepp/Chart.yaml
ocats-sepp/charts/
ocats-sepp/values.yaml
```

4. Create a copy of the custom values located at **Scripts/ocats_ocsepp_values_25.1.203.yaml** and update it for image name, tag and other parameters as per the requirement.

3.6.4 Creating Secrets and Supporting TLS 1.2 and TLS 1.3

> **Note**
>
> If SEPP ATS is deploying in ASM, the following procedures are not required.

3.6.4.1 Configuring Root Certificate

The following are the steps to configure the root certificate (caroot.cer):

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> - Use the same root certificate (`caroot.cer`) and key which is used for creating the SEPP certificates.
> - If the user has already generated the `caroot.cer` and `cakey.pem` file while deploying SEPP then user can skip to Generate ATS certificate step.
> - Both the ATS and SEPP must have same root certificate.

1. In case `caroot.cer` is not available for SEPP, create `ssl.conf file` for SEPP using the following format:

```
# Creation of CSEPP Certs, Fqdn should be changed

#ssl.conf
[ req ]default_bits = 4096
distinguished_name = req_distinguished_namereq_extensions = req_ext
[ req_distinguished_name ]
countryName = Country Name (2 letter code)
countryName_default = IN
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Karnataka
localityName = Locality Name (eg, city)
localityName_default = Bangalore
organizationName = Organization Name (eg, company)
organizationName_default = Oracle
commonName = sepp1.inter.oracle.com
commonName_max = 64
commonName_default = sepp1.inter.oracle.com
[ req_ext ]
subjectAltName = @alt_names
[alt_names]
IP = 127.0.0.1
DNS.1 = sepp1.inter.oracle.com
```

2. Set the following environment variables:

```
export PEM_PHRASE=NextGen1
export DEPLOYMENT_NAMESPACE=sepp1
```

3. Run the following command to create the required files:

```
openssl req -new -keyout cakey.pem -out careq.pem -passout pass:$
{PEM_PHRASE} -subj "/C=IN/ST=Karnataka/L=Bangalore/O=Oracle/
CN=sepp1.inter.oracle.com/emailAddress=xyz@oracle.com"

openssl x509 -signkey cakey.pem -req -days 3650 -in careq.pem -out
caroot.cer -extensions v3_ca -passin pass:${PEM_PHRASE}
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
openssl req -x509 -nodes -sha256 -days 365 -newkey rsa:2048 -keyout
rsa_private_key -out rsa_certificate.crt -subj '/C=IN/ST=Karnataka/
L=Bangalore/O=Oracle/CN=sepp1.inter.oracle.com/emailAddress=xyz@oracle.com'

openssl rsa -in rsa_private_key -outform PEM -out rsa_private_key_pkcs1.pem

openssl req -new -key rsa_private_key -out ocsepp.csr -config ssl.conf -
subj '/C=IN/ST=Karnataka/L=Bangalore/O=Oracle/CN=sepp1.inter.oracle.com/
emailAddress=xyz@oracle.com'

openssl x509 -CA caroot.cer -CAkey cakey.pem -CAserial serial.txt -req -in
ocsepp.csr -out ocsepp.cer -days 365 -extfile ssl.conf -extensions req_ext
-passin pass:${PEM_PHRASE}

openssl ecparam -genkey -name prime256v1 -noout -out ec_private_key.pem

openssl pkcs8 -topk8 -in ec_private_key.pem -inform pem -out
ecdsa_private_key.pem -outform pem -nocrypt

openssl req -new -key ecdsa_private_key.pem -x509 -nodes -days 365 -out
ecdsa_certificate_pkcs1.crt -subj '/C=IN/ST=Karnataka/L=Bangalore/O=Oracle/
CN=sepp1.inter.oracle.com/emailAddress=xyz@oracle.com'

openssl req -new -key ecdsa_private_key.pem -out ecdsa_certificate.csr -
subj '/C=IN/ST=Karnataka/L=Bangalore/O=Oracle/CN=sepp1.inter.oracle.com/
emailAddress=xyz@oracle.com'

echo NextGen1 > trust.txt
echo NextGen1 > key.txt
echo 1234 > serial.txt
```

3.6.4.2 Generating ATS Certificate

The following are the steps to configure the ATS certificate:

1. Create and edit the `ssl.conf` file as follows:

> **Note**
>
> - While trying to access the GUI with DNS, ensure that the common Name_default is the same as the DNS name being used.
> - Ensure that the DNS is in the format `<service_name>.<namespace>.<cluster_domain>`
> - The user can add multiple DNS like DNS.1, DNS.2 so on.
> - The ATS_HELM_RELEASE_NAME is the release name which will be used to deploy ATS.

   a. In the alt_names section of the `ssl.conf`, IPs has to be listed down through which ATS GUI will be opened. We can add multiple IPs like IP.1, IP.2 so on.

Copyright © 2019, 2025, Oracle and/or its affiliates.

b. All stubserver service names (`{ATS_HELM_RELEASE_NAME}-stubserver.{ats-namespace},{ATS_HELM_RELEASE_NAME}-stubserver-2.{ats-namespace},{ATS_HELM_RELEASE_NAME}-stubserver-3.{ats-namespace})` must be in Subject Alternative Name in certificate.

c. Update ocats service name (`${ATS_HELM_RELEASE_NAME}-ocats.${DEPLOYMENT_NAMESPACE}.svc.cluster.local`) in the commonName, commonName_default and DNS name in alt_names section.

Sample code:

```
#ssl.conf
[ req ]
default_bits = 4096
distinguished_name = req_distinguished_name
req_extensions = req_ext

[ req_distinguished_name ]
countryName = Country Name (2 letter code)
countryName_default = IN
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Karnataka
localityName = Locality Name (eg, city)
localityName_default = Bangalore
organizationName = Organization Name (eg, company)
organizationName_default = Oracle
commonName = ${ATS_HELM_RELEASE_NAME}-ocats.$
{DEPLOYMENT_NAMESPACE}.svc.cluster.local
commonName_max = 64
commonName_default = ${ATS_HELM_RELEASE_NAME}-ocats.$
{DEPLOYMENT_NAMESPACE}.svc.cluster.local

[ req_ext ]
subjectAltName = @alt_names

[alt_names]
IP.1 = 127.0.0.1
IP.2 = 10.75.217.5

#Mandtory values
DNS.1 = ${ATS_HELM_RELEASE_NAME}-ocats.$
{DEPLOYMENT_NAMESPACE}.svc.cluster.local
DNS.2 = ${ATS_HELM_RELEASE_NAME}-stubserver.${DEPLOYMENT_NAMESPACE}
DNS.3 = ${ATS_HELM_RELEASE_NAME}-stubserver-2.${DEPLOYMENT_NAMESPACE}
DNS.4 = ${ATS_HELM_RELEASE_NAME}-stubserver-3.${DEPLOYMENT_NAMESPACE}
DNS.5 = localhost
```

2. Run the following command to create a `certificate signing request` or csr:

```
$ openssl req -config ssl.conf -newkey rsa:2048 -days 1000 -nodes -keyout
rsa_private_key_pkcs1.key > ssl_rsa_certificate.csr
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Output:

```
Ignoring -days; not generating a certificate
Generating a RSA private key
...+++++
.........+++++
writing new private key to 'rsa_private_key_pkcs1.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IN]:
State or Province Name (full name) [KA]:
Locality Name (eg, city) [BLR]:
Organization Name (eg, company) [ORACLE]:
Common Name (e.g. server FQDN or YOUR name) [ocats]:
[cloud-user@star23-bastion-1 ocats]$
```

3. Run the following command to verify whether all configurations are done:

```
openssl req -text -noout -verify -in ssl_rsa_certificate.csr
```

4. Run the following command to sign the certificate signing request or csr file with root certificate:

```
$ openssl x509 -extfile ssl.conf -extensions req_ext -req -in
ssl_rsa_certificate.csr -days 1000 -CA caroot.cer -CAkey cakey.pem -
set_serial 04 > ssl_rsa_certificate.crt
```

Output:

```
Signature ok
subject=C = IN, ST = KA, L = BLR, O = ORACLE, CN = sepp-ats-rel-
ocats.testns.svc.cluster.local
Getting CA Private Key
[cloud-user@star23-bastion-1 ocats]$
```

> **Note**
>
> When the output prompts for the current password, enter the password was used to create`cakey.pem` file.

5. Verify whether the certificate is properly signed by root certificate:

```
$ openssl verify -CAfile caroot.cer ssl_rsa_certificate.crt
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Output:

```
ssl_rsa_certificate.crt:OK
```

6. For then jenkins to support GUI access through https, a jks file has to be created. Perform the following steps to generate jks file for Jenkins Server:
   a. Run the following command to generate the `.p12 keystore` file:

      ```
      $ openssl pkcs12 -inkey rsa_private_key_pkcs1.key -in
      ssl_rsa_certificate.crt -export -out certificate.p12
      ```

      Output:

      ```
      Enter Export Password:
      Verifying - Enter Export Password:
      ```

      > **Note**
      >
      > When the output prompts for the password, enter the password and note it down, as it is required for creating the jks file.

7. Run the following command to convert `.p12` file in jks format file to be used in Jenkins Server:

   > **Note**
   >
   > - Esure to use the same password used for creating jks file and for creating the .p12 file.
   > - Java should be pre-installed to run the keytool utility.

   ```
   $ keytool -importkeystore -srckeystore ./certificate.p12 -srcstoretype
   pkcs12 -destkeystore jenkinsserver.jks -deststoretype JKS
   ```

   Output:

   ```
   Importing keystore ./certificate.p12 to jenkinsserver.jks...
   Enter destination keystore password:
   Re-enter new password:
   Enter source keystore password:
   Entry for alias 1 successfully imported.
   Import command completed:  1 entries successfully imported, 0 entries
   failed or cancelled
   ```

   The generated file, `jenkinserver.jks` must be given to the jenkins server.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.6.4.3 Creating ATS Secret and Configuring Helm chart

The following are the steps to create ATS Secret and configure the Helm chart:

> **Note**
>
> The user can decide whether to use the generated CA signed certificate or self-signed certificate.

1. Run the following command to create secret:

   ```
   kubectl create secret generic ocats-sepp-secret --from-
   file=jenkinsserver.jks --from-file=ssl_rsa_certificate.crt --from-
   file=rsa_private_key_pkcs1.key --from-file=caroot.cer --from-
   literal=jks_password=${JKS_PASS} -n <deployment_namespace>
   ```

   > **Note**
   >
   > Use the same password for `JKS_PASS` that was used during the creation of the `jenkinserver.jks` file.

2. Run the following command to verify the secret:

   ```
   $ kubectl describe secret ocats-sepp-secret -n testns
   ```

   Output:

   ```
   Name:         ocats-sepp-secret
   Namespace:    seppsvc
   Labels:       <none>
   Annotations:  <none>

   Type:  Opaque

   Data
   ====
   caroot.cer:                      1147 bytes
   ssl_rsa_certificate.crt:         1424 bytes
   jenkinsserver.jks:               2357 bytes
   rsa_private_key_pkcs1.key:       1675 bytes
   ```

Changes to the Helm charts:

The following changes must be updated in the `ocats_ocsepp_values_<version>.yaml` file:

Copyright © 2019, 2025, Oracle and/or its affiliates.

- The Helm parameter `atsGuiTLSEnabled` must set to true for ATS to get the certificates and support HTTPS for GUI. If the user does not want to open ATS GUI in HTTPS mode, then `atsGuiTLSEnabled` flag should be set to false.

  ```
  atsGuiTLSEnabled: false
  ```

- The Helm parameter `atsCommunicationTLSEnabled` must set to true for for necessary context variable to be created which can be later used to communicate with other services in HTTPS.
  - For non-asm deployment `atsCommunicationTLSEnabled` flag should be set to true.
  - For asm deployment `atsCommunicationTLSEnabled` flag should be set to false.

  ```
  atsCommunicationTLSEnabled: true #If set to true,  ATS will get necessary
  variables to communicate with SUT, Stub
          or other  NFs with TLS enabled. It is not required in ASM
      environment.
  ```

- The certificates section of the ocats and stubserver section of the ats custom values file must be updated as follows:

  ```
  certificates:
     cert_secret_name: "ocats-sepp-secret"
     a_cert: "caroot.cer"
     client_cert: "ssl_rsa_certificate.crt"
     private_key: "rsa_private_key_pkcs1.key"

     # This parameter is needed when atsGuiTLSEnabled is set to true. This
  file is necessary for ATS GUI to be opend with secured TLS protocol. The
  file caroot.pem, used during creation of jks file needs to be passed for
  Jenkins/ATS API communication.
     jks_file: "jenkinsserver.jks"
  ```

  Add caroot certificate in browser to access the ATS GUI. The `caroot.cer` file certificate created by us will be added to Truststore, in this case the browser.

The following are the steps to add caroot certificate in the browser to access the ATS GUI, either Mozilla Firefox or Chrome:

> **Note**
>
> Future versions of these browsers may involve different menu options. For more information on importing root certificate, see the browser documentation to add a self-signed certificate to the browser as a trusted certificate.

1. In the Chrome browser, navigate to the settings and search for certificates.
2. Click the **security** option that appears next to search.
3. Click the **Manage Device Certificate** option. The Keychain Access window opens.
4. Search the tab certificate and drag and drop the downloaded caroot certificate.

Copyright © 2019, 2025, Oracle and/or its affiliates.

5. Find the uploaded certificate in the list, usually listed by a temporary name.
6. Double click the certificate and expand the Trust option.
7. In **When using this certificate** option, assign it to "always trust".
8. Close the window and validate if it asks for the password.
9. Save and restart the browser.

1. In the Mozilla Firefox browser, navigate to the settings and search for certificates.
2. Click the **View Certificate** that appears next to search. This opens a Certificate Manager window.
3. Navigate to the Authorities section, click the **Import** button, and upload the `caroot` certificate.
4. Click the **Trust** options in the pop-up window and click OK.
5. Save and restart the browser.

3.6.5 Create ATS Health Check secret

To enable the ATS health check pipeline, the following configurations needs to be updated in the `ocats_ocsepp_values_<version>.yaml` file:

Non OCI Environment

1. The following parameters needs to be updated in the base64 encoded version format, occnehostip, occnehostusername, occnehostpassword, envtype.
2. On installing ATS, health check secret will be created, and the health check pipeline will be shown in the ATS GUI. If `healthcheck` parameter is set to false, then the health check pipeline will not be visible in the ATS GUI.

```
ats-ocats
  atsFeatures:
    healthcheck: true

  sshDetails:
    secretname: "healthchecksecret"
    occnehostip: "" # $(echo -n '10.75.217.42' | base64), Where occne host ip
needs to be provided
    occnehostusername: "" # $(echo -n 'cloud-user' | base64), Where occne
host username needs to be provided
    occnehostpassword: "" # $(echo -n '*****' | base64), Where password of host
needs to be provided
    envtype: "" # $(echo -n 'OCCNE' | base64), Where occne keyword needs to
be provided
```

OCI Environment

For key based health check support in OCI, refer to ATS Health Check section under the ATS Framework Features.

1. The following parameters needs to be updated in the base64 encoded version in the `ocats_ocsepp_values_<version>.yaml` file:

```
ats-ocats
  atsFeatures:
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    healthcheck: true    envtype: "" # $(echo -n 'OCCNE' | base64), Where
occne keyword needs to be provided

  sshDetails
    ociHealthCheck:
      passwordAuthenticationEnabled: false
      bastion:
        ip: "" # $(echo -n '10.75.217.42' | base64), Where occne host ip
needs to be provided
        username: "" # $(echo -n '10.75.217.42' | base64), Where occne host
ip needs to be provided
        password: ""
      operatorInstance:
        ip: "" # $(echo -n '10.75.217.42' | base64), Where occne host ip
needs to be provided
        username: "" # $(echo -n '10.75.217.42' | base64), Where occne host
ip needs to be provided
        password: ""
```

3.6.6 Configuring ATS

This section describes how to configure ATS for SEPP.

3.6.6.1 Enabling Aspen Service Mesh

To enable Aspen Service Mesh (ASM) for ATS, complete the following procedure:

> **Note**
>
> By default, this feature is disabled.

1. If ASM is not enabled on the global level for the namespace, run the following command before deploying ATS:

   ```
   kubectl label --overwrite namespace <namespace_name> istio-
   injection=enabled
   ```

   Example:

   ```
   kubectl label --overwrite namespace seppsvc istio-injection=enabled
   ```

2. Add the following annotations in the **BDD client** section in the `ocats-sepp-custom-values.yaml` file to support ASM:
   To enable or disable the ASM, update the `ASMEnabled` flag to true or false.

   ```
   ASMEnabled: false
   asm:
     configMgrPort: 9090 # config manager service port
     pn32fPort: 9090 # pn32f service port
     stubServerPort: 8080 # stub server service port
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  plmnIgwPort: 80 # plmn ingress gateway service port
  n32IgwPort: 80 # n32 ingress gateway service port
```

3. Add the following value in the **stub-server** section in the `ocats-sepp-custom-values.yaml` file to support ASM:
   To enable or disable the ASM, update the `ASMEnabled` flag to true or false.

   ```
   ASMEnabled: false
   ```

4. For asm deployment, set the `atsCommunicationTLSEnabled` flag to false as given below:

   ```
   atsCommunicationTLSEnabled: false
    #If set to true, ATS will get necessary variables to communicate with
   SUT, Stub or other NFs with TLS enabled. It is not required in ASM
   environment.
   ```

5. (Optional) The user can configure the resources assigned to the Aspen Mesh (istio-proxy) sidecars in `ocats_ocsepp_values_<version>.yaml` file sections as follows:

   ```
   asm:
   istioResources:
       limits:
         cpu: 2
         memory: 1Gi
       requests:
         cpu: 100m
         memory: 128Mi
   ```

   > **Note**
   >
   > It is recommended to use the default values of the resources.

6. In `ocats_ocsepp_values_<version>.yaml` file, parameter `expose_tls_service` in stubserver section should be configured to false in ASM mode to deploy stub server in HTTP mode.

   ```
   stubserver:
     service:
       expose_tls_service: false
   ```

7. Update the following parameters in the `ocats_ocsepp_values_<values>.yaml` file to empty values, as secret creation is not required when operating in ASM mode.

   ```
   ocats:
       certificates:
       cert_secret_name: ""
       ca_cert: ""
       client_cert: ""
       private_key: ""
       jks_file: "" # This parameter is needed when atsGuiTLSEnabled is set
   to true. This file is necessary for ATS GUI to be opened with secured TLS
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
protocol.


In stubserver section:
stubserver:
  env:
    cert_secret_name: ""
    ca_cert: ""
    client_cert: ""
    private_key: ""
```

> **Note**
>
> - If the SEPP is deployed with ASM enabled and user disables the ASM on the global level, then user needs to redeploy the setup to work without ASM.
> - Mediation is supported in ASM from release 25.1.1xx onwards.

3.6.6.2 ATS API

The application programming interface (API) feature provides APIs to perform routine ATS tasks such as starting the ATS suite, monitoring and stopping the ATS suite etc.

By default, this feature is enabled in ATS framework.

For more details about the ATS API feature, refer to ATS API feature section.

3.6.6.3 Enabling Static Port

To enable static port:

> **Note**
>
> ATS supports static port. By default, this feature is disabled.

- In the `ocats_ocsepp_values_<version>.yaml` file under service section, set the **staticNodePortEnabled** parameter value to 'true' and **staticNodePort** parameter value with valid nodePort.

```
ocats:
  service:
      customExtension:
        labels: {}
        annotations: {}
      type: LoadBalancer
      ports:
        https:
          port: "8443"
          staticNodePortEnabled: false
          staticNodePort: ""
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
http:
  port: "8080"
  staticNodePortEnabled: false
  staticNodePort: ""
```

3.6.6.4 Enabling Roaming Hub Mode

SEPP ATS supports two types deployment modes:

- SEPP
- Roaming Hub or Hosted SEPP Mode

Flag has been introduced to select the deployment mode.

#Flag to enable Roaming Hub mode

```
RHenabled: True
```

3.6.6.5 Enabling Hosted SEPP Mode

The Hosted SEPP Mode can be enabled as follows:

#Flag to enable Hosted SEPP mode

```
RHenabled: True
```

Customizing Error Code Variable in Hosted SEPP Mode

For handling failure scenarios in Hosted SEPP mode, the following customized error code variable has been introduced:

#Customized error code variable for Hosted SEPP

```
HSErrCode: "400"
```

3.6.6.6 Configuring Egress Rate Limiting Feature

If the Egress Rate Limiting feature is enabled in SEPP deployment, to run the Egress Rate Limiter test cases in ATS, `EgressRateLimiterFlag` parameter has been introduced in the `ocats-sepp-custom-values.yaml` file. If the Egress Rate Limiting feature is disabled in SEPP deployment, ensure to set `EgressRateLimiterFlag` parameter to false.

```
EgressRateLimiterFlag: true/false
```

For more information about the feature, see the "Rate Limiting for Egress Roaming Signaling per PLMN" section in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide* and the "Configuration Parameters" section in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.6.6.7 Configuring Ingress Rate Limiting Feature

If the Ingress rate limiting feature is enabled in SEPP deployment, to run the Ingress rate limiter test cases in ATS, `IngressRateLimiterFlag` parameter has been introduced in the `ocats_ocsepp_values_<version>.yaml` file. If the Ingress rate limiting feature is disabled in SEPP deployment, ensure to set `IngressRateLimiterFlag` parameter to `false`.

Flag to enable or disable Ingress Rate Limiter:

```
IngressRateLimiterFlag: true/false
```

For more information about the feature, see the "Rate Limiting for Ingress Roaming Signaling per Remote SEPP Set" section in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy User Guide* and the "Configuration Parameters" section in *Oracle Communications Cloud Native Core, Security Edge Protection Proxy Installation, Upgrade, and Fault Recovery Guide*.

3.6.6.8 Configuring Single Service Account

> **Note**
>
> If the user doesn't want to configure a single service account, skip the following steps.

1. To configure the single service account across all microservices, the following parameter must be updated in the `ocats_ocsepp_values_<version>.yaml` file with the service account name that was created while deploying SEPP.

```
ocats:
    serviceAccountName: "" # Update service account name
```

2. In the stubserver section, update the following:

> **Note**
>
> Here, update the same service account name that was used while deploying SEPP.

```
stubserver:
     serviceAccount:
        create: false
        name: ""
```

3.6.6.9 Configuring the Cache Refresh Timeout Value

Copyright © 2019, 2025, Oracle and/or its affiliates.

SEPP supports Refresh ahead cache. To run the ATS cases, *cacheRefreshTimeout* value must be set to 1000 (ms) so that cache can be updated automatically after every test case is executed when ATS suite is triggered.

To set the cache refresh timeout value:

In the ocsepp-custom-values.yaml file, under cn32f-svc and pn32f-svc microservices, set the *cacheRefreshTimeout* parameter value to 1000 (ms).

```
configs:
 cacheRefreshTimeout: 1000 #(ms)
 cacheRefreshInitTimeout: 50000 #(ms)
```

> **Note**
>
> If ATS is not configured, the cacheRefreshTimeout value must be 30000 (ms).

3.6.6.10 Configuring the Topology Cache Refresh Timeout Value

SEPP supports Refresh ahead cache. To run the ATS cases, *topologycacheRefreshTimeout* value must be set to 1000 (ms) so that cache can be updated automatically after every test case is executed when ATS suite is triggered.

In the *ocsepp-custom-values.yaml* file, under cn32f-svc and pn32f-svc microservices, set the *topologycacheRefreshTimeout* parameter value to 1000 (ms).

```
topologyHiding:
  timerConfig:
     topologyhidingCacheRefreshTimeout: 30000
     topologyhidingCacheRefreshInitTimeout: 50000
     topologyhidingHistoryUpdateTimeout: 30000
     topologyhidingHistoryRefreshSeconds: 60
  config:
     topologyHidingStateCheck: true
```

3.6.6.11 Configuring the Security Counter Measure Cache Refresh Timeout

SEPP supports Refresh ahead cache. To run the ATS cases, you must set the value of the `securityCacheRefreshTimeout` parameter to 10 (ms) so that cache can be automatically updated after every test case is run when ATS suite is triggered.

Copyright © 2019, 2025, Oracle and/or its affiliates.

In the *ocsepp-custom-values.yaml* file, under cn32f-svc and pn32f-svc microservices, set the `securityCacheRefreshTimeout` parameter value to 1000 (ms).

```
configs:
 securityCacheRefreshTimeout: 1000 #(ms)
 securityCacheRefreshInitTimeout: 50000 #(ms)
```

3.6.6.12 Configuring n32cHandshakePlmnIdListValidationEnabled

SEPP supports validation of PLMN ID List in n32c capability exchange message which can be turned off. To run the ATS cases, `n32cHandshakePlmnIdListValidationEnabled` value must be set to true so that testcases validating PLMN ID List in n32c capability exchange message can be run successfully when ATS suite is triggered.

To set the `n32cHandshakePlmnIdListValidationEnabled` value, do the following:

In the `ocsepp_custom_values_<version>.yaml` file, under localProfile, set the `n32cHandshakePlmnIdListValidationEnabled parameter` value to true.

```
    localProfile:
    name: "SEPP-3"
    plmnIdList: [{"mcc":"111","mnc":"100"},{"mcc":"111","mnc":"101"},
{"mcc":"111","mnc":"102"},{"mcc":"111","mnc":"103"},{"mcc":"111","mnc":"104"},
{"mcc":"111","mnc":"105"},{"mcc":"111","mnc":"106"},{"mcc":"111","mnc":"107"},
{"mcc":"111","mnc":"108"},{"mcc":"111","mnc":"109"},{"mcc":"111","mnc":"110"},
{"mcc":"111","mnc":"111"},{"mcc":"111","mnc":"112"},{"mcc":"111","mnc":"113"},
{"mcc":"111","mnc":"114"},{"mcc":"111","mnc":"115"},{"mcc":"111","mnc":"116"},
{"mcc":"111","mnc":"117"},{"mcc":"111","mnc":"118"},{"mcc":"111","mnc":"119"},
{"mcc":"111","mnc":"120"},{"mcc":"111","mnc":"121"},{"mcc":"111","mnc":"122"},
{"mcc":"111","mnc":"123"},{"mcc":"111","mnc":"124"},{"mcc":"111","mnc":"125"},
{"mcc":"111","mnc":"126"},{"mcc":"111","mnc":"127"},{"mcc":"111","mnc":"128"},
{"mcc":"111","mnc":"129"}]
    # Do not change this value, this will be always true
    sbiTargetApiRootSupported: true
    # Enable PLMN ID List Validation in Exchange Capability Request, Default
set to true
    n32cHandshakePlmnIdListValidationEnabled: true
    # PLMN ID List Validation Type in Exchange Capability Request, can be
SUBSET or STRICT only
    n32cHandshakePlmnIdListValidationType: "SUBSET"
```

Update following PLMN ID list in localProfile section for Roaming Hub mode to run the ATS testcases:

```
[{"mcc":"111","mnc":"200"},{"mcc":"111","mnc":"201"},
{"mcc":"111","mnc":"202"},{"mcc":"111","mnc":"203"},{"mcc":"111","mnc":"204"},
{"mcc":"111","mnc":"205"},{"mcc":"111","mnc":"206"},{"mcc":"111","mnc":"207"},
{"mcc":"111","mnc":"208"},{"mcc":"111","mnc":"209"},{"mcc":"111","mnc":"210"},
{"mcc":"111","mnc":"211"},{"mcc":"111","mnc":"212"},{"mcc":"111","mnc":"213"},
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

{"mcc":"111","mnc":"214"},{"mcc":"111","mnc":"215"},{"mcc":"111","mnc":"216"},
{"mcc":"111","mnc":"217"},{"mcc":"111","mnc":"218"},{"mcc":"111","mnc":"219"},
{"mcc":"111","mnc":"220"},{"mcc":"111","mnc":"221"},{"mcc":"111","mnc":"222"},
{"mcc":"111","mnc":"223"},{"mcc":"111","mnc":"224"},{"mcc":"111","mnc":"225"},
{"mcc":"111","mnc":"226"},{"mcc":"111","mnc":"227"},{"mcc":"111","mnc":"228"},
{"mcc":"111","mnc":"229"},{"mcc":"111","mnc":"230"},{"mcc":"111","mnc":"231"},
{"mcc":"111","mnc":"232"},{"mcc":"111","mnc":"233"},{"mcc":"111","mnc":"234"},
{"mcc":"111","mnc":"235"},{"mcc":"111","mnc":"236"},{"mcc":"111","mnc":"237"},
{"mcc":"111","mnc":"238"},{"mcc":"111","mnc":"239"},{"mcc":"111","mnc":"240"},
{"mcc":"111","mnc":"241"},{"mcc":"111","mnc":"242"},{"mcc":"111","mnc":"243"},
{"mcc":"111","mnc":"244"},{"mcc":"111","mnc":"245"},{"mcc":"111","mnc":"246"},
{"mcc":"111","mnc":"247"},{"mcc":"111","mnc":"248"},{"mcc":"111","mnc":"249"},
{"mcc":"111","mnc":"250"},{"mcc":"111","mnc":"251"},{"mcc":"111","mnc":"252"},
{"mcc":"111","mnc":"253"},{"mcc":"111","mnc":"254"},{"mcc":"111","mnc":"255"},
{"mcc":"111","mnc":"256"},{"mcc":"111","mnc":"257"},{"mcc":"111","mnc":"258"},
{"mcc":"111","mnc":"259"},{"mcc":"111","mnc":"260"},{"mcc":"111","mnc":"261"},
{"mcc":"111","mnc":"262"},{"mcc":"111","mnc":"263"},{"mcc":"111","mnc":"264"},
{"mcc":"111","mnc":"265"},{"mcc":"111","mnc":"266"},{"mcc":"111","mnc":"267"},
{"mcc":"111","mnc":"268"},{"mcc":"111","mnc":"269"},{"mcc":"111","mnc":"270"},
{"mcc":"111","mnc":"271"},{"mcc":"111","mnc":"272"},{"mcc":"111","mnc":"273"},
{"mcc":"111","mnc":"274"},{"mcc":"111","mnc":"275"},{"mcc":"111","mnc":"276"},
{"mcc":"111","mnc":"277"},{"mcc":"111","mnc":"278"},{"mcc":"111","mnc":"279"},
{"mcc":"111","mnc":"280"},{"mcc":"111","mnc":"281"},{"mcc":"111","mnc":"282"},
{"mcc":"111","mnc":"283"},{"mcc":"111","mnc":"284"},{"mcc":"111","mnc":"285"},
{"mcc":"111","mnc":"286"},{"mcc":"111","mnc":"287"},{"mcc":"111","mnc":"288"},
{"mcc":"111","mnc":"289"},{"mcc":"111","mnc":"290"},{"mcc":"111","mnc":"291"},
{"mcc":"111","mnc":"292"},{"mcc":"111","mnc":"293"},{"mcc":"111","mnc":"294"},
{"mcc":"111","mnc":"295"},{"mcc":"111","mnc":"296"},{"mcc":"111","mnc":"297"},
{"mcc":"111","mnc":"298"},{"mcc":"111","mnc":"299"},{"mcc":"111","mnc":"300"},
{"mcc":"111","mnc":"301"},{"mcc":"111","mnc":"302"},{"mcc":"111","mnc":"303"},
{"mcc":"111","mnc":"304"},{"mcc":"111","mnc":"305"},{"mcc":"111","mnc":"306"},
{"mcc":"111","mnc":"307"},{"mcc":"111","mnc":"308"},{"mcc":"111","mnc":"309"},
{"mcc":"111","mnc":"310"},{"mcc":"111","mnc":"311"},{"mcc":"111","mnc":"312"},
{"mcc":"111","mnc":"313"},{"mcc":"111","mnc":"314"},{"mcc":"111","mnc":"315"},
{"mcc":"111","mnc":"316"},{"mcc":"111","mnc":"317"},{"mcc":"111","mnc":"318"},
{"mcc":"111","mnc":"319"},{"mcc":"111","mnc":"320"},{"mcc":"111","mnc":"321"},
{"mcc":"111","mnc":"322"},{"mcc":"111","mnc":"323"},{"mcc":"111","mnc":"324"},
{"mcc":"111","mnc":"325"},{"mcc":"111","mnc":"326"},{"mcc":"111","mnc":"327"},
{"mcc":"111","mnc":"328"},{"mcc":"111","mnc":"329"},{"mcc":"111","mnc":"330"},
{"mcc":"111","mnc":"331"},{"mcc":"111","mnc":"332"},{"mcc":"111","mnc":"333"},
{"mcc":"111","mnc":"334"},{"mcc":"111","mnc":"335"},{"mcc":"111","mnc":"336"},
{"mcc":"111","mnc":"337"},{"mcc":"111","mnc":"338"},{"mcc":"111","mnc":"339"},
{"mcc":"111","mnc":"340"},{"mcc":"111","mnc":"341"},{"mcc":"111","mnc":"342"},
{"mcc":"111","mnc":"343"},{"mcc":"111","mnc":"344"},{"mcc":"111","mnc":"345"},
{"mcc":"111","mnc":"346"},{"mcc":"111","mnc":"347"},{"mcc":"111","mnc":"348"},
{"mcc":"111","mnc":"349"},{"mcc":"111","mnc":"350"},{"mcc":"111","mnc":"351"},
{"mcc":"111","mnc":"352"},{"mcc":"111","mnc":"353"},{"mcc":"111","mnc":"354"},
{"mcc":"111","mnc":"355"},{"mcc":"111","mnc":"356"},{"mcc":"111","mnc":"357"},
{"mcc":"111","mnc":"358"},{"mcc":"111","mnc":"359"},{"mcc":"111","mnc":"360"},
{"mcc":"111","mnc":"361"},{"mcc":"111","mnc":"362"},{"mcc":"111","mnc":"363"},
{"mcc":"111","mnc":"364"},{"mcc":"111","mnc":"365"},{"mcc":"111","mnc":"366"},
{"mcc":"111","mnc":"367"},{"mcc":"111","mnc":"368"},{"mcc":"111","mnc":"369"},
{"mcc":"111","mnc":"370"},{"mcc":"111","mnc":"371"},{"mcc":"111","mnc":"372"},
{"mcc":"111","mnc":"373"},{"mcc":"111","mnc":"374"},{"mcc":"111","mnc":"375"},
{"mcc":"111","mnc":"376"},{"mcc":"111","mnc":"377"},{"mcc":"111","mnc":"378"},
{"mcc":"111","mnc":"379"},{"mcc":"111","mnc":"380"},{"mcc":"111","mnc":"381"},

Copyright © 2019, 2025, Oracle and/or its affiliates.

{"mcc":"111","mnc":"382"},{"mcc":"111","mnc":"383"},{"mcc":"111","mnc":"384"},
{"mcc":"111","mnc":"385"},{"mcc":"111","mnc":"386"},{"mcc":"111","mnc":"387"},
{"mcc":"111","mnc":"388"},{"mcc":"111","mnc":"389"},{"mcc":"111","mnc":"390"},
{"mcc":"111","mnc":"391"},{"mcc":"111","mnc":"392"},{"mcc":"111","mnc":"393"},
{"mcc":"111","mnc":"394"},{"mcc":"111","mnc":"395"},{"mcc":"111","mnc":"396"},
{"mcc":"111","mnc":"397"},{"mcc":"111","mnc":"398"},{"mcc":"111","mnc":"399"},
{"mcc":"111","mnc":"400"},{"mcc":"111","mnc":"401"},{"mcc":"111","mnc":"402"},
{"mcc":"111","mnc":"403"},{"mcc":"111","mnc":"404"},{"mcc":"111","mnc":"405"},
{"mcc":"111","mnc":"406"},{"mcc":"111","mnc":"407"},{"mcc":"111","mnc":"408"},
{"mcc":"111","mnc":"409"},{"mcc":"111","mnc":"410"},{"mcc":"111","mnc":"411"},
{"mcc":"111","mnc":"412"},{"mcc":"111","mnc":"413"},{"mcc":"111","mnc":"414"},
{"mcc":"111","mnc":"415"},{"mcc":"111","mnc":"416"},{"mcc":"111","mnc":"417"},
{"mcc":"111","mnc":"418"},{"mcc":"111","mnc":"419"},{"mcc":"111","mnc":"420"},
{"mcc":"111","mnc":"421"},{"mcc":"111","mnc":"422"},{"mcc":"111","mnc":"423"},
{"mcc":"111","mnc":"424"},{"mcc":"111","mnc":"425"},{"mcc":"111","mnc":"426"},
{"mcc":"111","mnc":"427"},{"mcc":"111","mnc":"428"},{"mcc":"111","mnc":"429"},
{"mcc":"111","mnc":"430"},{"mcc":"111","mnc":"431"},{"mcc":"111","mnc":"432"},
{"mcc":"111","mnc":"433"},{"mcc":"111","mnc":"434"},{"mcc":"111","mnc":"435"},
{"mcc":"111","mnc":"436"},{"mcc":"111","mnc":"437"},{"mcc":"111","mnc":"438"},
{"mcc":"111","mnc":"439"},{"mcc":"111","mnc":"440"},{"mcc":"111","mnc":"441"},
{"mcc":"111","mnc":"442"},{"mcc":"111","mnc":"443"},{"mcc":"111","mnc":"444"},
{"mcc":"111","mnc":"445"},{"mcc":"111","mnc":"446"},{"mcc":"111","mnc":"447"},
{"mcc":"111","mnc":"448"},{"mcc":"111","mnc":"449"},{"mcc":"111","mnc":"450"},
{"mcc":"111","mnc":"451"},{"mcc":"111","mnc":"452"},{"mcc":"111","mnc":"453"},
{"mcc":"111","mnc":"454"},{"mcc":"111","mnc":"455"},{"mcc":"111","mnc":"456"},
{"mcc":"111","mnc":"457"},{"mcc":"111","mnc":"458"},{"mcc":"111","mnc":"459"},
{"mcc":"111","mnc":"460"},{"mcc":"111","mnc":"461"},{"mcc":"111","mnc":"462"},
{"mcc":"111","mnc":"463"},{"mcc":"111","mnc":"464"},{"mcc":"111","mnc":"465"},
{"mcc":"111","mnc":"466"},{"mcc":"111","mnc":"467"},{"mcc":"111","mnc":"468"},
{"mcc":"111","mnc":"469"},{"mcc":"111","mnc":"470"},{"mcc":"111","mnc":"471"},
{"mcc":"111","mnc":"472"},{"mcc":"111","mnc":"473"},{"mcc":"111","mnc":"474"},
{"mcc":"111","mnc":"475"},{"mcc":"111","mnc":"476"},{"mcc":"111","mnc":"477"},
{"mcc":"111","mnc":"478"},{"mcc":"111","mnc":"479"},{"mcc":"111","mnc":"480"},
{"mcc":"111","mnc":"481"},{"mcc":"111","mnc":"482"},{"mcc":"111","mnc":"483"},
{"mcc":"111","mnc":"484"},{"mcc":"111","mnc":"485"},{"mcc":"111","mnc":"486"},
{"mcc":"111","mnc":"487"},{"mcc":"111","mnc":"488"},{"mcc":"111","mnc":"489"},
{"mcc":"111","mnc":"490"},{"mcc":"111","mnc":"491"},{"mcc":"111","mnc":"492"},
{"mcc":"111","mnc":"493"},{"mcc":"111","mnc":"494"},{"mcc":"111","mnc":"495"},
{"mcc":"111","mnc":"496"},{"mcc":"111","mnc":"497"},{"mcc":"111","mnc":"498"},
{"mcc":"111","mnc":"499"},{"mcc":"111","mnc":"500"},{"mcc":"111","mnc":"501"},
{"mcc":"111","mnc":"502"},{"mcc":"111","mnc":"503"},{"mcc":"111","mnc":"504"},
{"mcc":"111","mnc":"505"},{"mcc":"111","mnc":"506"},{"mcc":"111","mnc":"507"},
{"mcc":"111","mnc":"508"},{"mcc":"111","mnc":"509"},{"mcc":"111","mnc":"510"},
{"mcc":"111","mnc":"511"},{"mcc":"111","mnc":"512"},{"mcc":"111","mnc":"513"},
{"mcc":"111","mnc":"514"},{"mcc":"111","mnc":"515"},{"mcc":"111","mnc":"516"},
{"mcc":"111","mnc":"517"},{"mcc":"111","mnc":"518"},{"mcc":"111","mnc":"519"},
{"mcc":"111","mnc":"520"},{"mcc":"111","mnc":"521"},{"mcc":"111","mnc":"522"},
{"mcc":"111","mnc":"523"},{"mcc":"111","mnc":"524"},{"mcc":"111","mnc":"525"},
{"mcc":"111","mnc":"526"},{"mcc":"111","mnc":"527"},{"mcc":"111","mnc":"528"},
{"mcc":"111","mnc":"529"},{"mcc":"111","mnc":"530"},{"mcc":"111","mnc":"531"},
{"mcc":"111","mnc":"532"},{"mcc":"111","mnc":"533"},{"mcc":"111","mnc":"534"},
{"mcc":"111","mnc":"535"},{"mcc":"111","mnc":"536"},{"mcc":"111","mnc":"537"},
{"mcc":"111","mnc":"538"},{"mcc":"111","mnc":"539"},{"mcc":"111","mnc":"540"},
{"mcc":"111","mnc":"541"},{"mcc":"111","mnc":"542"},{"mcc":"111","mnc":"543"},
{"mcc":"111","mnc":"544"},{"mcc":"111","mnc":"545"},{"mcc":"111","mnc":"546"},
{"mcc":"111","mnc":"547"},{"mcc":"111","mnc":"548"},{"mcc":"111","mnc":"549"},

Copyright © 2019, 2025, Oracle and/or its affiliates.

```
{"mcc":"111","mnc":"550"},{"mcc":"111","mnc":"551"},{"mcc":"111","mnc":"552"},
{"mcc":"111","mnc":"553"},{"mcc":"111","mnc":"554"},{"mcc":"111","mnc":"555"},
{"mcc":"111","mnc":"556"},{"mcc":"111","mnc":"557"},{"mcc":"111","mnc":"558"},
{"mcc":"111","mnc":"559"},{"mcc":"111","mnc":"560"},{"mcc":"111","mnc":"561"},
{"mcc":"111","mnc":"562"},{"mcc":"111","mnc":"563"},{"mcc":"111","mnc":"564"},
{"mcc":"111","mnc":"565"},{"mcc":"111","mnc":"566"},{"mcc":"111","mnc":"567"},
{"mcc":"111","mnc":"568"},{"mcc":"111","mnc":"569"},{"mcc":"111","mnc":"570"},
{"mcc":"111","mnc":"571"},{"mcc":"111","mnc":"572"},{"mcc":"111","mnc":"573"},
{"mcc":"111","mnc":"574"},{"mcc":"111","mnc":"575"},{"mcc":"111","mnc":"576"},
{"mcc":"111","mnc":"577"},{"mcc":"111","mnc":"578"},{"mcc":"111","mnc":"579"},
{"mcc":"111","mnc":"580"},{"mcc":"111","mnc":"581"},{"mcc":"111","mnc":"582"},
{"mcc":"111","mnc":"583"},{"mcc":"111","mnc":"584"},{"mcc":"111","mnc":"585"},
{"mcc":"111","mnc":"586"},{"mcc":"111","mnc":"587"},{"mcc":"111","mnc":"588"},
{"mcc":"111","mnc":"589"},{"mcc":"111","mnc":"590"},{"mcc":"111","mnc":"591"},
{"mcc":"111","mnc":"592"},{"mcc":"111","mnc":"593"},{"mcc":"111","mnc":"594"},
{"mcc":"111","mnc":"595"},{"mcc":"111","mnc":"596"},{"mcc":"111","mnc":"597"},
{"mcc":"111","mnc":"598"},{"mcc":"111","mnc":"599"}]
```

3.6.6.13 Configuring Request Timeout

SEPP supports `requestTimeout` parameter to avoid the request timeout of services. The `requestTimeout` is increased in the given services to avoid any request timeout that may happen due to the processing of response by the stub server.

- The user has to set `requestTimeout` to 5000 in the cn32f-svc, pn32f-svc, plmn-egress-gateway, n32-egress-gateway, n32-ingress-gateway, and plmn-ingress-gateway sections of the `ocsepp_custom_values_<version>.yaml` file.

```
requestTimeout: 5000 #(ms)
```

> **Note**
>
> - If the ATS cases are failing due to the request timeout, increase the request timeout of a particular service which is creating the issue. For example set request timeout to "5000".
> - If ATS is not configured, `requestTimeout` must be set to the following values:

Table 3-24 requestTimeout

Service	REQUEST TIMEOUT
cn32f-svc	2000
pn32f-svc	1100
plmn-egress-gateway	1000
n32-egress-gateway	1500
n32-ingress-gateway	700

Updating n32-ingress-gateway config map

Copyright © 2019, 2025, Oracle and/or its affiliates.

In the ocsepp_custom_values.yaml file, under n32-ingress-gateway section, set the requestTimeout parameter value to 5000 (ms) for updating the config map.

```
   routesConfig:
    - id: n32f
      #Below field is used to provide an option to enable/disable route level
xfccHeaderValidation, it will override global configuration for
xfccHeaderValidation.enabled
      metadata:
        requestTimeout: 1200
```

Updating plmn-ingress-gateway config map

In the ocsepp_custom_values.yaml file, under plmn-ingress-gateway section, set the requestTimeout parameter value to 5000 (ms) for updating the config map.

```
     routesConfig:
    - id: cn32f
      #Below field is used to provide an option to enable/disable route level
xfccHeaderValidation, it will override global configuration for
xfccHeaderValidation.enabled
      metadata:
        requestTimeout: 2600
```

3.6.6.14 Idle Timeout

The Idle timeout must be increased in the given services to avoid any pending request transaction to timeout due to its idle state in the request queue.

In the ocsepp_custom_values_<version>.yaml file, under n32-ingress-gateway section, set the jettyIdleTimeout parameter value to 5000(ms).

```
  #Jetty Idle Timeout Settings (ms)
 jettyIdleTimeout: 5000 #(ms)
```

> **Note**
>
> - If the ATS is not configured, the jettyIdleTimeout value should be 3000 (ms).
> - If the ATS cases are failing due to the request timeout, increase idle timeout of a particular service which is failing. For example set idle timeout to "5000".

3.6.6.15 Configuring Polling Interval in N32 Egress Gateway

The pollingInterval parameter overrides polling interval in n32-egress-gateway. As a result of continuous configuration of Remote SEPPs while running ATS, peer, peerset, and routes configurations are constantly being updated in common_configuration for n32-egress-gateway. To minimize route-related failures, the cache for updated peers, peer sets, and routes needs to

Copyright © 2019, 2025, Oracle and/or its affiliates.

be refreshed in the n32-egress-gateway. For this, the polling interval must be updated to 2000ms in the `ocsepp_custom_values_<version>.yaml` file.

```
n32-egress-gateway:
   commonCfgServer:
      pollingInterval: 2000
```

3.6.6.16 EvictSanHeaderCacheDelay

The SEPP supports `evictSanHeaderCacheDelay` to update the cache on the PN32F service with the updated values of Remote SEPP and Remote SEPP set associated with the SAN header.

The user has to set `evictSanHeaderCacheDelay` parameter to 100 in pn32f-svc microservice section of custom yaml file.

```
configs:
    evictSanHeaderCacheDelay: 100 #(ms)
```

> **Note**
>
> If ATS is not configured `evictSanHeaderCacheDelay` must be set to 50000 in the pn32f-svc section of `ocsepp_custom_values_<values>.yaml` file.

3.6.6.17 Configuring the Reroute Attempts for Egress Gateway

SEPP allows configuring the number of reroute attempts for alternate routing. You can configure `attempts` parameter based on the requirement.

Under config manager section in `ocsepp_custom_values_<version>.yaml` file, configure attempts parameter before running ATS test execution. The default value for `attempts` is set to 0.

Set `attempts` to 3 before running ATS test execution.

```
alternateRoute:
    sbiReRoute:
      sbiRoutingErrorActionSets: [{"id": "action_0", "action": "reroute",
"attempts": 3, "blacklist": {"enabled": false, "duration": 60000}}]
```

3.6.6.18 Configuring Alternate Routing based on the DNS SRV Record for Home Network Functions

The Alternate Routing based on the DNS SRV Record for Home Network Functions feature can be enabled as follows:

Update the following parameters in `ocsepp_custom_values.yaml` file to configure the DNS SRV feature:

At Global Section:

Copyright © 2019, 2025, Oracle and/or its affiliates.

```
global:
  alternateRouteServiceEnable : true
```

At Alternate route section:

```
alternate-route:
  global:
     alternateRouteServiceEnable: true
```

Update the Target host:

```
alternate-route:   #Static virtual FQDN Config
   staticVirtualFqdns:
     - name: https://sepp.ats.test.routing.com
       alternateFqdns:
       #Below Target FQDNs needs to be updated for ATS DNS SRV scenarios
       - target: <ats-release-name>-stubserver
         port: 8443
         priority: 100
         weight: 90
       - target: <ats-release-name>-stubserver-2
         port: 8443
         priority: 100
         weight: 10
       - target: <ats-release-name>-stubserver-3
         port: 8443
         priority: 1
         weight: 90
```

Here, update the target parameter with the user-specific ATS release name.

Example: If the user defines the <ats-release-name> as `sepp-ats-rel` and <stub-server-name> as `stubserver`, then update the target as `sepp-ats-rel-stubserver`.

> **Note**
>
> For DNS SRV and SOR features, the plmn-egress-gateway must be deployed in REST mode. In the `ocsepp_custom_values.yaml` file, update:
>
> ```
> plmn-egress-gateway:
>             routeConfigMode: REST
> ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Note

In DNS SRV feature, the rerouting to next peer is decided on the basis of error codes or exceptions. If some exception arises which is not present in the json file `DNSSRVCriteriaSet.json` (may cause scenario failures), user must add those exceptions in the exceptions list at the following paths:

```
In the file /var/lib/jenkins/ocsepp_tests/data/DNSSRVCriteriaSet.json,
update

"exceptions": [
      "java.net.SocketException",
      "java.nio.channels.ClosedChannelException"
    ]


In the file /var/lib/jenkins/ocsepp_tests/cust_data/
DNSSRVCriteriaSet.json, update

"exceptions": [
      "java.net.SocketException",
      "java.nio.channels.ClosedChannelException"
    ]
```

3.6.6.19 Configuring Load Sharing among Multiple Remote SEPP Nodes

The load sharing among multiple Remote SEPP nodes feature can be configured as follows:

1. Enable the following parameters in `ocsepp_custom_values.yaml` file to configure the load sharing feature:
   a. Enable the following flag at alternate_route section:

```
alternate-route:
  global:
     alternateRouteServiceEnable: true
```

   b. Replace the release name of the Target host:

```
 alternate-route:   #Static virtual FQDN Config
   staticVirtualFqdns:
     - name: http://sepp.ats.loadsharing.com
       alternateFqdns:
       #Below Target FQDNs needs to be updated for ATS DNS SRV scenarios
       - target: <ats-release-name>-stubserver
         port: 8443
         priority: 100
         weight: 90
       - target: <ats-release-name>-stubserver-2
         port: 8443
         priority: 100
         weight: 10
       - target: <ats-release-name>-stubserver-3
         port: 8443
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
priority: 1
weight: 90
```

> **Note**
>
> If the user has chosen the ats release name as `sepp-ats-rel`
>
> Modify
>
> `target: <release-name>-stubserver` with actual release name of ATS deployment.
>
> Example: `sepp-ats-rel-stubserver`

2. Enable the following parameter at NrfClient Global parameters:

```
alternateRouteServiceEnable: true
```

3.6.6.20 Configuring Dual Stack

Using the dual stack mechanism, applications or NFs can establish connections with pods and services in a Kubernetes cluster using IPv4 or IPv6 or both simultaneously. Dual stack provides:

- coexistence strategy that allows hosts to reach IPv4 and IPv6 simultaneously.
- IPv4 and IPv6 allocation to the Kubernetes clusters during cluster creation. This allocation is applicable for all the Kubernetes resources unless explicitly specified during the cluster creation.

  To support dual stack functionality in the ATS Helm charts, the following updates are required in the `ocats_ocsepp_values_<version>.yaml` file:

```
global:
  #Deployment mode for ats and stub in dual stack support. Possible
values : IPv4, IPv6, IPv4_IPv6, IPv6_IPv4, ClusterPreferred
  deploymentMode: &deploymentMode IPv4
```

The following are the possible values based on the cluster type:

```
Single Stack cluster: IPv4, IPv6
Dual Stack cluster: IPv4, IPv6, IPv4_IPv6, IPv6_IPv4, ClusterPreferred
```

3.6.6.21 Enabling Persistant Volume Storage

ATS supports Persistent storage to retain ATS historical build execution data, test cases, and one-time environment variable configurations.

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> By default, this feature is disabled.

To enable persistent storage:

1. Create a PVC and associate the same to the ATS pod.
2. Set the **PVEnabled** flag to **true**.
3. Set **PVClaimName** to PVC that is created for ATS.

```
ocats:
  PVEnabled: false
  PVClaimName: "sepp-pvc"
  PVStorageClassName: "standard"
  PVStorage: "1Gi"
  RetainPVC: false
```

For more details on Persistent Volume Storage, you can refer to Persistent Volume for 5G ATS

3.6.6.22 ATS Tagging Support

The ATS tagging support feature assists in running the feature files after filtering features and scenarios based on tags. Instead of manually navigating through several feature files, the user can save time by using this feature.

For more details about tagging support feature, see ATS Tagging Support section.

By default, this feature is enabled in ATS framework but if user wants to enable or disable the feature, set `executionWithTagging` flag to true or false in the `ocats_ocsepp_values_<version>.yaml` file.

```
ocats:
  atsFeatures:
    executionWithTagging: true
```

Feature Level Tags in SEPP Mode

Table 3-25 Feature Level Tags in SEPP Mode

Sl.No.	Tag	Selected Feature Files
1.	MEDIATION	All the mediation feature files will be selected.
2	MULTIPART	All multipart feature files will be selected.
3	ORIGINATING_NETWORK_ID	All originating network id feature files will be selected.
4	ERL	All Egress rate limiting feature files will be selected.
5	TOPOLOGY	All topology feature files will be selected.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-25 (Cont.) Feature Level Tags in SEPP Mode

Sl.No.	Tag	Selected Feature Files
6	PREVIOUS_LOCATION_CHECK	All previous location check feature files will be selected.
7	MESSAGE_VALIDATION	All message validation feature files will be selected.
8	PROACTIVE_STATUS_UPDATE	All proactive status update feature files will be selected.
9	ALTERNATE_ROUTING	All alternate routing feature files will be selected.
10	SOR	All SOR feature files will be selected
11	Handshake	All the feature files which are written for testing handshake functionality will be selected.
12	PORT_SEGREGATION	All port segregation feature files will be selected.
13	DNS_SRV	All DNS SRV files will be selected.
14	NETWORK_ID_VALIDATION	All network id validation feature files will be selected.
15	SERVICE_API_VALIDATION	All service api validation feature files will be selected.
16	IRL	All Ingress rate limiting feature files will be selected.
17	API_PREFIX	All API Prefix feature files will be selected.
18	SI	All system integration feature files will be selected.
19	HEADER_VALIDATION	All header validation feature files will be selected.
20	TIME_LOCATION_CHECK	All Time Location check feature files will be selected.
21	VERBOSITY_VALIDATION	All the feature files which will have verbosity validation scenarios will be selected.
22	ASSERTED_PLMN_ID_METRIC_VALIDATION	SEPP dashboard support for detecting vulnerable messages.

Feature Level Tags in Roaming Hub Mode

Table 3-26 Feature Level Tags in Roaming Hub Mode

Sl.No.	Tag	Selected Feature Files
1	MEDIATION	All the mediation feature files will be selected.
2	TOPOLOGY	All topology feature files will be selected.
3	MESSAGE_VALIDATION	All message validation feature files will be selected.
4	SI	All system integration feature files will be selected.
5	SERVICE_API_VALIDATION	All service API validation feature files will be selected.
6	NETWORK_ID_VALIDATION	All network id validation feature files will be selected.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-26 (Cont.) Feature Level Tags in Roaming Hub Mode

Sl.No.	Tag	Selected Feature Files
7	Handshake	All the feature files which are written for testing handshake functionality will be selected.
8	DNS_SRV	All DNS SRV feature files will be selected.
9	VERBOSITY_VALIDATION	All the feature files which will have verbosity validation scenarios will be selected.
10	ERL	All egress rate limiting feature files will be selected.
11	PROACTIVE_STATUS_UPDATE	All proactive status update feature files will be selected.
12	ALTERNATE_ROUTING	All alternate routing feature files will be selected.
13	HOSTEDSEPP_VALIDATION	All the feature files will be selected which will have Hosted SEPP validation scenarios.
14	PORT_SEGREGATION	All Port Segregation feature files will be selected

Scenario Level Tags in SEPP and Roaming Hub Mode

Table 3-27 Scenario Level Tags in SEPP and Roaming Hub Mode

Sl.No.	Tag	Selected Scenarios
1	sanity	All the scenarios will be selected which are part of sanity testing.

3.6.6.23 Configuring LCI and OCI Header Support

Update the following configuration to run LCI and OCI header test cases:

In `ocsepp_custom_values_<version>.yaml` update the following:

Global section

```
global:
  # Flag to enable LCI & OCI Header support
  #Enabling LCI over N32-Ingress
  seppLciEnabledN32Ingress: &n32IngressLciEnable true
  #Enabling OCI over N32-Ingress
  seppOciEnabledN32Ingress: &n32IngressOciEnable true
  #Enabling LCI over PLMN-Ingress
  seppLciEnabledPlmnIngress: &plmnIngressLciEnable true
  #Enabling OCI over PLMN-Ingress
  seppOciEnabledPlmnIngress: &plmnIngressOciEnable true
```

n32-ingress-gateway section

```
n32-ingress-gateway:
    #oci header configuration at n32-ingress-gateway
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    ociHeaderConfig:
      overloadConfigRange: #Note - minor, major and critical conditions
should be broken down in buckets of range [0 to 100] only both inclusive for
it to be a valid config
        minor: "[0-70]"
```

plmn-ingress-gateway section:

```
plmn-ingress-gateway:
    #oci header configuration at plmn-ingress-gateway
    ociHeaderConfig:
      overloadConfigRange: #Note - minor, major and critical conditions
should be broken down in buckets of range [0 to 100] only both inclusive for
it to be a valid config
        minor: "[0-70]"
```

nrfclient section:

```
nrfclient:
    perf-info:
        configmapPerformance:
            prometheus: http://occne-kube-prom-stack-kube-prometheus.occne-
infra/{cluster_name}:80
```

3.6.7 Deploying ATS in Kubernetes Cluster

> **Note**
>
> It is important to ensure that all the three components; ATS, Stub, and SEPP are in the same namespace.

If the namespace does not exists, run the following command to create a namespace:

```
kubectl create namespace ocsepp
```

Deploying ATS:

```
helm install -name <release_name> ocats-sepp-<version>.tgz --namespace
<namespace_name> -f <values-yaml-file>
```

Example:

```
helm install -name ocats ocats-sepp-<version>.tgz --namespace ocsepp -f ocats-
sepp-custom-values.yaml
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

3.6.8 Verifying ATS Deployment

Run the following command to verify ATS deployment:

```
helm status <release_name>
```

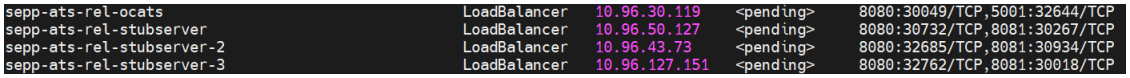
Checking Pod Deployment:

```
kubectl get pod -n seppsvc
```

Checking Service Deployment:

```
kubectl get service -n seppsvc
```

Figure 3-20 Checking Pod and Service Deployment



> **Note**
>
> bddclient name is now updated to ocats.

3.6.9 Post Installation Steps (If persistent volume is used)

If persistent volume is used, follow the post-installation steps as mentioned in the Persistent Volume for 5G ATS section.

3.7 Installing ATS for UDR

Before installing ATS for UDR, it is important to ensure the resource requirements are met. For information about the resource requirements, see Resource Requirements.

3.7.1 Resource Requirements

ATS resource requirements for UDR, SLF, and EIR are as follows:

Table 3-28 UDR - Total Number of Resources

Resource Name	vCPUs	Memory (GB)	Storage (GB)
UDR SUT Totals	38	38	**Ephemeral Storage**: 33 **PVC Storage**: 30

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-28 (Cont.) UDR - Total Number of Resources

Resource Name	vCPUs	Memory (GB)	Storage (GB)
DB Tier Totals	20	41	**Ephemeral Storage**: 0 **PVC Storage**: 37
ATS Totals	17	12	**Ephemeral Storage**: 0 **PVC Storage**: 160
Grand Total UDR-ATS	75	91	**Ephemeral Storage**: 33 **PVC Storage**: 227

Table 3-29 SLF - Total Number of Resources

Resource Name	vCPUs (excluding sidecar)	vCPUs (including sidecar)	Memory (GB) (excluding sidecar)	Memory (including sidecar)	Storage (GB)
SLF-SUT Totals	76	103	78	105	**Ephemeral Storage**: 56 **PVC Storage**: 25
DB Tier Totals	20	41	28	53	**Ephemeral Storage**: 0 **PVC Storage**: 37
ATS Totals	8	11	6	8	**Ephemeral Storage**: 0 **PVC Storage**: 160
Grand Total SLF-ATS	104	155	112	166	**Ephemeral Storage**: 56 **PVC Storage**: 222

Table 3-30 EIR - Total Number of Resources

Resource Name	vCPUs	Memory (GB)	Storage (GB)
EIR SUT Totals	26	26	**Ephemeral Storage**: 20 **PVC Storage**: 30
DB Tier Totals	20	41	**Ephemeral Storage**: 0 **PVC Storage**: 37
ATS Totals	8	7	**Ephemeral Storage**: 0 **PVC Storage**: 160
Grand Total EIR-ATS	54	74	**Ephemeral Storage**: 20 **PVC Storage**: 227

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-31 UDR Operational Resource Requirements

Microservice	vCPUs Required per Pod	Memory Required per Pod (GB)	Ephemeral Storage Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	vCPUs Required - Total	Memory Required - Total (GB)	Ephemeral Storage Required - Total (GB)	PVC Storage (GB)
ocudr-ingressgateway-sig	2	*2*	1 GB	2	1	2	2	1	NA
ocudr-ingressgateway-prov	2	*2*	1 GB	2	1	2	2	1	NA
ocudr-nudr-drservice	1	1	1 GB	2	1	1	1	1	NA
ocudr-nudr-dr-provservice	1	1	1 GB	2	1	1	1	1	NA
ocudr-nudr-notify-service	1	1	1 GB	2	1	1	1	1	NA
ocudr-oc-diam-gateway	1	1	1 GB	2	1	1	1	1	NA
ocudr-nudr-diameterproxy	1	1	1 GB	2	1	1	1	1	NA
ocudr-egressgateway	2	2	1 GB	1	1	2	2	1	NA
ocudr-nudr-config	1	1	1 GB	1	1	1	1	1	NA
ocudr-nudr-config-server	1	1	1 GB	1	1	1	1	1	NA
ocudr-nudr-nrf-client-nfmanagement	1	1	1 GB	1	1	1	1	1	NA
ocudr-appinfo	0.5	1	1 GB	1	1	0.5	1	1	NA
ocudr-alternate-route	1	1	1 GB	1	1	1	1	1	NA

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-31 (Cont.) UDR Operational Resource Requirements

Microservice	vCPUs Required per Pod	Memory Required per Pod (GB)	Ephemeral Storage Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	vCPUs Required - Total	Memory Required - Total (GB)	Ephemeral Storage Required - Total (GB)	PVC Storage (GB)
ocudr-nudr-migration	1	1	1 GB	1	1	1	1	1	NA
ocudr-nudr-bulk-import	1	1	1 GB	1	1	1	1	1	20
ocudr-nudr-ondemand-migration	1	1	1 GB	1	1	1	1	1	NA
ocudr-performance	1	1	1 GB	1	1	1	1	1	NA
ocudr-export-tool	1	1	1 GB	1	1	1	1	1	10
UDR Additional Resources (Hooks/ Init/ Update Containers)	-	-	-	-	-	6	6	6	-
provgw-prov-ingressgateway	2	2	0	2	1	2	2	1	NA
provgw-provgw-service	1	1	0	2	1	1	1	1	NA
provgw-provgw-config	1	1	0	1	1	1	1	1	NA
provgw-prov-egressgateway	2	2	0	2	1	2	2	1	NA

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-31 (Cont.) UDR Operational Resource Requirements

Microservice	vCPUs Required per Pod	Memory Required per Pod (GB)	Ephemeral Storage Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	vCPUs Required - Total	Memory Required - Total (GB)	Ephemeral Storage Required - Total (GB)	PVC Storage (GB)
Provgw Additional Resources (Hooks/ Init/ Update Containers)	-	-	-	-	-	5	5	5	NA
UDR-SUT Total (UDR and ProvGw)	-	-	-	-	-	38	38	33	30

Table 3-32 SLF Operational Resource Requirements

Microservice	vCPUs Required per Pod (excluding sidecar)	vCPUs required for sidecar container per pod	Memory Required per Pod (GB) (excluing sidecar)	Memory required for sidecar container per pod (GB)	Ephemeral Storage Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	vCPUs Required - Total (excludes sidecar)	vCPUs Required - Total (includes sidecar)	Memory Required - Total (GB) (excludes sidecar)	Memory Required - Total (GB) (includes sidecar)	Ephemeral Storage Required - Total (GB)	PVC Required Total (GB)
ocudr-ingressgateway-sig	2	0.5	2	0.5	1 GB	2	1	2	2.5	2	2.5	1	NA
ocudr-ingressgateway-prov	2	0.5	2	0.5	1 GB	2	1	2	2.5	2	2.5	1	NA
ocudr-nudr-drservice	1	0.5	1	0.5	1 GB	2	1	1	1.5	1	1.5	1	NA

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-32 (Cont.) SLF Operational Resource Requirements

Microservice	vCPUs Required per Pod (excluding sidecar)	vCPUs required for sidecar container per pod	Memory Required per Pod (GB) (excluing sidecar)	Memory required for sidecar container per pod (GB)	Ephemeral Storage Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	vCPUs Required - Total (excludes sidecar)	vCPUs Required - Total (includes sidecar)	Memory Required - Total (GB) (excludes sidecar)	Memory Required - Total (GB) (includes sidecar)	Ephemeral Storage Required - Total (GB)	PVC Required Total (GB)
ocudr-nudr-dr-provservice	1	0.5	1	0.5	1 GB	2	1	1	1.5	1	1.5	1	NA
ocudr-egressgateway	2	0.5	2	0.5	1 GB	2	1	2	2.5	2	2.5	1	NA
ocudr-nudr-config	1	0.5	1	0.5	1 GB	1	1	1	1.5	1	1.5	1	NA
ocudr-nudr-config-server	1	0.5	1	0.5	1 GB	1	1	1	1.5	1	1.5	1	NA
ocudr-nudr-nrf-client-nfmanagement	1	0.5	1	0.5	1 GB	1	1	1	1.5	1	1.5	1	NA
ocudr-appinfo	0.5	0.5	1	0.5	1 GB	1	1	0.5	1	1	1.5	1	NA
ocudr-alternate-route	2	0.5	2	0.5	1 GB	1	1	2	2.5	2	2.5	1	NA
ocudr-nudr-bulk-import	1	0.5	1	0.5	1 GB	1	1	1	1.5	1	1.5	1	10Gi

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-32 (Cont.) SLF Operational Resource Requirements

Microservice	vCPUs Required per Pod (excluding sidecar)	vCPUs required for sidecar container per pod	Memory Required per Pod (GB) (excluing sidecar)	Memory required for sidecar container per pod (GB)	Ephemeral Storage Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	vCPUs Required - Total (excludes sidecar)	vCPUs Required - Total (includes sidecar)	Memory Required - Total (GB) (excludes sidecar)	Memory Required - Total (GB) (includes sidecar)	Ephemeral Storage Required - Total (GB)	PVC Required Total (GB)
ocudr-performance	1	0.5	1	0.5	1 GB	1	1	1	1.5	1	1.5	1	NA
ocudr-nudr-export-tool	2	1	2	1	3 GB	1	1	2	3	2	3	3	10Gi
SLF Additional Resources (Hooks/Init/Update Containers)	-	-	-	-	-	-	-	6	7	6	7	3	NA
provgw-prov-ingressgateway	2	0.5	2	0.5	1 GB	2	1	2	2.5	2	2.5	1	NA
provgw-prov-egressgateway	2	0.5	2	0.5	1 GB	2	1	2	2.5	2	2.5	1	NA
provgw-provgw-service	1	0.5	1	0.5	1 GB	2	1	1	1.5	1	1.5	1	NA
provgw-provgw-config	1	0.5	1	0.5	1 GB	1	1	1	1.5	1	1.5	1	NA

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-32 (Cont.) SLF Operational Resource Requirements

Microservice	vCPUs Required per Pod (excluding sidecar)	vCPUs required for sidecar container per pod	Memory Required per Pod (GB) (excluing sidecar)	Memory required for sidecar container per pod (GB)	Ephemeral Storage Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	vCPUs Required - Total (excludes sidecar)	vCPUs Required - Total (includes sidecar)	Memory Required - Total (GB) (excludes sidecar)	Memory Required - Total (GB) (includes sidecar)	Ephemeral Storage Required - Total (GB)	PVC Required Total (GB)
provgw-provgw-config-server	1	0.5	1	0.5	1 GB	1	1	1	1.5	1	1.5	1	NA
provgw-auditor-service	1	0.5	*1*	0.5	1 GB	1	1	1	1.5	1	1.5	1	5Gi
Provgw Additional Resources (Hooks/Init/Update Containers)	-	-	-	-	-	-	-	5	6	5	6	3	NA
SLF-SUT Total Required (SLFs and ProvGw)	-	-	-	-	-	-	-	76	103	78	105	56	25

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-33 EIR Operational Resource Requirements

Microservice	vCPUs Required per Pod	Memory Required per Pod (GB)	Ephemeral Storage Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	vCPUs Required - Total	Memory Required - Total (GB)	Ephemeral Storage Required - Total (GB)	PVC Required Total (GB)
ocudr-ingressgateway-sig	2	2	1 GB	2	1	2	2	1	NA
ocudr-ingressgateway-prov	2	2	1 GB	2	1	2	2	1	NA
ocudr-nudr-drservice	1	1	1 GB	2	1	1	1	1	NA
ocudr-nudr-dr-provservice	1	1	1 GB	2	1	1	1	1	NA
ocudr-oc-diam-gateway	1	1	1 GB	2	1	1	1	1	NA
ocudr-nudr-diameterproxy	1	1	1 GB	2	1	1	1	1	NA
ocudr-egressgateway	2	2	1 GB	2	1	2	2	1	NA
ocudr-nudr-config	1	1	1 GB	1	1	1	1	1	NA
ocudr-nudr-config-server	1	1	1 GB	1	1	1	1	1	NA
ocudr-nudr-nrf-client-nfmanagement	1	1	1 GB	1	1	1	1	1	NA
ocudr-appinfo	0.5	1	1 GB	1	1	0.5	1	1	NA
ocudr-alternate-route	2	2	1 GB	1	1	2	2	1	NA
ocudr-nudr-bulk-import	1	1	1 GB	1	1	1	1	1	20

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-33 (Cont.) EIR Operational Resource Requirements

Microservice	vCPUs Required per Pod	Memory Required per Pod (GB)	Ephemeral Storage Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	vCPUs Required - Total	Memory Required - Total (GB)	Ephemeral Storage Required - Total (GB)	PVC Required Total (GB)
ocudr-performance	1	1	1 GB	1	1	1	1	1	NA
ocudr-nudr-export-tool	2	2	3 GB	1	1	2	2	3	10
EIR Additional Resources (Hooks/ Init/ Update Containers)	-	-	-	-	-	6	6	3	NA
EIR-SUT Total Required	-	-	-	-	-	26	26	20	30

Table 3-34 ATS Resource Requirements for UDR mode

Microservice	vCPUs Required per Pod	Memory Required per Pod (GB)	Storage PVC Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	vCPUs Required - Total	Memory Required - Total (GB)	Storage PVC Required - Total (GB)
ATS Behave	4	4	0	-	1	4	4	160
NRF-Stub (python stub)	2	1	0	-	2 (Two separate deployments)	4	2	0
Notify-Stub (python stub)	2	1	0	-	3 (Three separate deployments)	6	3	0
diam-stub	1	1	0	-	2 (Two separate deployments)	2	2	0
fourg-stub	1	1	0	-	1	1	1	0
ATS Totals	-	-	-	-	-	17	12	160

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-35 ATS Resource Requirements for SLF Mode

Microservice	vCPUs Required per Pod (excluding sidecar)	vCPUs required for sidecar container per pod	Memory Required per Pod (GB) (excluing sidecar)	Memory required for sidecar container per pod(GB)	Ephemeral Storage Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	vCPUs Required - Total (excludes sidecar)	vCPUs Required - Total (includes sidecar)	Memory Required - Total (GB) (excludes sidecar)	Memory Required - Total (GB) (includes sidecar)	Storage PVC Required - Total (GB)
ocats-udr	4	2	4	1	0	1	1	4	6	4	5	160
nrf-ocstub-python	2	.5	1	.5	0	1	2 (Two separate deployments)	4	5	2	3	0
ATS Total required	-	-	-	-	-	-	-	8	11	6	8	160

Table 3-36 ATS Resource Requirements for EIR mode

Microservice	vCPUs Required per Pod	Memory Required per Pod (GB)	Storage PVC Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	vCPUs Required - Total	Memory Required - Total (GB)	Storage PVC Required - Total (GB)
ocats-udr	3	4	0	1	1	3	4	160
nrf-ocstub-python (For deployments)	2	1	0	1	2 (Two separate deployments)	4	2	0
diam-stub	1	1	0	-	1	1	1	0
ATS Total required	-	-	-	-	-	7	6	160

Note

ATS total resource calculation includes 2 nrf-stub deployments, 1 diam-stub deployment, and 1 ATS deployment.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-37 UDR cnDBTier Pods

Micro service name	Container name	Number of Pods	vCPUs Requirement Per Pod	Memory Requirement Per Pod	PVC Requirement	Total Resources
Management node	mysqlndbcluster	2	2 CPUs	5 GB	3*2 → 6 GB	4 CPUs 10 GB Memory
Data node	mysqlndbcluster	2	2 CPUs	4 GB	8*2 → 16 GB	5 CPUs 9 GB Memory
	db-backup-executor-svc		100m CPU	128 MB		
APP SQL node	mysqlndbcluster	2	2 CPUs	5 GB	3*2 → 6 GB	5 CPUs 10.5 GB Memory
	init-sidecar		100m CPU	256 MB		
SQL node (Used for Replication)	mysqlndbcluster	2	2 CPUs	5 GB	3*2 → 6 GB	5 CPUs 10.5 GB Memory
	init-sidecar		100m CPU	256 MB		
DB Monitor Service	db-monitor-svc	1	200m CPUs	500 MB	-	0.5 CPUs 0.5 GB Memory
DB Backup Manager Service	replication-svc	1	200m CPU	500 MB	-	0.5 CPUs 0.5 GB Memory
Total	-	-	-	-	34+3 (Replication Svc) → 37 GB	20 CPUs 41 GB Memory

Table 3-38 SLF cnDBTier Pods

Micro service name	Container name	Number of Pods	vCPUs Requirement Per Pod	Memory Requirement Per Pod	PVC Requirement	Total Resources
Management node	mysqlndbcluster	2	2 CPUs	5 GB	3*2 → 6 GB	8 CPUs 12 GB Memory
	istio-proxy		2 CPUs	1 GB		
Data node	mysqlndbcluster	2	2 CPUs	4 GB	8*2 → 16 GB	9 CPUs 11 GB Memory
	istio-proxy		2 CPUs	1 GB		
	db-backup-executor-svc		100m CPU	128 MB		
APP SQL node	mysqlndbcluster	2	2 CPUs	5 GB	3*2 → 6 GB	9 CPUs 13 GB Memory
	init-sidecar		100m CPU	256 MB		
	istio-proxy		2 CPUs	1 GB		
SQL node (Used for Replication)	mysqlndbcluster	2	2 CPUs	5 GB	3*2 → 6 GB	9 CPUs 13 GB Memory
	init-sidecar		100m CPU	256 MB		
	istio-proxy		2 CPUs	1 GB		

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-38 (Cont.) SLF cnDBTier Pods

Micro service name	Container name	Number of Pods	vCPUs Requirement Per Pod	Memory Requirement Per Pod	PVC Requirement	Total Resources
DB Monitor Service	db-monitor-svc	1	200m CPUs	500 MB	-	3 CPUs 2 GB Memory
	istio-proxy		2 CPUs	1 GB		
DB Backup Manager Service	replication-svc	1	200m CPU	500 MB	-	3 CPUs 2 GB Memory
	istio-proxy		2 CPUs	1 GB		
Total	-	-	-	-	34+3 (Replication Svc) → 37 GB	41 CPUs 53 GB Memory

Table 3-39 EIR cnDBTier Pods

Micro service name	Container name	Number of Pods	vCPUs Requirement Per Pod	Memory Requirement Per Pod	PVC Requirement	Total Resources
Management node	mysqlndbcluster	2	2 CPUs	5 GB	3*2 → 6 GB	4 CPUs 10 GB Memory
Data node	mysqlndbcluster	2	2 CPUs	4 GB	8*2 → 16 GB	5 CPUs 9 GB Memory
	db-backup-executor-svc		100m CPU	128 MB		
APP SQL node	mysqlndbcluster	2	2 CPUs	5 GB	3*2 → 6 GB	5 CPUs 10.5 GB Memory
	init-sidecar		100m CPU	256 MB		
SQL node (Used for Replication)	mysqlndbcluster	2	2 CPUs	5 GB	3*2 → 6 GB	5 CPUs 10.5 GB Memory
	init-sidecar		100m CPU	256 MB		
DB Monitor Service	db-monitor-svc	1	200m CPUs	500 MB	-	1 CPUs 1 GB Memory
DB Backup Manager Service	replication-svc	1	200m CPU	500 MB	-	1 CPUs 1 GB Memory
Total	-	-	-	-	34+3 (Replication Svc) → 37 GB	20 CPUs 41 GB Memory

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-40 SLF Performance based Resource Details

-	vCPUs (excluding sidecar)	vCPUs (including sidecar)	Memory (GB) (excluding sidecar)	Memory (GB) (including sidecar)	Ephemeral Storage (GB)	PVC Storage (GB)	-	-	-	-	-	-
SLF-SUT Totals	59	105	47	93	21	0	-	-	-	-	-	-
DB Tier Totals	40	80	40	80	0	37	-	-	-	-	-	-
ATS Totals	6	4	1	3	0	0	-	-	-	-	-	-
Grand Total SLF-ATS	105	193	95	183	21	37	-	-	-	-	-	-
Microservice	**CPUs Required per Pod (excluding sidecar)**	**CPUs required for sidecar container**	**Memory Required per Pod (GB) (excluding sidecar)**	**Memory required for sidecar container per pod (GB)**	**Ephemeral Storage Required per Pod (GB)**	**Replicas (regular deployment)**	**Replicas (ATS deployment)**	**CPUs Required - Total (excluding sidecar)**	**CPUs Required - Total (including sidecar)**	**Memory Required - Total (GB) (excluding sidecar)**	**Memory Required - Total (GB) (including sidecar)**	**Ephemeral Storage Required - Total (GB)**
ocudr-ingressgateway-sig	6	4	4	4	1 GB	2	2	12	20	8	16	2
ocudr-ingressgateway-prov	6	4	4	4	1 GB	2	2	12	20	8	16	2
ocudr-nudr-drservice	5	4	4	4	1 GB	2	2	10	18	8	16	2
ocudr-nudr-dr-provservice	5	4	4	4	1 GB	2	2	10	18	8	16	2
ocudr-egressgateway	1	2	1	2	1 GB	1	1	1	3	1	3	1

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-40 (Cont.) SLF Performance based Resource Details

-	vCPUs (excluding sidecar)	vCPUs (including sidecar)	Memory (GB) (excluding sidecar)	Memory (GB) (including sidecar)	Ephemeral Storage (GB)	PVC Storage (GB)	-	-	-	-	-	-
ocudr-nudr-config	1	1	1	1	1 GB	1	1	1	2	1	2	1
ocudr-nudr-config-server	1	1	1	1	1 GB	1	1	1	2	1	2	1
ocudr-nudr-nrf-client-nfmanagement	1	1	1	1	1 GB	1	1	1	2	1	2	1
ocudr-appinfo	1	1	1	1	1 GB	1	1	1	2	1	2	1
ocudr-alternate-route	1	1	1	1	1 GB	1	1	1	2	1	2	1
ocudr-performance	1	1	1	1	1 GB	1	1	1	2	1	2	1
Additional SLF Resources (Hooks/Init/Update Containers)	-	-	-	-	-	-	-	6	12	6	12	6
SLF SUT Totals	-	-	-	-	-	-	-	59	105	47	93	21
Microservice	Container	Number of Pods	CPU Requirement Per Pod	Memory Requirement Per Pod	PVC Requirement	Total Resources	-	-	-	-	-	-

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-40 (Cont.) SLF Performance based Resource Details

-	vCPUs (excluding sidecar)	vCPUs (including sidecar)	Memory (GB) (excluding sidecar)	Memory (GB) (including sidecar)	Ephemeral Storage (GB)	PVC Storage (GB)	-	-	-	-	-	-
Management node	mysqlndbcluster	2	2 CPUs	5 GB	3*2 → 6 GB	8 CPUs 12 GB Memory	-	-	-	-	-	-
	istio-proxy		2 CPUs	1 GB								
Data node	mysqlndbcluster	2	2 CPUs	4 GB	8*2 → 16 GB	9 CPUs 11 GB Memory	-	-	-	-	-	-
	istio-proxy		2 CPUs	1 GB								
	db-backup-executor-svc		100m CPU	128 MB								
APP SQL node	mysqlndbcluster	2	2 CPUs	5 GB	3*2 → 6 GB	9 CPUs 13 GB Memory	-	-	-	-	-	-
	init-sidecar		100m CPU	256 MB								
	istio-proxy		2 CPUs	1 GB								
SQL node (Used for Replication)	mysqlndbcluster	2	2 CPUs	5 GB	3*2 → 6 GB	9 CPUs 13 GB Memory	-	-	-	-	-	-
	init-sidecar		100m CPU	256 MB								
	istio-proxy		2 CPUs	1 GB								
DB Monitor Service	db-monitor-svc	1	200m CPUs	1 GB	-	3 CPUs 2 GB Memory	-	-	-	-	-	-
	istio-proxy		2 CPUs	500 MB								
DB Backup Manager Service	replication-svc	1	200m CPU	500 MB	-	3 CPUs 2 GB Memory	-	-	-	-	-	-
	istio-proxy		2 CPUs	1 GB								

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 3-40 (Cont.) SLF Performance based Resource Details

-	vCPUs (excluding sidecar)	vCPUs (including sidecar)	Memory (GB) (excluding sidecar)	Memory (GB) (including sidecar)	Ephemeral Storage (GB)	PVC Storage (GB)	-	-	-	-	-	-
Total	-	-	-	-	-	-	-	-	-	-	34+3(Replication Svc) → 37 GB	41 CPUs 53 GB Memory

Table 3-41 ATS and stub requirement for SLF Performance

Microservice	vCPUs Required per Pod (excluding sidecar)	vCPUs required for sidecar container per pod	Memory Required per Pod (GB) (exclusing sidecar)	Memory required for sidecar container per pod(GB)	Storage PVC Required per Pod (GB)	Replicas (regular deployment)	Replicas (ATS deployment)	vCPUs Required - Total (excludes sidecar)	vCPUs Required - Total (includes sidecar)	Memory Required - Total (GB) (includes sidecar)	Memory Required - Total (GB) (includes sidecar)	Storage PVC Required - Total (GB)
ocats-udr	6	2	8	2	0	1	1	6	8	8	10	0
ATS Total required	-	-	-	-	-	-		6	8	8	10	0

3.7.2 Locating and Downloading ATS Image

To locate and download the ATS Image from MOS:

1. Log in to My Oracle Support using appropriate credentials.
2. Select the **Patches & Updates** tab.
3. In the **Patch Search** window, click **Product or Family (Advanced)**.
4. Enter *Oracle Communications Cloud Native Core - 5G* in the **Product** field.
5. Select *Oracle Communications Cloud Native Core Unified Data Repository <release_number>* from the **Release** drop-down.
6. Click **Search**. The **Patch Advanced Search Results** list appears.
7. Select the required ATS patch from the list. The **Patch Details** window appears.
8. Click **Download**. The **File Download** window appears.
9. Click the **<p********_<release_number>_Tekelec>.zip** file.

Copyright © 2019, 2025, Oracle and/or its affiliates.

10. Untar the zip file to access all the ATS images. The `ocats_udr_pkg_25_1_200_0_0.tgz` file has following files:
    - ocats_udr_pkg_25_1_200_0_0.tgz
    - ocats_udr_pkg_25_1_200_0_0-README.txt
    - ocats_udr_pkg_25_1_200_0_0.tgz.sha256
    - ocats-udr-custom-configtemplates-25.1.200.0.0.zip
    - ocats-udr-custom-configtemplates-25.1.200.0.0-README.txt
    - ocats_udr_pkg_25_1_200_0_0.mcafee.sw_packaging.log

    The ocats_udr_pkg_25_1_200_0_0.tgz file contains:

```
|_ _ ocats-udr-ats-pkg-25.1.200.tgz

|      |_ _ _ _ _ _ ocats-udr-25.1.200.tgz (Helm Charts)

|      |_ _ _ _ _ _ ocats-udr-images-25.1.200.tar(Docker Images)

|      |_ _ _ _ _ _ OCATS-UDR-Readme.txt

|      |_ _ _ _ _ _ ocats-udr-25.1.200.tgz.sha256

|      |_ _ _ _ _ _ ocats-udr-images-25.1.200.tar.sha256

|      |_ _ _ _ _ _ _ ocats-udr-data-25.1.200.tgz

|      |_ _ _ _ _ _ _ ocats-udr-data-25.1.200.tgz.sha256

|_ _ ocats-udr-stub-pkg-25.1.200.tgz

     |_ _ _ _ _ _ ocstub-py-25.1.202.tgz (Helm Charts)

     |_ _ _ _ _ _ fourg-stub-25.1.200.tgz (Helm Charts)

     |_ _ _ _ _ _ diam-stub-25.1.200.tgz (Helm Charts)

     |_ _ _ _ _ _ ocstub-py-25.1.202.tar(Docker Images)

     |_ _ _ _ _ _ ocats-udr-fourg-stub-images-25.1.200.tar(Docker Images)

     |_ _ _ _ _ _ ocats-udr-diam-stub-images-25.1.200.tar(Docker Images)

     |_ _ _ _ _ _ OCATS-UDR-STUB-Readme.txt

     |_ _ _ _ _ _ ocstub-py-25.1.202.tgz.sha256

     |_ _ _ _ _ _ fourg-stub-25.1.200.tgz.sha256

     |_ _ _ _ _ _ diam-stub-25.1.200.tgz.sha256

     |_ _ _ _ _ _ ocstub-py-25.1.202.tar.sha256

     |_ _ _ _ _ _ ocats-udr-fourg-stub-images-25.1.200.tar.sha256
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
      |_ _ _ _ _ _ ocats-udr-diam-stub-images-25.1.200.tar.sha256
```

The ocats-udr-custom-configtemplates-25.1.200.0.0.zip file contains:

```
ocats-udr-custom-configtemplates-25.1.200.0.0.zip

      |_ _ _ _ _ _ ocats-udr-custom-values-25.1.200.yaml(Custom values for
UDR-ATS)

      |_ _ _ _ _ _ ocstub-py-custom-values-25.1.202.yaml (Custom values
for COMMON-PYTHON-STUB)

      |_ _ _ _ _ _ fourg-stub-custom-values-25.1.200.yaml (Custom values
for FOURG-STUB)

      |_ _ _ _ _ _ diam-stub-custom-values-25.1.200.yaml(Custom values for
DIAMETER-STUB)
```

Copy the ocats_udr_pkg_25_1_200_0_0.tgz tar file and ocats-udr-custom-configtemplates-25.1.200.0.0.zip file to OCI or Kubernetes cluster, where you want to deploy ATS.

3.7.3 Preparing to Deploy ATS and Stub Pods

To deploy ATS and stub pods in Kubernetes cluster:

> **Note**
>
> Deploy ATS and Subscriber Location Function (SLF) in the same namespace.

- Loading UDR ATS Images
- Loading UDR Stub Images for SLF, UDR, and EIR:
  - SLF-NewFeatures or SLF-Regression pipeline
  - UDR-NewFeatures or UDR-Regression
  - Loading EIR Stub Images in the EIR-NewFeatures or EIR-Regression Pipeline

3.7.4 Loading UDR ATS Image

To load UDR ATS image:

1. Run the following command to extract the tar file content.
   ```
   tar -xvf ocats_udr_pkg_25_1_200_0_0.tgz
   ```
   The output of this command is:
   - ocats-udr-ats-pkg-25.1.200.tgz
   - ocats-udr-stub-pkg-25.1.200.tgz
2. Run the following command to extract the helm charts and docker images of ATS.
   ```
   tar -xvf ocats-udr-ats-pkg-25.1.200.tgz
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

The output of this command is:

- ocats-udr-25.1.200.tgz
- ocats-udr-images-25.1.200.tar
- ocats-udr-data-25.1.200.tgz

3. Run the following command to extract the helm charts and docker images of ATS.
   `tar -xvf ocats-udr-stub-pkg-25.1.200.tgz`

   The output of this command is:

   - ocstub-py-25.1.202.tgz (Helm charts)
   - fourg-stub-25.1.200.tgz (Helm charts)
   - diam-stub-25.1.200.tgz (Helm charts)
   - ocats-udr-notify-stub-images-25.1.200.tar (Docker image)
   - ocats-udr-fourg-stub-images-25.1.200.tar(Docker Images
   - ocats-udr-diam-stub-images-25.1.200.tar(Docker Images)
   - ocstub-py-25.1.202.tar (Docker image)

   The `ocats-udr-images-25.1.200.tar` file contains docker images (ocats-udr-images-25.1.200.tar) of ATS for UDR 25.1.200.

4. Run the following command in your cluster to load the ATS image.

   ```
   docker load --input ocats-udr-images-25.1.200.tar
   ```

   > **Note**
   >
   > For CNE 1.8.0 and above, you can use Podman instead of Docker. See the following sample Podman command:
   > `sudo podman load --input ocats-udr-images-25.1.200.tar`

5. Run the following commands to tag and push the ATS image to your registry.

   ```
   docker tag ocats-udr-images:25.1.200 <registry>/ocats-udr-images:25.1.200
   ```

   ```
   docker push <registry>/ocats-udr-images:25.1.200
   ```

   In the previous command, <registry> is the name of docker image repository.

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> For CNE 1.8.0 and above, you can use Podman instead of Docker to tag and push the Docker image. Run the following sample Podman command to tag and push Docker image:
>
> ```
> sudo podman tag ocats-udr-images:25.1.200 <customer repo>/ <image
> name>:<image version>
>
>
> sudo podman push <customer repo>/<image name>:<image version>
> ```

6. Run the following command to untar the helm charts (ocats-udr-25.1.200.tgz) and update the registry name, image name, and tag (if required) in the ocats-udr-custom-values-25.1.200.yaml file.

   ```
   tar -xvf ocats-udr-25.1.200.tgz
   ```

   Output:

   ```
   ocats-udr/Chart.yaml

   ocats-udr/values.yaml

   ocats-udr/templates/NOTES.txt

   ocats-udr/templates/_helpers.tpl

   ocats-udr/templates/deployment.yaml

   ocats-udr/templates/ingress.yaml

   ocats-udr/templates/service.yaml

   ocats-udr/templates/serviceaccount.yaml
   ```

   Output:

   ```
   ocats-udr
   ├── Chart.yaml
   ├── templates
   │   ├── deployment.yaml
   │   ├── _helpers.tpl
   │   ├── ingress.yaml
   │   ├── NOTES.txt
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
│   ├── serviceaccount.yaml
│   └── service.yaml
└── values.yaml
```

3.7.5 Loading UDR Stub Images in the SLF-NewFeatures or SLF-Regression Pipeline

To load the UDR stub images:

> **Note**
>
> For the SLF-NewFeatures or SLF-Regression pipeline, deploy only ocstub-py Helm chart. The ocstub-py-25.1.202.tar contains common-python-stub image (ocstub-py:25.1.202).

1. Run the following command to extract the tar file content.
   `tar -xvf ocats_udr_pkg_25_1_200_0_0.tgz`

   The output of this command is:
   - ocats-udr-ats-pkg-25.1.200.tgz
   - ocats-udr-stub-pkg-25.1.200.tgz
2. Run the following command to extract the stub tar file content.
   `tar -xvf ocats-udr-stub-pkg-25.1.200.tgz`

   The output of this command is:
   - ocstub-py-25.1.202.tgz (Helm charts)
   - fourg-stub-25.1.200.tgz (Helm charts)
   - diam-stub-25.1.200.tgz (Helm charts)
   - ocats-udr-notify-stub-images-25.1.200.tar (Docker image)
   - ocats-udr-fourg-stub-images-25.1.200.tar(Docker Images)
   - ocats-udr-diam-stub-images-25.1.200.tar(Docker Images)
   - ocstub-py-25.1.202.tar (Docker image)
3. To load the UDR stub images (ocstub-py-25.1.202.tar) in the SLF-NewFeatures and SLF-Regression pipeline, run the following command in your cluster.

   ```
   docker load --input ocstub-py-25.1.202.tar
   ```
4. Run the following commands to tag and push the stub image to your registry.

   ```
   docker tag ocstub-py:25.1.202 <registry>/ocstub-py:25.1.202
   ```

   ```
   docker push <registry>/ocstub-py:25.1.202
   ```

   In the previous command, <registry> is the name of docker image repository.

Copyright © 2019, 2025, Oracle and/or its affiliates.

5. Run the following command to untar the common python stub helm charts (ocstub-py-25.1.202.tgz) to get ocstub-py charts:

```
tar -xvf ocstub-py-25.1.202.tgz
```

Output:

```
ocstub-py/Chart.yaml
ocstub-py/values.yaml
ocstub-py/templates/_helpers.tpl
ocstub-py/templates/deployment.yaml
ocstub-py/templates/ingress.yaml
ocstub-py/templates/service.yaml
ocstub-py/templates/serviceaccount.yaml
ocstub-py/README.md
```

Output:

```
ocstub-py
├── Chart.yaml
├── templates
│   ├── deployment.yaml
│   ├── _helpers.tpl
│   ├── ingress.yaml
│   ├── serviceaccount.yaml
│   └── service.yaml
├── values.yaml
└── README.md
```

3.7.6 Loading UDR Stub Images in the UDR-NewFeatures or UDR-Regression Pipeline

> **Note**
>
> - To run UDR-NewFeatures/UDR-Regression, ocstub-py, fourg-stub, and diam-stub must be deployed
> - ocstub-py-25.1.202.tar contains common-python-stub image (ocstub-py:25.1.202)
> - ocats-udr-fourg-stub-images-25.1.200.tar contains fourg-stub image (ocats-udr-fourg-stub-images:25.1.200)
> - ocats-udr-diam-stub-images-25.1.200.tar contains diam-stub image (ocats-udr-diam-stub-images:25.1.200)

Copyright © 2019, 2025, Oracle and/or its affiliates.

To load the UDR Stub Images:

1. Run the following command to extract the tar file content.
   ```
   tar -xvf ocats-udr-ats-pkg-25.1.200.tgz
   ```
   The output of this command is:
   - ocats-udr-ats-pkg-25.1.200.tgz
   - ocats-udr-stub-pkg-25.1.200.tgz
2. Run the following command to extract the stub tar file content.
   ```
   tar -xvf ocats-udr-stub-pkg-25.1.200.tgz
   ```
   - ocstub-py-25.1.202.tgz (Helm charts)
   - fourg-stub-25.1.200.tgz (Helm charts)
   - diam-stub-25.1.200.tgz (Helm charts)
   - ocats-udr-notify-stub-images-25.1.200.tar (Docker image)
   - ocats-udr-fourg-stub-images-25.1.200.tar(Docker Images)
   - ocats-udr-diam-stub-images-25.1.200.tar(Docker Images)
   - ocstub-py-25.1.202.tar (Docker image)
3. Run the following commands in your cluster to load the stub images.

   ```
   docker load --input ocstub-py-25.1.202.tar
       docker load --input ocats-udr-fourg-stub-images-25.1.200.tar
       docker load --input ocats-udr-diam-stub-images-25.1.200.tar
   ```

4. Run the following commands to tag and push the stub image to your registry.

   ```
   docker tag ocstub-py:25.1.202 <registry>/ocstub-py:25.1.202
       docker push <registry>/ocstub-py:25.1.202
       docker tag ocats-udr-fourg-stub-images:25.1.200 <registry>/ocats-udr-
   fourg-stub-images:25.1.200
       docker push <registry>/ocats-udr-fourg-stub-images:25.1.200
       docker tag ocats-udr-diam-stub-images:25.1.200 <registry>/ocats-udr-
   diam-stub-images:25.1.200
       docker push <registry>/ocats-udr-diam-stub-images:25.1.200
   ```

5. Run the following command to untar all the stub charts:

   ```
   tar -xvf ocstub-py-25.1.202.tgz
       tar -xvf fourg-stub-25.1.200.tgz
       tar -xvf diam-stub-25.1.200.tgz
   ```

   Output of each helm chart is as follows:

   ```
   ocstub-py:

   ├── Chart.yaml

   ├── templates

   │   ├── deployment.yaml
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
│   ├── _helpers.tpl
│   ├── ingress.yaml
│   ├── serviceaccount.yaml
│   └── service.yaml
├── values.yaml
└── README.md

fourg-stub
├── Chart.yaml
├── templates
│   ├── deployment.yaml
│   ├── _helpers.tpl
│   ├── ingress.yaml
│   ├── NOTES.txt
│   └── service.yaml
└── values.yaml

diam-stub
├── Chart.yaml
├── templates
│   ├── deployment.yaml
│   ├── _helpers.tpl
│   ├── ingress.yaml
│   ├── NOTES.txt
│   └── service.yaml
└── values.yaml
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

3.7.7 Loading EIR Stub Images in the EIR-NewFeatures or EIR-Regression Pipeline

> **Note**
>
> - To run EIR-NewFeatures/EIR-Regression, ocstub-py, and diam-stub must be deployed
> - ocstub-py-25.1.202.tar contains common-python-stub image (ocstub-py:25.1.202)
> - ocats-udr-diam-stub-images-25.1.200.tar contains diam-stub image (ocats-udr-diam-stub-images:25.1.200)

To load the EIR Stub Images:

1. Run the following commands in your cluster to load the stub images.

   ```
   docker load --input ocstub-py-25.1.202.tar
       docker load --input ocats-udr-diam-stub-images-25.1.200.tar
   ```

2. Run the following commands to tag and push the stub image to your registry.

   ```
       docker tag ocstub-py:25.1.202 <registry>/ocstub-py:25.1.202
       docker push <registry>/pythonstub:25.1.200
       docker tag ocats-udr-diam-stub-images:25.1.200 <registry>/ocats-udr-
   diam-stub-images:25.1.200
       docker push <registry>/ocats-udr-diam-stub-images:25.1.200
   ```

3. Run the following command to untar all the stub charts:

   ```
       tar -xvf ocstub-py-25.1.202.tgz
       tar -xvf diam-stub-25.1.200.tgz
   ```

   Output of each helm chart is as follows:

   ```
   ocstub-py:
   ├── Chart.yaml
   ├── templates
   │   ├── deployment.yaml
   │   ├── _helpers.tpl
   │   ├── ingress.yaml
   │   ├── serviceaccount.yaml
   │   └── service.yaml
   ├── values.yaml
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
└── README.md

diam-stub
├── Chart.yaml
├── templates
│   ├── deployment.yaml
│   ├── _helpers.tpl
│   ├── ingress.yaml
│   ├── NOTES.txt
│   └── service.yaml
└── values.yaml
```

3.7.8 Configuring ATS

It is important to configure the following features before deploying ATS for UDR:

- Configuring Docker Registry
- Enabling Static Port
- Enabling Persistent Volume
- Settings for OAuth2 Test Cases
- Enabling IPv6 on ATS
- Configuring ATS to Run Health-Check Pipeline
- Creating Service Account
- Enabling Service Mesh
- Configuring NRF Stub
- Configuring Notify Stub
- Configuring Fourg Stub
- Configuring Diameter Stub

> **Note**
>
> - The deployment of notify-stub, fourg-stub, and diam-stub are not applicable to SLF pipelines.
> - Service name used by each of the stubs must be unique for successful deployment.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3.7.8.1 Configuring Docker Registry

Update the docker registry file as follows:

```
image:
  repository: <docker registry>:<docker port>/ocats-udr-images
```

3.7.8.2 Enabling Static Port

To enable static port:

> **Note**
>
> ATS supports static port. By default, this feature is not enabled.

In the ocats-udr-custom-values-25.1.200.yaml file under **service** section, set the `staticNodePortEnabled` parameter as `true` and `staticNodePort` parameter with valid nodePort value.

Here is a sample configuration for enabling static port in the ocats-udr-custom-values-25.1.200.yaml.file:

```
service:
  customExtension:
    labels: {}
    annotations: {}
  type: LoadBalancer
  port: "8080"
  staticNodePortEnabled: true
  staticNodePort: "31083"
```

3.7.8.3 Enabling Persistent Volume

To enable persistent volume, create a PVC and associate the same to the ATS pod.

> **Note**
>
> - For information about creating PVC, see Persistent Volume for 5G ATS.
> - For information about PVC size calculation, see UDR Application Log Collection.

To enable persistent volume, set the following parameters in the values.yaml file:

1. Set the **PVEnabled** flag to 'true'.
2. Set **PVClaimName** to 'PVC' that user has created for ATS.

```
deployment:
  customExtension:
    labels: {}
    annotations: {}
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
PVEnabled: true
PVClaimName: "ocats-udr-pvc"
```

3.7.8.4 Settings for OAuth2 Test Cases

UDR-ATS supports test cases that verifies OAuth2 validation scenarios at both UDR Provisioning Ingress Gateway and UDR Signaling Ingress Gateway. To run the OAuth2 related test cases on ATS:

1. Generate four ECDSA private keys in pem format (two each for UDR Provisioning Ingress Gateway and UDR Signaling Ingress Gateway) and enter the keys name in the **privateKey** field.
2. Generate four public certificates using the private keys (two each for UDR Provisioning Ingress Gateway and UDR Signaling Ingress Gateway) and enter the certificate names under **publicKey**.
   Sample Commands to Generate ECDSA Private Keys and Certificates

   ```
   openssl ecparam -genkey -name prime256v1 -noout -out ec_private_key1.pem
   openssl pkcs8 -topk8 -in ec_private_key1.pem -inform pem -out
   ec_private_key_pkcs8.pem -outform pem -nocrypt
   openssl req -new -key ec_private_key_pkcs8.pem -x509 -nodes -days 365 -out
   4bc0c762-0212-416a-bd94-b7f1fb348bd4.crt -subj "/C=IN/ST=KA/L=BLR/O=ORACLE/
   OU=CGBU/CN=ocnrf-endpoint.ocnrf.svc.cluster.local"
   ```

   > **Note**
   >
   > - You can configure the above mentioned inputs only if OAuth2 is configured on UDR. For information to configure OAuth2 on UDR, see the "Configuring OAuth2.0" section in the *Oracle Communications Cloud Native Core Unified Data Repository User Guide*.
   > - ATS configures Ingress Gateway with secret name, keyId, certificate name, and instanceid based on the inputs provided in the ocats-udr-custom-values-25.1.200.yaml file.
   > - ATS supports only ES256 algorithm to generate token for this release. User should generate ECDSA private key to test OAuth2 feature.

3. Update the ocats-udr-custom-values-25.1.200.yaml file with the public and private keys generated in the previous steps. A sample code snippet is as follows:
   Sample: OAuth Configuration

   ```
   deployment:   oauthKeys:
       - keyId: '664b344e74294c8fa5d2e7dfaaaba407'
         udrSecret: 'oauthsecret1'
         privateKey: 'ec_private_key1.pem'
         publicKey: '4bc0c762-0212-416a-bd94-b7f1fb348bd4.crt'
         issuerId: '4bc0c762-0212-416a-bd94-b7f1fb348bd4'
         reqType: 'igw-sig'
       - keyId: '664b344e74294c8fa5d2e7dfaaaba408'
         udrSecret: 'oauthsecret2'
         privateKey: 'ec_private_key2.pem'
         publicKey: '4bc0c762-0212-416a-bd94-b7f1fb348bd5.crt'
         issuerId: '4bc0c762-0212-416a-bd94-b7f1fb348bd5'
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    reqType: 'igw-sig'
  - keyId: '664b344e74294c8fa5d2e7dfaaaba409'
    udrSecret: 'oauthsecret3'
    privateKey: 'ec_private_key3.pem'
    publicKey: '4bc0c762-0212-416a-bd94-b7f1fb348bd6.crt'
    issuerId: '4bc0c762-0212-416a-bd94-b7f1fb348bd6'
    reqType: 'igw-prov'
  - keyId: '664b344e74294c8fa5d2e7dfaaaba410'
    udrSecret: 'oauthsecret4'
    privateKey: 'ec_private_key4.pem'
    publicKey: '4bc0c762-0212-416a-bd94-b7f1fb348bd7.crt'
    issuerId: '4bc0c762-0212-416a-bd94-b7f1fb348bd7'
    reqType: 'igw-prov'
```

In the above code snippet,

- **issuerId** is any uuid that follows NfInstanceId format.
- **keyId** is user-defined value.
- **reqType** indicates the mapping between secret created and Ingress Gateway. igw-sig indicates secrets are to be used for UDR Signaling Ingress Gateway and igw-prov indicates UDR Provisioning Ingress Gateway.

4. Create four secrets in the namespace where ATS and UDR are installed, with public certificate and enter the value for the **udrSecret** field.

3.7.8.5 Enabling IPv6 on ATS

If you are deploying ATS setup on IPv6 system, then enable the following flag in the ocats-udr-custom-values.yaml file:

Enabling IPv6 on ATS

```
deployment:
  ipv6enabled: true
```

3.7.8.6 Configuring ATS to Run Health-Check Pipeline

To run the SLF-Health-Check, UDR-Health-Check, or EIR-HealthCheck features, provide the following inputs in the values.yaml file if ATS is being deployed on OCCNE.

```
deployment:
  Webscale: false
  occnehostip: <base64 encoded occne bastion ip>
  occnehostusername: <base64 encoded occne login user name>
  occnehostpassword: <base64 encoded occne login password>
```

And, provide the following inputs to run the SLF-Health-Check, UDR-Health-Check, or EIR-HealthCheck features, if ATS is being deployed on Webscale.

```
deployment:
  Webscale: true
  #Provide Webscale Environment details with base64 encoding
  webscalejumpserverip: <base64 encoded jump server ip>
  webscalejumpserverusername: <base64 encoded jump server username>
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  webscalejumpserverpassword: <base64 encoded jump server password>
  webscaleprojectname: <base64 encoded webscale project name>
  webscalelabserverFQDN: <base64 encoded lab server fqdn>
  webscalelabserverport: <base64 encoded lab server port>
  webscalelabserverusername: <base64 encoded lab server username>
  webscalelabserverpassword: <base64 encoded lab server password>
```

You can configure the name of the secret that the ATS creates for healthcheck pipeline:

```
healthchecksecretname: ats-healthcheck-secret
```

If CNE supports dual stack feature and UDR, SLF, EIR, and ProvGw are deployed in IPv6_IPv4 mode, then you can enable the dual stack feature on ATS by setting the `deploymentMode` parameter to IPv6_IPv4. This will enable ATS service to have both IPv6 and IPv4 address. IPv6 address will be preferred.

```
deplopymentMode: IPv6_IPv4
```

For more information, see Health Check.

> **Note**
>
> UDR-ATS creates a secret with the name 'healthcheck-secret' on Kubernetes to store the above inputs.

3.7.8.7 Creating Service Account

To run SLF-ATS and EIR-ATS, it is mandatory to create a service account using the following inputs:

```
rules:
- apiGroups: ["apps"]
  resources: ["deployments", "replicasets", "deployments/scale","statefulsets/
scale"]
  verbs: ["get","watch","list","update"]
- apiGroups: [""]
  resources: ["pods", "deployments","pods/log","configmaps","pods/exec"]
  verbs: ["get","watch","list","update","create"]
```

To run UDR-ATS, it is mandatory to create a service account using the following inputs:

```
rules:
- apiGroups: ["apps"]
  resources: ["deployments", "replicasets", "deployments/scale","statefulsets/
scale"]
  verbs: ["get","watch","list","update"]
- apiGroups: [""]
  resources: ["pods", "deployments","pods/log","configmaps","pods/
exec","services"]
  verbs: ["get","watch","list","update","create","delete"]
- apiGroups: ["batch"]
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
resources: ["jobs"]
verbs: ["get","create","delete","update","list"]
```

> **Note**
>
> For information about creating service account, see the *Oracle Communications Cloud Native Core Unified Data Repository Installation and Upgrade Guide* available on MOS.

3.7.8.8 Enabling Service Mesh

> **Note**
>
> The UDR-NewFeatures, UDR-Regression, EIR-NewFeatures, and EIR-Regression pipelines do not support deployment on service mesh enabled system.

To enable service mesh:

1. If service mesh is not enabled at the global level for namespace, then run the following command to enable service mesh at the namespace level before deploying UDR-ATS.
   `kubectl label --overwrite namespace <namespace_name> istio-injection=enabled`

   Example:

   `kubectl label --overwrite namespace ocudr istio-injection=enabled`

2. Add the following annotation in the **lbDeployment** section under global section of the ocats-udr-custom-values-25.1.200.yaml file:

```
global:
  # ********  Sub-Section Start: Custom Extension Global Parameters
********

#************************************************************************

  customExtension:
    allResources:
      labels: {}
      annotations: {}

    lbServices:
      labels: {}
      annotations: {}

    lbDeployments:
      labels: {}
      annotations:
        traffic.sidecar.istio.io/excludeInboundPorts: "8080"
        traffic.sidecar.istio.io/excludeOutboundPorts: "443,9000,22,9090"

    nonlbServices:
      labels: {}
      annotations: {}
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    nonlbDeployments:
      labels: {}
      annotations: {}
```

3. Use the following code snippet to create an envoy filter for both UDR and ATS:

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  name: <user defined name for envoy filter>
  namespace: <namespace where ATS is deployed>
spec:
  workloadSelector:
    labels:
      app: ocats-udr
  configPatches:
  - applyTo: NETWORK_FILTER
    match:
      listener:
        filterChain:
          filter:
            name: "envoy.http_connection_manager"
    patch:
      operation: MERGE
      value:
        typed_config:
          '@type': type.googleapis.com/
envoy.config.filter.network.http_connection_manager.v2.HttpConnectionManage
r
          server_header_transformation: PASS_THROUGH

---
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  name: <user defined name for envoy filter>
  namespace: <namespace where ATS is deployed>
spec:
  workloadSelector:
    labels:
      app: ocats-udr
  configPatches:
  - applyTo: NETWORK_FILTER
    match:
      listener:
        filterChain:
          filter:
            name: "envoy.http_connection_manager"
    patch:
      operation: MERGE
      value:
        typed_config:
          '@type': type.googleapis.com/
envoy.config.filter.network.http_connection_manager.v2.HttpConnectionManage
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
r
          forward_client_cert_details: ALWAYS_FORWARD_ONLY
```

For more information about envoy filter and service mesh configuration, see the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.

4. After deploying service mesh, create Peer Authentication on the pods for inter pod communication. A sample template is as follows:

```
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: <ATS Peer Authentication name>
  namespace: <ATS deployment namespace>
spec:
  selector:
    matchLabels:
      app: ocats-udr
  mtls:
    mode: PERMISSIVE
---

apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: <ATS Stubs Peer Authentication name>
  namespace: <ATS Stubs deployment namespace>
spec:
  selector:
    matchLabels:
      app: ocstub-py   mtls:
    mode: PERMISSIVE
```

5. ATS sidecar must have at least 2 CPUs and 2 Gi memory. ATS sidecar is configured as follows:

```
deployment:
  customExtension:
    annotations:
      sidecar.istio.io/proxyCPU: "2000m"
      sidecar.istio.io/proxyCPULimit: "2000m"
      sidecar.istio.io/proxyMemory: "2Gi"
      sidecar.istio.io/proxyMemoryLimit: "2Gi"
      proxy.istio.io/config: |
        concurrency: 4
```

3.7.8.9 Configuring Stubs

From ATS release 24.2.0:

- The common python stub chart and image must be used for two NRF-stubs, Notify-stub, TLS notify stub (TLS notify must be deployed separately), and SCP stub.
- Two different deployments of NRF (primary and secondary) using common python stub chart and image must be deployed for SLF-ATS and EIR-ATS.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- Two different deployments of NRF (primary and secondary), one deployment for notify stub (non-TLS scenarios), one deployment for tls-notify-stub (TLS scenarios) and one deployment for SCP stub must be deployed for UDR-ATS.

3.7.8.9.1 Configuring NRF Stub

UDR-ATS, SLF-ATS, and EIR-ATS requires two NRF stubs, one as primary NRF and other as secondary NRF. These NRF stubs is deployed using ocstub-py-custom-values-25.1.200.yaml.

To configure NRF stub:

- Provide the docker registry details with ocstub-py-custom-values-25.1.202.yaml file where images are pushed as follows:

```
image:
  repository: <docker registry>:<docker port>
```

- Set the env.NF to SLF, UDR, or 5G EIR depending upon the deployed NF:

```
env:
  ...
  NF: SLF


env:
  ...
  NF: UDR


env:
  ...
  NF: 5G_EIR
```

- Set the `appendReleaseName` parameter to **false** for backward compatibility and `bypass_additional_port` parameter to **true** to avoid creating HTTP1 port. Each deployment should have unique service name:

```
service:
  ...
  name: <user-defined nrf stub service name> #Example nrf-stub-service1
for primary NRF stub and nrf-stub-service2 for secondary NRF stub
  appendReleaseName: false #this is to avoid adding release name at the
beginning of the service name thereby supporting backward compatibility
  ...
  bypass_additional_port: true #to avoid creating HTTP1 port.
```

- If CNE supports dual stack feature and UDR, SLF, EIR, and ProvGw are deployed in IPv6_IPv4 mode, then to run the dual stack scenarios on ATS, you must deploy NRF stub by setting the `deploymentMode` as IPv6_IPv4. This will enable NRF stub service to have both IPv6 and IPv4 address. IPv6 address will be preferred.

```
deplopymentMode: IPv6_IPv4
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

3.7.8.9.2 Configuring Notify Stub

UDR-ATS requires notify-stub to be deployed in three modes using ocstub-py-custom-values-25.1.202.yaml:

- HTTP mode: To receive non-TLS notifications from UDR
- HTTPS mode: To receive TLS notifications from UDR
- SCP mode: To validate SCP routing scenarios

To configure notify stub for Notify (TLS and non-TLS) and SCP stub modes:

- Update the docker registry where common python stub image is pushed as follows:

```
image:
  repository: <docker registry>:<docker port>
```

- Set the env.NF to UDR depending upon the whether the deployed NF is UDR:

```
env:
  ...
  NF: UDR
```

- If CNE supports dual stack feature and UDR, SLF, EIR, and ProvGw are deployed in IPv6_IPv4 mode, then to run the dual stack scenarios on ATS, you must deploy Notify stub by setting the `deploymentMode` as IPv6_IPv4. This will enable Notify stub service to have both IPv6 and IPv4 address. IPv6 address will be preferred.

```
deplopymentMode: IPv6_IPv4
```

- Set the appendReleaseName parameter to false for backward compatibility and bypass_additional_port parameter to true to avoid creating HTTP1 port and expose_tls_service parameter to false to disable TLS mode of stub deployment. Common python stub deployed as HTTP and TLS server must have unique service name:

```
service:
  ...
  name: <user-defined notify stub service name> #Example notify-stub-
service for non-TLS stub
  appendReleaseName: false #this is to avoid adding release name at the
beginning of the service name thereby supporting backward compatibility
  ...
  bypass_additional_port: true #to avoid creating HTTP1 port.
  expose_tls_service: false
```

- Set the appendReleaseName parameter to false for backward compatibility and bypass_additional_port parameter to true to avoid creating HTTP1 port. Common python stub deployed as HTTP server must have unique service name:

```
service:
  ...
  name: <user-defined TLS notify stub service name> #Example tlsnotify-
stub-service for TLS stub
  appendReleaseName: false #this is to avoid adding release name at the
beginning of the service name thereby supporting backward compatibility
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  ...
  bypass_additional_port: true #to avoid creating HTTP1 port.
```

- Configure the stub with private key, certificates and secret that is used by UDR Egress Gateway for TLS notification validation and set expose_tls_service to true:

```
env:
  ...
  cert_secret_name: <TLS secret of UDR egressgateway> #E.g., ocudr-gateway-
secret
  ca_cert: <CA certificate used for UDR egressgateway deployment> #E.g.,
caroot.cer
  client_cert: <Client certificate used for UDR egressgateway deployment>
#E.g., apigatewayrsa.cer
  private_key: <Private key used for UDR egressgateway deployment> #E.g.,
rsa_private_key_pkcs1.pem
  ...
  CLIENT_CERT_REQ: true
  APACHE_KEEP_ALIVE_TIMEOUT: 1000
...
service:
 ...
  expose_tls_service: true
```

- To deploy notify-stub as SCP set the appendReleaseName parameter to false for backward compatibility and bypass_additional_port parameter to true to avoid creating HTTP1 port and set expose_tls_service to false. Common python stub deployed as HTTP and TLS server must have unique service name:

```
service:
  ...
  name: <user-defined scp stub service name> #Example scp-stub-service for
SCP stub
  appendReleaseName: false #this is to avoid adding release name at the
beginning of the service name thereby supporting backward compatibility
  ...
  bypass_additional_port: true #to avoid creating HTTP1 port.
  expose_tls_service: false
```

3.7.8.9.3 Configuring Fourg Stub

UDR-ATS requires one fourg stub to test the migration scenarios. Configure the docker registry where diameter stub image is pushed as follows:

```
image:
  repository: <docker registry>:<docker port>/ocats-udr-fourg-stub-images
```

If CNE supports dual stack feature and UDR, SLF, EIR, and ProvGw are deployed in IPv6_IPv4 mode, then to run the dual stack scenarios on ATS, you must deploy Fourg stub by setting the `deploymentMode` as IPv6_IPv4. This will enable Fourg stub service to have both IPv6 and IPv4 address. IPv6 address will be preferred.

```
deplopymentMode: IPv6_IPv4
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

3.7.8.9.4 Configuring Diameter Stub

UDR-ATS requires two diameter stubs as two different diameter peers.

- diam-toola: peernode1 (seagull1a.seagull.com)
- diam-toolb: peernode2 (seagull1b.seagull.com)

> **Note**
>
> UDR-ATS uses diam-stub-25.1.200.yaml file for diameter stub configuration.

EIR-ATS requires one diam-stub to be deployed as single diameter peer.

- diam-toola → peernode1 (seagull1a.seagull.com)

> **Note**
>
> You must follow only configuring diam-toola section. The diam-toolb section is only for UDR-ATS.

To configure diameter stubs:

- Update the docker registry where diameter stub image is pushed as follows:

```
image:
  repository: <docker registry>:<docker port>/ocats-udr-diam-stub-images
```

- If CNE supports dual stack feature and UDR, SLF, EIR, and ProvGw are deployed in IPv6_IPv4 mode, then to run the dual stack scenarios on ATS, you must deploy diam stub by setting the `deploymentMode` as IPv6_IPv4. This will enable diam stub service to have both IPv6 and IPv4 address. IPv6 address will be preferred.

```
deplopymentMode: IPv6_IPv4
```

- Configure diam-toola as follows:

```
deployment:
  SourceIdentity: seagull1a.seagull.com
  SourceRealm: seagulla.com
  mode: nudr-dr
```

- Configure diam-toolb as follows:

```
deployment:
  SourceIdentity: seagull1b.seagull.com
  SourceRealm: seagullb.com
  mode: nudr-dr
```

- Configure diam-tool for EIR-ATS as follows:

```
deployment:
  SourceIdentity: seagull1a.seagull.com
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
SourceRealm: seagulla.com
mode: n5g-eir-eic
```

3.7.9 Deploying ATS in Kubernetes Cluster

You can deploy ATS Pod in Kubernetes cluster using Helm commands.

Run the following command to deploy ATS.

```
helm install --name <release_name> --namespace <namespace_name> -f <values-yaml-file> ocats-udr
```

Example:

```
helm install --name ocats-udr-25.1.100 --namespace ocudr -f ocats-udr-custom-values-25.1.100.yaml ocats-udr
```

3.7.9.1 Deploying ATS Pod and Stubs in Kubernetes Cluster

ATS resource allocation must be done by referring to the Resource Requirements section to support parallel test execution feature. For more information on Parallel Test Execution, see Parallel Test Execution. If you must enable the Application Log Collection, see UDR Application Log Collection.

The CPU and memory utilization depends on the number of behave command executed at given point of time. UDR-ATS runs six behave commands at a time and SLF-ATS runs seven behave commands at a time.

You can deploy stub pod in Kubernetes cluster using Helm commands.

Run the following command to deploy to deploy ATS and stubs:

```
helm install --name <release_name> --namespace <namespace_name> -f <values-yaml-file> ocats-udr
```

Example:

```
 helm install --name ocats-udr-25.1.200 --namespace ocudr -f ocats-udr-custom-values-25.1.200.yaml ocats-udr
```

SLF-NewFeatures, EIR-NewFeatures, SLF-Regression, or EIR-Regression Pipeline

Run the following command to deploy NRF-STUB:

```
helm install --name <release_name> --namespace <namespace_name> -f <values-yaml-file> ocstub-py
```

Example:

```
helm install --name stub1 --namespace ocudr -f ocstub-py-custom-values-25.1.202.yaml ocstub-py
helm install --name stub2 --namespace ocudr -f ocstub-py-custom-values-25.1.202.yaml ocstub-py
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> To test DNS SRV feature in SLF-Regression and EIR-NewFeatures, NRF stub needs to be deployed two times to act as primary and secondary NRF.

UDR-NewFeatures or UDR-Regression Pipelines using Helm

Run the following commands to deploy each of the required stubs using Helm:

```
helm install --name <release_name> --namespace <namespace_name> -f <values-yaml-file> ocstub-py

helm install --name <release_name> --namespace <namespace_name> -f <fourg-stub-values-yaml-file> fourg-stub

helm install --name <release_name> --namespace <namespace_name> -f <diam-tool-values-yaml-file> diam-stub
```

Example:

```
 helm install --name nrfstub --namespace ocudr -f ocstub-py-custom-values-25.1.202.yaml ocstub-py

 helm install --name notify --namespace ocudr -f ocstub-py-custom-values-25.1.202.yaml ocstub-py

 helm install --name tlsnotify--namespace ocudr -f ocstub-py-custom-values-25.1.202.yaml ocstub-py

 helm install --name scp --namespace ocudr -f ocstub-py-custom-values-25.1.202.yaml  ocstub-py

 helm install --name fourgstub --namespace ocudr -f fourg-stub-custom-values-25.1.200.yaml fourg-stub

 helm install --name diamtoola --namespace ocudr -f diam-tool-custom-values-25.1.200.yaml diam-stub

 helm install --name diamtoolb --namespace ocudr -f diam-tool-custom-values-25.1.200.yaml diam-stub
```

3.7.10 Verifying ATS Deployment

To verify ATS deployment, run the following command:

```
helm status <release_name>
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 3-21 Sample Output of Verifying ATS Deployment

```
[cloud-user@cne-24-3-bastion-1 ~]$ helm status ats -n ocudr
NAME: ats
LAST DEPLOYED: Fri Jul 11 10:35:18 2025
NAMESPACE: ocudr
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
# Copyright 2018 (C), Oracle and/or its affiliates. All rights reserved.

Thank you for installing ocats-udr.

Your release is named ats , Release Revision: 1.
To learn more about the release, try:

  $ helm status ats
  $ helm get ats
[cloud-user@cne-24-3-bastion-1 ~]$
```

To view UDR, four UDRs, Provisioning Gateway, two NRF stubs, one bulk import and ATS deployed in the SLF namespace, run `kubectl get pods -n <ns>` command. The output is as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 3-22 Sample SLF Namespace

```
[cloud-user@udr2-cne243-bastion-2 ~]$ kubectl get pods -n udr-ravi
NAME                                                  READY   STATUS      RESTARTS   AGE
bulk-import-nudr-bulk-import-0                        2/2     Running     0          4h21m
oso-prom-ravi-svr-bc75fd59-hkclk                      2/2     Running     0          4h51m
provgw-auditor-service-0                              2/2     Running     0          4h47m
provgw-prov-egressgateway-d79c6c4b8-hwfmp             2/2     Running     0          4h47m
provgw-prov-egressgateway-d79c6c4b8-qvdvd             2/2     Running     0          4h47m
provgw-prov-ingressgateway-57bddd7bbb-tvg9c           2/2     Running     0          4h47m
provgw-prov-ingressgateway-57bddd7bbb-vg859           2/2     Running     0          4h47m
provgw-provgw-config-d4d48bfd9-m28tx                  2/2     Running     0          4h47m
provgw-provgw-config-d4d48bfd9-nxsqn                  2/2     Running     0          4h47m
provgw-provgw-service-5db6bcb498-8lggb                2/2     Running     0          4h47m
provgw-provgw-service-5db6bcb498-l8dtv                2/2     Running     0          4h47m
provgw-test                                           0/1     Completed   0          4h20m
slfats-ocats-udr-877f54b64-d7ld7                      2/2     Running     0          3h32m
slfexport-nudr-export-tool-0                          2/2     Running     0          4h20m
stub1-ocstub-py-5659cd5746-lqvtk                      2/2     Running     0          4h51m
stub2-ocstub-py-5dd79884fd-rs2zc                      2/2     Running     0          4h51m
udr1-alternate-route-664f7f5769-9kps4                 2/2     Running     0          4h41m
udr1-alternate-route-664f7f5769-xjlrf                 2/2     Running     0          4h41m
udr1-appinfo-6f695f5866-bxxjx                         2/2     Running     0          4h41m
udr1-egressgateway-7b4794b86c-6q4nr                   2/2     Running     0          4h41m
udr1-egressgateway-7b4794b86c-8s2vv                   2/2     Running     0          4h41m
udr1-ingressgateway-prov-cffb68d44-6f8ms              2/2     Running     0          4h41m
udr1-ingressgateway-prov-cffb68d44-c9dlp              2/2     Running     0          4h41m
udr1-ingressgateway-sig-69ddddbd75-7xvsm              2/2     Running     0          4h41m
udr1-ingressgateway-sig-69ddddbd75-l74nb              2/2     Running     0          4h41m
udr1-nudr-config-5dd8f48dd5-dftj5                     2/2     Running     0          4h41m
udr1-nudr-config-5dd8f48dd5-n6cjn                     2/2     Running     0          4h41m
udr1-nudr-config-server-645bd99b4c-b9fv5              2/2     Running     0          4h41m
udr1-nudr-config-server-645bd99b4c-pxgj2              2/2     Running     0          4h41m
udr1-nudr-dr-provservice-8665f48b45-hklsr             2/2     Running     0          4h41m
udr1-nudr-dr-provservice-8665f48b45-nz4rj             2/2     Running     0          4h41m
udr1-nudr-drservice-cfd69bb7b-2vmpg                   2/2     Running     0          4h41m
udr1-nudr-drservice-cfd69bb7b-hnxrt                   2/2     Running     0          4h41m
udr1-nudr-nrf-client-nfmanagement-56c8cddbb6-nmhzv    2/2     Running     0          4h41m
udr1-performance-f66b68659-57c5k                      2/2     Running     0          4h41m
udr1-performance-f66b68659-rh9w6                      2/2     Running     0          4h41m
udr1-test                                             0/1     Completed   0          4h20m
udr2-alternate-route-59957fcbcc-rkwdf                 2/2     Running     0          4h35m
udr2-appinfo-845d7bc796-k8tk2                         2/2     Running     0          4h35m
udr2-appinfo-845d7bc796-vj2jz                         2/2     Running     0          4h35m
udr2-egressgateway-5c4d469679-l49vc                   2/2     Running     0          4h35m
udr2-ingressgateway-prov-5b7687589f-7f9z5             2/2     Running     0          4h35m
udr2-ingressgateway-sig-7cdbc5cf84-47nnd              2/2     Running     0          4h35m
udr2-nudr-config-55fbcb88dc-s94z4                     2/2     Running     0          4h35m
udr2-nudr-config-server-55d68d6c99-5kg4f              2/2     Running     0          4h35m
udr2-nudr-config-server-55d68d6c99-x9cwv              2/2     Running     0          4h35m
udr2-nudr-dr-provservice-5b4c479c69-54f65             2/2     Running     0          4h35m
udr2-nudr-drservice-775678b69b-p42bs                  2/2     Running     0          4h35m
udr2-performance-776b9955dd-4mqj6                     2/2     Running     0          4h35m
udr2-performance-776b9955dd-tjthw                     2/2     Running     0          4h35m
udr2-test                                             0/1     Completed   0          4h20m
udr3-alternate-route-7f97f7d64f-8dp7h                 2/2     Running     0          4h29m
udr3-appinfo-59d78d7dfc-9s9nh                         2/2     Running     0          4h29m
udr3-appinfo-59d78d7dfc-9whf9                         2/2     Running     0          4h29m
udr3-egressgateway-8c96b6b8d-dfmsl                    2/2     Running     0          4h29m
udr3-ingressgateway-prov-7fbbf99555-2jrx4             2/2     Running     0          4h29m
udr3-ingressgateway-sig-684798677c-xh69t              2/2     Running     0          4h29m
udr3-nudr-config-66596ccfd5-s25cv                     2/2     Running     0          4h29m
udr3-nudr-config-server-66dd568f55-d79qc              2/2     Running     0          4h29m
udr3-nudr-config-server-66dd568f55-phrcj              2/2     Running     0          4h29m
udr3-nudr-dr-provservice-66f94f4cc9-plqlx             2/2     Running     0          4h29m
udr3-nudr-drservice-6db8f967d4-vdcc2                  2/2     Running     0          4h29m
udr3-performance-664cf75b6-58zfd                      2/2     Running     0          4h29m
udr3-performance-664cf75b6-ljfn5                      2/2     Running     0          4h29m
udr3-test                                             0/1     Completed   0          4h20m
udr4-alternate-route-78dcc485fc-jg44k                 2/2     Running     0          4h23m
udr4-appinfo-796b87c58b-4nbcj                         2/2     Running     0          4h23m
udr4-appinfo-796b87c58b-nx6lw                         2/2     Running     0          4h23m
udr4-egressgateway-7d85888497-nf9q6                   2/2     Running     0          4h23m
udr4-ingressgateway-prov-7d8c69f9cf-gvw4v             2/2     Running     0          4h23m
udr4-ingressgateway-sig-5d6bfd7689-79z6m              2/2     Running     0          4h23m
udr4-nudr-config-7c7989849b-4lknn                     2/2     Running     0          4h23m
udr4-nudr-config-server-776d84495-45jdl               2/2     Running     0          4h23m
udr4-nudr-config-server-776d84495-sxl62               2/2     Running     0          4h23m
udr4-nudr-dr-provservice-775b87577b-f5ghn             2/2     Running     0          4h23m
udr4-nudr-drservice-798bf84c7-hshsd                   2/2     Running     0          4h23m
udr4-performance-676f97658d-4djch                     2/2     Running     0          4h23m
```


Copyright © 2019, 2025, Oracle and/or its affiliates

Following is the sample output of the command, '`kubectl get pods -n <ns>`'. It shows OCUDR namespace with one UDR, one Provisioning Gateway, two diam-tool stubs, one http-server stub, one SCP stub, one bulk import, two NRF stubs, one fourg-stub and ATS after installing UDR-ATS for UDR-Pipelines:

Figure 3-23 Sample UDR Namespace

```
[cloud-user@cne-24-3-bastion-1 ~]$ kubectl get pods -n ats-ravi
NAME                                                READY   STATUS    RESTARTS       AGE
ats-ravi-ocats-udr-9fb99dd44-rsd9s                  1/1     Running   0              27h
diamtoola-ravi-diam-stub-6459d7ff55-zjph4           1/1     Running   0              27h
diamtoolb-ravi-diam-stub-55565d5cbd-jh4qh           1/1     Running   0              27h
fourgstub-ravi-fourg-stub-749bdd896c-p4m9t          1/1     Running   0              27h
notifystub-ravi-ocstub-py-6bfcb9dc4c-gjqj6          1/1     Running   0              27h
nrfstub1-ravi-ocstub-py-bc9d86ff9-9g7zs             1/1     Running   0              27h
nrfstub2-ravi-ocstub-py-56975fb4bc-b92h9            1/1     Running   0              27h
ocudr-ravi-alternate-route-7b5cd6b7b9-6q76x         1/1     Running   0              27h
ocudr-ravi-alternate-route-7b5cd6b7b9-7m99w         1/1     Running   0              27h
ocudr-ravi-appinfo-669d7d7998-kz4cd                 1/1     Running   1 (27h ago)    27h
ocudr-ravi-appinfo-669d7d7998-wcptl                 1/1     Running   0              27h
ocudr-ravi-egressgateway-55f54454dc-njgtb           1/1     Running   0              27h
ocudr-ravi-egressgateway-55f54454dc-z7g8v           1/1     Running   0              27h
ocudr-ravi-ingressgateway-prov-79c87c7784-7k2kw     1/1     Running   0              27h
ocudr-ravi-ingressgateway-prov-79c87c7784-wg87x     1/1     Running   0              27h
ocudr-ravi-ingressgateway-sig-7b5b96fddb-7tsgm      1/1     Running   0              27h
ocudr-ravi-ingressgateway-sig-7b5b96fddb-fvgck      1/1     Running   0              27h
ocudr-ravi-nudr-config-fc9f5b9b5-ck929              1/1     Running   0              27h
ocudr-ravi-nudr-config-fc9f5b9b5-kr275              1/1     Running   0              27h
ocudr-ravi-nudr-config-server-6cdd4845bd-4rwsz      1/1     Running   0              27h
ocudr-ravi-nudr-config-server-6cdd4845bd-dq79w      1/1     Running   0              27h
ocudr-ravi-nudr-diam-gateway-0                      1/1     Running   1 (27h ago)    27h
ocudr-ravi-nudr-diam-gateway-1                      1/1     Running   0              27h
ocudr-ravi-nudr-diameterproxy-768965fd94-k9knj      1/1     Running   0              27h
ocudr-ravi-nudr-diameterproxy-768965fd94-p6mzs      1/1     Running   2 (27h ago)    27h
ocudr-ravi-nudr-dr-provservice-7bb574699b-r9ln6     1/1     Running   1 (27h ago)    27h
ocudr-ravi-nudr-dr-provservice-7bb574699b-xj24c     1/1     Running   1 (27h ago)    27h
ocudr-ravi-nudr-drservice-67566cc9d6-97pbk          1/1     Running   1 (27h ago)    27h
ocudr-ravi-nudr-drservice-67566cc9d6-l4jxw          1/1     Running   0              27h
ocudr-ravi-nudr-notify-service-7f96564c88-9tkrg     1/1     Running   21 (25h ago)   27h
ocudr-ravi-nudr-notify-service-7f96564c88-n6qsw     1/1     Running   1 (27h ago)    27h
ocudr-ravi-nudr-nrf-client-nfmanagement-856b44b654-tc48s   1/1     Running   0              27h
ocudr-ravi-nudr-ondemand-migration-7c5c7d4bd7-5w585 1/1     Running   3 (27h ago)    27h
ocudr-ravi-nudr-ondemand-migration-7c5c7d4bd7-nxx6b 1/1     Running   4 (27h ago)    27h
ocudr-ravi-performance-6989cf7dbb-jd245             1/1     Running   1 (27h ago)    27h
ocudr-ravi-performance-6989cf7dbb-k5vlk             1/1     Running   2 (27h ago)    27h
scpstub-ravi-ocstub-py-674c67856f-zzbl6             1/1     Running   0              27h
tlsnotifystub-ravi-ocstub-py-846cd8d785-5jk7c       1/1     Running   0              27h
[cloud-user@cne-24-3-bastion-1 ~]$
```

Following is the sample output of the command, '`kubectl get pods -n <ns>` - It shows OCUDR namespace with one UDR, two nrf stubs, one diam-stub, one bulk import and ATS after installation for EIR pipeline:

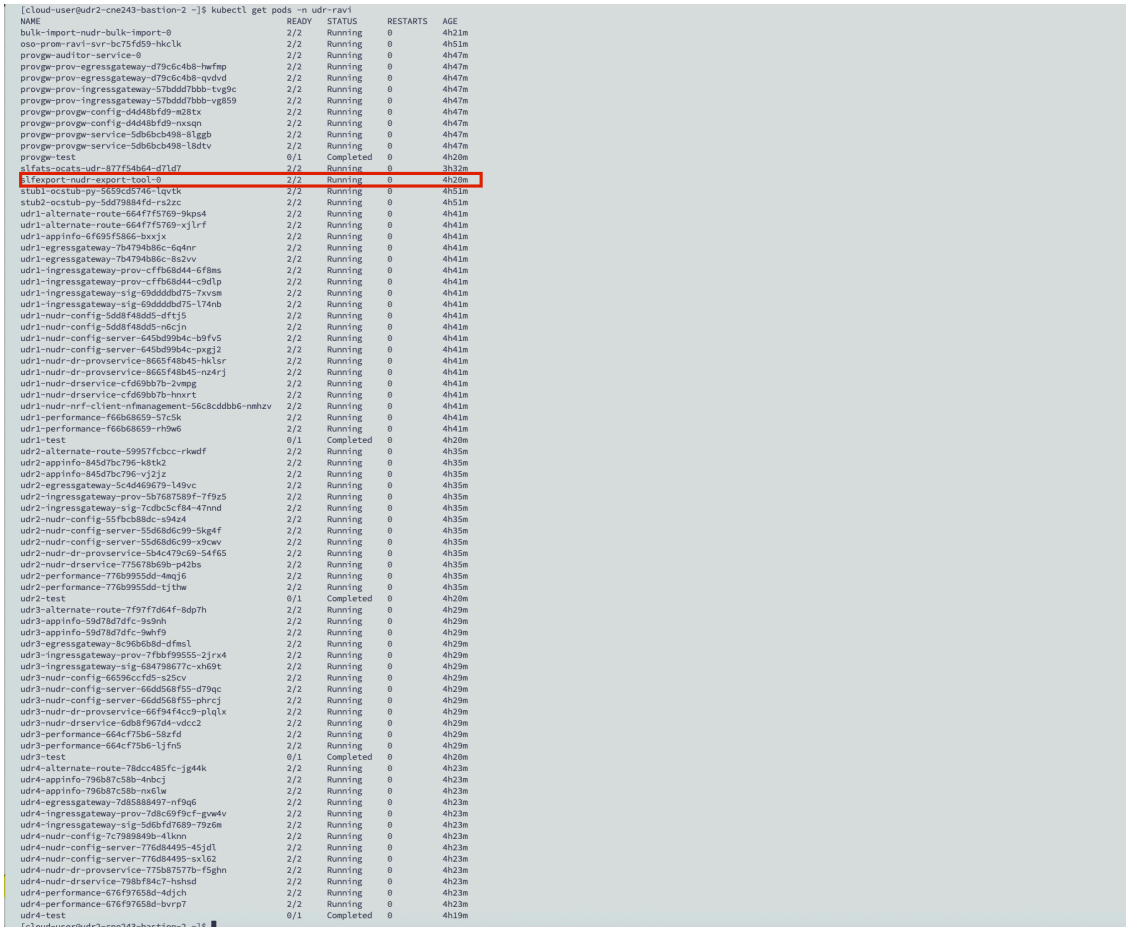
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 3-24 Sample EIR Namespace

```
[cloud-user@cne-24-1-bastion-1 5G-UDR]$ kubectl get pod -n eir-ats
NAME                                                  READY   STATUS    RESTARTS   AGE
bulk-import-nudr-bulk-import-0                        2/2     Running   0          4h49m
diamtoola-ats-diam-stub-69d49c7788-ggd7j              1/1     Running   0          5h11m
eirats-ocats-udr-69b7989588-cmn8k                     1/1     Running   0          5h11m
export-tool-nudr-export-tool-0                        1/1     Running   0          4h48m
oceir-ats-alternate-route-5f76f8fcf6-5phpr            1/1     Running   0          5h2m
oceir-ats-appinfo-789d84b86b-98c52                    1/1     Running   0          5h2m
oceir-ats-egressgateway-65d9bfcd45-bt7f8              1/1     Running   0          5h2m
oceir-ats-ingressgateway-prov-86db664564-vtdjw        1/1     Running   0          5h2m
oceir-ats-ingressgateway-sig-65c4d78c56-9c6zx         1/1     Running   0          5h2m
oceir-ats-nudr-config-778d6688fd-97pqw                1/1     Running   0          5h2m
oceir-ats-nudr-config-server-746c894b8c-zxgmx         1/1     Running   0          5h2m
oceir-ats-nudr-diam-gateway-0                         1/1     Running   0          5h2m
oceir-ats-nudr-diameterproxy-6fd7d5c96b-c4hmf         1/1     Running   0          5h2m
oceir-ats-nudr-dr-provservice-56bb7f7498-nbtm7        1/1     Running   0          5h2m
oceir-ats-nudr-drservice-55bfc5d876-l2vm8             1/1     Running   0          5h2m
oceir-ats-nudr-nrf-client-nfmanagement-584996b459-7v5xw   1/1     Running   0          5h2m
oceir-ats-performance-868b9c8577-jb246                1/1     Running   0          5h2m
stub1-ocstub-py-54d558c7fc-s4hq8                      1/1     Running   0          5h11m
stub2-ocstub-py-65cf6bbff7-xg895                      1/1     Running   0          5h11m
```

If you have installed ATS with sidecar, ensure ATS shows two containers in the READY state as "2/2". A sample output of the command, '`kubectl get pods -n <ns>`' for SLF-Pipelines is as follows:

Figure 3-25 Sample ATS Deployed with Sidecar



Copyright © 2019, 2025, Oracle and/or its affiliates.

3.7.11 Post Installation Steps

If Provisioning Gateway is upgraded using helm upgrade command

Following are the post installation steps:

1. Perform the post installation steps as described in Cloud Native Core Provisioning Gateway Guide to change segDetails to UDR Provisioning Ingress Gateway fqdn and port. For more information, see *Cloud Native Core Provisioning Gateway Guide*.
2. Run the following command:

   ```
   kubectl exec -it -n <ns> <ATS pod> bash
   ```

3. Run the following command:

   ```
   curl -X PUT http://<provgw helm release>-provgw-config.<ns>:5001/provgw-
   config/v1/udr.provgwservice.cfg/PROVGW-SERVICE -d '{"tracingEnabled":
   false,"soapService_udrIp": "ocudr-ingressgateway-
   prov.ocudr","soapService_udrSignallingIp": "ocudr-ingressgateway-
   sig.ocudr","retryErrorCodes": [500,503],"retryCount": 3,"retryInterval":
   2}'
   ```

4. Run the `exit` command.

If PV (Persistent Volume) is enabled for UDR ATS

Following are the post installation steps:

1. Run the following command to extract the ocslf_tests (for SLF Pipelines), ocudr_tests (for UDR Pipelines), or oceir_tests (for EIR Pipeline) and jobs from ocats-udr-data-25.1.200.tgz.
   `tar -xvf ocats-udr-data-25.1.200.tgz`
2. Run the following command to create certs and oauth_keys in ocslf_tests folder (for SLF pipeline runs), oceir_tests (for EIR pipeline runs), or ocudr_tests (for UDR pipeline runs):

   ```
   mkdir -p ocslf_tests/certs ocslf_tests/oauth_keys
   ```

   ```
   mkdir -p oceir_tests/certs oceir_tests/oauth_keys
   ```

   ```
   mkdir -p ocudr_tests/certs ocudr_tests/oauth_keys
   ```

3. Run the following commands to copy the ocslf_tests and jobs folder to the ATS pod only if it is intended to run SLF Pipelines.

   ```
   kubectl cp ocslf_tests <namespace>/<pod-name>:/var/lib/jenkins
   kubectl cp jobs <namespace>/<pod-name>:/var/lib/jenkins/.jenkins
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

4. Run the following commands to copy the ocudr_tests and jobs folder to the ATS pod only if it is intended to run UDR Pipelines.

   ```
   kubectl cp ocudr_tests <namespace>/<pod-name>:/var/lib/jenkins
   kubectl cp jobs <namespace>/<pod-name>:/var/lib/jenkins/.jenkins
   ```

5. Run the following commands to copy the oceir_tests and jobs folder to the ATS pod only if it is intended to run EIR Pipelines.

   ```
   kubectl cp oceir_tests <namespace>/<pod-name>:/var/lib/jenkins
   kubectl cp jobs <namespace>/<pod-name>:/var/lib/jenkins/.jenkins
   ```

6. Run the following command to restart the pod:
   `kubectl delete pod <pod-name> -n <namespace>`
7. For SLF-NewFeatures and SLF-Regression pipelines, copy the root certificate authority (CA), signed server certificate with root CA private key, and private keys used to create secret for TLS support on Provisioning Gateway to `ocslf_tests/certs` folder in the path as follows:

   > **Note**
   >
   > Provisioning Gateway must use all three files as part of TLS support.

   a. `kubectl cp <root CA> <namespace>/<pod-name>:/var/lib/jenkins/ocslf_tests/certs`
      Example: `kubectl cp caroot.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocslf_tests/certs`
   b. `kubectl cp <server certificate> <namespace>/<pod-name>:/var/lib/jenkins/ocslf_tests/certs`
      Example: `kubectl cp apigatewayrsa.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocslf_tests/certs`
   c. `kubectl cp <private key> <namespace>/<pod-name>:/var/lib/jenkins/ocslf_tests/certs`
      Example: `kubectl cp rsa_private_key_pkcs1.pem ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocslf_tests/certs`

   > **Note**
   >
   > Provisioning Gateway should use the above three files as part of TLS support.

8. For UDR-NewFeatures and UDR-Regression pipelines, copy the root certificate authority (CA), signed server certificate with root CA private key, and private key that are used to create secret for TLS support on Provisioning Gateway to `ocudr_tests or ocudr_certs` folder in the path as follows:

   > **Note**
   >
   > For TLS validation, use the same set of copied certificates for Provisioning Gateway Ingress Gateway, UDR Ingress Gateway, and UDR Egress Gateway.

Copyright © 2019, 2025, Oracle and/or its affiliates.

a. `kubectl cp <root CA> <namespace>/<pod-name>:/var/lib/jenkins/ocudr_tests/certs`
   Example: `kubectl cp caroot.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocudr_tests/certs`

b. `kubectl cp <server certificate> <namespace>/<pod-name>:/var/lib/jenkins/ocudr_tests/certs`
   Example: `kubectl cp apigatewayrsa.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocudr_tests/certs`

c. `kubectl cp <private key> <namespace>/<pod-name>:/var/lib/jenkins/ocudr_tests/certs`
   Example: `kubectl cp rsa_private_key_pkcs1.pem ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocudr_tests/certs`

9. For EIR-NewFeatures and EIR-Regression pipelines, copy the root certificate authority (CA), signed server certificate with root CA private key, and private key that are used to create secret for TLS support on UDR ingressgateway-prov and ingressgateway-sig to `oceir_tests/certs` folder in the path as follows:

   a. `kubectl cp <root CA> <namespace>/<pod-name>:/var/lib/jenkins/oceir_tests/certs`
      Example: `kubectl cp caroot_sig.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/certs`

   b. `kubectl cp <server certificate> <namespace>/<pod-name>:/var/lib/jenkins/oceir_tests/certs`
      Example: `kubectl cp apigatewayrsa_sig.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/certs`

   c. `kubectl cp <private key> <namespace>/<pod-name>:/var/lib/jenkins/oceir_tests/certs`
      Example: `kubectl cp rsa_private_key_pkcs1_sig.pem ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/certs`

   d. `kubectl cp <root CA> <namespace>/<pod-name>:/var/lib/jenkins/oceir_tests/certs`
      Example: `kubectl cp caroot_prov.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/certs`

   e. `kubectl cp <server certificate> <namespace>/<pod-name>:/var/lib/jenkins/oceir_tests/certs`
      Example: `kubectl cp apigatewayrsa_prov.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/certs`

   f. `kubectl cp <private key> <namespace>/<pod-name>:/var/lib/jenkins/oceir_tests/certs`
      Example: `kubectl cp rsa_private_key_pkcs1_prov.pem ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/certs`

10. To run Oauth2 validation scenarios on SLF-Regression, copy each private keys in pem format to oauth_keys folder in `/var/lib/jenkins/ocslf_tests` path.

    ```
    kubectl cp <private key pem file> <namespace>/<pod-name>:/var/lib/jenkins/ocslf_tests/oauth_keys
    ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

Example:

```
kubectl cp ec_private_key1.pem ocudr/ats-ocats-
udr-696df7c84d-2qc7h:/var/lib/jenkins/ocslf_tests/oauth_keys
```

11. To run OAuth2 validation scenarios on UDR-Regression, copy each private keys in pem format to oauth_keys folder in `/var/lib/jenkins/ocudr_tests` path.

    ```
    kubectl cp <private key pem file> <namespace>/<pod-name>:/var/lib/jenkins/
    ocudr_tests/oauth_keys
    ```

    Example:

    ```
    kubectl cp ec_private_key1.pem ocudr/ats-ocats-
    udr-696df7c84d-2qc7h:/var/lib/jenkins/ocudr_tests/oauth_keys
    ```

12. To run OAuth2 validation scenarios on EIR-Regression, copy each private keys in pem format to oauth_keys folder in `/var/lib/jenkins/oceir_tests` path.

    ```
    kubectl cp <private key pem file> <namespace>/<pod-name>:/var/lib/jenkins/
    oceir_tests/oauth_keys
    ```

    Example:

    ```
    kubectl cp ec_private_key1.pem ocudr/ats-ocats-
    udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/oauth_keys
    ```

If PV is disabled:

Following are the post installation steps:

1. For SLF-NewFeatures and SLF-Regression pipelines, copy the root certificate authority (CA), signed server certificate with root CA private key, and private key used to create secret for TLS support on Provisioning Gateway to ocslf_tests/certs folder in the path as follows:

   > **Note**
   >
   > Provisioning Gateway must use all three files as part of TLS support.

   a. `kubectl cp <root CA> <namespace>/<pod-name>:/var/lib/jenkins/ocslf_tests/certs`
      Example: `kubectl cp caroot.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocslf_tests/certs`

   b. `kubectl cp <server certificate> <namespace>/<pod-name>:/var/lib/jenkins/ocslf_tests/certs`
      Example: `kubectl cp apigatewayrsa.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocslf_tests/certs`

   c. `kubectl cp <private key> <namespace>/<pod-name>:/var/lib/jenkins/ocslf_tests/certs`

Copyright © 2019, 2025, Oracle and/or its affiliates.

Example: `kubectl cp rsa_private_key_pkcs1.pem ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocslf_tests/certs`

2. For UDR-NewFeatures and UDR-Regression pipelines, copy the root certificate authority (CA), signed server certificate with root CA private key, and private keys used to create secret for TLS support on Provisioning Gateway to `ocslf_tests/certs` folder in the path as follows:

> **Note**
>
> For TLS validation, use the same set of copied certificates for Provisioning Gateway Ingress Gateway, UDR Ingress Gateway, and UDR Egress Gateway.

   a. `kubectl cp <root CA> <namespace>/<pod-name>:/var/lib/jenkins/ocudr_tests/certs`
   Example: `kubectl cp caroot.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocudr_tests/certs`

   b. `kubectl cp <server certificate> <namespace>/<pod-name>:/var/lib/jenkins/ocudr_tests/certs`
   Example: `kubectl cp apigatewayrsa.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocudr_tests/certs`

   c. `kubectl cp <private key> <namespace>/<pod-name>:/var/lib/jenkins/ocudr_tests/certs`
   Example: `kubectl cp rsa_private_key_pkcs1.pem ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocudr_tests/certs`

> **Note**
>
> The Ingress Gateway of UDR should use the above three files as part of TLS support.

3. For EIR-NewFeatures and EIR-Regression pipelines, copy the root certificate authority (CA), signed server certificate with root CA private key, and private keys used to create secret for TLS support on UDR ingressgateway-prov and ingressgateway-sig to `oceir_tests/certs` folder in the path as follows:

   a. `kubectl cp <root CA> <namespace>/<pod-name>:/var/lib/jenkins/oceir_tests/certs`
   Example: `kubectl cp caroot_sig.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/certs`

   b. `kubectl cp <server certificate> <namespace>/<pod-name>:/var/lib/jenkins/oceir_tests/certs`
   Example: `kubectl cp apigatewayrsa_sig.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/certs`

   c. `kubectl cp <private key> <namespace>/<pod-name>:/var/lib/jenkins/oceir_tests/certs`
   Example: `kubectl cp rsa_private_key_pkcs1_sig.pem ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/certs`

   d. `kubectl cp <root CA> <namespace>/<pod-name>:/var/lib/jenkins/oceir_tests/certs`

Copyright © 2019, 2025, Oracle and/or its affiliates.

Example: `kubectl cp caroot_prov.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/certs`

e. `kubectl cp <server certificate> <namespace>/<pod-name>:/var/lib/jenkins/oceir_tests/certs`
Example: `kubectl cp apigatewayrsa_prov.cer ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/certs`

f. `kubectl cp <private key> <namespace>/<pod-name>:/var/lib/jenkins/oceir_tests/certs`
Example: `kubectl cp rsa_private_key_pkcs1_prov.pem ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/certs`

4. To run OAuth2 validation scenarios on SLF-Regression, copy each private keys in pem format to oauth_keys folder in `/var/lib/jenkins/ocslf_tests` path.

```
kubectl cp <private key pem file> <namespace>/<pod-name>:/var/lib/jenkins/ocslf_tests/oauth_keys
```

Example:

```
kubectl cp ec_private_key1.pem ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocslf_tests/oauth_keys
```

5. To run OAuth2 validation scenarios on UDR-Regression, copy each private keys in pem format to oauth_keys folder in `/var/lib/jenkins/ocudr_tests` path.

```
kubectl cp <private key pem file> <namespace>/<pod-name>:/var/lib/jenkins/ocudr_tests/oauth_keys
```

Example:

```
kubectl cp ec_private_key1.pem ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/ocudr_tests/oauth_keys
```

6. To run OAuth2 validation scenarios on EIR-Regression, copy each private keys in pem format to oauth_keys folder in `/var/lib/jenkins/oceir_tests` path.

```
kubectl cp <private key pem file> <namespace>/<pod-name>:/var/lib/jenkins/oceir_tests/oauth_keys
```

Example:

```
kubectl cp ec_private_key1.pem ocudr/ats-ocats-udr-696df7c84d-2qc7h:/var/lib/jenkins/oceir_tests/oauth_keys
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

4 Running NF Test Cases Using ATS

This section describes how to run NF test cases using ATS. It includes:

- Running BSF Test Cases using ATS
- Running NRF Test Cases using ATS
- Running NSSF Test Cases using ATS
- Running Policy Test Cases using ATS
- Running SCP Test Cases using ATS
- Running SEPP Test Cases using ATS
- Running UDR Test Cases using ATS

4.1 Running BSF Test Cases using ATS

This section describes how to run BSF test cases using ATS.

> **Note**
>
> Restart the NRF-client pod of BSF for UDR and CHF discovery as part of each test case.

4.1.1 Prerequisites

To run Binding Support Function test cases, ensure that the following prerequisites are met:

- Deploy BSF 25.1.200 with default helm configurations using helm charts to run all test cases. The ATS version must be compatible with BSF 25.1.200.
  For more information on how to install BSF, see *Oracle Communications Cloud Native Core, Binding Support Function Installation, Upgrade, and Fault Recovery Guide*.
- Go-STUB must be installed in the same namespace where ocbsf is installed.
- Add the following to Kubernetes namespace to grant role access:

```
PolicyRule:
  Resources               Non-Resource URLs  Resource Names  Verbs
  ---------               -----------------  --------------  -----
  pods/log                []                 []              [get list]
  configmaps              []                 []              [watch get
list delete update create]
  pods                    []                 []              [watch get
list delete update create]
  secrets                 []                 []              [watch get
list delete update create]
  services                []                 []              [watch get
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
list delete update create]
  deployments.apps        []                []              [watch get
list update]
  replicasets.apps        []                []              [watch get
list update]
  deployments.extensions  []                []              [watch get
list update]
  replicasets.extensions  []                []              [watch get
list update]
```

- ATS Prometheus metrics validation works only when:
  - the metrics suffixes are not configured
  - installation has a single pod for each microservice in the BSF deployment
- You can customize test cases in the custom test case folders (cust_newfeatures, cust_regression and cust_performance). You can add new test cases, remove unwanted test cases and modify existing test cases. It does not impact the original product packaged test cases available in the newfeatures, regression and performance folders respectively. For more details about custom test case folders, see Custom Folder Implementation.
- Install Prometheus server in the cluster.
- Database cluster is in the running state with all required tables. Verify that there are no previous entries in the database before running test cases.
- Do not initiate a job in two different pipelines at the same time.
- For running ATS features, ensure to update the following mandatory parameters in ocbsf_custom_values_25.1.200.yaml file only when you are not using the minimal custom values.yaml file.

```
logging:
  burst:
    rate: 500
    max: 3000
    onMismatch: DENY
    logLevel: DEBUG
```

> **Note**
>
> Please ensure that you use the latest version of the Custom Values file when installing BSF initially.

- For using Controlled Shutdown feature, ensure that `enableControlledShutdown` parameter is enabled on BSF during installation. If this parameter is not enabled, the test case for this feature will fail.
- In the `application-config` configmap, configure the following parameters with the respective values:
  - `primaryNrfApiRoot=nf1stub.<namespace_gostubs_are_deployed_in>.svc:8080`

    For example:

    `primaryNrfApiRoot=nf1stub.ocats.svc:8080`
  - `secondaryNrfApiRoot=nf11stub.<namespace_gostubs_are_deployed_in>.svc:8080`

    For example:

Copyright © 2019, 2025, Oracle and/or its affiliates.

```
secondaryNrfApiRoot=nf11stub.ocats.svc:8080
```

- `virtualNrfFqdn = nf1stub.<namespace_gostubs_are_deployed_in>.svc`

  For example:

  ```
  virtualNrfFqdn=nf1stub.ocats.svc
  ```

- `retryAfterTime=PT30S`

- To enable ATS BSF GUI with the HTTPS protocol, the above mentioned `application-config` configmap related following parameters should be configured with the respective values:

```
# Please edit the object below. Lines beginning with '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
  profile: | -
    [appcfg]
    primaryNrfApiRoot=nf1stub.ocats.svc:8443
    secondaryNrfApiRoot=nf11stub.ocats.svc:8443
    nrfScheme=https
    virtualNrfPort=8443
    virtualNrfScheme=https
```

- Before running ATS Test suite, restart `nrf-client-nfdiscovery` and `nrf-client-nfmanagement` pods.

  - Run the following command to get all the configmaps in your namespace.

    ```
    kubectl get configmaps -n <BSF_namespace>
    ```

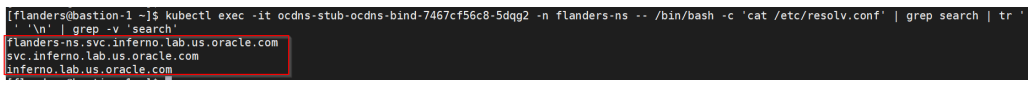
  - Edit the `alternate route service` deployment pointing towards DNS Stub.

    Run the following command to get searches information from `dns-bind` pod to enable communication between Alternate Route and `dns-bind` service.

    ```
    kubectl exec -it <dns-bind pod> -n <NAMESPACE> -- /bin/bash -c
    'cat /etc/resolv.conf' | grep search | tr ' ' '\n' | grep -v 'search'
    ```

    Example:

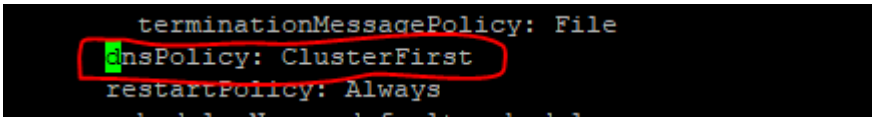
Figure 4-1 Editing Alternate Route Service deployment pointing towards DNS Stub



By default, Alternate Route Service points to CoreDNS.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-2 Alternate Route Service settings in deployment file



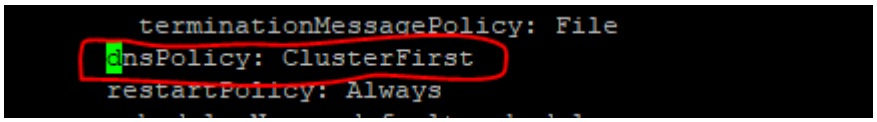
Change the deployment file to add following content in alternate service to query DNS stub.

```
kubectl edit deployment ocbsf-occnp-alternate-route -n ocbsf
```

* Add the nameservers IPaddress.
* Add all the search information.
* Set dnsPolicy to `None`.

```
dnsConfig:
  nameservers:
   - <dns_stub_cluster_ip_address>
   searches:
   - dns-bind search
   - dns-bind search
   - dns-bind search
   dnsPolicy: None
```

Figure 4-3 dnsConfig



– If Service Mesh check is enabled, create a destination rule to fetch the metrics from the Prometheus. For destination rule to communicate between TLS enabled entity (ATS) and non-TLS entity (Prometheus), Prometheus is kept outside of the service mesh. The rule can be created as follows:

```
kubectl apply -f - <<EOF

apiVersion:networking.istio.io/v1alpha3
kind:DestinationRule
metadata:
  name:prometheus-dr
  namespace:ocats
spec:
  host:oso-prometheus-server.ocbsf.svc.cluster.local
  trafficPolicy:
    tls:
      mode:DISABLE
EOF
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

    - `name`: indicates the name of destination rule.
    - `namespace`: indicates where the ATS is deployed.
    - `host`: indicates the hostname of the prometheus server.

- For Bsf_To_Nrf_Late_Arrival feature file to be executed, nfInstanceId should be configured in Bsf_To_Nrf_Late_Arrival.yaml parameterization file. This nfInstanceId must be the same as the one under BSF nfInstanceId, that is configured during BSF installation. The nfInstanceId must to be configured in ATS UI under pipeline configuration (BSF_NFINSTANCE_ID).
- To ensure consistent functioning of ATS related to the audit service, modify the Audit deployment by reducing the value of AUDIT_NOTIFY_SCHEDULER_POLLING_INTERVAL_MILLISEC from 30000 to 1000 (changing from 30 seconds to 1 second).

  ```
  $ kubectl get deploy -n ocbsf | grep 'audit' ocbsf-ocpm-audit-service
  1/1     1      1     10h


  $ kubectl edit deploy ocbsf-ocpm-audit-service -n ocbsf
  ```

  The deployment will open. Scroll down until below fields are seen

  ```
  - name:  AUDIT_NOTIFY_SCHEDULER_POLLING_INTERVAL_MILLISEC value: "30000"
  ```

  Update the value of AUDIT_NOTIFY_SCHEDULER_POLLING_INTERVAL_MILLISEC to 1000 if it isn't already.

Application Config map changes for BSF registrations over TLS

To enable TLS communication make the following changes by editing the config map of application-config:

- NRF port from 8080 to 8443
- nrfScheme to https

Sample:

```
apiVersion: v1
data:
  profile: |-
    [appcfg]
    primaryNrfApiRoot=nf1stub.ocats.svc:8443
    secondaryNrfApiRoot=nf11stub.ocats.svc:8443
    nrfScheme=https
    virtualNrfPort=8443
    virtualNrfScheme=https
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> In the config map of application-config, delete the lines which has *supportedDataSetId* or *secondaryNrfApiRoot* strings.

4.1.2 Logging into ATS

Before logging into ATS GUI, it is important to get the worker node external IP and node port of the service, 'ocats-bsf'.

Run the following command to get the external IP for the worker node:

Example:

```
kubectl get nodes -owide
```

The output of the command is:

```
ocbsf-k8s-node-1        Ready    <none>   111d   v1.16.7   192.168.200.26
10.75.152.111   Oracle Linux Server 7.8
4.14.35-1902.303.5.3.el7uek.x86_64   containerd://1.2.10
```

Run the following command to get the nodeport:

```
kubectl get svc -n <BSF_namespace>
```

Example:

```
kubectl get svc -n ocbsf
```

The output of the command is:

```
ocbsf-ocats-ocats-bsf     LoadBalancer   10.233.53.144   10.75.225.49
8080:31944/TCP      19h
```

To log in to ATS, open the web browser and type the following URL:

http://<Worker-Node-IP>:<Node-Port-of-ATS>

If the 'ocats-bsf' Service has an external IP available, <SVC external IP> can also be used to log in to ATS.

```
http://<External IP of ATS Service>:8080
```

Example:

```
http://10.75.225.49:8080
```

Running ATS

To run ATS test cases, perform the following steps:

Copyright © 2019, 2025, Oracle and/or its affiliates.

1. Enter the **username** as `bsfuser` and **Password** as `bsfpasswd`.
2. Click **Sign in**.

> **Note**
>
> To modify default login password, see Modifying Login Password.

Figure 4-4 Pre-configured Pipelines



On successful log in, users should see the following pipelines:

- **BSF-NewFeatures:** This pipeline has all the new test cases delivered for BSF 25.1.200.
- **BSF-Performance:** This pipeline is not operational as of now. It is reserved for future releases of ATS.
- **BSF-HealthCheck:** This pipeline checks if BSF and ATS are deployed correctly. This shows only when the user has enabled this feature at the time of installing BSF ATS.
- **BSF-Regression:** This pipeline has all the test cases delivered in BSF ATS - 25.1.200.

4.1.3 Running BSF_NewFeatures Pipeline

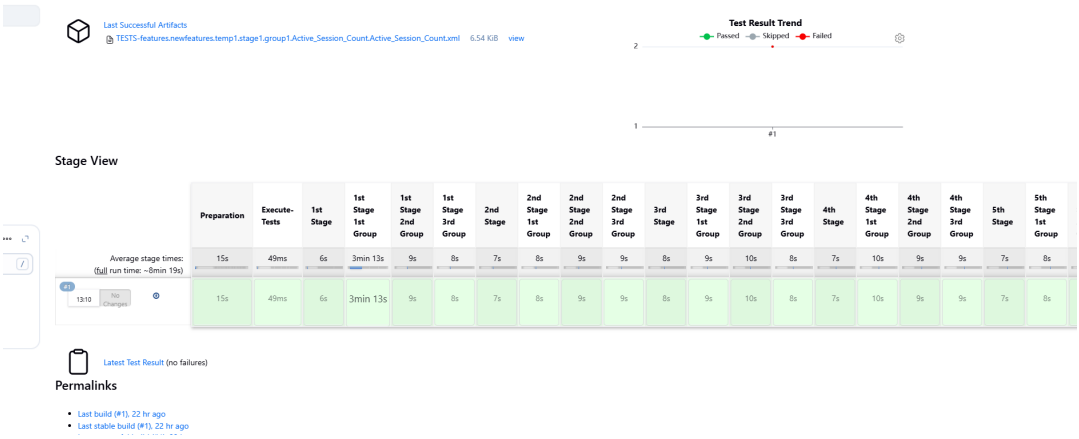
BSF_NewFeatures Pipeline

This is a pre-configured pipeline where users can run all the BSF new test cases. To configure its parameters, which is a one time activity, perform the following steps:

1. Click **BSF_NewFeatures** in the **Name** column and then, click **Configure** in the left navigation pane as shown below:

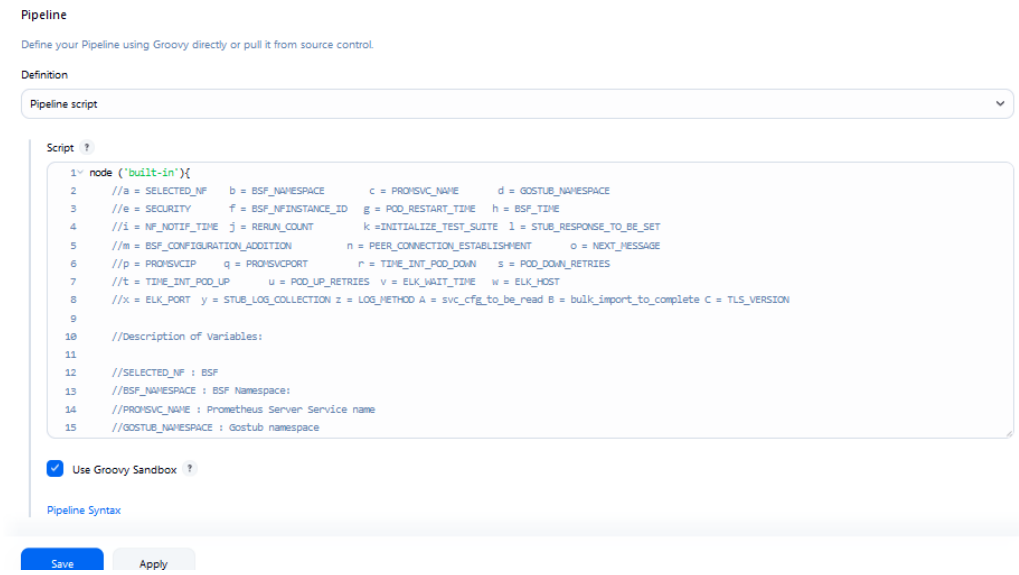
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-5 BSF New Feature Pipeline



2. The BSF_NewFeatures, **General** tab appears. Make sure that the screen loads completely.
3. Scroll-down to the end. The control moves from **General** tab to the **Pipeline** tab as shown below:

Figure 4-6 BSF New Features Pipeline Configuration



You can change the parameter values from **"a"** - to - **"x"** as per user requirement. The parameter details are available as comments from line number **2** - to - **7**. In the **Script** area of the Pipeline section, you can change the values of the following parameters:

- **a:** Name of the NF to be tested in capital (BSF)
- **b:** Change this parameter to update the namespace where BSF was deployed in your bastion.
- **c:** Name of Prometheus service namespace (occne-prometheus-server)
- **d:** Change this parameter to update the namespace where your gostubs are deployed in your bastion.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- **e:** Set this parameter as 'unsecure', if you intend to run ATS in TLS disabled mode. Else, set this parameter as 'secure'.
- **f:** Configure this parameter to set BSF_NFINSTANCE_ID.
- **g:** Set a value more than 45 seconds for this parameter. The default wait time for the pod to come up is 45 seconds. Every TC requires restart of the nrf-client-management pod.
- **h:** Set a value more than 60 seconds for this parameter. The default wait time to add a configurations to the database is 60 secs.
- **i:** Set this parameter to more than 140 secs. The default wait time for Nf_Notification Test Cases is given as 140 secs.
- **k:** Use this parameter to set the waiting time to initialize Test Suite.
- **l:** Use this parameter to set the waiting time to get response from Stub.
- **m:** Use this parameter to set the waiting time after adding BSF Configuration.
- **n:** Use this parameter to set the waiting time for Peer connection establishment.
- **o:** Use this parameter to set the waiting time before sending next message.
- **p:** Use this parameter to set Prometheus Server IP.
- **q:** Use this parameter to set Prometheus Server Port.
- **r:** Use this parameter to set the interval after which the POD status is checked when it is down.
- **s:** Use this parameter to set number of retry attempt in which will check the pod down status
- **t:** Use this parameter to set the interval after which we check the POD status if its UP.
- **u:** Use this parameter to set number of retry attempt in which will check the pod up status.
- **v:** Use this parameter to set Wait time to connect to Elastic Search.
- **w:** Use this parameter to set Elastic Search HostName.
- **x:** Use this parameter to set Elastic Search Port.
- **y:** Use this parameter to set To Enable/Disable Stub logs collection.
- **z:** Use this parameter to set Log collection endpoint either Elasticsearch or Kubernetes.
- **A:** Use this parameter to set Timer to wait for importing service configurations.
- **B:** Use bulk_import_to_complete to add custom time in Jenkins post bulk imports.
- **C:** Use this parameter to set TLS version (1.2 or 1.3). The default value is 1.2.

(*Optional*)To collect application logs per failed scenario, user can configure the values for the following parameters:

- **z:** If you want log collection to happen through Elastic search, set the value for this parameter as `Elasticsearch`. If not, specify the value as `Kubernetes`.
  If you want to collect logs through Elastic search, it is required to configure the values for the following parameters:
  - **v:** Specifies the wait time to connect to Elastic search (`ELK_WAIT_TIME`).
  - **w:** Specifies the host name of Elastic search (`ELK_HOST`). For example, `occne-elastic-elasticsearch-master.occne-infra/`

Copyright © 2019, 2025, Oracle and/or its affiliates.

  - **x:** Specifies the port for Elastic search (`ELK_PORT`). For example, 9200.
- **y:** If you want to collect stub logs, set the value for this parameter as `yes`. If not, specify the value as `no`.

4. Click **Save** after updating the parameter values. The **BSF_NewFeatures** Pipeline page appears.

> **Note**
>
> It is recommended to save a copy of the pipeline script in your local machine that you may refer while restarting ATS pods.

> **NOT_SUPPORTED**
>
> Do not modify anything other than the parameter values described in this section.

Extracting Application Logs

If any scenarios fail after running the pipeline job, perform the following steps:

1. Log in to the ATS pod:

   ```
   kubectl exec -it pod/ocats-bsf-6f6dfc76b5-jbgzt -n ocbsf bash
   ```

2. Go to Jenkins build directory:

   ```
   cd $JENKINS_HOME/jobs/$JOB_NAME/builds/$BUILD_NUMBER/
   ```

   For example:

   ```
   cd /var/lib/jenkins/.jenkins/jobs/BSF_Regression/builds/2
   ```

3. Extract the `applogs.zip` file:

   ```
   unzip applogs.zip
   ```

4. After successfully unzipping the file, open the `applog` folder to view the pod logs for failed scenarios:

   ```
   (env) [jenkins@ocats-bsf-6f6dfc76b5-jbgzt applog]$ pwd
   /var/lib/jenkins/.jenkins/jobs/BSF_Regression/builds/2/applog
   (env) [jenkins@ocats-bsf-6f6dfc76b5-jbgzt applog]$ ls -ltrh
   total 760K
   -rw-r--r--. 1 jenkins jenkins 250K Nov 19 11:08 Initial_Run-
   Register_BSF_With_NFSetIDList.log
   -rw-r--r--. 1 jenkins jenkins 249K Nov 19 11:08 1st_Rerun-
   Register_BSF_With_NFSetIDList.log
   -rw-r--r--. 1 jenkins jenkins 255K Nov 19 11:09 2nd_Rerun-
   Register_BSF_With_NFSetIDList.log
   ```

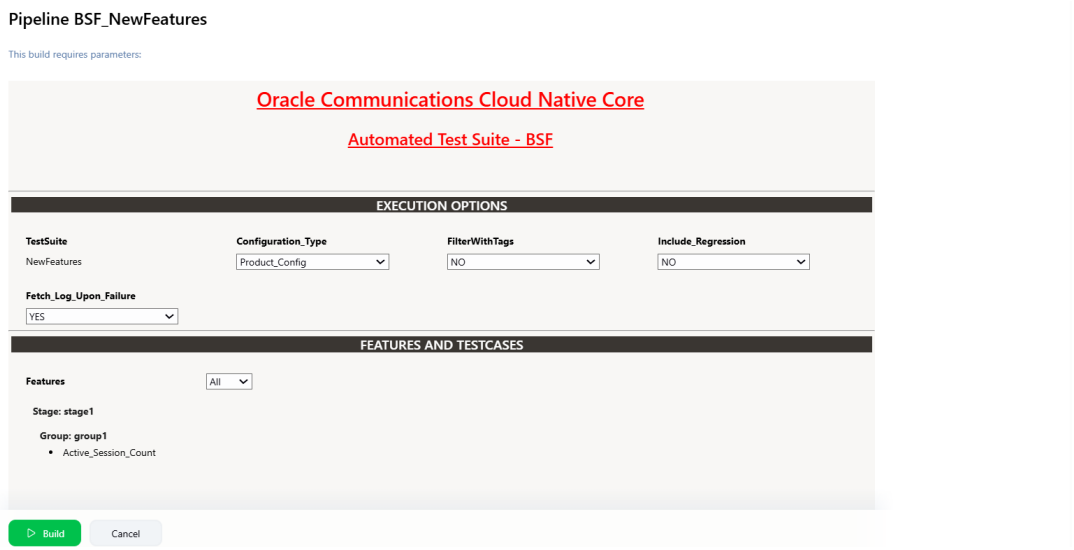

Copyright © 2019, 2025, Oracle and/or its affiliates.

Running BSF Test Cases

To run BSF test cases, perform the following steps:

1. Click the **Build with Parameters** link available in the left navigation pane of the **BSF_NewFeatures** Pipeline screen. The following page appears.

Figure 4-7 BSF New Features Build with Parameters



> **Note**
>
> Make sure that the value of **FilterWithTags** and **Include_NewFeatures** is selected as **NO**.

2. If you want to collect logs for any given build, select YES from the drop-down menu of **Fetch_Log_Upon_Failure**.
3. Click **Build** and select **Console Output** to view the test results. The following is a sample test result ouput:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-8 Sample: Test Result Output in Console



> **Note**
>
> For more details on consolidated test report, see Managing Final Summary Report, Build Color, and Application Log.

Queuing Jenkins Jobs

Using this feature, you can queue a second job even when current job is still running. The second job can be triggered either from the same or a different pipeline.

The following table lists various scenarios on how queuing works:

Table 4-1 Queuing Jenkins Jobs

Concurrent Builds	New Features Current Build	New Features New Build	Regression Current Build	Regression New Build Build	Result
Enabled	Running	Triggered	NA	NA	New-Build of New-Features is added to queue.
Enabled	Running	NA	NA	Triggered	New-Build of Regression is added to queue.
Disabled	NA	NA	Running	Triggered	New-Build of Regression is added to queue.
Disabled	NA	Triggered	Running	NA	New-Build of New-Features is added to queue.

Copyright © 2019, 2025, Oracle and/or its affiliates.

4.1.4 Running BSF_HealthCheck Pipeline

This is a pre-configured pipeline where ATS performs a test probe with SUT. It triggers helm test and provides the results in Jenkins Console Logs.

You can run BSF_HealthCheck pipeline to check if all BSF pods are up and running. If yes, it provides the status as successful. If any pod is down due to any reason, then the pipeline fails.

To configure the parameters for **BSF_HealthCheck** pipeline, perform the following steps:

1. Click **BSF_HealthCheck** in the **Name** column.
2. Click **Configure** in the left navigation pane.
3. When you scroll-down, the **General** tab becomes active. Be sure that the screen loads completely.
4. Continue scrolling down until the **Pipeline** tab becomes active. The following is a screen capture that shows the Pipeline script:

Figure 4-9 Helm Test Script



5. In the `Script` area under the Pipeline section, users may customize the values for the following parameters:

> **NOT_SUPPORTED**
>
> Do not modify values of parameters other than the ones described in this section.

- **a:** Change this parameter to update the helm release name where BSF is deployed in your bastion.
- **b:** Change this parameter to update the namespace where BSF is deployed in your bastion.

6. Click **Save** to update the values.

Running Helm Test

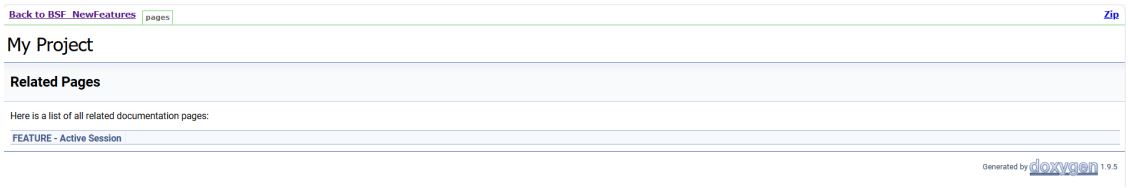
To run BSF test cases, click **Build Now**.

4.1.5 BSF_NewFeatures Documentation

To view BSF functionalities, go to **BSF_NewFeatures** pipeline. and click **Documentation** link in the left navigation pane.

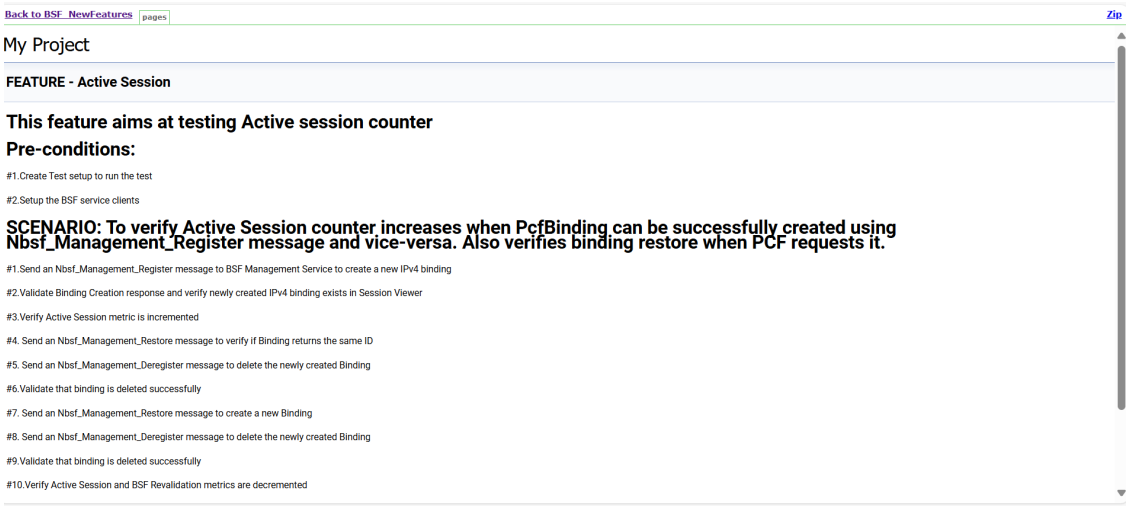
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-10 BSF_NewFeatures Feature List Documentation



Click any functionality to view its test cases and scenarios of each test case. For example, when you click **FEATURE - BSF_Error_Response_Enhancements**, the following test description appears:

Figure 4-11 Test Cases and Scenarios of Feature - BSF_Error_Response_Enhancements



Based on the functionalities covered under **Documentation**, the **Build Requires Parameters** screen displays test cases. To navigate back to the pipeline BSF-NewFeatures screen, click **Back to BSF_NewFeatures** link available on top left corner of the screen.

Test Result Analyzer

Using the Test Result Analyzer plug-in available in ATS, user can view consolidated and detailed reports. For more information, see Test Results Analyzer section.

Test Case Mapping to Features and Display Total Counts

With this feature, users can view Total Count of Features, TestCases/Scenarios and TestCase mapping to each Feature of BSF in ATS View. For more information, see Support for Test Case Mapping and Count section.

Stub Predefined_priming Support

Stub Predefined_priming configuration in ATS enables ATS to respond back with the payload message instead of default message when prime configuration does not match with the feature level priming or when the sub is not primed.

Copyright © 2019, 2025, Oracle and/or its affiliates.

When the Predefined_priming is not configured and when a request is received at the stub that mimics NRF:

1. Stub checks against the feature level prime configuration.
2. If a match is found in feature level prime, stub replies with the payload message.
3. If there is no match found, stub replies with the default response.

```
stub_log: No match found in prime configuration for sending the response.
Sending default - 200
ATS log: {default_response}
```

When the Predefined_priming is configured and when a request is received at the stub that mimics NRF:

1. Stub checks against the feature level prime configuration.
2. If a match is found in feature level prime, feature level prime is used for responding to the request.
3. If a match is not found in the feature level prime, but found in the pre-configured prime, pre-configured prime is used for responding to the request.
4. If a match is found in both pre-primed as well as feature level prime, feature level prime configuration is given priority and the same is used for responding to the request.
5. If a match is not found in both pre-primed as well as feature level prime, stub sends a default response.

4.1.6 Running BSF Regression Pipeline

This section describes how to run test cases for Binding Support Function (BSF) Regression pipeline.

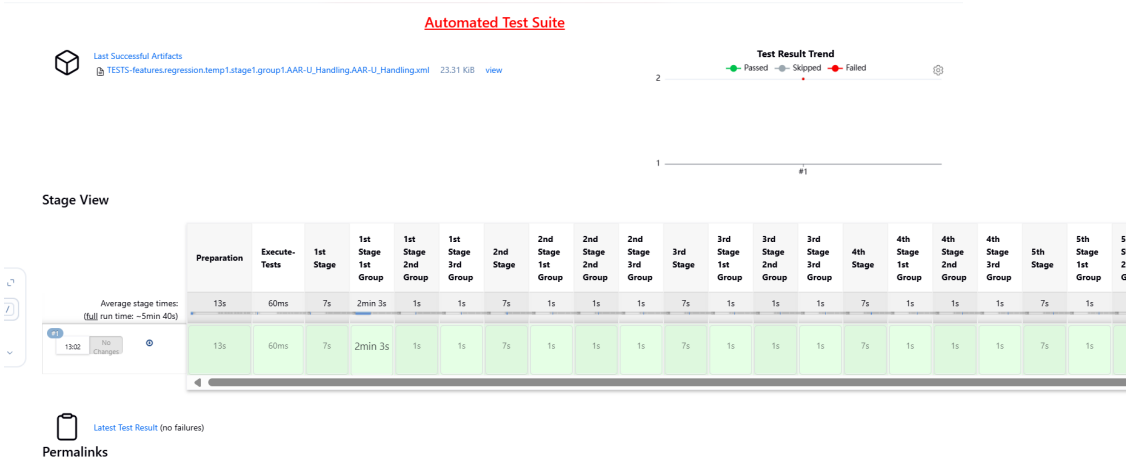
The **BSF_Regression** pipeline is a pre-configured pipeline where all the test cases of previous releases are available. For example, for BSF 25.1.200, this pipeline has all the test cases released till BSF 25.1.200.

To view **BSF_Regression** pipeline details, perform the following steps:

1. Click **BSF_Regression** in the **Name** column.
2. Click **Build with Parameters** in the left navigation pane.
3. Copy the required test cases that are available in the BSF folder and place them appropriately within the **custom** folder for BSF_Regression.
4. Reload the page to view the test cases available in the **custom** Regression folder.
5. Click **Build**.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-12 BSF_Regression Pipeline



> **Note**
>
> The Regression pipeline does not have any sanity option. However, users must perform all the steps performed in the **BSF_NewFeatures** pipeline. Ensure that the pipeline script is configured according to the environment variables.

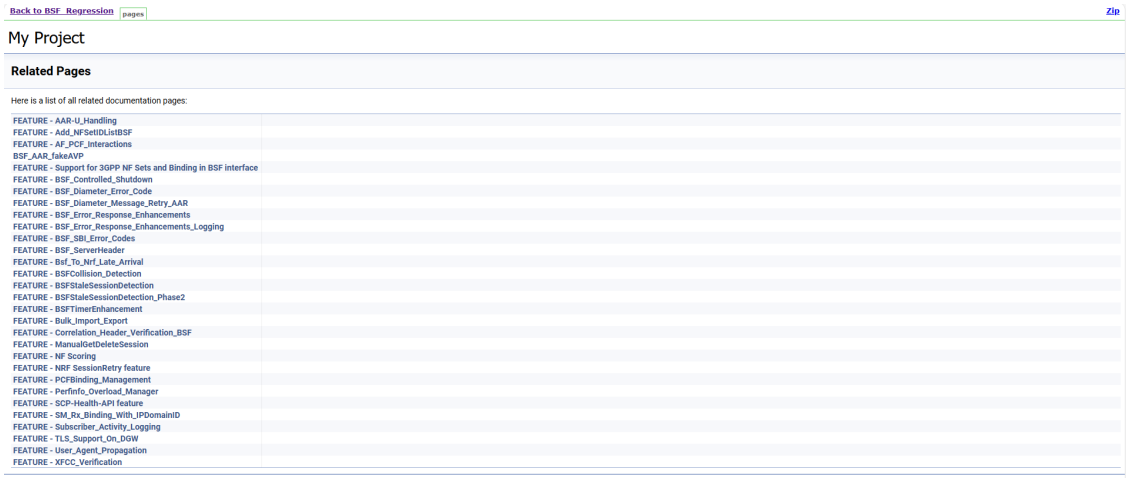
4.1.7 BSF_Regression Documentation

This section describes the documentation for **BSF_Regression** pipeline.

To view the documentation for any of the BSF features, on the ATS home page, click BSF_Regression. Then, click **Documentation** in the left navigation pane.

This page shows features of only those test cases that are released in previous releases.

Figure 4-13 BSF_Regression Features Documentation



Copyright © 2019, 2025, Oracle and/or its affiliates.

4.2 Running NRF Test Cases using ATS

4.2.1 Prerequisites

To run NRF test cases using NRF ATS 25.1.203, you need to ensure that following prerequisites are met.

- For three-site georedundancy test cases, deploy NRF 25.1.203 on three sites with database replication enabled. The georedundancy feature must be enabled on all NRF sites before running these test cases.
- For two-site georedundancy test cases, deploy NRF 25.1.203 on two sites with database replication enabled. These test cases are run separately, as they require two different NRFs. They can also be run in a three-site georedundancy setup. The georedundancy feature must be enabled on all NRF sites before running these test cases.
- For NRF-Growth test cases, deploy two standalone NRF 25.1.203 instances. These test cases are run separately, as they require two different NRFs.
- All the ATS pipelines other than georedundancy, three-site georedundancy, and NRF Growth, must be run with only one NRF deployment (standalone NRF).
- The following NRF database tables should not have any entries:
  - nrfApplicationDB.NfInstances
  - nrfApplicationDB.NfStatusMonitor
  - nrfApplicationDB.NfSubscriptions
- NRF instances should be deployed with the default Helm and REST configurations, except for `ocnrfHost` and `ocnrfPort`**.**
- For NF-FQDN-Authentication-Feature test cases, deploy both NRF and NRF ATS, with the NF Authentication feature changes.
- All microservices of NRF must be up and running including **Alternate route** and **Artisan** microservices.
- To run Alerts test cases, configure NRF Alerts on the AlertManager on the Prometheus server. For more details on configuring alerts, see *Oracle Communications Cloud Native Core, Network Repository Function Installation, Upgrade, and Fault Recovery Guide*.
- Before deploying NRF, create RSA and ECDSA certificates and keys (public and private) for AccessToken microservice.
- Deploy NRF 25.1.203 with default helm configurations using helm charts to run all test cases except for **NF-FQDN-Authentication-Feature** cases.
- Ensure all NRF microservices are up and running including Accesstoken microservice. Keys with RSA and ECDSA algorithm are mandatory.
- Copy the ECDSA and RSA public keys to the ATS pod at **/var/lib/jenkins/ocnrf_tests/public_keys**, naming them **ec_public.pem** and **rs_public.pem** , respectively. Then, create a duplicate of **rs_public.pem** and save it as **new_public.pem** in the same directory.
- Create JSON files for key and certificate details and copy them to ATS at **/var/lib/jenkins/ocnrf_tests/** location. Refer to KID feature specific Prerequisite.
- Deploy ATS using helm charts.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- For NRF 25.1.203, deploy five stub servers for running Roaming, SLF, Forwarding, and SLF via SCP functionality test cases. The service name for the stub servers must be **notify-stub-service**, **notify-stub-service02**, **notify-stub-service03**, **amf1-cluster1-net2-amf-5gc-mnc016-mcc310-3gppnetwork-org**, and **slf-stub-service01**.
- The **slf-stub-service01** stub must be deployed with **service port as 80**. For more information, see Installing ATS for NRF.
- Replace the existing content with the following content in the alternate-route section of the `ocnrf_custom_values.yaml` file:
  The following configuration changes are required to accommodate alternate routing test cases based on the latest configuration.

```
staticVirtualFqdns:
    - name: https://abc.test.com
      alternateFqdns:
        - target: notify-stub-service03
          port: 8080
          priority: 10
          weight: 20
        - target: notify-stub-service03
          port: 8080
          priority: 20
          weight: 30
    - name: http://xyz.test.com
      alternateFqdns:
        - target: notify-stub-service03
          port: 8080
          priority: 10
          weight: 20
        - target: notify-stub-service03
          port: 8080
          priority: 20
          weight: 30

  dnsSrvEnabled: false
```

- Ensure Prometheus service is up and running.
- Deploy ATS and Stubs in the same namespace as NRF, as default ATS deployment is with role binding. In addition, deploy test stubs in the same namespace as NRF.
- User must not initiate a job in two different pipelines at the same time.
- User must not abort a running job, this may lead to data corruption in database.
- For getting PVC support to retain NRF ATS environment variables and pipeline console histories, follow the installation step. By default PVC support is disabled.
- If installation is done with Service mesh, the NRF must be deployed with below annotation in **lbDeployments** and **nonlbDeployments** section:

```
oracle.com/cnc: "true"
traffic.sidecar.istio.io/excludeInboundPorts: "9090,8095,8096,7,53"
traffic.sidecar.istio.io/excludeOutboundPorts: "9090,8095,8096,7,53"
```

  where, first one is for connecting to Operations Services Overlay (OSO) for Alerts test cases. Second one is for connecting to ATS for fetching the metrics from pods.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- If Service Mesh is enabled, then create a destination rule to fetch the metrics from the Prometheus. In most of the deployments, Prometheus is kept outside the service mesh and a destination rule is required to communicate between TLS enabled entity (ATS) and non-TLS enabled entity (Prometheus).
  To create a rule:

  ```
  kubectl apply -f - <<EOF
  apiVersion:networking.istio.io/v1alpha3
  kind:DestinationRule
  metadata:
    name:prometheus-dr
    namespace:ocnrf
  spec:
    host:oso-prometheus-server.ocnrf.svc.cluster.local
    trafficPolicy:
      tls:
        mode:DISABLE
  EOF
  ```

  In the above rule,

  - **name** indicates the name of destination rule.
  - **namespace** indicates where the ATS is deployed.
  - **host** indicates the hostname of the Prometheus server.
- In case NRF is deployed with network policy enabled, then before running NRF-ATS a new network policy must be created to access NRF-ATS Jenkins GUI. Following is the sample:

  ```
  kubectl apply -f nodePort.yaml -n ocnrf
  apiVersion: networking.k8s.io/v1
  kind: NetworkPolicy
  metadata:
    name: allow-from-node-port
  spec:
    podSelector:
      matchLabels:
        app: ocats-nrf
    policyTypes:
    - Ingress
    ingress:
    - ports:
      - protocol: TCP
        port: 8080
  ```

- To run with CNE HA Prometheus, update the **prom_type** value in `product_config/global.yaml` and `custom_config/global.yaml` file to "**cne**".

Custom Folder Implementation

ATS provides custom test case folders (cust_newfeatures, cust_regression, and cust_performance) using which you can add new test cases, remove unwanted test cases, and modify existing test cases. It does not impact the original product packaged test cases available in the newfeatures, regression and performance folders. For more details, refer to Custom Folder Implementation.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Apart from cust_newfeatures and cust_regression, NRF ATS has cust_FQDNauthentication, cust_GeoRedundancy, and cust_3SiteGeo directories that contains test cases related to NF-FQDN and georedundancy respectively.

Key Identifier (KID) feature specific Prerequisite

User should create the following files with exact names and update the file content as per the naming convention used for keys and certificate along with secret name and namespace. The file names are:

- set_keyDetailsList.json
- set_currenKeyId.json
- tokenSigningDetails.json

Sample Files

```
$ cat set_keyDetailsList.json
{
        "tokenSigningDetails": {
                "defaultK8SecretDetails": {
                        "k8SecretNameSpace": "ocnrf",
                        "k8SecretName": "ocnrfaccesstoken-secret"
                },
                "keyDetailsList": [{
                                "keyID": "EcdsaKid",
                                "algorithm": "ES256",
                                "privateKey": {
                                        "k8SecretName": "ocnrfaccesstoken-
secret",
                                        "k8SecretNameSpace": "ocnrf",
                                        "fileName": "ecdsa_private_key.pem"
                                },
                                "certificate": {
                                        "k8SecretName": "ocnrfaccesstoken-
secret",
                                        "k8SecretNameSpace": "ocnrf",
                                        "fileName": "ecdsa_certificate.crt"
                                }
                        },
                        {
                                "keyID": "RsaKid",
                                "algorithm": "RS256",
                                "privateKey": {
                                        "k8SecretName": "ocnrfaccesstoken-
secret",
                                        "k8SecretNameSpace": "ocnrf",
                                        "fileName": "rsa_private_key.pem"
                                },
                                "certificate": {
                                        "k8SecretName": "ocnrfaccesstoken-
secret",
                                        "k8SecretNameSpace": "ocnrf",
                                        "fileName": "rsa_certificate.crt"
                                }
                        },
                        {
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
                                "keyID": "newKey",
                                "algorithm": "RS256",
                                "privateKey": {
                                        "k8SecretName": "ocnrfaccesstoken-
secret",
                                        "k8SecretNameSpace": "ocnrf",
                                        "fileName": "rsa_private_key.pem"
                                },
                                "certificate": {
                                        "k8SecretName": "ocnrfaccesstoken-
secret",
                                        "k8SecretNameSpace": "ocnrf",
                                        "fileName": "rsa_certificate.crt"
                                }
                        }
                ]
        }
}
```

> **Note**
>
> Update the following parameters in the above files:
>
> - k8SecretName
> - k8SecretNameSpace
> - fileName (name of the private keys and corresponding certificates)
>
> For more information about parameter values, see *Oracle Communications Cloud Native Core, Network Repository Function Installation, Upgrade, and Fault Recovery Guide*.

After updating the files, copy them under the **/var/lib/jenkins/ocnrf_tests/** directory.

```
$ cat set_currenKeyId.json
{
        "tokenSigningDetails":
                {"currentKeyID":"EcdsaKid"
        }
}
```

> **Note**
>
> User need not update the `currentKeyID` name (for example: `EcdsaKid`), if the same file name as in `keyID` is used while configuring the `set_currenKeyId.json` file.

```
$ cat tokenSigningDetails.json
{
 "presentCurrentKey" : {
    "algorithm" : "ES256",
    "privateKey" : {
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
            "k8sSecretName" : "ocnrfaccesstoken-secret",
            "fileName" : "ecdsa_private_key.pem"
        },
        "certificate" : {
            "k8sSecretName" : "ocnrfaccesstoken-secret",
            "fileName" : "ecdsa_certificate.crt"
        }
    },
    "newCurrentKey" : {
        "algorithm" : "RS256",
        "privateKey" : {
            "k8sSecretName" : "ocnrfaccesstoken-secret",
            "fileName" : "rsa_private_key.pem"
        },
        "certificate" : {
            "k8sSecretName" : "ocnrfaccesstoken-secret",
            "fileName" : "rsa_certificate.crt"
        }
    }
}
```

```
$ cat set_currenKeyId.json
{
 "tokenSigningDetails":
   {"currentKeyID":"EcdsaKid"
   }
}
```

> **Note**
>
> User doesn't want to update the name of the key if the same name was used while configuring.

```
$ cat tokenSigningDetails.json
{
 "presentCurrentKey" : {
    "algorithm" : "ES256",
    "privateKey" : {
        "k8sSecretName" : "ocnrfaccesstoken-secret",
        "fileName" : "ecdsa_private_key.pem"
    },
    "certificate" : {
        "k8sSecretName" : "ocnrfaccesstoken-secret",
        "fileName" : "ecdsa_certificate.crt"
    }
 },
 "newCurrentKey" : {
    "algorithm" : "RS256",
    "privateKey" : {
        "k8sSecretName" : "ocnrfaccesstoken-secret",
        "fileName" : "rsa_private_key.pem"
    },
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    "certificate" : {
        "k8sSecretName" : "ocnrfaccesstoken-secret",
        "fileName" : "rsa_certificate.crt"
    }
 }
}
```

> **Note**
>
> Here, user should update k8SecretName fileName (name of private keys and the corresponding certificates.

> **Note**
>
> Update the following parameters in the above code snippet:
>
> - k8SecretName
> - fileName (name of the private keys and corresponding certificates)
>
> For more information about parameter values, see *Oracle Communications Cloud Native Core, Network Repository Function Installation, Upgrade, and Fault Recovery Guide*.

After updating all the files, copy them to the ATS pod using **"kubectl cp"** command. Refer to the following sample commands:

```
$ kubectl cp tokenSigningDetails.json ocats-ocats-nrf-58fc5dcbb9-
zjsks:/var/lib/jenkins/ocnrf_tests/tokenSigningDetails.json -n
ocnrf
$ kubectl cp set_keyDetailsList.json ocats-ocats-nrf-58fc5dcbb9-
zjsks:/var/lib/jenkins/ocnrf_tests/set_keyDetailsList.json -n ocnrf
$ kubectl cp set_currenKeyId.json ocats-ocats-nrf-58fc5dcbb9-zjsks:/var/lib/
jenkins/ocnrf_tests/set_currenKeyId.json -n ocnrf
$ kubectl cp ec_public.pem ocats-ocats-nrf-58fc5dcbb9-zjsks:/var/lib/jenkins/
ocnrf_tests/public_keys/ec_public.pem -n ocnrf
$ kubectl cp rs_public.pem ocats-ocats-nrf-58fc5dcbb9-zjsks:/var/lib/jenkins/
ocnrf_tests/public_keys/rs_public.pem -n ocnrf
$ kubectl cp rs_public.pem ocats-ocats-nrf-58fc5dcbb9-zjsks:/var/lib/jenkins/
ocnrf_tests/public_keys/new_public.pem -n ocnrf
```

Prerequisites to Run Testcases for the CCA-Header Feature

To run the pipeline for the CCA-Header feature, NRF must be deployed with the following changes:

1. During installation of NRF, the namespace, secret name and the `caroot.cer` file name must be provided in NRF's custom values, in ingress-gateway section, as follows:

   ```
   ingress-gateway:
     ccaHeaderValidation:
       k8SecretName: ocingress-secret
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    k8NameSpace: ocnrf
    fileName: caroot.cer
```

2. In NRF's custom values, `metadata.ccaHeaderValidation.enabled` should be set to **true** in the '**accesstoken_mapping**' object from the `ingress-gateway.routesConfig` list.

> **Note**
>
> If ccaHeaderValidation is enabled in NRF, only **CCA-Header** pipeline should be run. The other pipelines (Newfeatures, Regression, 2SiteGeo, 3SiteGeo, or FQDN-Authentication) will not work with the feature enabled.

```
ingress-gateway:
  routesConfig:
  - id: accesstoken_mapping
    uri: http://{{ template "accesstoken.service.fullname" . }}:
{{ template "accesstoken.service.port" . }}
    path: /oauth2/token
    order: 4
    filters:
      controlledShutdownFilter:
        applicableShutdownStates:
            - COMPLETE_SHUTDOWN
    metadata:
        ccaHeaderValidation:
          enabled: true
```

3. Create a secret with the `caroot.cer` file provided in the ocats_ocnrf_tests_jenkinsjobs_25.1.203.tgz, using the following command. This same `caroot.cer` file is required to be configured in the secret to run the CCA-Header cases.

   ```
   kubectl create secret generic <secret name> --from-file=<ca root file
   name> -n <namespace>
   ```

   For example:

   ```
   kubectl create secret generic ocingress-secret --from-file=caroot.cer -n
   ocnrf
   ```

4. Update the secret name from step 3 in **global.yaml** file in either **/var/lib/jenkins/ocnrf_tests/product_config** or **/var/lib/jenkins/ocnrf_tests/custom_config** based on the whether the pipeline will be run with **Product_Config** option or **Custom_Config** option, under the **ccaHeaderSecretName** parameter.

Prerequisites to Run Testcases for the NRF Growth Features

To run the pipeline for the NRF Growth feature, the following prerequisites have to be done.

1. Install two NRFs that are not georedundant with each other.

Copyright © 2019, 2025, Oracle and/or its affiliates.

2. The pipeline configuration has to be set to the below before running the pipeline. The values of options "**u**" to **"x"** have to be set to 0.1.

```
node ('master'){
    //a = SELECTED_NF    b = NF_NAMESPACE        c = FT_ENDPOINT      d =
GATEWAY_IP        e = GATEWAY_PORT
    //f = CONFIG_IP      g = CONFIG_PORT         h = STUB_IP          i =
STUB_PORT         j = NFINSTANCEID
    //k = PROMETHEUS_IP  l = PROMETHEUS_PORT     m = RERUN_COUNT      n =
PROMETHEUS_URI    o = HELM_RELEASE_NAME
    //p = MYSQL_HOST     q = PRIVILEGED_USER_SECRET_NAME              r =
DISCOVERY_WAIT_TIME
    //s = REPLICATION_STATUS_URI                 t = CLUSTER_DOMAIN   u =
REGISTER_WRITE_WAIT_TIME
    //v = SUBSCRIPTION_WAIT_TIME                 w =
ACCESSTOKEN_WAIT_TIME                  x = REGISTER_READ_WAIT_TIME
    sh '''
        sh /var/lib/jenkins/ocnrf_tests/preTestConfig.sh \
        -a NRF \
        -b occne-cndbtierone,occne-cndbtiertwo \
        -c ocnrf-ingressgateway.ocnrf.svc.cluster.local:80,ocnrf-1-
ingressgateway.ocnrf.svc.cluster.local:80 \
        -d ocnrf-ingressgateway.occne-cndbtierone,ocnrf-1-
ingressgateway.occne-cndbtiertwo \
        -e 80,80 \
        -f ocnrf-nrfconfiguration.occne-cndbtierone,ocnrf-1-
nrfconfiguration.occne-cndbtiertwo \
        -g 8080,8080 \
        -h notify-stub-service.occne-cndbtierone,notify-stub-
service02.occne-cndbtierone \
        -i 8080,8080 \
        -j 6faf1bbc-6e4a-4454-a507-a14ef8e1bc5c,6faf1bbc-6e4a-4454-a507-
a14ef8e1bc5d \
        -k oso-prom-svr.oso \
        -l 80 \
        -m 2 \
        -n /prometheus/api/ \
        -o ocnrf,ocnrf-1 \
        -p mysql-connectivity-service.occne-cndbtierone,mysql-connectivity-
service.occne-cndbtiertwo \
        -q privilegeduser-secret     \

        -r 0.5 \
        -s http://ocnrf-ocnrf-app-info:5906/status/category/
replicationstatus,http://ocnrf-1-ocnrf-app-info:5906/status/category/
replicationstatus \
        -t svc.cluster.local \
        -u 0.1 \
        -v 0.1 \
        -w 0.1 \
        -x 0.1
    '''
    if(env.Include_NewFeatures && "${Include_NewFeatures}" == "YES"){
        sh '''sh /var/lib/jenkins/common_scripts/merge_jenkinsfile.sh'''
        load "/var/lib/jenkins/ocnrf_tests/jenkinsData/Jenkinsfile-Merged"
    }
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    else{
        load "/var/lib/jenkins/ocnrf_tests/jenkinsData/Jenkinsfile-
Regression"
    }

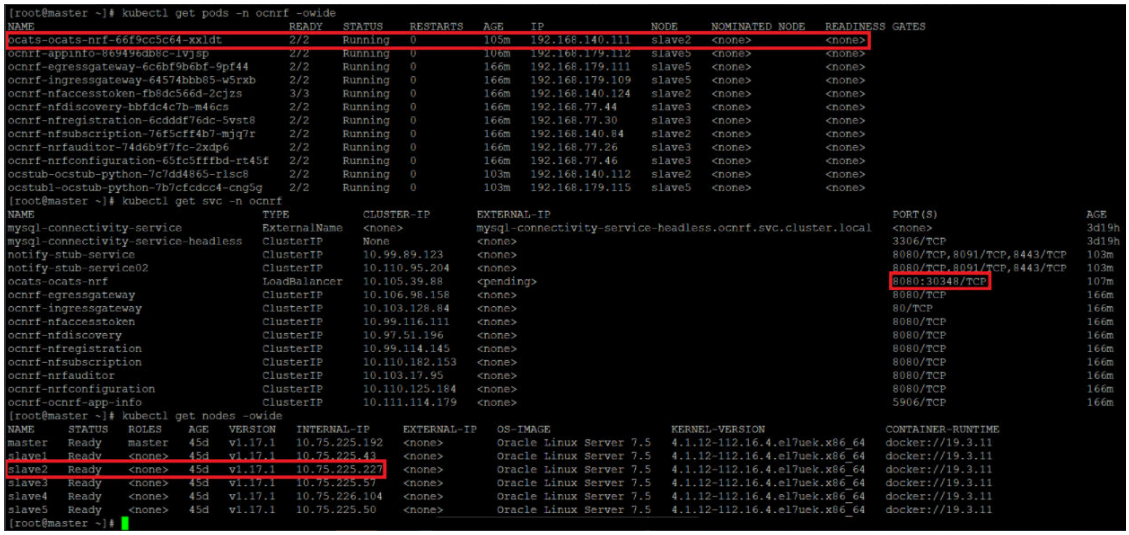
    }
}
```

4.2.2 Logging into ATS

Before logging into ATS, you need to ensure that ATS is deployed successfully using Helm charts as shown below:

> **Note**
>
> To modify default log in password, refer to Modifying Login Password.

Figure 4-14 Verifying ATS Pod



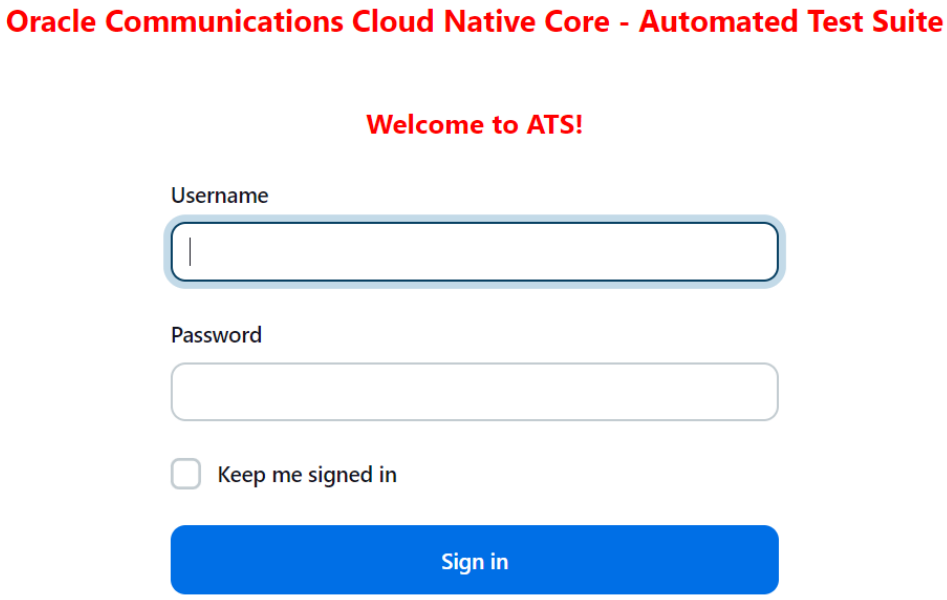
For more information on verifying ATS deployment, see Verifying ATS Deployment.

> **Note**
>
> In the **Verifying ATS Pod** screen, **slave2** is the node where ATS is deployed, **30348** is the ATS nodeport and **10.75.225.227** is the worker node IP, highlighted in red. For more details on ATS deployment, refer to Installing ATS for NRF.

To log in to ATS, open a browser and provide the IP Address and port details as <Worker-Node-IP>:<Node-Port-of-ATS>. As per above screen, it is **10.75.225.227:30348**.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-15 ATS Login

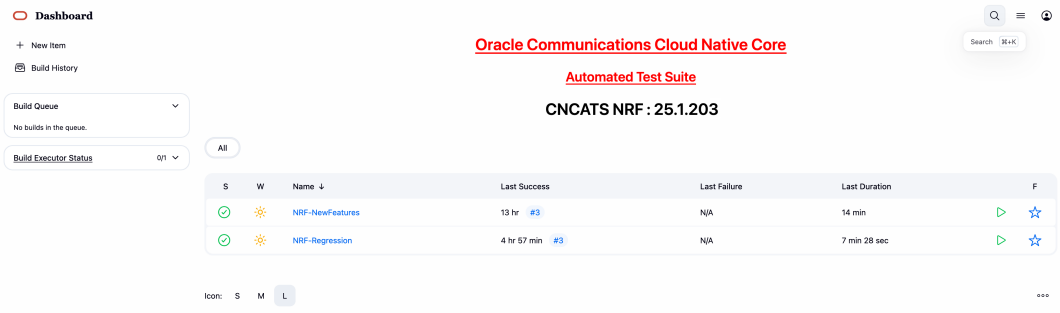


Running ATS

To run ATS:

1. Enter the login credentials. Click **Sign in**. The following screen appears.

Figure 4-16 NRF Pre-Configured Pipelines



NRF ATS has two pre-configured pipelines.

- **NRF-NewFeatures:** This pipeline has all the test cases delivered as part of NRF ATS - 25.1.203.

Copyright © 2019, 2025, Oracle and/or its affiliates.

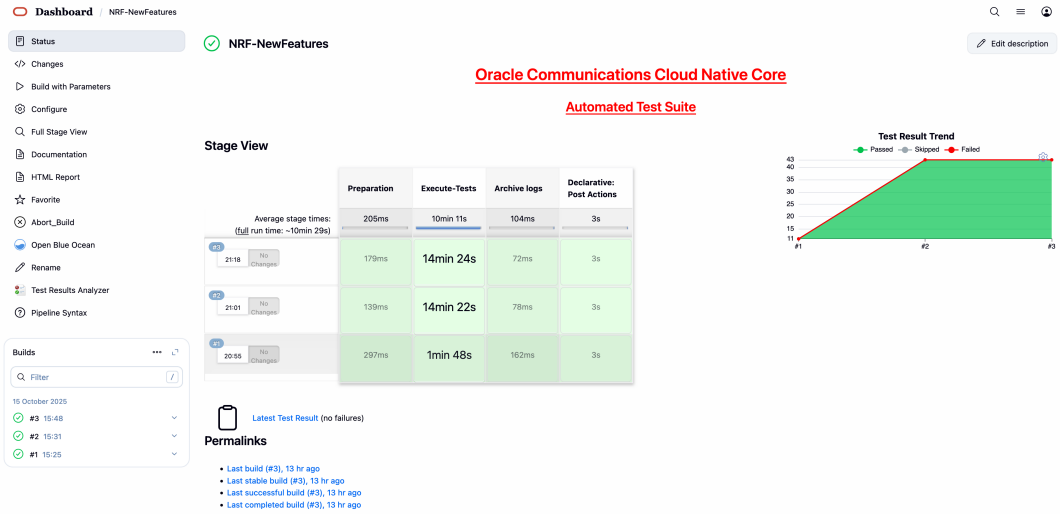
- **NRF-Regression:** This pipeline has all the test cases delivered so far in the previous releases.

4.2.3 NRF-NewFeatures Pipeline

After identifying the NRF pipelines, configure ATS as one-time activity as per NRF deployment. In this pipeline, all the new testcases related to NRF are deployed. To configure its parameters:

1. Click **NRF-NewFeatures** in the **Name** column. The following screen appears:

Figure 4-17 Configuring NRF-NewFeatures



In the above screen:

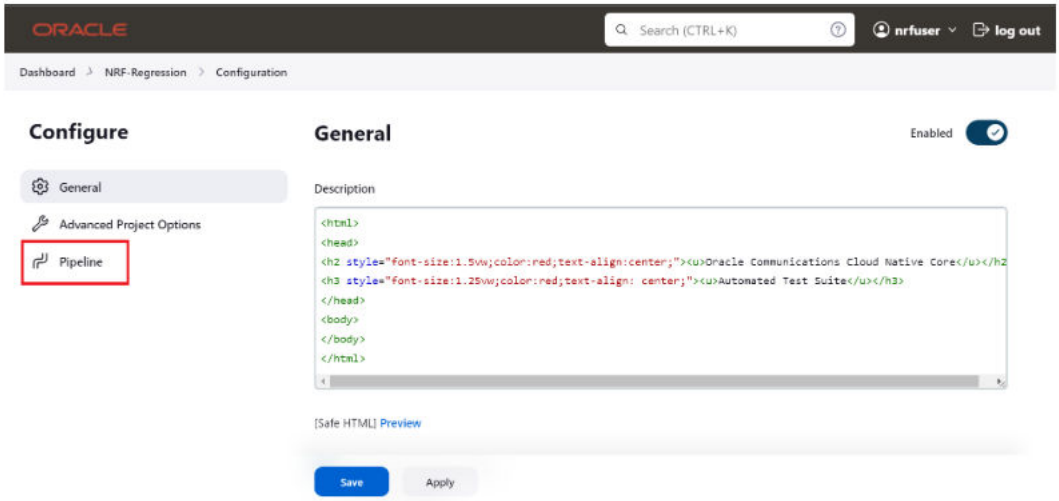
- Click **Configure** to configure NRF-New Features.
- Click **Documentation** to view the documented test cases, which are part of this NRF release.
- Click the blue dots inside **Build History** box to view the success console logs of the "Sanity", "All-NewFeatures" respectively.
- The **Stage View** represents the already deployed pipeline for the customer reference.
- The **Test Results Analyzer** is the new plugin integrated in the NRF-ATS. This option can be used to display the previous history of all the execution build-wise. It will provide a graphical representation of the past execution together.

2. Click **Configure**. User must wait for the page to load completely. Once the page loads completely, click the **Pipeline** tab:

> **Note**
>
> Make sure that the following page loads completely before you perform any action on it. Also, do not modify any configuration other than shown below.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-18 Pipeline Tab



The **Pipeline** section of the configuration page appears as follows:

Figure 4-19 Pipeline Section

```
node ('built-in'){
    //a = SELECTED_NF     b = NF_NAMESPACE        c = FT_ENDPOINT      d = GATEWAY_IP        e = GATEWAY_PORT
    //f = CONFIG_IP       g = CONFIG_PORT         h = STUB_IP          i = STUB_PORT         j = NFINSTANCEID
    //k = PROMETHEUS_IP   l = PROMETHEUS_PORT     m = RERUN_COUNT      n = PROMETHEUS_URI    o = HELM_RELEASE_NAME
    //p = MYSQL_HOST      q = PRIVILEGED_USER_SECRET_NAME.             r = DISCOVERY_WAIT_TIME
    //s = REPLICATION_STATUS_URI                  t = CLUSTER_DOMAIN   u = REGISTER_WRITE_WAIT_TIME
    //v = SUBSCRIPTION_WAIT_TIME                  w = ACCESSTOKEN_WAIT_TIME                  x = REGISTER_READ_WAIT_TIME
    withEnv([

    ]){

    sh '''
        sh /var/lib/jenkins/ocnrf_tests/preTestConfig.sh \
        -a NRF \
        -b ocnrf \
        -c ocnrf-ingressgateway.ocnrf.svc.cluster.local:80 \
        -d ocnrf-ingressgateway.ocnrf \
        -e 80 \
        -f ocnrf-nrfconfiguration.ocnrf \
        -g 8080 \
        -h notify-stub-service.ocnrf \
        -i 8080 \
        -j 6faf1bbc-6e4a-4454-a507-a14ef8e1bc5c \
        -k oso-prom-svr.oso \
        -l 80 \
        -m 2 \
        -n /prometheus/api/ \
        -o ocnrf \
        -p mysql-connectivity-service.ocnrf \
        -q privilegeduser-secret \
        -r 0.5 \
        -s http://ocnrf-ocnrf-app-info:5906/status/category/replicationstatus \
        -t svc.cluster.local \
        -u 0\
        -v 0\
        -w 0\
        -x 0
    '''
```

In the above screenshot, change the values of the '**Pipeline script**'. The content of the pipeline script is as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-20 Pipeline Script



```
node ('built-in'){
    //a = SELECTED_NF     b = NF_NAMESPACE        c = FT_ENDPOINT      d = GATEWAY_IP        e = GATEWAY_PORT
    //f = CONFIG_IP       g = CONFIG_PORT         h = STUB_IP          i = STUB_PORT         j = NFINSTANCEID
    //k = PROMETHEUS_IP   l = PROMETHEUS_PORT     m = RERUN_COUNT      n = PROMETHEUS_URI    o = HELM_RELEASE_NAME
    //p = MYSQL_HOST      q = PRIVILEGED_USER_SECRET_NAME              r = DISCOVERY_WAIT_TIME
    //s = REPLICATION_STATUS_URI                  t = CLUSTER_DOMAIN   u = REGISTER_WRITE_WAIT_TIME
    //v = SUBSCRIPTION_WAIT_TIME                  w = ACCESSTOKEN_WAIT_TIME                 x = REGISTER_READ_WAIT_TIME
    withEnv([

    ]){

    sh '''
        sh /var/lib/jenkins/ocnrf_tests/preTestConfig.sh \
        -a NRF \
        -b ocats \
        -c ocnrf-ingressgateway.ocats:80 \
        -d ocnrf-ingressgateway.ocats \
        -e 80 \
        -f ocnrf-nrfconfiguration.ocats \
        -g 8080 \
        -h notify-stub-service.ocats \
        -i 8080 \
        -j 6faf1bbc-6e4a-4454-a507-a14ef8e1bc5c \
        -k oso-prom-svr.oso \
        -l 80 \
        -m 0 \
        -n /prometheus/api/ \
        -o ocnrf \
        -p database-mysql-cne \
        -q privilegeduser-secret \
        -r 0.5 \
        -s http://ocnrf-ocnrf-app-info:5906/status/category/replicationstatus \
        -t thrust6 \
        -u 0\
        -v 0\
        -w 0\
        -x 0
    '''
    if(env.Include_Regression && "${Include_Regression}" == "YES"){
        sh '''sh /var/lib/jenkins/common_scripts/merge_jenkinsfile.sh'''
        load "/var/lib/jenkins/ocnrf_tests/jenkinsData/Jenkinsfile-Merged"
    }
    else{
        load "/var/lib/jenkins/ocnrf_tests/jenkinsData/Jenkinsfile-NewFeatures"
    }

    }
}
```

> **Note**
>
> The user must not change any other value apart from **line number 15** to **line 37**.

You can change the parameter values from **"a"** - to - **"x"** as per user requirement. The parameter details are available as comments from line number **2** - to - **7**.

- **a:** Name of the NF to be tested in capital (NRF).
- **b:** Namespace in which the NRF is deployed.
- **c:** endPointIP: endPointPort value used while deploying NRF with the help of helm chart.
- **d:** Comma separated values of NRF1 and NRF2 ingress gateway service (For example: ocnrf-ingressgateway.ocnrf,1.1.1.1). It is also known as cluster_domain.
- **e:** Comma separated values of NRF1 and NRF2 port of ingressgateway service (For example: 80,31000).
- **f:** Comma separated values of NRF1 and NRF2 configuration service (For example: ocnrf-nrfconfiguration.ocnrf,1.1.1.1). It is also known as cluster_domain.
- **g:** Comma separated values of NRF1 and NRF2 port of configuration service (For example: 8080, 31001).
- **h:** Name_of_stub_service.namespace (notify-stub-service.ocnrf).
- **i:** Port of stub service (8080).
- **j:** NRF_Instance ID (6faf1bbc-6e4a-4454-a507-a14ef8e1bc5c).

Copyright © 2019, 2025, Oracle and/or its affiliates.

- **k:** Name_of_Prometheus_service.namespace (occne-prometheus-server.occne-infra).
- **l:** Port of Prometheus service (80).
- **m:** Number of times the re-run of failed case is allowed (default as 2).
- **n:** URI of Prometheus service (/api/) -
  - Use "/api/" for CNE version till 1.6.0 and "/prometheus/api/" for CNE version 1.7.0 and 1.8.0.
  - For, **CNE 1.9.x, CNE 1.10.x**, and **CNE 22.1.0,** cluster name is needed in the URI. The URI must be **<cluster_name>/prometheus/api**. For example: **/bmw/prometheus/api** (For more details on the Prometheus URI, check corresponding CNE version's user guide.)
  - If OSO is being used, use **"/api/"** for OSO-1.6.1 or earlier and use **"/prometheus/api/"** for OSO-1.6.2 or later.
  - For OSO-1.10.0, the URI must be **<cluster_name>/prometheus/api**. For example: **/bmw/prometheus/api** (For more details on the Prometheus URI, check corresponding OSO version's user guide)
- **o:** Helm_release_name used to deploy NRF
- **p:** Host name of MySQL (For example: mysql-connectivity-service)
- **q:** Privileged User secret name (For example: privilegeduser-secret)
- **r**: wait time before sending discovery request (0.5)
- **s**: comma-separated values of replicationStatusUri(s) of the appinfo(s).
- **t**: the cluster domain
- **u**: wait time before sending registration requests for write operations
- **v**: wait time before sending subscription requests
- **w**: wait time before sending accesstoken requests
- **x**: wait time before sending registration requests for read operation

Copyright © 2019, 2025, Oracle and/or its affiliates.

Note

- If the **2-Site-Geo** cases are run, the user has to provide values for NRF2 in corresponding environment variables in the above script as per the deployment. A sample configuration is given as below:

```
node ('master'){
    //a = SELECTED_NF    b = NF_NAMESPACE        c =
FT_ENDPOINT      d = GATEWAY_IP        e = GATEWAY_PORT
    //f = CONFIG_IP      g = CONFIG_PORT         h =
STUB_IP          i = STUB_PORT         j = NFINSTANCEID
    //k = PROMETHEUS_IP  l = PROMETHEUS_PORT     m =
RERUN_COUNT      n = PROMETHEUS_URI    o = HELM_RELEASE_NAME
    //p = MYSQL_HOST     q =
PRIVILEGED_USER_SECRET_NAME              r = DISCOVERY_WAIT_TIME
    //s = REPLICATION_STATUS_URI                t =
CLUSTER_DOMAIN   u = REGISTER_WRITE_WAIT_TIME
    //v = SUBSCRIPTION_WAIT_TIME                w =
ACCESSTOKEN_WAIT_TIME                 x =
REGISTER_READ_WAIT_TIME
    sh '''
        sh /var/lib/jenkins/ocnrf_tests/preTestConfig.sh \
        -a NRF \
        -b occne-cndbtierone,occne-cndbtiertwo \
        -c ocnrf-
ingressgateway.ocnrf.svc.cluster.local:80,ocnrf-1-
ingressgateway.ocnrf.svc.cluster.local:80 \
        -d ocnrf-ingressgateway.occne-cndbtierone,ocnrf-1-
ingressgateway.occne-cndbtiertwo \
        -e 80,80 \
        -f ocnrf-nrfconfiguration.occne-cndbtierone,ocnrf-1-
nrfconfiguration.occne-cndbtiertwo \
        -g 8080,8080 \
        -h notify-stub-service.occne-cndbtierone,notify-stub-
service02.occne-cndbtierone \
        -i 8080,8080 \
        -j 6faf1bbc-6e4a-4454-a507-
a14ef8e1bc5c,6faf1bbc-6e4a-4454-a507-a14ef8e1bc5d \
        -k oso-prom-svr.oso \
        -l 80 \
        -m 2 \
        -n /prometheus/api/ \
        -o ocnrf,ocnrf-1 \
        -p mysql-connectivity-service \
        -q privilegeduser-secret     \

        -r 0.5 \
        -s http://ocnrf-ocnrf-app-info:5906/status/category/
replicationstatus,http://ocnrf-1-ocnrf-app-info:5906/status/
category/replicationstatus \
        -t svc.cluster.local \
        -u 0 \
        -v 0 \
        -w 0 \
        -x 0
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    '''
    if(env.Include_NewFeatures && "${Include_NewFeatures}" ==
"YES"){
        sh '''sh /var/lib/jenkins/common_scripts/
merge_jenkinsfile.sh'''
        load "/var/lib/jenkins/ocnrf_tests/jenkinsData/
Jenkinsfile-Merged"
    }
    else{
        load "/var/lib/jenkins/ocnrf_tests/jenkinsData/
Jenkinsfile-Regression"
    }

    }
}
```

- If **3SiteGeo** and georedundancy suite in Regression pipeline cases are run, the user has to provide values for NRF2, NRF3 in corresponding environment variables in the above script as per the deployment. A sample configuration is given as below:

```
node ('master'){
    //a = SELECTED_NF    b = NF_NAMESPACE        c =
FT_ENDPOINT      d = GATEWAY_IP        e = GATEWAY_PORT
    //f = CONFIG_IP      g = CONFIG_PORT         h =
STUB_IP          i = STUB_PORT         j = NFINSTANCEID
    //k = PROMETHEUS_IP  l = PROMETHEUS_PORT     m =
RERUN_COUNT      n = PROMETHEUS_URI    o = HELM_RELEASE_NAME
    //p = MYSQL_HOST     q =
PRIVILEGED_USER_SECRET_NAME              r = DISCOVERY_WAIT_TIME
    //s = REPLICATION_STATUS_URI                t =
CLUSTER_DOMAIN   u = REGISTER_WRITE_WAIT_TIME
    //v = SUBSCRIPTION_WAIT_TIME                w =
ACCESSTOKEN_WAIT_TIME                  x =
REGISTER_READ_WAIT_TIME
    sh '''
        sh /var/lib/jenkins/ocnrf_tests/preTestConfig.sh \
        -a NRF \
        -b occne-cndbtierone,occne-cndbtiertwo,occne-
cndbtierthree \
        -c ocnrf-
ingressgateway.ocnrf.svc.cluster.local:80,ocnrf-1-
ingressgateway.ocnrf.svc.cluster.local:80,ocnrf-2-
ingressgateway.ocnrf.svc.cluster.local:80 \
        -d ocnrf-ingressgateway.occne-cndbtierone,ocnrf-1-
ingressgateway.occne-cndbtiertwo,ocnrf-2-ingressgateway.occne-
cndbtierthree \
        -e 80,80,80 \
        -f ocnrf-nrfconfiguration.occne-cndbtierone,ocnrf-1-
nrfconfiguration.occne-cndbtiertwo,ocnrf-2-
nrfconfiguration.occne-cndbtierthree \
        -g 8080,8080,8080 \
        -h notify-stub-service.occne-cndbtierone,notify-stub-
service02.occne-cndbtierone,notify-stub-service03.occne-
cndbtierone \
        -i 8080,8080,8080 \
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
            -j 6faf1bbc-6e4a-4454-a507-
        a14ef8e1bc5c,6faf1bbc-6e4a-4454-a507-
        a14ef8e1bc5d,6faf1bbc-6e4a-4454-a507-a14ef8e1bc5e \
            -k oso-prom-svr.oso \
            -l 80 \
            -m 2 \
            -n /prometheus/api/ \
            -o ocnrf,ocnrf-1,ocnrf-2 \
            -p mysql-connectivity-service \
            -q privilegeduser-secret     \

            -r 0.5 \
            -s http://ocnrf-ocnrf-app-info:5906/status/category/
        replicationstatus,http://ocnrf-1-ocnrf-app-info:5906/status/
        category/replicationstatus,http://ocnrf-2-ocnrf-app-info:5906/
        status/category/replicationstatus \
            -t svc.cluster.local \
            -u 0 \
            -v 0 \
            -w 0 \
            -x 0
        '''
        if(env.Include_NewFeatures && "${Include_NewFeatures}" ==
        "YES"){
            sh '''sh /var/lib/jenkins/common_scripts/
        merge_jenkinsfile.sh'''
            load "/var/lib/jenkins/ocnrf_tests/jenkinsData/
        Jenkinsfile-Merged"
        }
        else{
            load "/var/lib/jenkins/ocnrf_tests/jenkinsData/
        Jenkinsfile-Regression"
        }

        }
    }
```

> **Note**
>
> If user is using CNE1.6 or OSO 1.6.1 or earlier, change the Prometheus service URI (-n) to "/api/" and Prometheus service name and port to appropriate values. For CNE 1.7.0 or OSO 1.6.2 or later, retain the default value **"/prometheus/api/"**. If OSO 1.10.0 is used, use the URI along with the cluster_name (for example: **/bmw/prometheus/api**).

3. Click **Save** after making necessary changes. The NRF-NewFeatures screen appears. Click the **Build with Parameters** link available in the left navigation pane of the **NRF-NewFeatures** Pipeline screen.
   In the Pipeline screen, there are three **Select_Option**(s), which are:
   - **All:** This is the default option. It runs all the NRF test cases. Scroll down and click **Build** to execute all the test cases.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- **Sanity:** It is recommended to run sanity before running any test case. It ensures all the deployments are done properly.
- **Single/MultipleFeatures:** This option allows you to select any number of test cases that you want to run from the list of total test cases available for execution.
  After selecting the test cases, scroll down and click **Build** to run the selected NRF test cases.

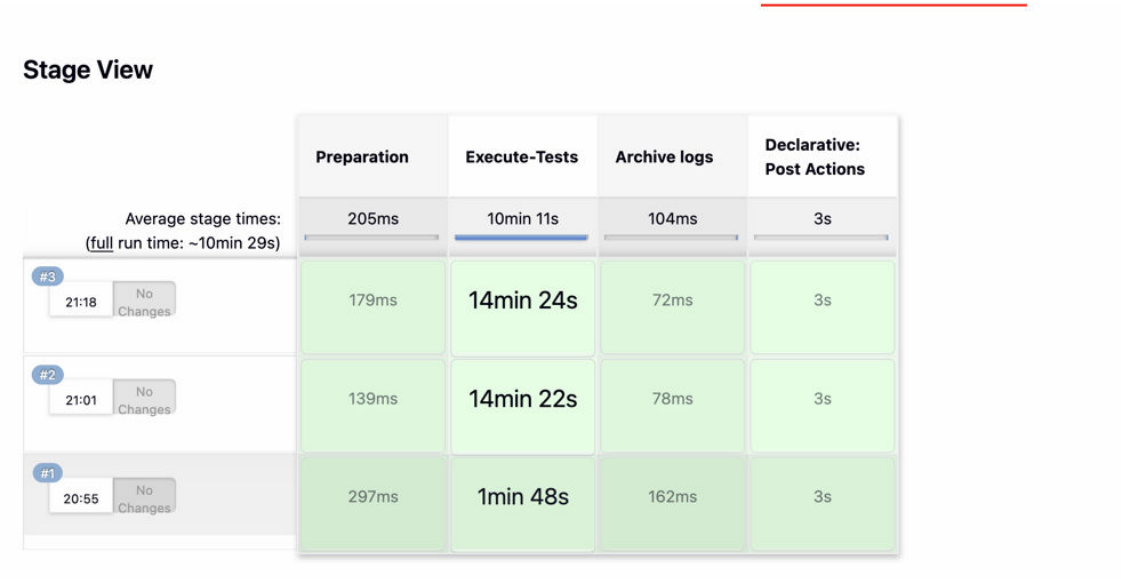
4. Select one of the following configuration types:
   - **Product_Config**: On selecting this option, test cases from product folders are populated on ATS UI and product configuration is applied to them via the key-value pair and yaml files defined or present in the "Product Config" folder.
   - **Custom_Config**: On selecting this option, test cases from custom folders are populated on ATS UI and custom configuration is applied to them via the key-value pair and yaml files defined or present in the "Custom Config" folder. To use the Parameterization feature, always select the Custom_Config option. User can copy, add, or delete the required test cases that are available for the NRF and place them appropriately within the custom folder for NRF-NewFeatures. Reload the page to view the test cases available in the custom NewFeatures folder.
     For more information, see Parameterized approach for SUT custom configuration.

Based on the requirement, user can select any of the above option and click on **"Build"**. The NRF testcases are divided into NRF Service operations. The classification is as follows:

- **NRF Sanity** - This feature file contains all the basic sanity cases for NRF ATS to validate the deployment is correct or not. It is advisable for user to run these cases before starting a complete suite.
- **Discovery** - These feature files are listed with a prefix as **"Disc"**. All the discovery microservice related cases will be listed once this option is selected.
- **NRF Functional** - These feature files are listed with a prefix as **"Feat"**. All the functional cases will be listed once this option is selected.
- **Registration** - These feature files are listed with a prefix as **"Upd"**. These are related to update operation of registered profiles. All the registration cases will be listed once this option is selected.
- **Subscription**- These feature files are listed with a prefix as **"Subs"**. All the subscription microservice related cases will be listed once this option is selected.
- **Roaming** -These feature files are listed with a prefix as **"Roaming"**. All the roaming cases will be listed once this option is selected.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-21 Sample Screen: NRF-ATS Full Execution



A sample consolidated test report is shown below, when rerun is set to 2:

Figure 4-22 Test Cases Result - Sanity

```
1 feature passed, 0 failed, 0 skipped
11 scenarios passed, 0 failed, 0 skipped
193 steps passed, 0 failed, 0 skipped, 0 undefined
Took 1m1.068s
[Pipeline] sh
+ cd /var/lib/jenkins/ocnrf_tests
++ cat /var/lib/jenkins/ocnrf_tests/environ.sh
++ grep RERUN
++ cut '-d;' -f1
++ cut -d= -f2
+ rerun=0
+ sh re-run.sh 0 /var/lib/jenkins/.jenkins/workspace/NRF-NewFeatures
1
Scenario passed
```

> **Note**
>
> For more details on consolidated test report, see Managing Final Summary Report, Build Color, and Application Log.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-23 Test Cases Result - All-NewFeatures

```
2025-10-22 21:35:43,343[32m     INFO LOG.AllHooks:296 [0m| [32mInside after_all[0m
2025-10-22 21:35:43,343[32m     INFO LOG.AllHooks:406 [0m| [32mExiting after_all[0m
13 features passed, 0 failed, 0 skipped
43 scenarios passed, 0 failed, 0 skipped
696 steps passed, 0 failed, 0 skipped, 0 undefined
Took 7m17.860s
[Pipeline] sh
+ cd /var/lib/jenkins/ocnrf_tests
++ cat /var/lib/jenkins/ocnrf_tests/environ.sh
++ grep RERUN
++ cut -d= -f2
++ cut '-d;' -f1
+ rerun=1
+ sh re-run.sh 1 /var/lib/jenkins/.jenkins/workspace/NRF-NewFeatures
/var/lib/jenkins/ocnrf_tests/rerun_failing.features does not exist
1
Scenario passed
[Pipeline] sh
```

Parameterized approach for SUT custom configuration

Using this feature, user can make the following customizations to custom folders

- Add new test cases by adding datafiles
- Remove test cases
- Modify the parameters and their values in the key-value pair or `<feature>.yaml` files

To run ATS test cases, user can maintain as many versions of **Custom_Config** folder by using the following naming convention:

Cust ConfigN

where N can be any number

At the time of execution, ensure to rename the required folder to **Cust Config** folder as Jenkins always retrieves data from this folder when user selects **Custom_Config**.

Updating Global Parameters
To use Custom_Config, it is required to change the value of ***cust_folder*** from ***data*** to ***cust_data*** in `global.yaml` file. In addition, you can customize the parameters and their respective values in the `global.yaml` as per the requirements.

Updating Feature Parameters
Consider the following points when customizing `<feature>.yaml` files for parameterized feature:

- In addition to `global.yaml` parameters, feature files may also contain parameters for which user can update values at the time of running pipelines.
- Changing the values of parameters tagged as "Feature Specific Value" may cause failures at the time of running pipelines.
- Values for parameters tagged with #START_GLOBAL and #END_GLOBAL tags take values from `global.yaml`.

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> For NRF-ATS release 25.1.203, parameterization is supported for NRF NewFeatures pipeline only.

4.2.4 NRF-NewFeatures Documentation

To view NRF test cases, go to **NRF-NewFeatures** pipeline and click the **Documentation** link in the left navigation pane. It lists all the test cases provided as part of NRF ATS -25.1.203 along with sanity cases.

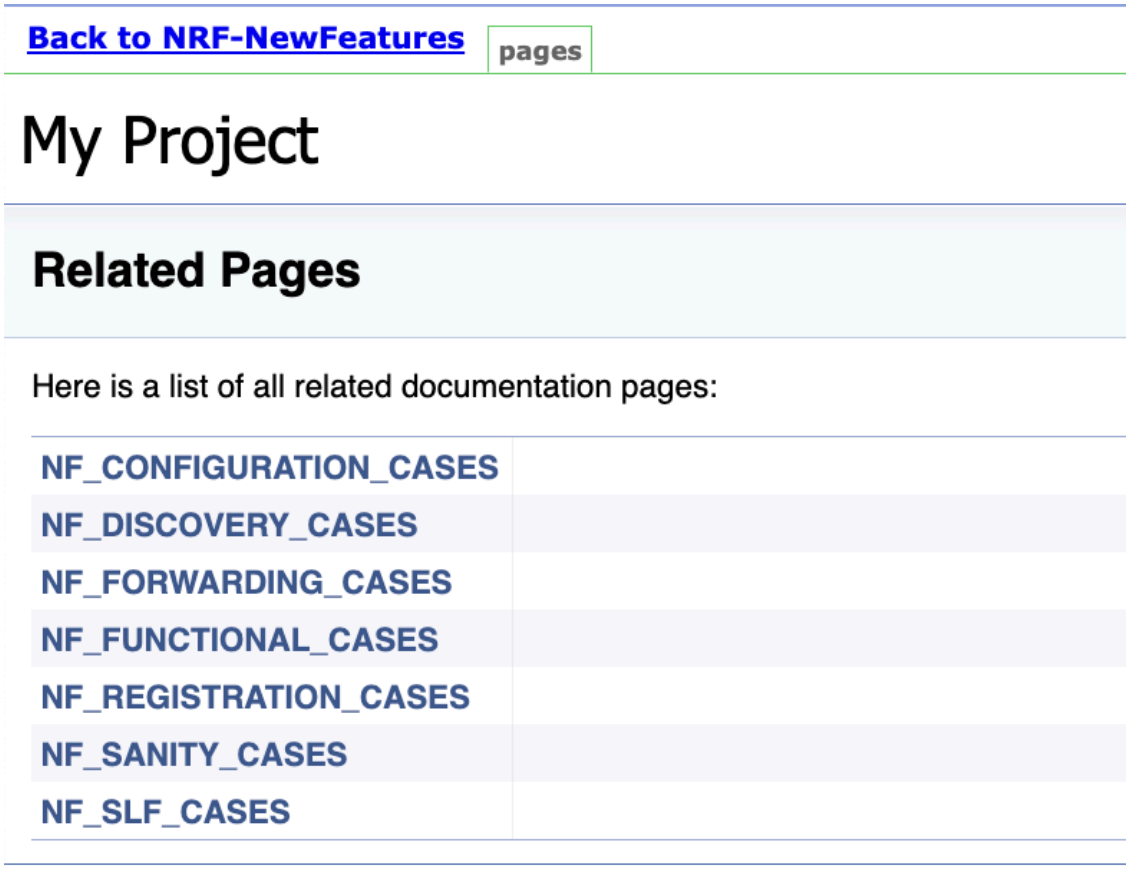
The NRF test cases are divided into multiple groups based on the functionality.

- **NF_CONFIGURATION_CASES**- All the cases related to NRF configuration will be listed once this option is selected.
- **NF_DISCOVERY_CASES** - All the discovery microservice related cases will be listed once this option is selected.
- **NRF_FORWARDING_CASES** - All the forwarding related cases will be listed once this option is selected.
- **NF_FUNCTIONAL_CASES** - All the functional cases will be listed once this option is selected.
- **NF_REGISTRATION_CASES**- All the registration cases will be listed once this option is selected.
- **NF_SANITY_CASES**- All the sanity cases will be listed once this option is selected.
- **NF_SLF_CASES** - All the SLF related cases will be listed once this option is selected.

The following screenshot shows all the NRF features:

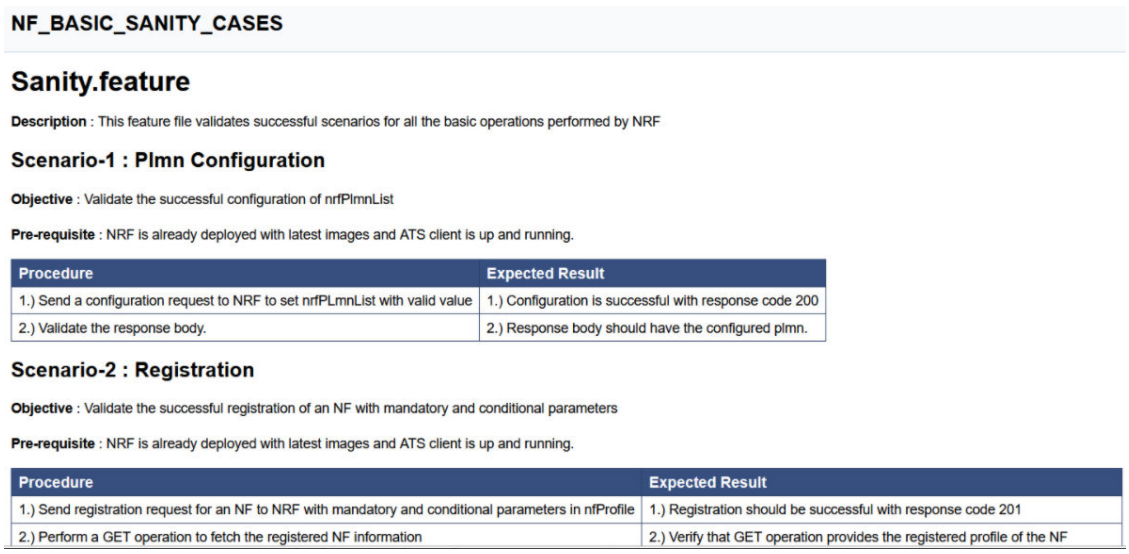
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-24 NRF-NewFeatures Documentation



Click any functionality to view its test cases and scenarios of each test case as shown below:

Figure 4-25 Sample Feature: NF_BASIC_SANITY_CASES



Copyright © 2019, 2025, Oracle and/or its affiliates.

Based on the functionalities covered under **Documentation**, the **Build Requires Parameters** screen displays test cases. To navigate back to the Pipeline **NRF-NewFeatures** screen, click the **Back to NRF-NewFeatures** link available on top left corner of the screen.

4.2.5 NRF-Regression Pipeline

This pre-configured pipeline contains all the test cases delivered till NRF ATS 25.1.203. However, some test cases are updated as per new implementation of NRF.

The configuration method and parameters are same as the **NewFeatures** pipeline.

Only difference in this pipeline is, while executing **2-Site-Geo** and **3-Site-Geo** test cases, user has to provide appropriate values for NRF2 and NRF3 in the Pipeline script. In this pipeline, user will not get option to run **Sanity**.

The NRF-Regression test cases are divided into following service operations:

- **2-Site-Geo** - These feature files are listed with a prefix as **"Geo_Redundancy"**.
- **3-Site-Geo**- These feature files contain all the cases which are executed on 3 Site GEO NRF.
- **CCA-Header**- These feature files contain all cases related to CCA Header feature.
- **NF-FQDN-Authentication** - These feature files are listed with a prefix as **"NfAuthentication"**.
- **AccessToken** - These feature files are listed with a prefix as **"oAuth" and "AccessToken"**.
- **Alerts -** These feature files are list with a prefix as **"Alert"**.
- **Configuration** - These feature files are listed with a prefix as **"Config" and "SystemOptions"**.
- **Discovery** - These feature files are listed with a prefix as **"Disc"**.
- **NRF Forwarding** - These feature files are listed with a prefix as **"Forwarding"**.
- **NRF Functional** - These feature files are listed with a prefix as **"Feat"**.
- **Key-ID** - These feature files are listed with a prefix as **"Kid"**.
- **NRF State Data** - These feature files are listed with a prefix as **"State"**.
- **Preferred Locality** - These feature files are listed with a prefix as **"Pref"**.
- **Registration** - These feature files are listed with a prefix as **"HBTimerEnhancement", "Reg"** and **"Upd"**. These are related to update operation of registered profiles.
- **NRF SLF** - These feature files are listed with a prefix as **"SLF"**.
- **Subscription** - These feature files are listed with a prefix as **"Subs"**.
- **NRF Growth** - These feature files contain all cases related to NRF Growth feature.

The following screenshot shows the NRF-Regression.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-26 NRF-Regression



The following screenshot shows successful execution as part of ATS image.

Figure 4-27 NRF-Regression - All-default-Regression

```
632 features passed, 0 failed, 0 skipped
1777 scenarios passed, 0 failed, 0 skipped
28726 steps passed, 0 failed, 0 skipped, 0 undefined
Took 250m37.072s
[Pipeline] sh
+ cd /var/lib/jenkins/ocnrf_tests
++ cat /var/lib/jenkins/ocnrf_tests/environ.sh
++ grep RERUN
++ cut -d= -f2
++ cut '-d;' -f1
+ rerun=1
+ sh re-run.sh 1 /var/lib/jenkins/.jenkins/workspace/NRF-Regression
/var/lib/jenkins/ocnrf_tests/rerun_failing.features does not exist
1
Scenario passed
```

4.2.6 NRF-Regression Documentation

Click **Documentation** in the left navigation pane of the **NRF-Regression** pipeline to view all the test cases provided till NRF ATS 25.1.203.

The NRF test cases are divided into multiple groups based on following functionalities:

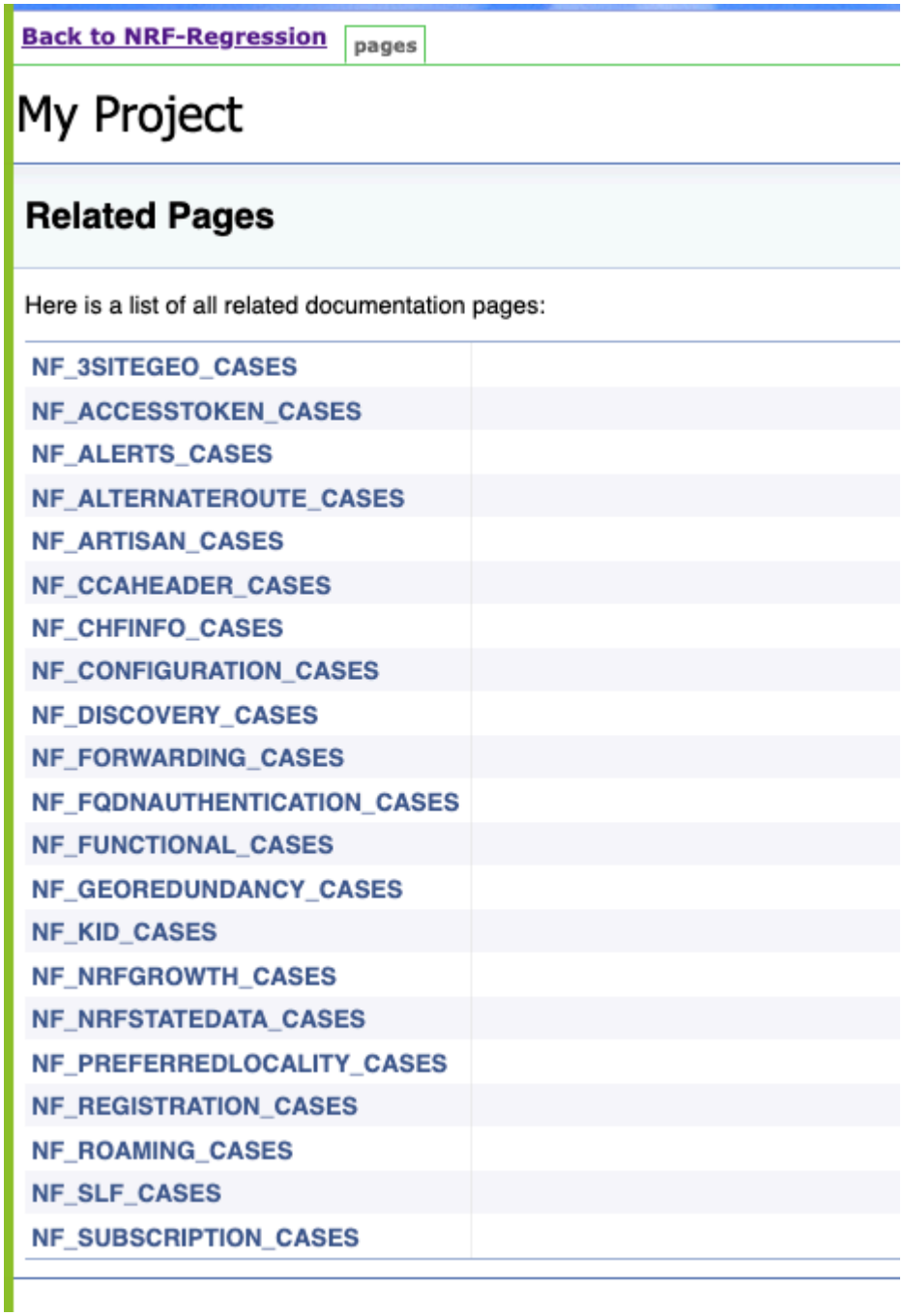
- **NF_3SITEGEO_CASES**- All the 3-Site Geo-Redundancy related cases will be listed once this option is selected.
- **NF_ACCESSTOKEN_CASES** - All the test cases for validating accesstoken operation will be listed once this option is selected.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- **NF_ALERTS_CASES** - Test case for validating alerts will be listed once this option is selected.
- **NF_ALTERNATEROUTE_CASES**- All the alternate route microservice related cases will be listed once this option is selected.
- **NF_ARTISAN_CASES**- All the NRF artisan microservice related cases will be listed once this option is selected.
- **NF_CCAHEADER_CASES** - Cases related to CCAHeader feature will be listed once this option is selected.
- **NF_CHFINFO_CASES** - All the ChfInfo enhancement cases will be listed once this option is selected.
- **NF_CONFIGURATION_CASES** - Cases related to NRF configuration will be listed once this option is selected.
- **NF_DISCOVERY_CASES** - All the discovery microservice related cases will be listed once this option is selected.
- **NF_FORWARDING_CASES** - All the forwarding related cases will be listed once this option is selected.
- **NF_FQDNAUTHENTICATION_CASES** - All the NF FQDN authentication cases will be listed once this option is selected.
- **NF_FUNCTIONAL_CASES** - All the functional cases will be listed once this option is selected.
- **NF_GEOREDUNDANCY_CASES** - All the Geo-Redundancy related cases (2-Site-GEO and 3-Site-GEO both) will be listed once this option is selected.
- **NF_KID_CASES** - All the K-ID feature related cases will be listed once this option is selected.
- **NF_NRFGROWTH_CASES** - All growth related cases will be listed once this option is selected.
- **NF_NRFSTATE_DATA_CASES** - All the nfStateData related cases will be listed once this option is selected.
- **NF_PREFERRED_LOCALITY_CASES** - All the preferred locality feature related cases will be listed once this option is selected.
- **NF_REGISTRATION_CASES** - All the registration related cases will be listed once this option is selected.
- **NF_ROAMING_CASES** - All the roaming related cases will be listed once this option is selected.
- **NF_SLF_CASES** - All the SLF related cases will be listed once this option is selected.
- **NF_SUBSCRIPTION_CASES** - All subscription related cases will be listed once this option is selected.

Copyright © 2019, 2025, Oracle and/or its affiliates.

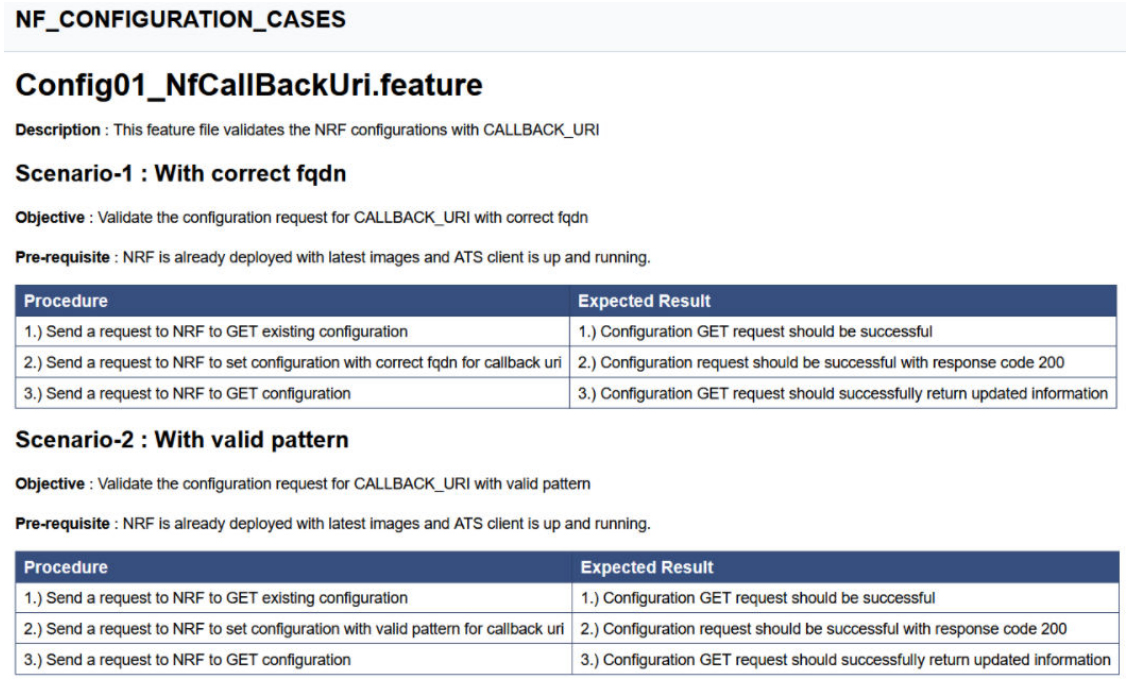
Figure 4-28 NRF-Regression Documentation



The following sample screen shows documentation of Regression pipeline for NRF ATS.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-29 Sample Screen: NRF-Regression Documentation



NF_CONFIGURATION_CASES

Config01_NfCallBackUri.feature

Description : This feature file validates the NRF configurations with CALLBACK_URI

Scenario-1 : With correct fqdn

Objective : Validate the configuration request for CALLBACK_URI with correct fqdn

Pre-requisite : NRF is already deployed with latest images and ATS client is up and running.

Procedure	Expected Result
1.) Send a request to NRF to GET existing configuration	1.) Configuration GET request should be successful
2.) Send a request to NRF to set configuration with correct fqdn for callback uri	2.) Configuration request should be successful with response code 200
3.) Send a request to NRF to GET configuration	3.) Configuration GET request should successfully return updated information

Scenario-2 : With valid pattern

Objective : Validate the configuration request for CALLBACK_URI with valid pattern

Pre-requisite : NRF is already deployed with latest images and ATS client is up and running.

Procedure	Expected Result
1.) Send a request to NRF to GET existing configuration	1.) Configuration GET request should be successful
2.) Send a request to NRF to set configuration with valid pattern for callback uri	2.) Configuration request should be successful with response code 200
3.) Send a request to NRF to GET configuration	3.) Configuration GET request should successfully return updated information

4.3 Running NSSF Test Cases using ATS

This section describes how to run NSSF test cases using ATS.

4.3.1 Prerequisites

The prerequisites to run NSSF Test Cases using NSSF ATS 25.1.201 are:

- Deploy NSSF 25.1.201 with default helm configurations using helm charts.
- All NSSF microservices must be up and running.
- Both NSSF ATS and ATS Deployment needs to be same namespace.
- For NSSF ATS 25.1.201, deploy one stub server and the service name should be **"amf-stubserver"**. It is required to run AMF-subscription Notification functionality test cases.
- For NSSF ATS 25.1.201, deploy one stub server and the service name should be **"nrf-stubserver"**. It is required to run NRF-subscription Notification functionality test cases.
- For NSSF ATS 25.1.201, deploy one stub server and the service name should be **"nrf-stubserver1"**. It is required to run NRF-selection based on DNS SRV.
- For NSSF ATS 25.1.201, deploy one stub server and the service name should be **"nrf-stubserver2"**. It is required to run NRF-selection based on DNS SRV.
- For NSSF 25.1.201, deploy one DNS stub server and the service name should be **"ocdns-bind"**. It is required to run NRF-selection based on DNS SRV.

4.3.2 Logging into ATS

Before logging into ATS, deploy ATS using HELM charts as shown below:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Verify ATS deployment

```
[opc@ocnssf-oci-phx-einstein-bastion-01 ~]$ helm status ocats

NAME: ocats
LAST DEPLOYED: Thu May 30 10:21:40 2024
NAMESPACE: cicdnssf-240530102024
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

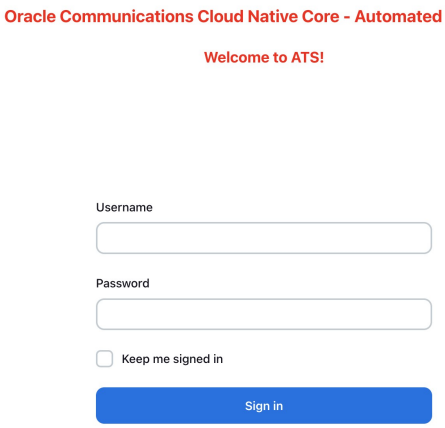
There are two ways to log in to ATS GUI.

- When an external load balancer (metalLB in case of OCCNE) is available and you provide an external IP to the ATS service, the user can log in to ATS GUI using <External-IP>:8080.
- When you do not provide an external IP to the ATS service, open the browser and enter the external IP of the worker node and nodeport of the ATS service to log in to the ATS GUI.
  `<Worker-Node-IP>:<Node-Port-of-ATS>`

> **Note**
>
> In the **Verifying ATS Deployment** screenshot, the ATS nodeport is highlighted in red as 32013. For more details on ATS deployment, refer to **NSSF ATS Installation Procedure.**

Open a browser and enter the IP Address and port details as <Worker-Node-IP>:<NodePort-of-ATS> (In the above example, the Worker-Node-IP and NodePort-of-ATS are 10.98.101.177:32013, which are shown as highlighted in the screenshot above).

The ATS login screen appears.



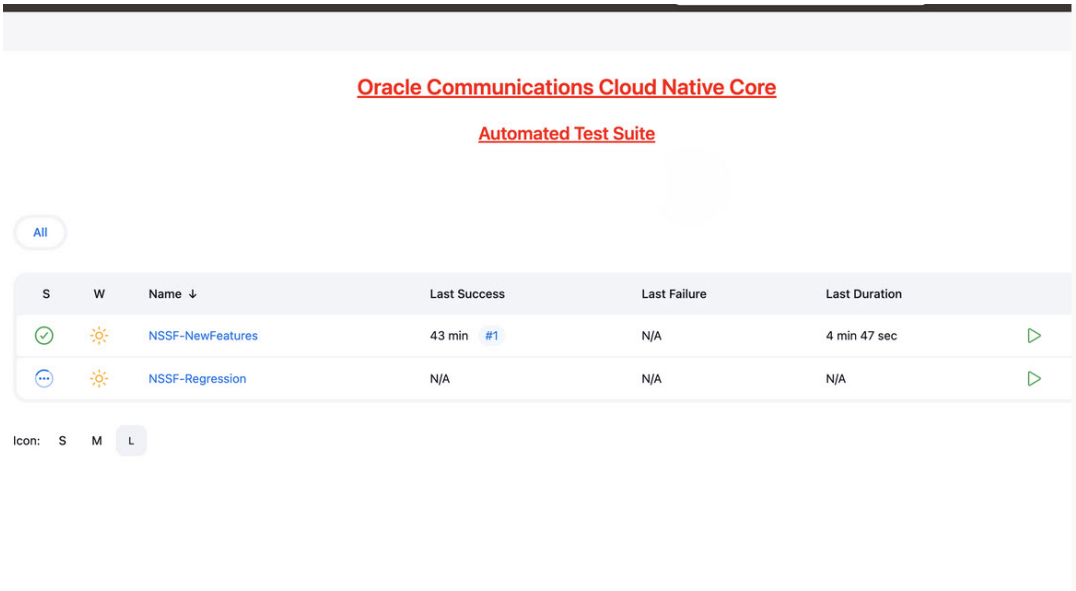
Copyright © 2019, 2025, Oracle and/or its affiliates.

To run ATS:

1. Enter the username as 'nssfuser' and password as 'nssfpasswd'. Click **Sign in**. A page with preconfigured pipelines appears.

> **Note**
>
> To modify the default login password, refer to Modifying Login Password.

A page with preconfigured pipelines appears.



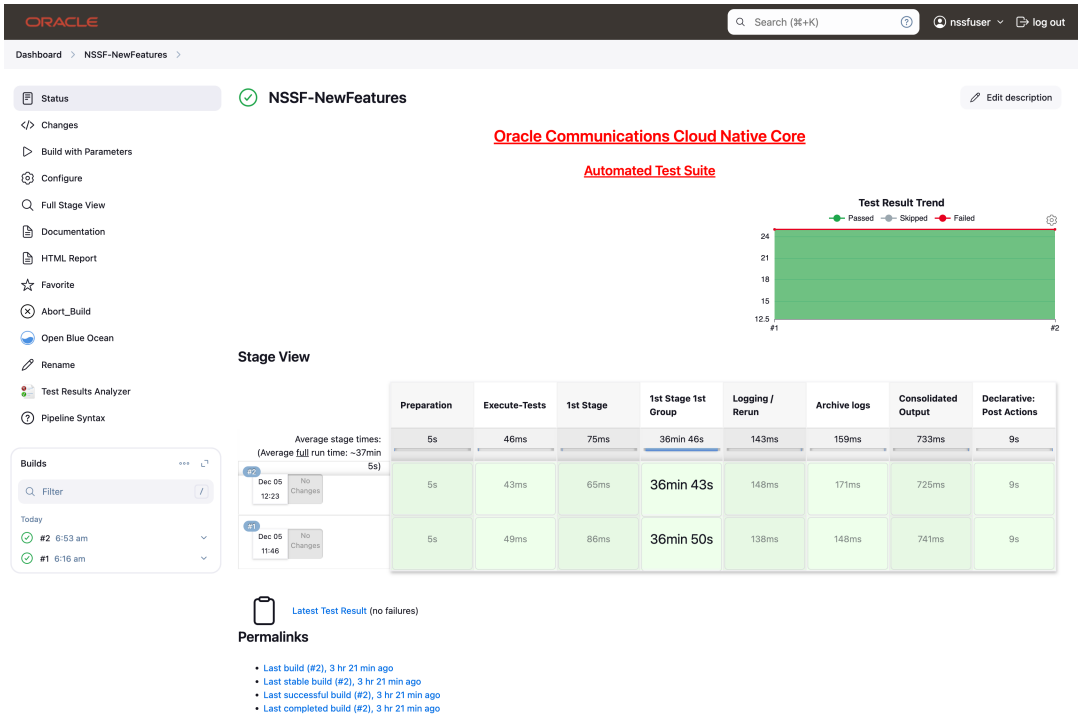
- **NSSF-New-Features:** This pipeline has all the test cases delivered as part of NSSF ATS - 25.1.201.
- **NSSF-Regression:** This pipeline has the test cases of all the previous releases.

4.3.3 NSSF-NewFeatures Pipeline

In this pipeline, you can configure ATS, which is a one-time activity as per System Under Test (SUT) deployment. You can also run all the new NSSF test cases using the pipeline. To configure its parameters:

1. Click **NSSF-NewFeatures** in the **Name** column. The following screen appears:

Copyright © 2019, 2025, Oracle and/or its affiliates.



In the above screen:

a. Click **Configure** to access the configuration screen.

b. Click **Documentation** to view the documented test cases.

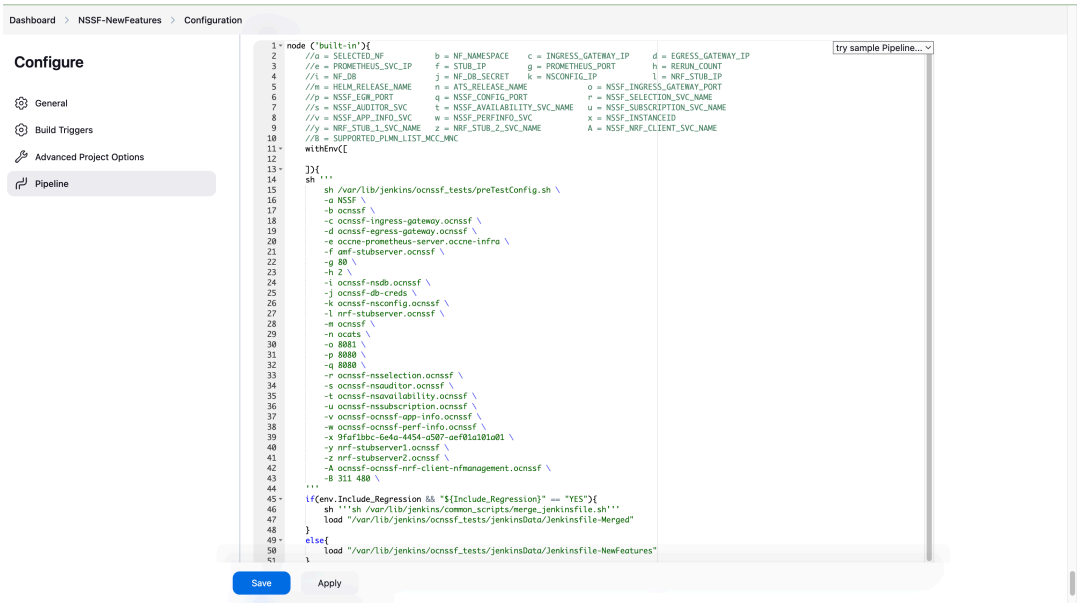
c. Click blue dots inside the **Build History** box to view the success console logs of the "All" and "Sanity" respectively.

d. The **Stage View** represents the already run pipeline for customer reference.

2. Click **Configure**. Users MUST wait for the page to load completely. Once the page loads completely, click the **Pipeline** tab to reach the Pipeline configuration as shown below:

> **Warning**
>
> Make sure that the screen shown above loads completely before you perform any action on it. Also, do not modify any configuration other than that discussed below.

Copyright © 2019, 2025, Oracle and/or its affiliates.



3. You can modify script pipeline parameters from "a" to "B" on the basis of your deployment environment and click Save. The content of the pipeline script is as follows:

```
node ('built-in'){
    //a = SELECTED_NF            b = NF_NAMESPACE    c =
INGRESS_GATEWAY_IP     d = EGRESS_GATEWAY_IP
    //e = PROMETHEUS_SVC_IP      f = STUB_IP         g =
PROMETHEUS_PORT        h = RERUN_COUNT
    //i = NF_DB                  j = NF_DB_SECRET    k =
NSCONFIG_IP            l = NRF_STUB_IP
    //m = HELM_RELEASE_NAME      n = ATS_RELEASE_NAME             o =
NSSF_INGRESS_GATEWAY_PORT
    //p = NSSF_EGW_PORT          q = NSSF_CONFIG_PORT             r =
NSSF_SELECTION_SVC_NAME
    //s = NSSF_AUDITOR_SVC       t = NSSF_AVAILABILITY_SVC_NAME   u =
NSSF_SUBSCRIPTION_SVC_NAME
    //v = NSSF_APP_INFO_SVC      w = NSSF_PERFINFO_SVC            x =
NSSF_INSTANCEID
    //y = NRF_STUB_1_SVC_NAME    z = NRF_STUB_2_SVC_NAME          A =
NSSF_NRF_CLIENT_SVC_NAME
    //B = SUPPORTED_PLMN_LIST_MCC_MNC
    withEnv([

    ]){
    sh '''
        sh /var/lib/jenkins/ocnssf_tests/preTestConfig.sh \
        -a NSSF \
        -b ocnssf \
        -c ocnssf-ingress-gateway.ocnssf \
        -d ocnssf-egress-gateway.ocnssf \
        -e occne-prometheus-server.occne-infra \
        -f amf-stubserver.ocnssf \
        -g 80 \
        -h 2 \
        -i ocnssf-nsdb.ocnssf \
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
            -j ocnssf-db-creds \
            -k ocnssf-nsconfig.ocnssf \
            -l nrf-stubserver.ocnssf \
            -m ocnssf \
            -n ocats \
            -o 8081 \
            -p 8080 \
            -q 8080 \
            -r ocnssf-nsselection.ocnssf \
            -s ocnssf-nsauditor.ocnssf \
            -t ocnssf-nsavailability.ocnssf \
            -u ocnssf-nssubscription.ocnssf \
            -v ocnssf-ocnssf-app-info.ocnssf \
            -w ocnssf-ocnssf-perf-info.ocnssf \
            -x 9faf1bbc-6e4a-4454-a507-aef01a101a01 \
            -y nrf-stubserver1.ocnssf \
            -z nrf-stubserver2.ocnssf \
            -A ocnssf-ocnssf-nrf-client-nfmanagement.ocnssf \
            -B 311 480 \
        '''
        if(env.Include_Regression && "${Include_Regression}" == "YES"){
            sh '''sh /var/lib/jenkins/common_scripts/merge_jenkinsfile.sh'''
            load "/var/lib/jenkins/ocnssf_tests/jenkinsData/Jenkinsfile-Merged"
        }
        else{
            load "/var/lib/jenkins/ocnssf_tests/jenkinsData/Jenkinsfile-
NewFeatures"
        }
        }
}
```

> **Note**
>
> The User MUST NOT change any other value apart from these parameters.

The description of these parameters is as follows:

- a: Name of the NF to be tested in capital (NSSF).
- b: Namespace in which the NSSF is deployed (default is ocnssf)
- c: Ingress Gateway IP address (default is ocnssf-ingress-gateway.ocnssf)
- d: Egress Gateway IP address (default is ocnssf-egress-gateway.ocnssf)
- e: Prometheus service IP address (default is prometheus.cne-infra)
- f: Stub service IP address (default is ocats-amf-stubserver.ocnssf)
- g: Port of Prometheus service (default is 80)
- h: Number of times the re-run of failed case is allowed (default is 2).
- i: Database name (default is ocnssf-nsdb.ocnssf)
- j Database secrets (default ocnssf-db-creds)
- k: NSSF config ip address (ocnssf-nsconfig.ocnssf)
- l: NRF stub server IP address (ocats-nrf-stubserver.ocnssf)

Copyright © 2019, 2025, Oracle and/or its affiliates.

- m: NSSF release name (ocnssf)
- n: ATS release name (ocats)
- o: NSSF Ingress Gateway Port
- p: NSSF Egress Gateway Port
- q: NSSF Config Port
- r: NSSF Selection Service Name
- s: NSSF Auditor Service Name
- t: NSSF Availability Service Name
- u: NSSF Subscription Service Name
- v: APP Info Service Name
- w:Perf info Service Name
- x: NSSF Supported PLMN
- A: NRF Client Managment Service Name (ocnssf-ocnssf-nrf-client-nfmanagement.ocnssf)
- B: NSSF PLMN List (311 480)

Copyright © 2019, 2025, Oracle and/or its affiliates.

Note

- Do not change any value if the OCCNE cluster is used and NSSF, ATS, and STUB are deployed in the ocnssf namespace.
- In the above image NSSF Helm release name is "ocnssf", ATS Helm Release Name is "ocats" and namespace is "ocnssf". If any change in helm release name of NSSF, ATS and namespace needs to be updated accordingly.
  For example, NSSF Helm release name is "ocnssfats" and ATS Helm release name is "ocatsnssf" and namespace is "ocnssf2510", then above Pipeline Configuration should be edited as shown below.
- NSSF Helm release name should be updated in "c d i k m r s t u v w" if any change in NSSF helm release name apart from "ocnssf".
- The NSSF Instance ID needs to be updated in the ATS Jenkins pipeline parameters if there are any changes in the NSSF custom values (CV) file. By default, the NSSF Instance ID is set to "`9faf1bbc-6e4a-4454-a507-aef01a101a01`". If the Instance ID is modified in the NSSF CV file, the same changes must be reflected in the Jenkins pipeline parameters.
  By default, NSSF Instance ID in NSSF custom values file as below:

  ```
  #InstanceId of NSSF used in case of GR
  nfInstanceId: &nfInstanceId "9faf1bbc-6e4a-4454-a507-
  aef01a101a01"
  ```

  For example, if the nfInstanceId in the NSSF custom values file is modified from "`9faf1bbc-6e4a-4454-a507-aef01a101a01`" to "`9faf1bbc-6e4a-4454-a507-aef01a101a20`", the same change must be updated in the "x" Jenkins pipeline parameters as shown below:

  ```
  node ('built-in'){
      //a = SELECTED_NF           b = NF_NAMESPACE    c =
  INGRESS_GATEWAY_IP     d = EGRESS_GATEWAY_IP
      //e = PROMETHEUS_SVC_IP     f = STUB_IP         g =
  PROMETHEUS_PORT        h = RERUN_COUNT
      //i = NF_DB                 j = NF_DB_SECRET    k =
  NSCONFIG_IP            l = NRF_STUB_IP
      //m = HELM_RELEASE_NAME     n = ATS_RELEASE_NAME
  o = NSSF_INGRESS_GATEWAY_PORT
      //p = NSSF_EGW_PORT         q = NSSF_CONFIG_PORT
  r = NSSF_SELECTION_SVC_NAME
      //s = NSSF_AUDITOR_SVC      t = NSSF_AVAILABILITY_SVC_NAME
  u = NSSF_SUBSCRIPTION_SVC_NAME
      //v = NSSF_APP_INFO_SVC     w = NSSF_PERFINFO_SVC
  x = NSSF_INSTANCEID
      //y = NRF_STUB_1_SVC_NAME   z = NRF_STUB_2_SVC_NAME
  A = NSSF_NRF_CLIENT_SVC_NAME
      //B = SUPPORTED_PLMN_LIST_MCC_MNC
      withEnv([

      ]){
      sh '''
          sh /var/lib/jenkins/ocnssf_tests/preTestConfig.sh \
          -a NSSF \
  ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
                -b ocnssf2510 \
                -c ocnssfats-ingress-gateway.ocnssf2510 \
                -d ocnssfats-egress-gateway.ocnssf2510 \
                -e occne-prometheus-server.occne-infra \
                -f amf-stubserver.ocnssf2510 \
                -g 80 \
                -h 2 \
                -i ocnssf-nsdb.ocnssf \
                -j ocnssf-db-creds \
                -k ocnssfats-nsconfig.ocnssf2510 \
                -l nrf-stubserver.ocnssf2510 \
                -m ocnssfats \
                -n ocats \
                -o 8081 \
                -p 8080 \
                -q 8080 \
                -r ocnssfats-nsselection.ocnssf2510 \
                -s ocnssfats-nsauditor.ocnssf2510 \
                -t ocnssfats-nsavailability.ocnssf2510 \
                -u ocnssfats-nssubscription.ocnssf2510 \
                -v ocnssfats-ocnssf-app-info.ocnssf2510 \
                -w ocnssfats-ocnssf-perf-info.ocnssf2510 \
                -x 9faf1bbc-6e4a-4454-a507-aef01a101a20 \
                -y nrf-stubserver1.ocnssf2510 \
                -z nrf-stubserver2.ocnssf2510 \
                -A ocnssfats-ocnssf-nrf-client-nfmanagement.ocnssf2510 \
                -B 311 480 \
            '''
            if(env.Include_Regression && "${Include_Regression}" ==
        "YES"){
                sh '''sh /var/lib/jenkins/common_scripts/
        merge_jenkinsfile.sh'''
                load "/var/lib/jenkins/ocnssf_tests/jenkinsData/
        Jenkinsfile-Merged"
            }
            else{
                load "/var/lib/jenkins/ocnssf_tests/jenkinsData/
        Jenkinsfile-NewFeatures"
            }
            }
        }
```

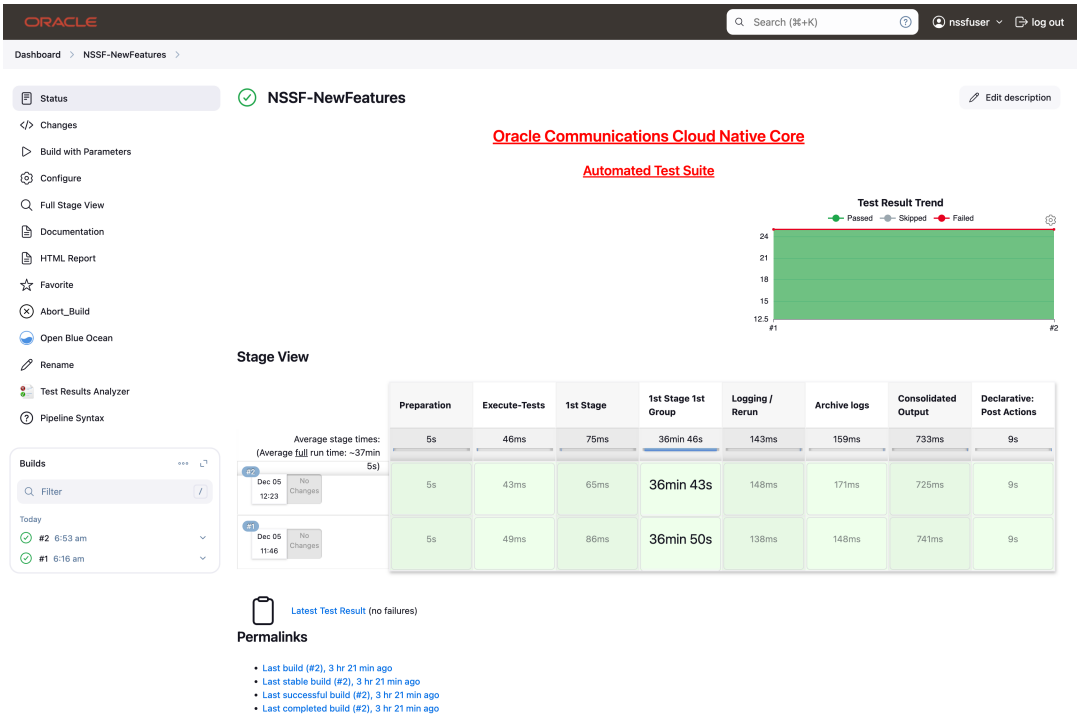
4. Click **Save** after making necessary changes. The **Pipeline NSSF-NewFeatures** screen appears.

Running NSSF New Features Test Cases

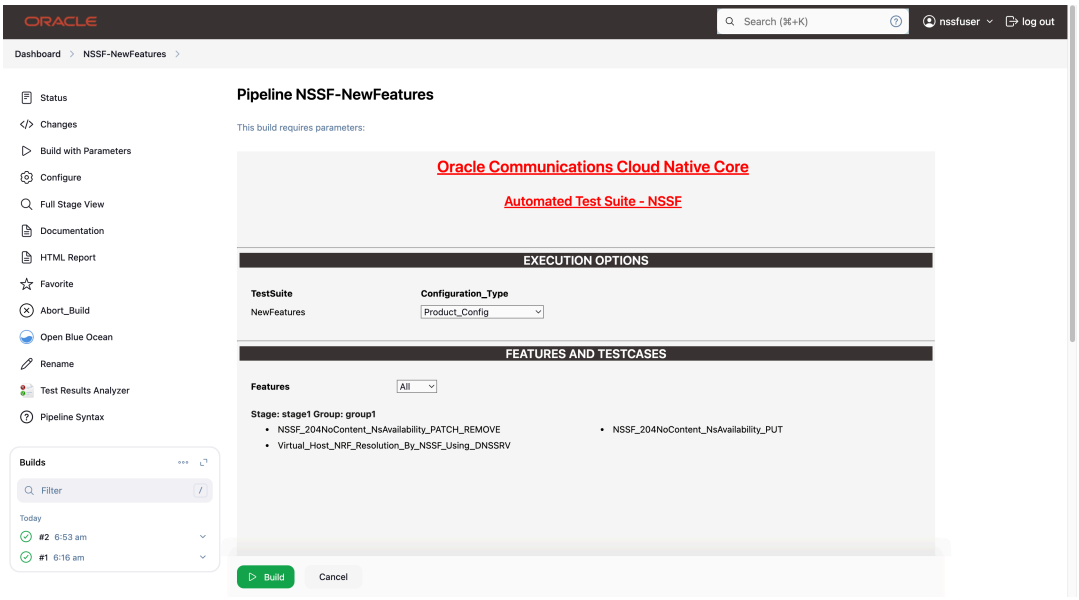
To run NSSF New Features test cases:

1. Go back to new features and Click **Build with Parameters** present on the NSSF-NewFeatures screen in the extreme left column corresponding to NSSF-NewFeatures row as shown below:

Copyright © 2019, 2025, Oracle and/or its affiliates.



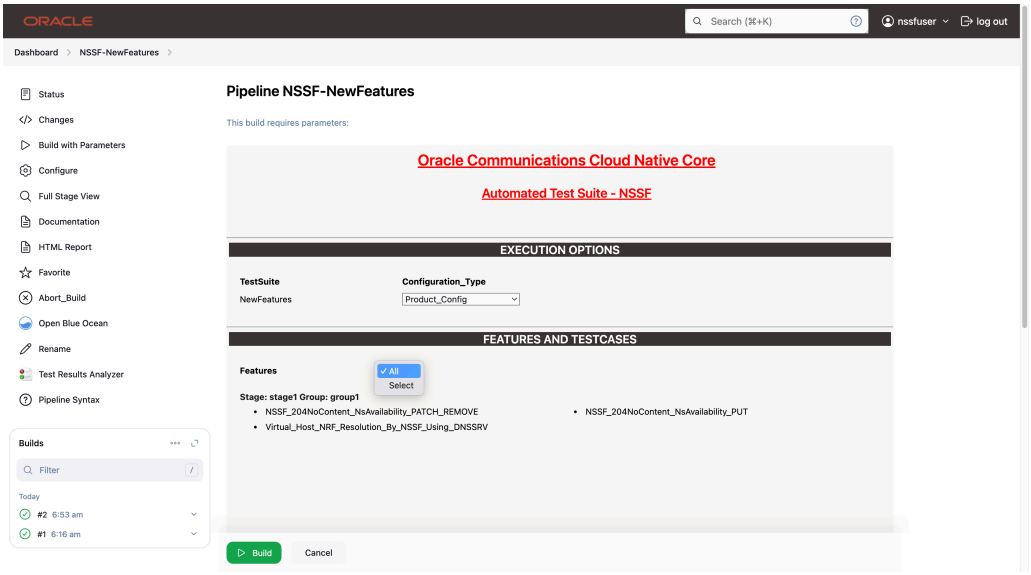
2. The following screen appears:

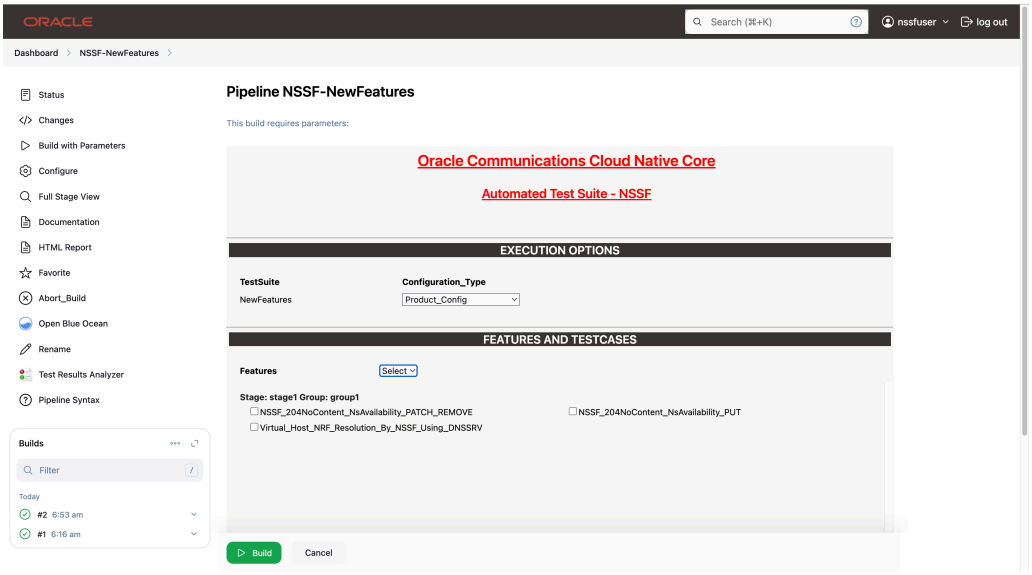


In the above screen, there are three **Select_Option(s)**, which are:

- By default, features 'ALL', configuration type 'Product_config', include regression 'NO' will be selected, and all test cases will be executed once clicked on BUILD, as shown above.
- Features: ALL. Once you click on ALL, a dropdown 'Select' option will appear. After selecting 'Select,' options will appear to choose feature files. To run the selected feature files, click on 'Build,' as shown below.

Copyright © 2019, 2025, Oracle and/or its affiliates.





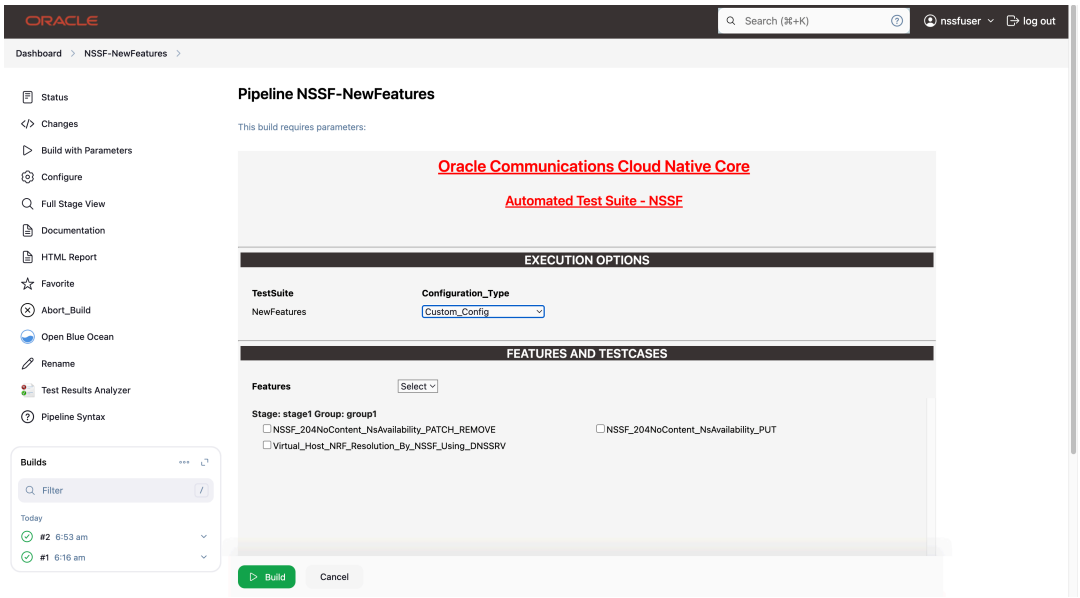
3. Select one of the following configuration types:
   - **Product_Config**: On selecting this option, test cases from product folders are populated on ATS UI and product configuration is applied to them via the keyvalue pair and yaml files defined or present in the "Product Config" folder.
   - **Custom_Config**: On selecting this option, test cases from custom folders are populated on ATS UI and custom configuration is applied to them via the keyvalue pair and yaml files defined or present in the "Custom Config" folder. To use the Parameterization feature, always select the Custom_Config option. User can copy, add, or delete the required test cases that are available for the NSSF and place them appropriately within the custom folder for NSSF-NewFeatures. Reload the page to view the test cases available in the custom NewFeatures folder. For more information, see Parameterized approach for SUT custom configuration.

   The NSSF test cases are divided into NSSF Service operations as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

- NSSF_204NoContent_NsAvailability_PATCH_REMOVE- This feature file contains test cases related to PATCH or PUT request for deleting all slices.
- Virtual_Host_NRF_Resolution_By_NSSF_Using_DNSSRV- This feature file contains test cases related to DNS SRV based selection of NRF in NSSF.

If you want to run custom_config, select below parameter as custom_config, where the framework automatically points out to custom_config and cust_data.




```
3 features passed, 0 failed, 0 skipped
25 scenarios passed, 0 failed, 0 skipped
546 steps passed, 0 failed, 0 skipped, 0 undefined
Took 36m30.275s
```

Parameterized approach for SUT custom configuration

Using this feature, user can make the following customizations to custom folders:

- Add new test cases by adding datafiles
- Remove test cases
- Modify the parameters and their values in the key-value pair or `<feature>`.yaml files

To run ATS test cases, user can maintain as many versions of Custom_Config folder by using the following naming convention:

Cust ConfigN

where N can be any number

At the time of execution, ensure to rename the required folder to Custom Config folder as Jenkins always retrieves data from this folder when user selects Custom_Config.

Updating Global Parameters

To use Custom_Config, it is required to change the value of cust_folder from data to cust_data in global.yaml file. In addition, you can customize the parameters and their respective values in the global.yaml as per the requirements.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Updating Feature Parameters

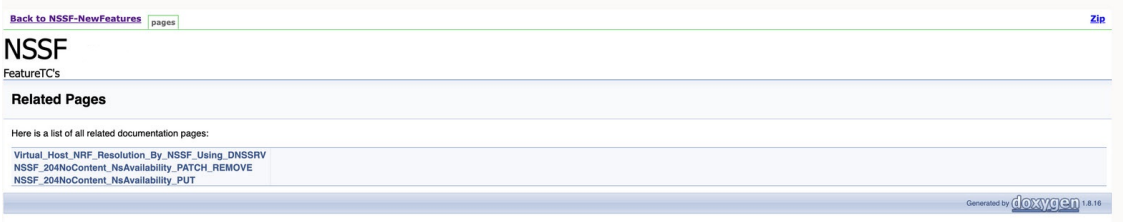
Consider the following points when customizing <feature>.yaml files for parameterized feature:

- In addition to global.yaml parameters, feature files may also contain parameters for which user can update values at the time of running pipelines.
- Changing the values of parameters tagged as "Feature Specific Value" may cause failures at the time of running pipelines.
- Values for parameters tagged with #START_GLOBAL and #END_GLOBAL tags take values from global.yaml.

> **Note**
>
> For NSSF-ATS release 25.1.201, parameterization is supported for NSSF NewFeatures pipeline only.

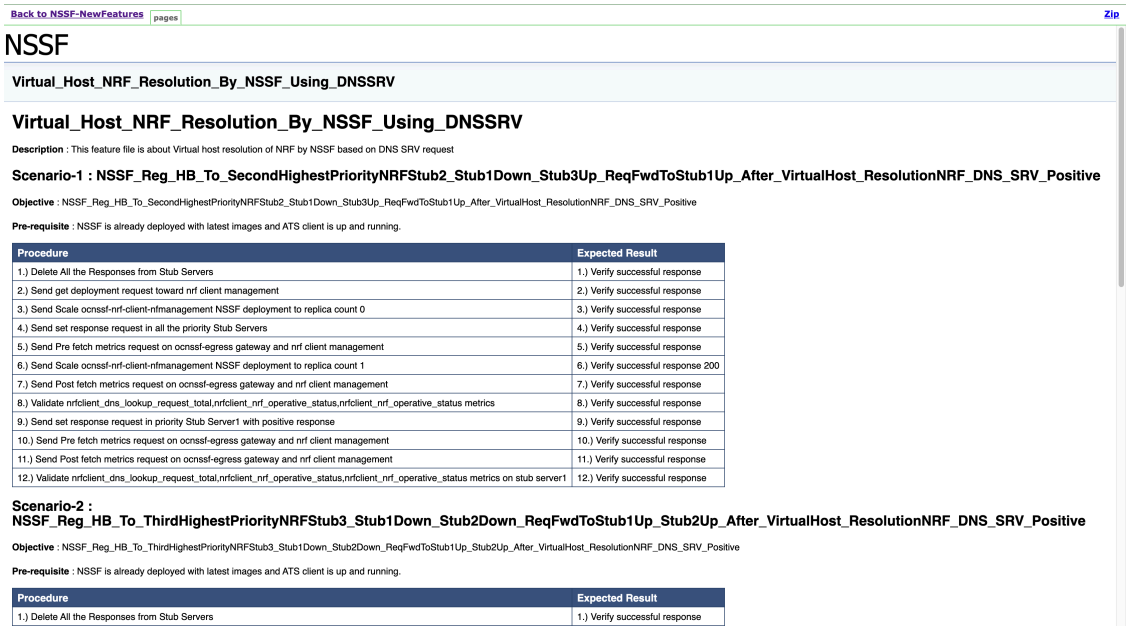
4.3.4 NSSF-NewFeatures Documentation

To view NSSF functionalities, go to **NSSF-NewFeatures** pipeline latest build and click the **Documentation** link in the left navigation pane. The following screen appears. Click any functionality to view its test cases and scenarios of each test case as shown in the sample screenshot below:



A sample of a few documentation features are as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.



Back to NSSF-NewFeatures pages Zip

NSSF

Virtual_Host_NRF_Resolution_By_NSSF_Using_DNSSRV

Virtual_Host_NRF_Resolution_By_NSSF_Using_DNSSRV

Description : This feature file is about Virtual host resolution of NRF by NSSF based on DNS SRV request

Scenario-1 : NSSF_Reg_HB_To_SecondHighestPriorityNRFStub2_Stub1Down_Stub3Up_ReqFwdToStub1Up_After_VirtualHost_ResolutionNRF_DNS_SRV_Positive

Objective : NSSF_Reg_HB_To_SecondHighestPriorityNRFStub2_Stub1Down_Stub3Up_ReqFwdToStub1Up_After_VirtualHost_ResolutionNRF_DNS_SRV_Positive

Pre-requisite : NSSF is already deployed with latest images and ATS client is up and running.

Procedure	Expected Result
1.) Delete All the Responses from Stub Servers	1.) Verify successful response
2.) Send get deployment request toward nrf client management	2.) Verify successful response
3.) Send Scale ocnssf-nrf-client-nfmanagement NSSF deployment to replica count 0	3.) Verify successful response
4.) Send set response request in all the priority Stub Servers	4.) Verify successful response
5.) Send Pre fetch metrics request on ocnssf-egress gateway and nrf client management	5.) Verify successful response
6.) Send Scale ocnssf-nrf-client-nfmanagement NSSF deployment to replica count 1	6.) Verify successful response 200
7.) Send Post fetch metrics request on ocnssf-egress gateway and nrf client management	7.) Verify successful response
8.) Validate nrfclient_dns_lookup_request_total,nrfclient_nrf_operative_status,nrfclient_nrf_operative_status metrics	8.) Verify successful response
9.) Send set response request in priority Stub Server1 with positive response	9.) Verify successful response
10.) Send Pre fetch metrics request on ocnssf-egress gateway and nrf client management	10.) Verify successful response
11.) Send Post fetch metrics request on ocnssf-egress gateway and nrf client management	11.) Verify successful response
12.) Validate nrfclient_dns_lookup_request_total,nrfclient_nrf_operative_status,nrfclient_nrf_operative_status metrics on stub server1	12.) Verify successful response

Scenario-2 : NSSF_Reg_HB_To_ThirdHighestPriorityNRFStub3_Stub1Down_Stub2Down_ReqFwdToStub1Up_Stub2Up_After_VirtualHost_ResolutionNRF_DNS_SRV_Positive

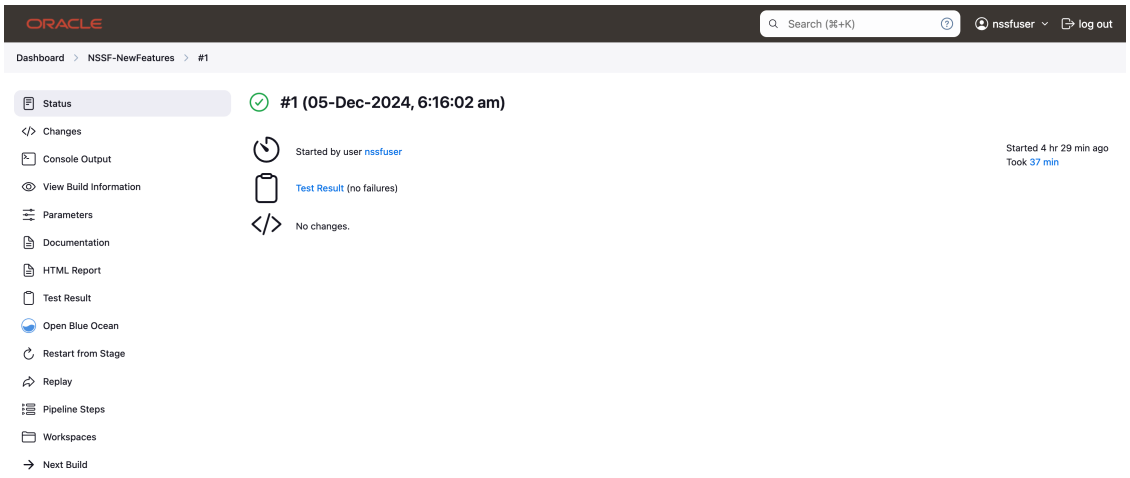
Objective : NSSF_Reg_HB_To_ThirdHighestPriorityNRFStub3_Stub1Down_Stub2Down_ReqFwdToStub1Up_Stub2Up_After_VirtualHost_ResolutionNRF_DNS_SRV_Positive

Pre-requisite : NSSF is already deployed with latest images and ATS client is up and running.

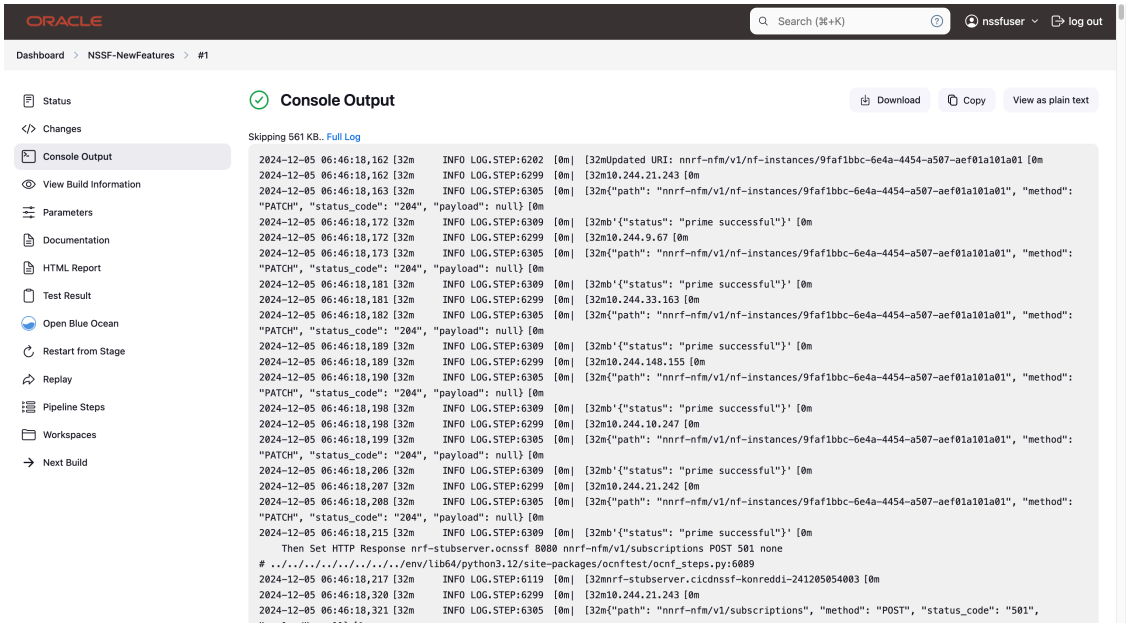
Procedure	Expected Result
1.) Delete All the Responses from Stub Servers	1.) Verify successful response

Once the run of new features done successfully, click on the build number whose logs are to be viewed and then click on **Console Output** on left side navigation.

Wait till the page is loaded completely, then click **Download** to download the new features logs.



Copyright © 2019, 2025, Oracle and/or its affiliates.



4.3.5 NSSF-Regression Pipeline

This pipeline contains test cases from the previous versions.

Some of the test cases are updated as per the new implementation of NSSF.

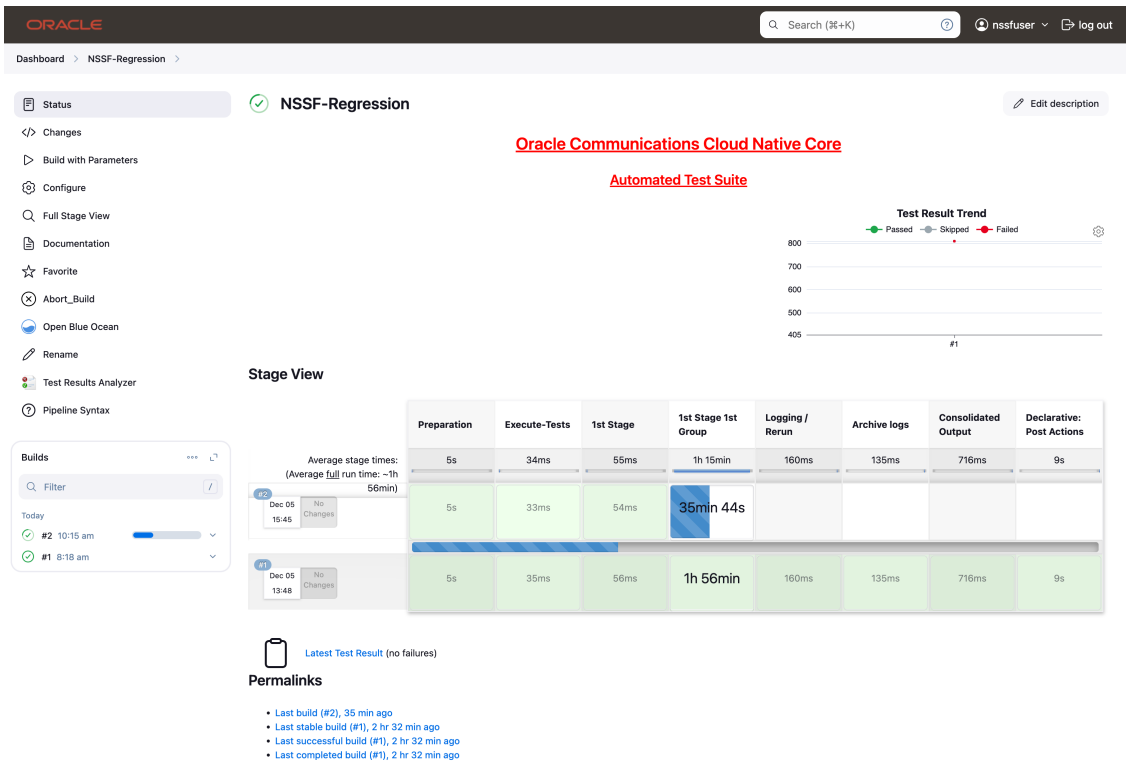
The configuration method and parameters are the same as the **NewFeatures** pipeline.

The NSSF test cases are divided into NSSF Service operations. The classification is as follows:

- **Availability Update:** These feature files are listed with a prefix as "Update".
- **Configuration:** These feature files are listed with a prefix as "failure".
- **Registration:** These feature files are listed with a prefix as "NsSelection_Registration".
- **PDU Session:** These feature files are listed with a prefix as "NsSelection_PDU".
- **SuportedFeatures:** This feature file has Feature Negotiation test cases.
- **subscription_patch:** This feature file has SUBMOD test cases.
- **eanan:** This feature file has EANAN test cases.
- **TAI Range**: This feature file has TAI Range supported NsAvailability and NsSelection test cases.
- **NSSF_Sending_Notification_via_ARS_SCP**: This feature file has test cases related to NRF discovery and subscription.
- **NSSF Sanity**: This feature file contains all the basic sanity cases for NSSF ATS 24.3.0.
- **NSSF_Allowed_NSSAI_computation_RegUeConf**: This feature file contains test cases related to NSSF Allowed SNSSAI Computation based on enabling and disabling of System Options PLMN based.
- **NSSF_Allowed_NSSAI_computation_ueconfig_update**: This feature file contains test cases related to NSSF Allowed SNSSAI Computation based on enabling and disabling of System Options PLMN based.

Copyright © 2019, 2025, Oracle and/or its affiliates.

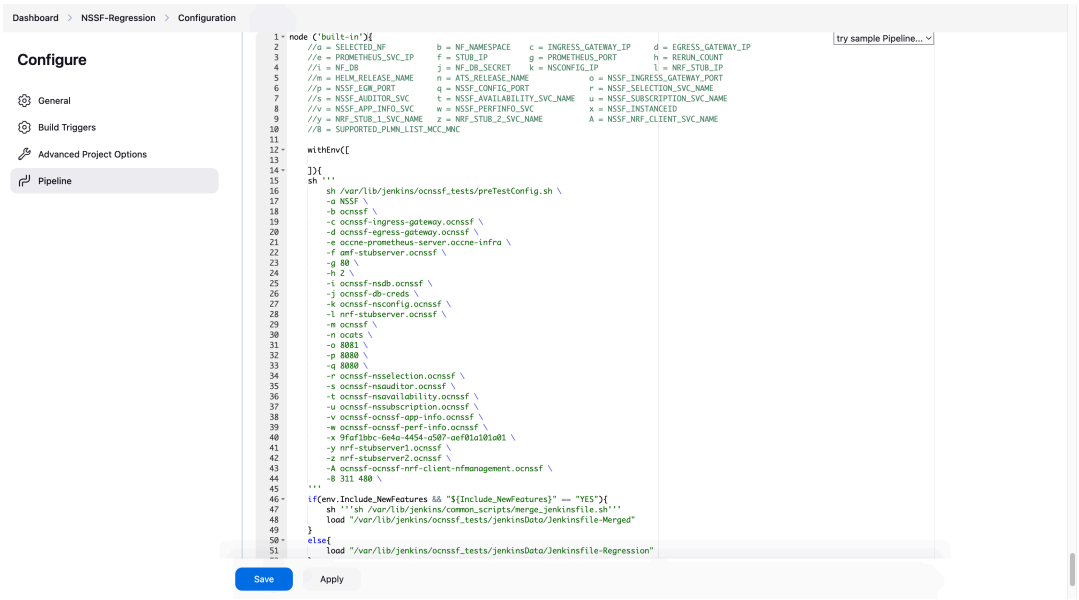
- **NSSF_Authorization_NonTrustedAMF_NsAvailPATCH**: This feature file contains test cases related to PATCH request for Update operation of NSSAI Availability in cases of Non Trusted AMFs.
- **NSSF_Authorization_NonTrustedAMF_NsAvailPUT**: This feature file contains test cases related to PUT request for Update operation of NSSAI Availability in cases of Non Trusted AMFs.
- **AMF_TrustedListing**: This feature file contains test cases related to Trusted of AMFs and Configuration of NsAvailability PUT request.



1. In the above screen:
   - Click **Configure** to access the configuration screen.
   - Click **Documentation** to view the documented test cases.
   - Click blue dots inside the **Build History** box to view the success console logs of the "All" and "Sanity" respectively.
   - The **Stage View** represents the already run pipeline for customer reference.
2. Click **Configure**. Users MUST wait for the page to load completely. Once the page loads completely, click the **Pipeline** tab to reach the Pipeline configuration as shown below:

> **Warning**
>
> Make sure that the screen shown above loads completely before you perform any action on it. Also, do not modify any configuration other than that discussed below.

Copyright © 2019, 2025, Oracle and/or its affiliates.



3. You can modify script pipeline parameters from "a" to "B" on the basis of your deployment environment and click Save. The content of the pipeline script is as follows:

```
node ('built-in'){
    //a = SELECTED_NF            b = NF_NAMESPACE    c =
INGRESS_GATEWAY_IP     d = EGRESS_GATEWAY_IP
    //e = PROMETHEUS_SVC_IP     f = STUB_IP         g =
PROMETHEUS_PORT        h = RERUN_COUNT
    //i = NF_DB                 j = NF_DB_SECRET    k =
NSCONFIG_IP            l = NRF_STUB_IP
    //m = HELM_RELEASE_NAME     n = ATS_RELEASE_NAME            o =
NSSF_INGRESS_GATEWAY_PORT
    //p = NSSF_EGW_PORT         q = NSSF_CONFIG_PORT            r =
NSSF_SELECTION_SVC_NAME
    //s = NSSF_AUDITOR_SVC      t = NSSF_AVAILABILITY_SVC_NAME  u =
NSSF_SUBSCRIPTION_SVC_NAME
    //v = NSSF_APP_INFO_SVC     w = NSSF_PERFINFO_SVC           x =
NSSF_INSTANCEID
    //y = NRF_STUB_1_SVC_NAME   z = NRF_STUB_2_SVC_NAME         A =
NSSF_NRF_CLIENT_SVC_NAME
    //B = SUPPORTED_PLMN_LIST_MCC_MNC

    withEnv([

    ]){
    sh '''
        sh /var/lib/jenkins/ocnssf_tests/preTestConfig.sh \
        -a NSSF \
        -b ocnssf \
        -c ocnssf-ingress-gateway.ocnssf \
        -d ocnssf-egress-gateway.ocnssf \
        -e occne-prometheus-server.occne-infra \
        -f amf-stubserver.ocnssf \
        -g 80 \
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
            -h 2 \
            -i ocnssf-nsdb.ocnssf \
            -j ocnssf-db-creds \
            -k ocnssf-nsconfig.ocnssf \
            -l nrf-stubserver.ocnssf \
            -m ocnssf \
            -n ocats \
            -o 8081 \
            -p 8080 \
            -q 8080 \
            -r ocnssf-nsselection.ocnssf \
            -s ocnssf-nsauditor.ocnssf \
            -t ocnssf-nsavailability.ocnssf \
            -u ocnssf-nssubscription.ocnssf \
            -v ocnssf-ocnssf-app-info.ocnssf \
            -w ocnssf-ocnssf-perf-info.ocnssf \
            -x 9faf1bbc-6e4a-4454-a507-aef01a101a01 \
            -y nrf-stubserver1.ocnssf \
            -z nrf-stubserver2.ocnssf \
            -A ocnssf-ocnssf-nrf-client-nfmanagement.ocnssf \
            -B 311 480 \
        '''
        if(env.Include_NewFeatures && "${Include_NewFeatures}" == "YES"){
            sh '''sh /var/lib/jenkins/common_scripts/merge_jenkinsfile.sh'''
            load "/var/lib/jenkins/ocnssf_tests/jenkinsData/Jenkinsfile-Merged"
        }
        else{
            load "/var/lib/jenkins/ocnssf_tests/jenkinsData/Jenkinsfile-
Regression"
        }
        }
}
```

> **Note**
>
> The User MUST NOT change any other value apart from these parameters.

The description of these parameters is as follows:

- a: Name of the NF to be tested in capital (NSSF).
- b: Namespace in which the NSSF is deployed (default is ocnssf)
- c: Ingress Gateway IP address (default is ocnssf-ingress-gateway.ocnssf)
- d: Egress Gateway IP address (default is ocnssf-egress-gateway.ocnssf)
- e: Prometheus service IP address (default is prometheus.cne-infra)
- f: Stub service IP address (default is ocats-amf-stubserver.ocnssf)
- g: Port of Prometheus service (default is 80)
- h: Number of times the re-run of failed case is allowed (default is 2).
- i: Database name (default is ocnssf-nsdb.ocnssf)
- j Database secrets (default ocnssf-db-creds)

Copyright © 2019, 2025, Oracle and/or its affiliates.

- k: NSSF config ip address (ocnssf-nsconfig.ocnssf)
- l: NRF stub server IP address (ocats-nrf-stubserver.ocnssf)
- m: NSSF release name (ocnssf)
- n: ATS release name (ocats)
- o: NSSF Ingress Gateway Port
- p: NSSF Egress Gateway Port
- q: NSSF Config Port
- r: NSSF Selection Service Name
- s: NSSF Auditor Service Name
- t: NSSF Availability Service Name
- u: NSSF Subscription Service Name
- v: APP Info Service Name
- w:Perf info Service Name
- x: NSSF NF Instance ID
- y: NRF stub Server Service name(nrf-stubserver1.ocnssf)
- z: NRF stub Server Service name(nrf-stubserver1.ocnssf)
- A: NRF Client Managment Service Name (ocnssf-ocnssf-nrf-client-nfmanagement.ocnssf)
- B: NSSF PLMN List (311 480)

Copyright © 2019, 2025, Oracle and/or its affiliates.

Note

- Do not change any value if the OCCNE cluster is used, and NSSF, ATS, and STUB are deployed in the ocnssf namespace.
- In the above image, the NSSF Helm Release Name is "ocnssf", ATS Helm Release Name is "ocats", and Namespace is "ocnssf". If any change in the helm release name of NSSF, ATS, and namespace needs to be updated accordingly.
- For example, if the NSSF Helm Release Name is "ocnssfats" and ATS Helm Release Name is "ocatsnssf", and Namespace is "ocnssf2510", then the above Pipeline Configuration should be edited as shown below:
  - NSSF Helm Release Name should be updated in "c d i k m r s t u v w" if any change in NSSF helm release name apart from "ocnssf".
- The NSSF Instance ID needs to be updated in the ATS Jenkins pipeline parameters if there are any changes in the NSSF custom values (CV) file. By default, the NSSF Instance ID is set to "`9faf1bbc-6e4a-4454-a507-aef01a101a01`". If the Instance ID is modified in the NSSF CV file, the same changes must be reflected in the Jenkins pipeline parameters.
  By default NSSF Instance ID in NSSF custom values file as below:

  ```
  #InstanceId of NSSF used in case of GR
  nfInstanceId: &nfInstanceId "9faf1bbc-6e4a-4454-a507-
  aef01a101a01"
  ```

  For example, if the nfInstanceId in the NSSF custom values file is modified from "`9faf1bbc-6e4a-4454-a507-aef01a101a01`" to "`9faf1bbc-6e4a-4454-a507-aef01a101a20`", the same change must be updated in the "x" Jenkins pipeline parameters as shown below.

```
node ('built-in'){
    //a = SELECTED_NF           b = NF_NAMESPACE    c =
INGRESS_GATEWAY_IP     d = EGRESS_GATEWAY_IP
    //e = PROMETHEUS_SVC_IP     f = STUB_IP         g =
PROMETHEUS_PORT        h = RERUN_COUNT
    //i = NF_DB                 j = NF_DB_SECRET    k =
NSCONFIG_IP            l = NRF_STUB_IP
    //m = HELM_RELEASE_NAME     n = ATS_RELEASE_NAME             o
= NSSF_INGRESS_GATEWAY_PORT
    //p = NSSF_EGW_PORT         q = NSSF_CONFIG_PORT             r
= NSSF_SELECTION_SVC_NAME
    //s = NSSF_AUDITOR_SVC      t = NSSF_AVAILABILITY_SVC_NAME   u
= NSSF_SUBSCRIPTION_SVC_NAME
    //v = NSSF_APP_INFO_SVC     w = NSSF_PERFINFO_SVC            x
= NSSF_INSTANCEID
    //y = NRF_STUB_1_SVC_NAME   z = NRF_STUB_2_SVC_NAME          A
= NSSF_NRF_CLIENT_SVC_NAME
    //B = SUPPORTED_PLMN_LIST_MCC_MNC
    withEnv([

    ]){
    sh '''
        sh /var/lib/jenkins/ocnssf_tests/preTestConfig.sh \
        -a NSSF \
```

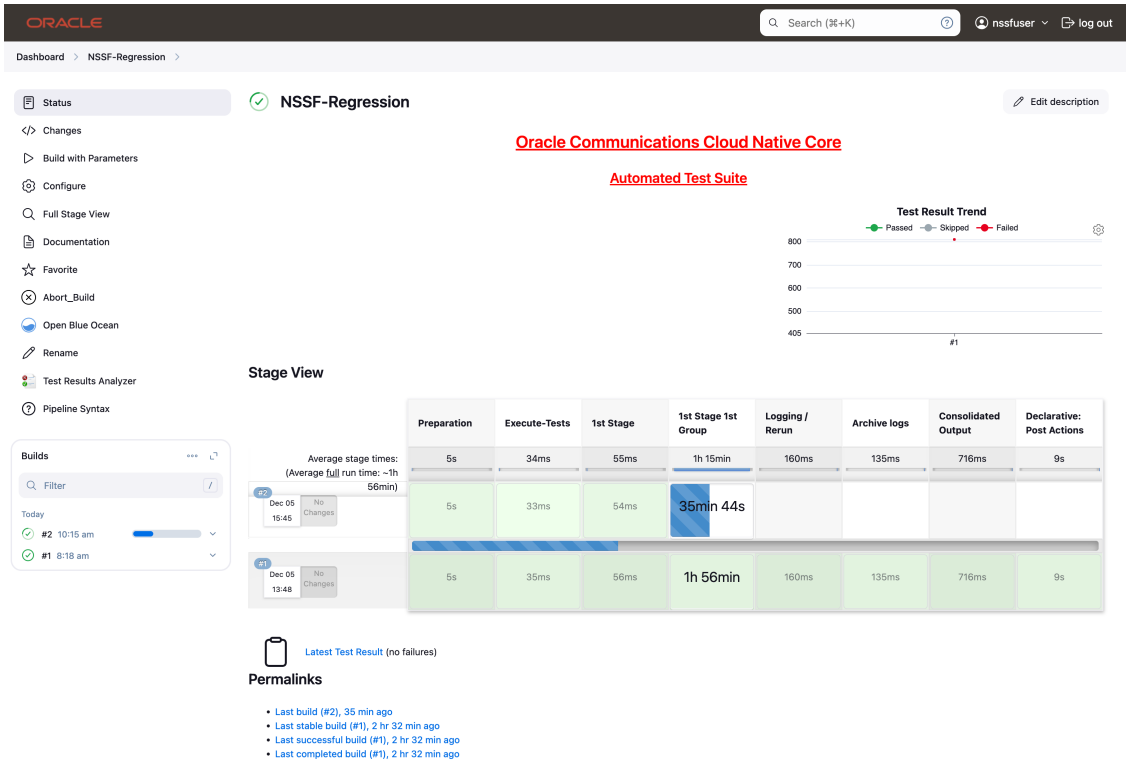

Copyright © 2019, 2025, Oracle and/or its affiliates.

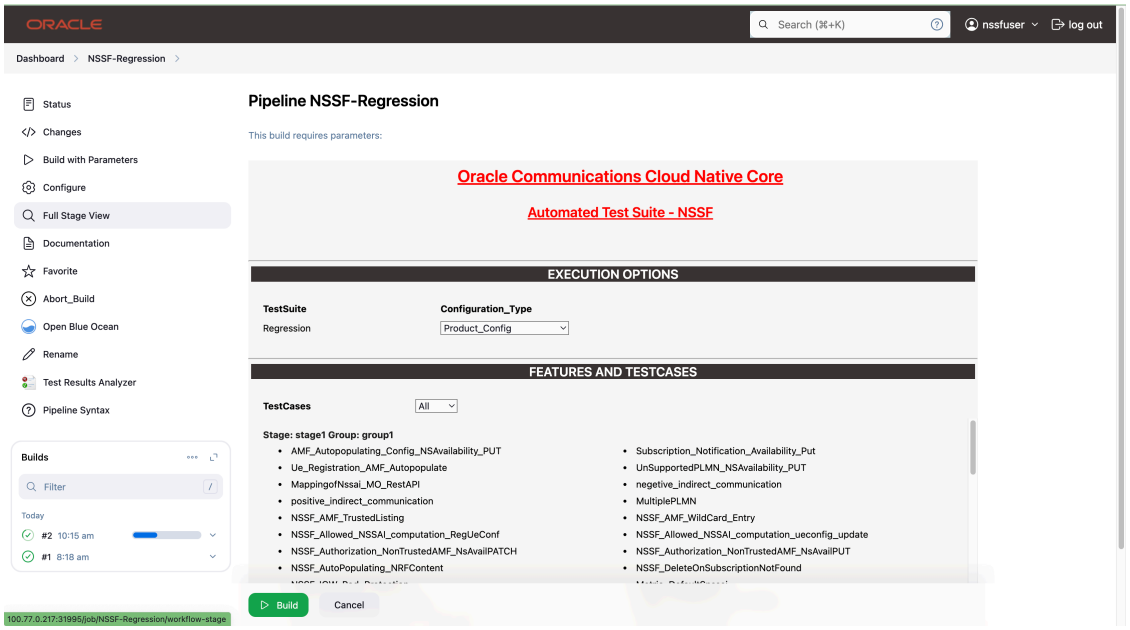
```
            -b ocnssf2510 \
            -c ocnssfats-ingress-gateway.ocnssf2510 \
            -d ocnssfats-egress-gateway.ocnssf2510 \
            -e occne-prometheus-server.occne-infra \
            -f amf-stubserver.ocnssf2510 \
            -g 80 \
            -h 2 \
            -i ocnssf-nsdb.ocnssf \
            -j ocnssf-db-creds \
            -k ocnssfats-nsconfig.ocnssf2510 \
            -l nrf-stubserver.ocnssf2510 \
            -m ocnssfats \
            -n ocats \
            -o 8081 \
            -p 8080 \
            -q 8080 \
            -r ocnssfats-nsselection.ocnssf2510 \
            -s ocnssfats-nsauditor.ocnssf2510 \
            -t ocnssfats-nsavailability.ocnssf2510 \
            -u ocnssfats-nssubscription.ocnssf2510 \
            -v ocnssfats-ocnssf-app-info.ocnssf2510 \
            -w ocnssfats-ocnssf-perf-info.ocnssf2510 \
            -x 9faf1bbc-6e4a-4454-a507-aef01a101a20 \
            -y nrf-stubserver1.ocnssf2510 \
            -z nrf-stubserver2.ocnssf2510 \
            -A ocnssfats-ocnssf-nrf-client-nfmanagement.ocnssf2510 \
            -B 311 480 \
        '''
        if(env.Include_Regression && "${Include_Regression}" == "YES"){
            sh '''sh /var/lib/jenkins/common_scripts/
merge_jenkinsfile.sh'''
            load "/var/lib/jenkins/ocnssf_tests/jenkinsData/Jenkinsfile-
Merged"
        }
        else{
            load "/var/lib/jenkins/ocnssf_tests/jenkinsData/Jenkinsfile-
Regression"
        }
        }
        }
}
```

4. Click **Save** after making necessary changes. The **Pipeline NSSF-Regression** screen appears.

NSSF-Regression - Build with Parameters

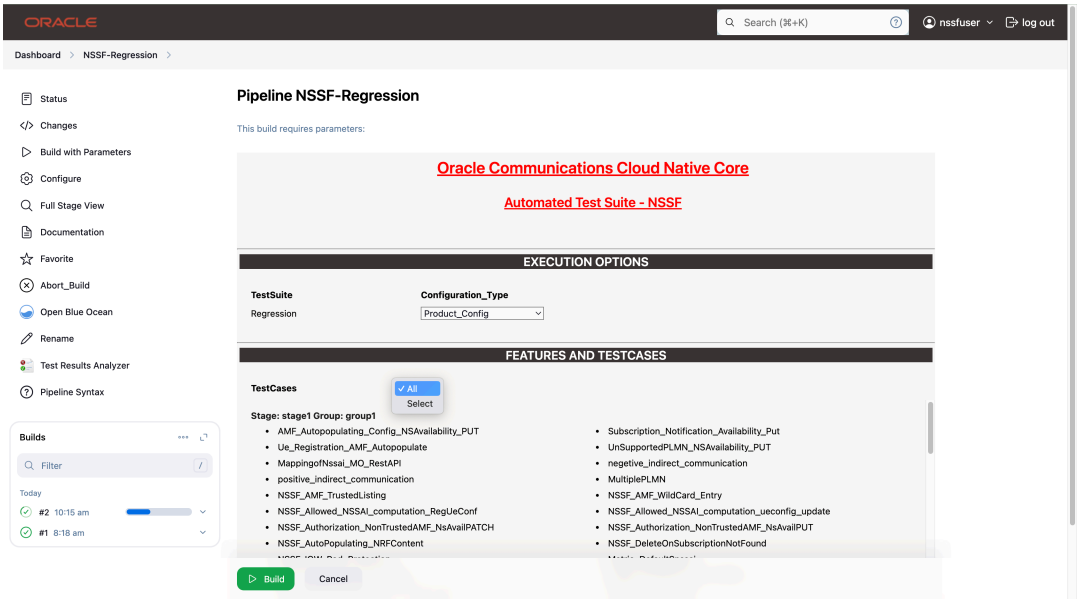
Copyright © 2019, 2025, Oracle and/or its affiliates.

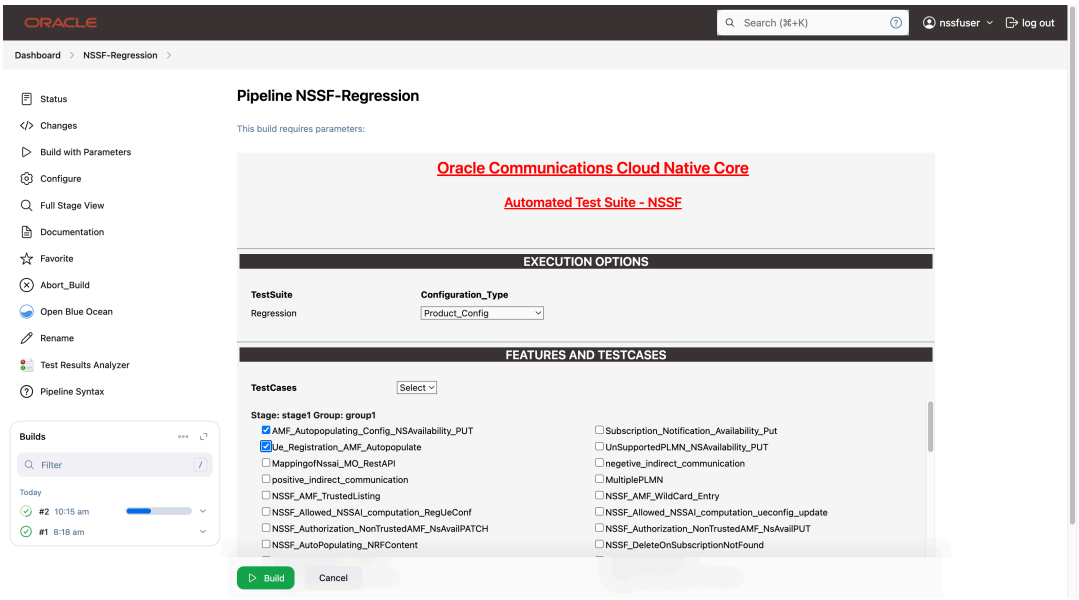




- By default, features "ALL," configuration type "Product_config," and include new features "NO" will be selected. All test cases will be executed once you click on BUILD, as shown above.
- Features: ALL. Once you click on ALL, a dropdown "Select" option will appear. After selecting the "Select" option, as shown below, options will appear to select feature files. To run the selected feature files, click on "Build," as shown below.

Copyright © 2019, 2025, Oracle and/or its affiliates.



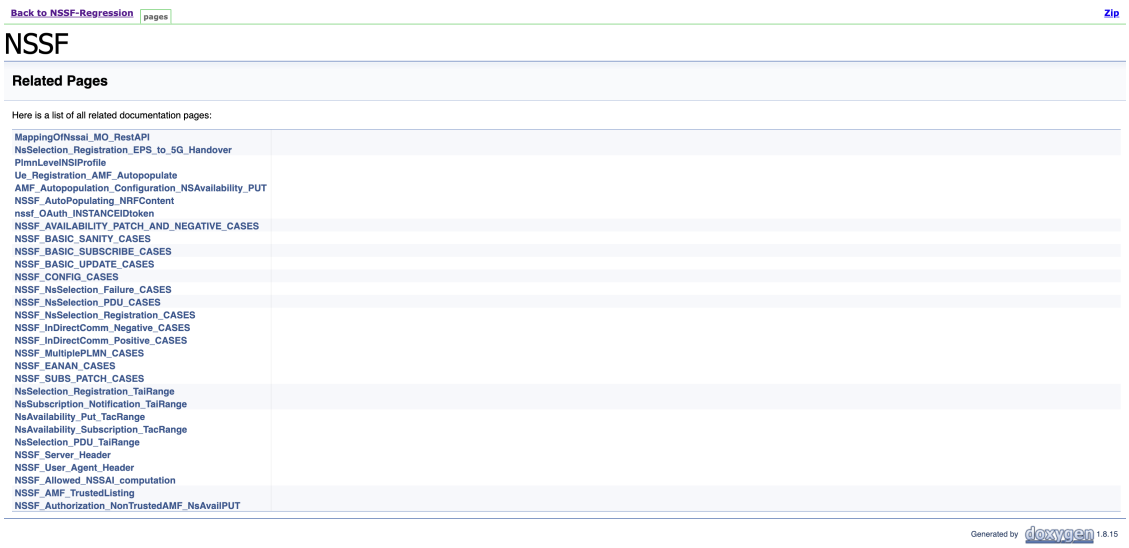



```
80 features passed, 1 failed, 0 skipped
812 scenarios passed, 1 failed, 0 skipped
10129 steps passed, 1 failed, 4 skipped, 0 undefined
```

4.3.6 NSSF-Regression Documentation

To view NSSF Regression cases, go to **NSSF-Regression** latest pipeline build and click the **Documentation** link in the left navigation pane. The following screen appears. Click any functionality to view its test cases and scenarios of each test case as shown below:

Copyright © 2019, 2025, Oracle and/or its affiliates.


Back to NSSF-Regression pages Zip

NSSF

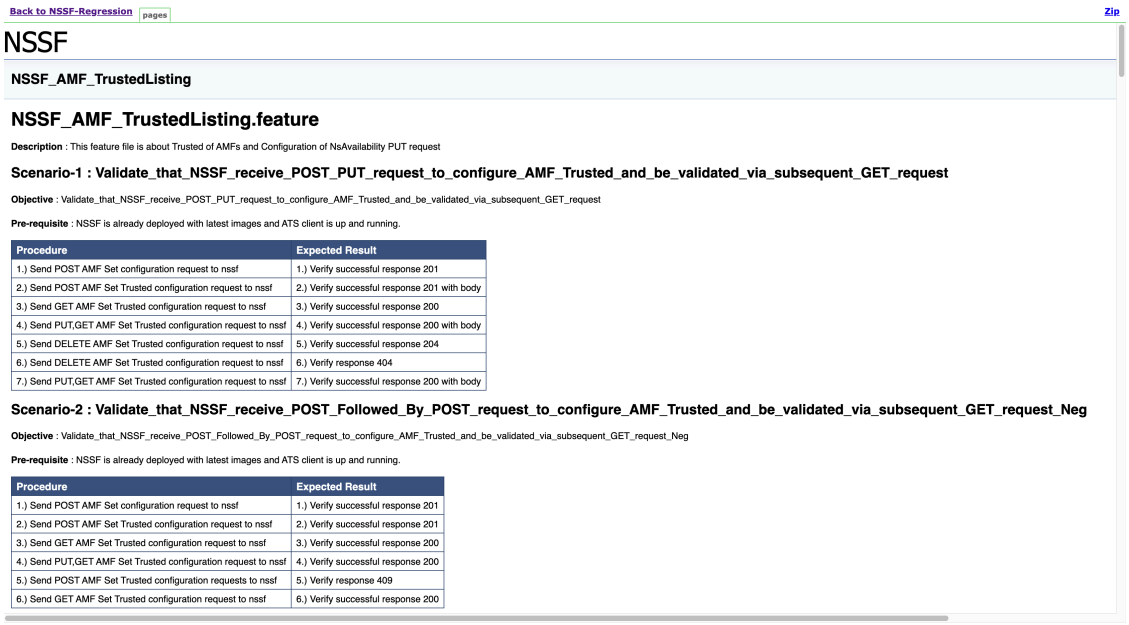
Related Pages

Here is a list of all related documentation pages:

MappingOfNssai_MO_RestAPI
NsSelection_Registration_EPS_to_5G_Handover
PlmnLevelNSIProfile
Ue_Registration_AMF_Autopopulate
AMF_Autopopulation_Configuration_NSAvailability_PUT
NSSF_AutoPopulating_NRFContent
nssf_OAuth_INSTANCEIDtoken
NSSF_AVAILABILITY_PATCH_AND_NEGATIVE_CASES
NSSF_BASIC_SANITY_CASES
NSSF_BASIC_SUBSCRIBE_CASES
NSSF_BASIC_UPDATE_CASES
NSSF_CONFIG_CASES
NSSF_NsSelection_Failure_CASES
NSSF_NsSelection_PDU_CASES
NSSF_NsSelection_Registration_CASES
NSSF_InDirectComm_Negative_CASES
NSSF_InDirectComm_Positive_CASES
NSSF_MultiplePLMN_CASES
NSSF_EANAN_CASES
NSSF_SUBS_PATCH_CASES
NsSelection_Registration_TaiRange
NsSubscription_Notification_TaiRange
NsAvailability_Put_TacRange
NsAvailability_Subscription_TacRange
NsSelection_PDU_TaiRange
NSSF_Server_Header
NSSF_User_Agent_Header
NSSF_Allowed_NSSAI_computation
NSSF_AMF_TrustedListing
NSSF_Authorization_NonTrustedAMF_NsAvailPUT

Generated by doxygen 1.8.15

A sample of a few documentation features are as follows:


Back to NSSF-Regression pages Zip

NSSF

NSSF_AMF_TrustedListing

NSSF_AMF_TrustedListing.feature

Description : This feature file is about Trusted of AMFs and Configuration of NsAvailability PUT request

Scenario-1 : Validate_that_NSSF_receive_POST_PUT_request_to_configure_AMF_Trusted_and_be_validated_via_subsequent_GET_request

Objective : Validate_that_NSSF_receive_POST_PUT_request_to_configure_AMF_Trusted_and_be_validated_via_subsequent_GET_request

Pre-requisite : NSSF is already deployed with latest images and ATS client is up and running.

Procedure	Expected Result
1.) Send POST AMF Set configuration request to nssf	1.) Verify successful response 201
2.) Send POST AMF Set Trusted configuration request to nssf	2.) Verify successful response 201 with body
3.) Send GET AMF Set Trusted configuration request to nssf	3.) Verify successful response 200
4.) Send PUT,GET AMF Set Trusted configuration request to nssf	4.) Verify successful response 200 with body
5.) Send DELETE AMF Set Trusted configuration request to nssf	5.) Verify successful response 204
6.) Send DELETE AMF Set Trusted configuration request to nssf	6.) Verify response 404
7.) Send PUT,GET AMF Set Trusted configuration request to nssf	7.) Verify successful response 200 with body

Scenario-2 : Validate_that_NSSF_receive_POST_Followed_By_POST_request_to_configure_AMF_Trusted_and_be_validated_via_subsequent_GET_request_Neg

Objective : Validate_that_NSSF_receive_POST_Followed_By_POST_request_to_configure_AMF_Trusted_and_be_validated_via_subsequent_GET_request_Neg

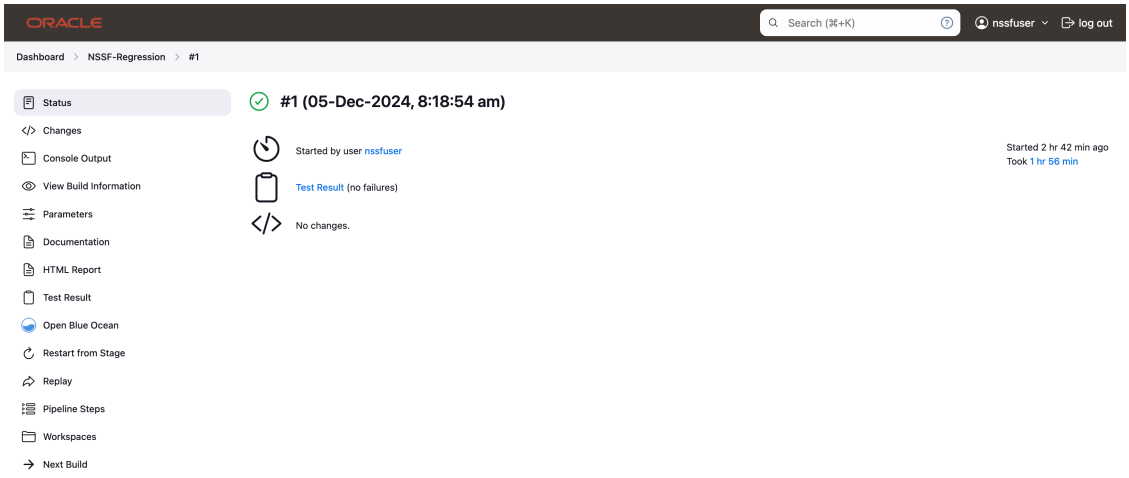
Pre-requisite : NSSF is already deployed with latest images and ATS client is up and running.

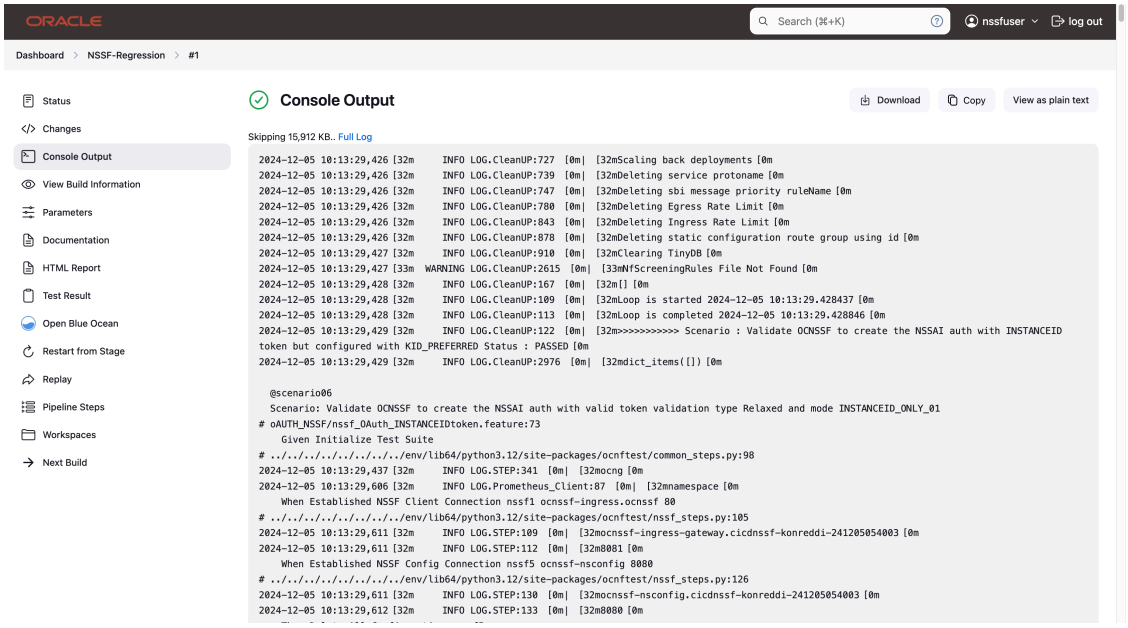
Procedure	Expected Result
1.) Send POST AMF Set configuration request to nssf	1.) Verify successful response 201
2.) Send POST AMF Set Trusted configuration request to nssf	2.) Verify successful response 201
3.) Send GET AMF Set Trusted configuration request to nssf	3.) Verify successful response 200
4.) Send PUT,GET AMF Set Trusted configuration request to nssf	4.) Verify successful response 200
5.) Send POST AMF Set Trusted configuration requests to nssf	5.) Verify response 409
6.) Send GET AMF Set Trusted configuration request to nssf	6.) Verify successful response 200

Once the run of regression features is done successfully, click on the build number whose logs are to be viewed and then click on **Console Output** on left side navigation.

Wait till the page is loaded completely, then click **Download** to download the regression features logs.

Copyright © 2019, 2025, Oracle and/or its affiliates.





4.4 Running Policy Test Cases using ATS

This section describes how to run CNC Policy test cases using ATS.

For this ATS-Policy release, scenarios or test cases are included for the following three deployment modes:

- Converged Policy
- PCF only for both TLS enabled (server side) and disabled mode
- PCRF only

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> Restart the NRF-client pod of CNC Policy for UDR and CHF discovery as part of each test case.

4.4.1 Prerequisites

PCF only deployment

To run CNC Policy test cases in PCF mode, ensure that the following prerequisites are met:

- Deploy CNC Policy 25.1.200 with default helm configurations using helm charts to run all test cases. The ATS version must be compatible with CNC Policy 25.1.200. For more information on how to install CNC Policy, see *Oracle Communications Cloud Native Core Policy Installation, Upgrade and Fault Recovery Guide*.
- For running ATS features for optional services such as LDAP, Notifier, and Usage Monitoring, ensure that the respective service is enabled on CNC Policy. If this service is not enabled, the test case for this feature will fail.
- For running SOAP notification service, ensure that the SOAP connector service is enabled on CNC Policy. If this service is not enabled, the test case for this feature will fail.
- For running ATS features, ensure to update the following mandatory parameters in ocpcf_custom_values_25.1.200.yaml file only when you are not using the minimal custom values.yaml file.

```
logging:
  burst:
    rate: 500
    max: 3000
    onMismatch: DENY
    logLevel: DEBUG
```

> **Note**
>
> Please ensure that you use the latest version of the Custom Values file when installing Policy initially.

- For using Controlled Shutdown feature, ensure that `enableControlledShutdown` parameter is enabled on CNC Policy. If this parameter is not enabled, the test case for this feature will fail.
- Install `py-stub`, `ocamf-stub`, `ocdns-bind stub`, and `oc-ldap stub` for Converged Policy and PCF deployment modes.
- Ensure there is a single pod for each microservice in the CNC Policy deployment for ATS Prometheus metrics validation to work.
- Users can add new test cases, remove unwanted test cases, and modify existing test cases in the custom test case folders (**cust_newfeatures**, **cust_regression**, and **cust_performance**). It does not impact the original test cases packaged with the product under newfeatures, regression, and performance folders. For more details about custom test case folders, see Custom Folder Implementation.
- The ATS prometheus metrics validation works only when:

Copyright © 2019, 2025, Oracle and/or its affiliates.

- the metrics suffixes are not configured
- installation has a single pod for each microservice in the BSF deployment

- For PRE and Policy_addition timer, Update PRE deployment to reduce the value of POLLING_INTERVAL from 10s to 5s and POLLING_INTERVAL_POLICYTABLE from 30s to 10s.

Figure 4-30 PRE and Policy_addition timer

```
[cloud-user@platform-bastion-1 ~]$ kubectl get deploy -n ocpcf | grep 'pre'
ocpcf-ocpm-pre                      1/1     1            1           10h
```

Run the following command to reduce the value of polling deployment:

```
[cloud-user@platform-bastion-1 ~]$ kubectl edit deploy ocpcf-ocpm-pre -n
ocpcf
```

- The following deployment is displayed:

Figure 4-31 POLLING_INTERVAL and POLLING_INTERVAL_POLICYTABLE Deployment

```
- name: POLLING_INTERVAL
  value: "5000"
- name: POLLING_INTERVAL_POLICYTABLE
  value: "10000"
```

> **Note**
>
> The value of POLLING_INTERVAL can be updated to 5000 and POLLING_INTERVAL_POLICYTABLE to 10000 if it is not updated.

For the policy_additon timer, the following changes have been made in the config.xml file for NewFeatures, Performance, and Regression.

Figure 4-32 Policy Addition Timer

```
//a = SELECTED_NF     b = PCF_NAMESPACE         c = PROMSVC_NAME        d = GOSTUB_NAMESPACE
//e = SECURITY        f = PCF_NFINSTANCE_ID     g = POD_RESTART_TIME    h = POLICY_TIME
//i = NF_NOTIF_TIME   j = RERUN_COUNT           k =INITIALIZE_TEST_SUITE  l = STUB_RESPONSE_TO_BE_SET
//m = POLICY_CONFIGURATION_ADDITION             n = POLICY_ADDITION       o = NEXT_MESSAGE
//p = PROMSVCIP       q = PROMSVCPORT           r = TIME_INT_POD_DOWN     s = POD_DOWN_RETRIES
//t = TIME_INT_POD_UP    u = POD_UP_RETRIES     v = ELK_WAIT_TIME     w = ELK_HOST
//x = ELK_PORT   y = STUB_LOG_COLLECTION z = LOG_METHOD  A = enable_snapshot B = svc_cfg_to_be_read C = PCF_API_ROOT
```

For every SUT, change the of value of -n from 15 to 7, as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-33 Change in SUT Value

```
-k 0 \
-l 1 \
-m 1 \
-n 7 \
-o 1 \
-p occne-prometheus-
-q 80\
-r 30\
```

- In the `application-config` configmap, configure the following parameters with the respective values:
  - Set the value of `primaryNrfApiRoot` as `primaryNrfApiRoot=nf1stub.<namespace_pystubs_are_deployed_in>.svc:8080`. For example:

    `primaryNrfApiRoot=nf1stub.ocats.svc:8080`
  - Remove `secondaryNrfApiRoot`.
    For example: `secondaryNrfApiRoot=nf1stub.ocats.svc:8080`
  - Set the value of `nrfClientSubscribeTypes` to `UDR,CHF,NWDAF`.
  - Remove `supportedDataSetId`.
    For example: `supportedDataSetId=POLICY`

    > **Note**
    >
    > These values can also be configured during Policy deployment.
    >
    > Run the following command to get all configmaps in your namespace:
    >
    > `kubectl get configmaps -n <Policy_namespace>`
  - Set the value of `retryAfterTime` equal to `PT30S`.
- Before running Policy test cases using ATS, restart `nrf-client-nfdiscovery` and `nrf-client-nfmanagement` pods.
- Edit the Alternate Route Service deployment pointing towards DNS Stub.
  1. Run the following command to get searches information from dns-bind pod to enable communication between Alternate Route and dns-bind service:

     ```
     kubectl exec -it <dns-bind pod> -n <NAMESPACE> -- /bin/bash -c
     'cat /etc/resolv.conf' | grep search | tr ' ' '\n' | grep -v 'search'
     ```

     The following output is displayed after running the command:

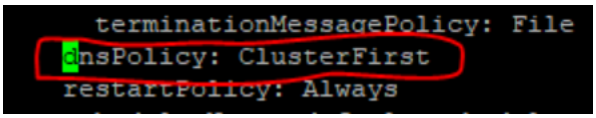
Figure 4-34 Sample Output

```
[flanders@bastion-1 ~]$ kubectl exec -it ocdns-stub-ocdns-bind-7467cf56c8-5dqg2 -n flanders-ns -- /bin/bash -c 'cat /etc/resolv.conf' | grep search | tr '
' '\n' | grep -v 'search'
flanders-ns.svc.inferno.lab.us.oracle.com
svc.inferno.lab.us.oracle.com
inferno.lab.us.oracle.com
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

By default alternate service will point to CoreDNS and you will see following settings in deployment file:

Figure 4-35 Alternate Route Service Deployment File



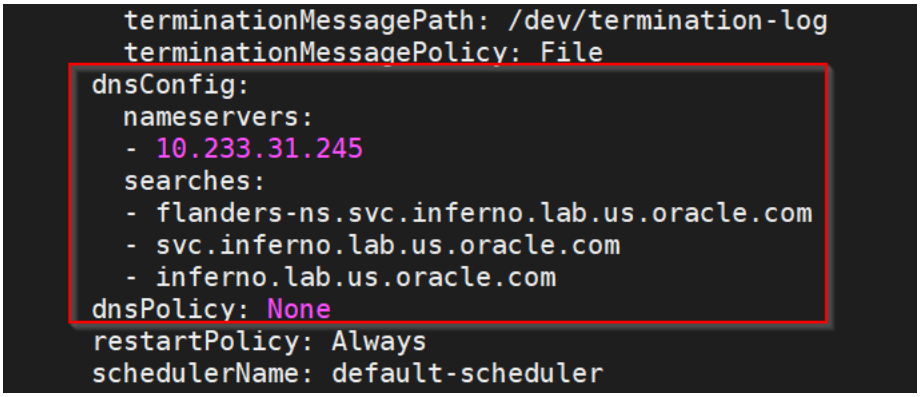
2. Run the following command to edit the deployment file and add the following content in alternate service to query DNS stub:

```
$kubectl edit deployment ocpcf-occnp-alternate-route -n ocpcf
```

   a. Add the IP Address of the nameserver that you have recorded after installing the DNS stub (cluster IP Address of DNS Stub).
   b. Add the search information one by one which you recorded earlier.
   c. Set `dnsPolicy` to "None".

```
dnsConfig:
  nameservers:
  - 10.233.33.169      // cluster IP of DNS Stub
  searches:
  - ocpcf.svc.occne15-ocpcf-ats
  - svc.occne15-ocpcf-ats
  - occne15-ocpcf-ats
dnsPolicy: None
```

Figure 4-36 Example



PCRF only deployment

To run CNC Policy test cases in PCRF mode, ensure that the following prerequisites are met:

- Run the following command to set the log level to Debug in Diam-GW statefulset:

  ```
  kubectl edit statefulset <diam-gw statefulset name> -n <namespace>
  ```

  name: LOGGING_LEVEL_APP

Copyright © 2019, 2025, Oracle and/or its affiliates.

value: DEBUG

- Set the peer Configuration using configuration map for backend peers.
  Edit and set the default configuration of Diameter peer in the diam configuration map and set `responseOnly` flag to true.

```
kubectl edit cm oc-diam-gateway-config-peers -n <namespace>

nodes:
  - name: 'ocpcf-occnp-pcrf-core'
    type: 'pcrf'
    responseOnly: true
    host: ocpcf-occnp-pcrf-core-headless
    port: 3868
    realm: ''
    identity: ''
```

Converged mode deployment

To run CNC Policy test cases in Converged mode, ensure that the following prerequisites are also met:

- Deploy CNC Policy 25.1.200 with default helm configurations using helm charts to run all test cases. The ATS version must be compatible with CNC Policy 25.1.200. For more information on how to install CNC Policy, see *Oracle Native Core, Converged Policy Installation, Upgrade, and Fault Recovery Guide*.
- For running ATS features for optional services such as LDAP, Notifier, and Usage Monitoring, ensure that the respective service is enabled on CNC Policy.
- Install Prometheus server in cluster.
- Database cluster should be in a running state with all the required tables. Ensure there are no previous entries in database before executing test cases.
- Deploy ATS in the same namespace as Policy using Helm Charts.
- User must not initiate a job in two different pipelines at the same time.
- The installation should have only one pod for each microservice related to ATS Prometheus Metrics validation to work in the CNC Policy deployment.
- If Service Mesh check is enabled, apply the below settings:
  For fetching the metrics from the Prometheus, create a destination rule. In most deployments, Prometheus is kept outside of the service mesh. Hence, a destination rule is required to communicate between TLS enabled entity (ATS) and non-TLS entity (Prometheus). Created the rule as follows:

```
kubectl apply -f - <<EOF

apiVersion:networking.istio.io/v1alpha3
kind:DestinationRule
metadata:
  name:prometheus-dr
  namespace:ocats
spec:
  host:oso-prometheus-server.pcf.svc.cluster.local
  trafficPolicy:
    tls:
      mode:DISABLE
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
EOF

Here,

1.) name,indicates the name of destination rule.
2.) namespace,indicates where the ATS is deployed.
3.) host,this indicates the hostname of the prometheus server.
```

- For running ATS features for Model D Indirect Communication Model, ensure that the custom values yaml of CNC Policy has the following parameters with the specified values:

```
routeConfigMode: REST
configureDefaultRoute: true
```

- Update the `occnp-servicemesh-config-custom-values-25.1.200.yaml` of CNC Policy with the following `virtualService` configuration for Egress Gateway:

```
virtualService:
#  - name: nrfvirtual1
#    host: ocpcf-occnp-egress-gateway
#    destinationhost: ocpcf-occnp-egress-gateway
#    port: 8000
#    exportTo: |-
#      [ "." ]
#    attempts: "0"
```

- Restart the ocpm-ldap-gateway pod manually before running the LDAP-specific ATS test cases.
- To ensure the consistent functioning of ATS related to the audit service, modify the audit deployment by reducing the value of AUDIT_NOTIFY_SCHEDULER_POLLING_INTERVAL_MILLISEC from 30000 to 1000 (changing from 30 seconds to 1 second).

```
$ kubectl get deploy -n ocpcf | grep 'audit' ocpcf-ocpm-audit-service
1/1     1      1      10h


$ kubectl edit deploy ocpcf-ocpm-audit-service -n ocpcf
```

The deployment will open. Scroll down until below fields are seen.

```
- name:  AUDIT_NOTIFY_SCHEDULER_POLLING_INTERVAL_MILLISEC value: "30000"
```

Update the value of AUDIT_NOTIFY_SCHEDULER_POLLING_INTERVAL_MILLISEC to 1000 if it isn't already.

> **Note**
>
> In order to run stage2 or stage3 cases using configuration_type of custom_config, replace the "data" folder with "cust_data" folder in stage<x>.txt under /var/lib/jenkins/ocpcf_tests/cust_data/common/stage_hooks/stage<x>. directory.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Application Config map changes for Policy registrations over TLS

To enable TLS communication make the following changes by editing the config map of application-config:

- NRF port from 8080 to 8443
- nrfScheme to https

Sample:

```
apiVersion: v1
data:
  profile: |-
    [appcfg]
    primaryNrfApiRoot=nf1stub.ocats.svc:8443
    nrfScheme=https
```

> **Note**
>
> In the config map of application-config, delete the lines which has *supportedDataSetId* or *secondaryNrfApiRoot* strings.

4.4.2 Logging into ATS

Before logging into ATS GUI, it is important to get the nodeport of the service, 'ocats-policy'.

To get the nodeport, perform the following steps:

```
kubectl get svc -n <Policy_namespace>
```

Example:

```
kubectl get svc -n ocpcf
```

The output of the command is:

```
ocats-ocats-pcf     LoadBalancer   10.233.56.56   10.75.225.49
8080:31944/TCP      19h
```

To log in to ATS, open the web browser and type the following URL:

http://<Worker-Node-IP>:<Node-Port-of-ATS>

If the ocats-policy Service has an external IP available, <SVC external IP> can also be used to log in to ATS.

```
http://<SVC external IP>:8080
```

Example: http://10.75.225.49:8080

Copyright © 2019, 2025, Oracle and/or its affiliates.

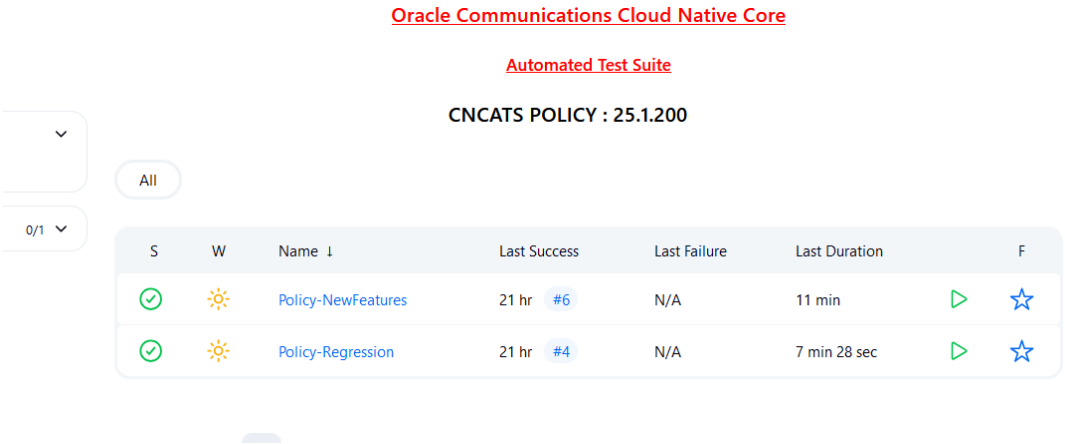
Running ATS

To run ATS test cases, perform the following steps:

1. Enter the **username** as `policyuser` and **password** as `policypasswd`.
2. Click **Sign in**.

> **Note**
>
> To modify default login password, see Modifying Login Password.

Figure 4-37 Pre-configured Pipelines



The description of each pipeline is as follows:

- **Policy-HealthCheck:** This pipeline checks if CNC Policy and ATS are deployed correctly. This shows only when the user has enabled this feature at the time of installing CNC Policy ATS.
- **Policy-NewFeatures:** This pipeline has all the new test cases delivered for CNC Policy 25.1.200.
- **Policy-Regression:** This pipeline has all the test cases delivered in Policy ATS - 24.3.x.
- **Policy-Performance:** This pipeline is not operational as of now. It is reserved for future releases of ATS.

4.4.3 Running Policy-HealthCheck Pipeline

This is a pre-configured pipeline where ATS performs a test probe with SUT. It triggers helm test & provides the results in Jenkins Console Logs.

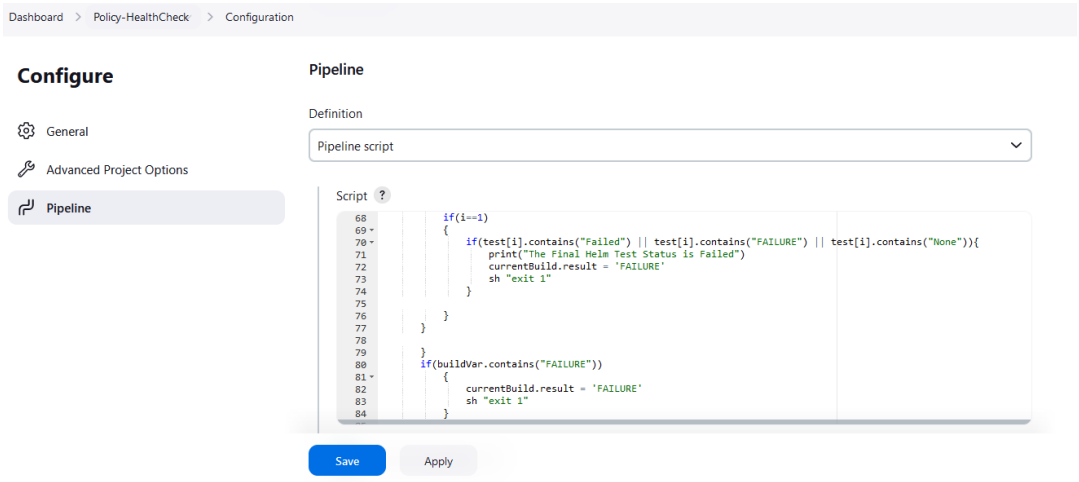
You can run Policy-HealthCheck pipeline to check if all CNC Policy pods are up and running. If yes, it provides the status as successful. If any pod is down due to any reason, then the pipeline fails.

To configure the parameters for **Policy-HealthCheck** pipeline, perform the following steps:

Copyright © 2019, 2025, Oracle and/or its affiliates.

1. Click **Policy-HealthCheck** in the **Name** column.
2. Click **Configure** in the left navigation pane.
3. When you scroll-down, the **General** tab becomes active. Be sure that the screen loads completely.
4. Continue scrolling down until the **Pipeline** tab becomes active. The following is a screen capture that shows the Pipeline script:

Figure 4-38 Helm Test Script



5. In the `Script` area under the Pipeline section, users may customize the values for the following parameters:

> **NOT_SUPPORTED**
>
> Do not modify values of parameters other than the ones described in this section.

- **a:** Change this parameter to update the helm release name where Policy is deployed in your bastion.
- **b:** Change this parameter to update the namespace where Policy is deployed in your bastion.

6. Click **Save** to update the values.

Running Helm Test

To run CNC Policy test cases, click **Build Now**.

4.4.4 Running Policy-NewFeatures Pipeline

Policy-New Features Pipeline

This is a pre-configured pipeline where all the Policy new test cases are executed.

To configure the parameters for **Policy-NewFeatures** pipeline, perform the following steps:

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> Configuring the parameters is a one-time activity.

1. Click **Policy-NewFeatures** in the **Name** column.
2. Click **Configure** in the left navigation pane.
3. When you scroll-down, the **General** tab becomes active. Be sure that the screen loads completely.
4. (Optional) If you want to retain a specified number of builds in the persistent volume, select the check box for **Discard old Builds**. It is recommended to configure this option; otherwise, Persistent Volume may fail when the builds are exceedingly high.
   Enter values for the following parameters:
   - **Days to keep builds**: Specifies the number of days for which build records are stored.
   - **Max # of builds to keep**: Specifies the number of build records to store.
5. Continue scrolling down until the **Pipeline** tab becomes active. The following is a screen capture that shows the Pipeline script:

Figure 4-39 Policy New Pipeline Configuration



6. In the `Script` area under the Pipeline section, users may customize the values for the following parameters:

> **NOT_SUPPORTED**
>
> Do not modify values of parameters other than the ones described in this section.

- **a:** Name of the NF to be tested in capital (PCF).
- **b:** Change this parameter to update the namespace where Policy is deployed in your bastion.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- **c:** Name of Prometheus service namespace (occne-prometheus-server)
- **d:** Change this parameter to update the namespace where gostubs are deployed in your bastion.
- **e:** Set this parameter as 'unsecure', if you intend to run ATS in TLS disabled mode. Else, set this parameter as 'secure'.
- **g:** Set this parameter to more than 45 secs. The default wait time for the pod is 45 secs. Every TC requires restart of the nrf-client-management pod.
- **h:** Set this parameter to more than 60 secs. The default wait time to add a configured policy to the database is 60 secs.
- **i:** Set this parameter to more than 140 secs. The default wait time for Nf_Notification Test Cases is given as 140 secs.
- **k:** Use this parameter to set the waiting time to initialize Test Suite.
- **l:** Use this parameter to set the waiting time to get response from Stub.
- **m:** Use this parameter to set the waiting time after adding Policy Configuration.
- **n:** Use this parameter to set the waiting time after adding Policy.
- **o:** Use this parameter to set the waiting time before sending next message.
- **p:** Use this parameter to set Prometheus Server IP.
- **q:** Use this parameter to set Prometheus Server Port.
- **r:** Use this parameter to set the interval after which the POD status is checked when it is down.
- **s:** Use this parameter to set number of retry attempt in which will check the pod down status
- **t:** Use this parameter to set the interval after which we check the POD status if its UP.
- **u:** Use this parameter to set number of retry attempt in which will check the pod up status.
- **v:** Use this parameter to set Wait time to connect to Elastic Search
- **w:** Use this parameter to set Elastic Search HostName
- **x:** Use this parameter to set Elastic Search Port
- **y:** Use this parameter to set To Enable/Disable Stub logs collection
- **z:** Use this parameter to set Log collection endpoint either Elasticsearch or Kubernetes
- **A:** Use this parameter to enable or disable snapshots that are created at the start and restored at the end of each test run
- **B:** Use this parameter to for timer to wait for importing service configurations
- **C:** Use this parameter to set PCF_API_ROOT information for Ingress gateway service name and port.
- **D:** Use this parameter to set bulk_import_to_complete to add custom time in Jenkins post bulk imports.
- **E:** Use this parameterto configure TLS_VERSION to define the supported TLS version (1.2 or 1.3).

(*Optional*)To collect application logs per failed scenario, user can configure the values for the following parameter:

Copyright © 2019, 2025, Oracle and/or its affiliates.

- **z:** If you want log collection to happen through Elastic search, set the value for this parameter as `Elasticsearch`. If not, specify the value as `Kubernetes`.
  If you want to collect logs Elastic search, it is required to configure the values for the following parameters:
  - **v:** Specifies the wait time to connect to Elastic search (`ELK_WAIT_TIME`).
  - **w:** Specifies the host name of Elastic search (`ELK_HOST`). For example, `occne-elastic-elasticsearch-master.occne-infra/`
  - **x:** Specifies the port for Elastic search (`ELK_PORT`). For example, 9200.
- **y:** If you want to collect stub logs, set the value for this parameter as `yes`. If not, specify the value as `no`.

7. **A**: To enable or disable snapshots that are created at the start and restored at the end of each test run.
8. Click **Save** to update the values. The Policy_NewFeatures Pipeline page appears.

> **Note**
>
> It is recommended to save the pipeline script in your local machine as it is needed at the time of ATS pod restart.

Extracting Application Logs

If any scenarios failed after running the pipeline job, perform the following steps:

1. Log in to the ATS pod:

   ```
   kubectl exec -it pod/occnp1-ocats-ocats-policy-xxxxxxxx -n ocpcf bash
   ```

2. Go to build directory:

   ```
   cd $JENKINS_HOME/jobs/$JOB_NAME/builds/$BUILD_NUMBER/
   ```

   For example:

   ```
   cd /var/lib/jenkins/.jenkins/jobs/Policy-Regression/builds/1
   ```

3. Extract the `applogs.zip` file:

   ```
   unzip applogs.zip
   ```

4. After successfully unzipping the file, open the `applog` folder to view pod logs for failed scenarios:

   ```
   (env) [jenkins@gapcf-ocats-ocats-policy-6df5cd84c-c6cm4 applog]$ pwd
   /var/lib/jenkins/.jenkins/jobs/Policy-Regression/builds/3/applog
   (env) [jenkins@gapcf-ocats-ocats-policy-6df5cd84c-c6cm4 applog]$ ls -ltrh
   total 1.4M
   -rw-r--r--. 1 jenkins jenkins 461K Sep  1 14:49 Initial_Run-
   Send_Npcf_SMPolicyControl_Create_request_message_to_PCF_and_verify_the_praI
   nfos_structure_is_downloaded_in_the_response_message_to_SMF_also_check_requ
   ests_total_metric_incremented_in_the_PCF.log
   -rw-r--r--. 1 jenkins jenkins 459K Sep  1 14:49 1st_Rerun-
   ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

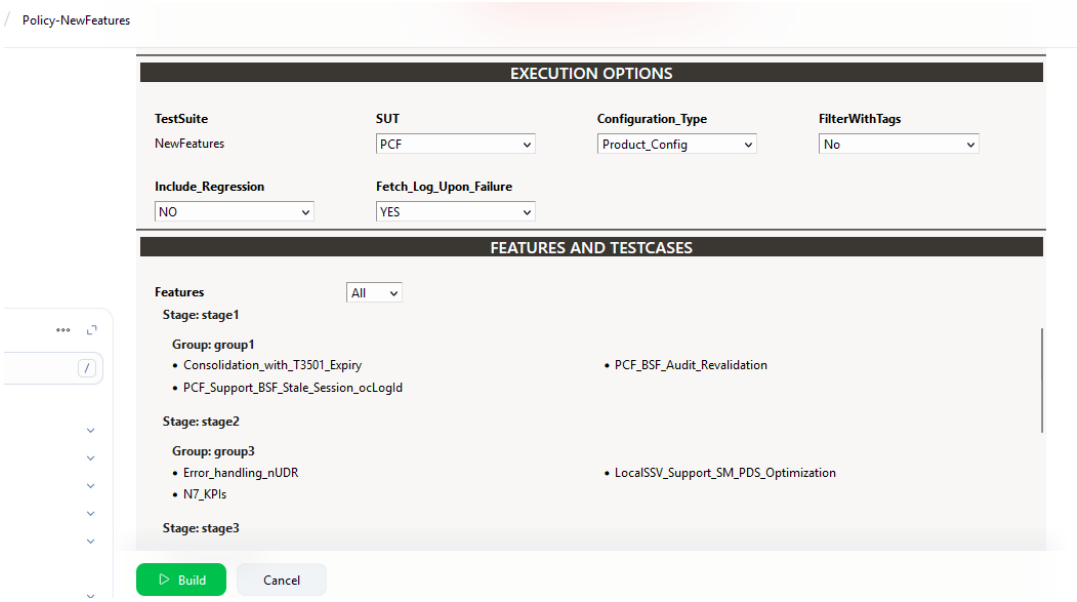
```
Send_Npcf_SMPolicyControl_Create_request_message_to_PCF_and_verify_the_praI
nfos_structure_is_downloaded_in_the_response_message_to_SMF_also_check_requ
ests_total_metric_incremented_in_the_PCF.log
-rw-r--r--. 1 jenkins jenkins 459K Sep  1 14:49 2nd_Rerun-
Send_Npcf_SMPolicyControl_Create_request_message_to_PCF_and_verify_the_praI
nfos_structure_is_downloaded_in_the_response_message_to_SMF_also_check_requ
ests_total_metric_incremented_in_the_PCF.log
```

Running Policy Test Cases

To run CNC Policy test cases, perform the following steps:

1. Click the **Build with Parameters** link available in the left navigation pane of the **Policy-NewFeatures** Pipeline screen. The following screen appears.

Figure 4-40 Pipeline Policy-NewFeatures



2. Select any of following valid values from the drop-down for **SUT**:
   - PCF
   - cnPCRF
   - Converged Policy
3. Select any of following valid values from the drop-down for **Configuration_Type**:
   - **Product_Config**: On selecting this option, test cases from product folders are populated on ATS UI and product configuration is applied to them via the key-value pair and yaml files defined or present in the "Product Config" folder.
   - **Custom_Config**: On selecting this option, test cases from custom folders are populated on ATS UI and custom configuration is applied to them via the key-value pair and yaml files defined or present in the "Custom Config" folder.
     To use the parameterization feature, always select the **Custom_Config** option. User can copy, add, or delete the required test cases that are available for the Converged Policy, PCF or PCRF and place them appropriately within the **custom** folder for **Policy-NewFeatures**. Reload the page to view the test cases available in the **custom** NewFeatures folder.

Copyright © 2019, 2025, Oracle and/or its affiliates.

For more information, see Parameterized approach for SUT custom configuration.

4. For the **Select_Option** field, select any of the following drop-down values:
   - **All:** By default, all the Policy test cases are selected for execution.
   - **Single/MultipleFeatures:** This option allows you to select any number of test cases you want to run from the list of total test cases. Select the check-box for each feature you want to run.

   Based on your selection, related **Test Cases** appear on the page.
5. To collect logs for any given build, select YES from the drop-down menu of **Fetch_Log_Upon_Failure**.
6. Click **Build** and select **Console Output** to view the test results. The following is a sample test result ouput:

Figure 4-41 Test Result Output in Console

```
# ../../../../../../../../../env/lib64/python3.9/site-packages/ocnftest/common_steps.py:194
2023-06-22 05:43:21,589[32m     INFO LOG.stage1.group1.STEP:331 [0m| [32mocng[0m
2023-06-22 05:43:21,596[31m    ERROR LOG.stage1.group1.Prometheus_Client:71 [0m| [31mCould not fetch scrapeInterval from prometheus[0m
2023-06-22 05:43:21,599[31m    ERROR LOG.stage1.group1.Prometheus_Client:91 [0m| [31mCould not fetch target_namespace from prometheus[0m
    Then Waiting for test_suite_to_initialize
# ../../../../../../../../../env/lib64/python3.9/site-packages/ocnftest/pcf_steps.py:2515
2023-06-22 05:43:21,606[32m     INFO LOG.stage1.group1.STEP:2532 [0m| [32mWait for 0s[0m
    Then Unset Variable
# ../../../../../../../../../env/lib64/python3.9/site-packages/ocnftest/pcf_steps.py:286
    Given Check And Set Policy Namespace
# ../../../../../../../../../env/lib64/python3.9/site-packages/ocnftest/pcf_steps.py:105
2023-06-22 05:43:21,608[32m     INFO LOG.stage1.group1.STEP:110 [0m| [32mPCF Namespace set to vega-ns5[0m
    Given Create Service Client
# ../../../../../../../../../env/lib64/python3.9/site-packages/ocnftest/pcf_steps.py:115
2023-06-22 05:43:21,883[32m     INFO LOG.stage1.group1.STEP:133 [0m| [32mATS Pod IP 10.233.102.218[0m
2023-06-22 05:43:21,883[32m     INFO LOG.stage1.group1.STEP:128 [0m| [32mDiam pod name vega-ns5-oc-diam-gateway-0[0m
2023-06-22 05:43:21,884[32m     INFO LOG.stage1.group1.STEP:130 [0m| [32mvega-ns5-oc-diam-gateway-0[0m
2023-06-22 05:43:21,884[32m     INFO LOG.stage1.group1.STEP:134 [0m| [32mDiam gw pod info - {'vega-ns5-oc-diam-gateway-0': '10.233.70.70'}[0m
2023-06-22 05:43:21,886[32m     INFO LOG.stage1.group1.STEP:140 [0m| [32mATS svc name ocats-policy[0m
2023-06-22 05:43:21,888[32m     INFO LOG.stage1.group1.STEP:212 [0m| [32mLdap org1 service name is: vega-ns5-oc-ldap-org1-service[0m
2023-06-22 05:43:21,889[32m     INFO LOG.stage1.group1.STEP:214 [0m| [32mLdap org1 service cluster IP is: 10.233.56.158[0m
2023-06-22 05:43:21,889[32m     INFO LOG.stage1.group1.STEP:218 [0m| [32mLdap org1 service port is: 8080[0m
2023-06-22 05:43:21,889[32m     INFO LOG.stage1.group1.STEP:221 [0m| [32mLdap org1 service secure port is: 8081[0m
```

> **Note**
>
> For more details on consolidated test report, see Managing Final Summary Report, Build Color, and Application Log.

Parameterized approach for SUT custom configuration

Using this feature, user can make the following customizations to custom folders

- Add new test cases by adding datafiles
- Remove test cases
- Modify the parameters and their values in the key-value pair or <feature>.yaml files

To run ATS test cases, user can maintain as many versions of Custom Config folder by using the following naming convention:

Cust ConfigN

where N can be any number

Copyright © 2019, 2025, Oracle and/or its affiliates.

At the time of execution, ensure to rename the required folder to **Cust Config** folder as Jenkins always retrieves data from this folder when user selects **Custom_Config**.

Updating Global Parameters
To use Custom_Config, it is required to change the value of ***cust_folder*** from ***data*** to ***cust_data*** in global.yaml file. In addition, you can customize the parameters and their respective values in the `global.yaml` as per the requirements.

To use custom diameter identity, update the value of parameter "diam_gw_id" under global.yaml file to the one configured for "DIAMETER_Identity" under diam-gw's statefulset.

Updating Feature Parameters
Consider the following points when customizing `<feature>.yaml` files for parameterized feature:

- In addition to `global.yaml` parameters, feature files may also contain parameters for which user can update values at the time of running pipelines.
- Changing the values of parameters tagged as "Feature Specific Value" may cause failures at the time of running pipelines.
- Values for parameters tagged with #START_GLOBAL and #END_GLOBAL tags take values from `global.yaml`.

> **Note**
>
> For CNC Policy-ATS release 25.1.200, parameterization is supported for PCF features only.

Incorrect Values
If the value to be parameterized is incorrect or not updated correctly, the following error is generated:

```
2021-12-17 09:02:43,908[32m     INFO LOG.featureHooks:14 [0m| [32mInside
before_feature[0m
2021-12-17 09:02:43,908[32m     INFO LOG.goldenConfig:305 [0m| [32mInside
goldenConfig.py[0m
2021-12-17 09:02:43,912[31m    ERROR LOG.goldenConfig:93 [0m| [31mPlease
check if the key metricsEnabled is valid or if it is present under correct
section[0m
2021-12-17 09:02:43,913[32m     INFO LOG.goldenConfig:324 [0m| [32mExiting
goldenConfig.py[0m
2021-12-17 09:02:43,913[32m     INFO LOG.featureHooks:27 [0m| [32mExiting
before_feature[0m
```

The execution continues to happen without any changes to the data files.

Config-Rollback (Snapshot Backup and Restore)

Using this feature, user can take a snapshot of initial configuration of the system, and then restore it at a later stage. On enabling this feature, a snapshot of system configurations is taken before ATS run starts and is reverted on completion of ATS job.

Queuing Jenkins Jobs

Using this feature, you can queue a second job even when current job is still running. The second job can be triggered either from the same or a different pipeline.

Copyright © 2019, 2025, Oracle and/or its affiliates.

The following table lists various scenarios on how queuing works:

Table 4-2 Queuing Jenkins Jobs

Concurrent Builds	New Features Current Build	New Features New Build	Regression Current Build	Regression New Build Build	Result
Enabled	Running	Triggered	NA	NA	New-Build of New-Features is added to queue.
Enabled	Running	NA	NA	Triggered	New-Build of Regression is added to queue.
Disabled	NA	NA	Running	Triggered	New-Build of Regression is added to queue.
Disabled	NA	Triggered	Running	NA	New-Build of New-Features is added to queue.

Test Result Analyzer

Using the Test Result Analyzer plug-in, user can view consolidated and detailed reports. For more information, see Test Results Analyzer section.

Test Case Mapping to Features and Display Total Counts

With this feature, users can view total count of features, test cases, and TestCase mapping for each feature of CNC Policy on ATS GUI. For more information, see Support for Test Case Mapping and Count.

4.4.5 Policy-NewFeatures Documentation

This section describes the documentation for **Policy-NewFeatures** pipeline.

To view the documentation for any of the CNC Policy features, on the ATS home page, click Policy-NewFeatures. Then, click **Documentation** in the left navigation pane.

On clicking Documentation, the following page opens with the list of new features:

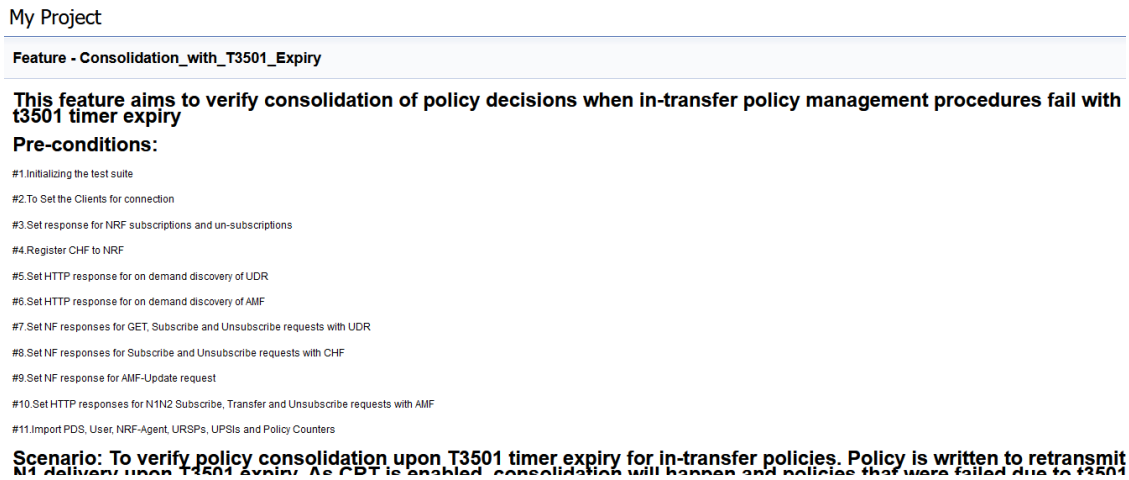
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-42 Policy-NewFeatures Feature List



Example: On clicking **Feature - Consolidation_with_T3501_Expiry**, the following screen opens:

Figure 4-43 Feature - Consolidation_with_T3501_Expiry



Based on the features covered under Documentation, the **Build Requires Parameters** screen displays test cases.

To navigate back to the Policy-NewFeatures pipeline, click **Back to Policy-NewFeatures** link available on the top left corner of the screen.

4.4.6 Running Policy-Regression Pipeline

This section describes how to run test cases for CNC Policy Regression pipeline.

The **Policy-Regression** pipeline is a pre-configured pipeline where all the test cases of previous releases are available. For an example, for Policy 25.1.200, this pipeline has all the test cases released till Policy 24.3.x.

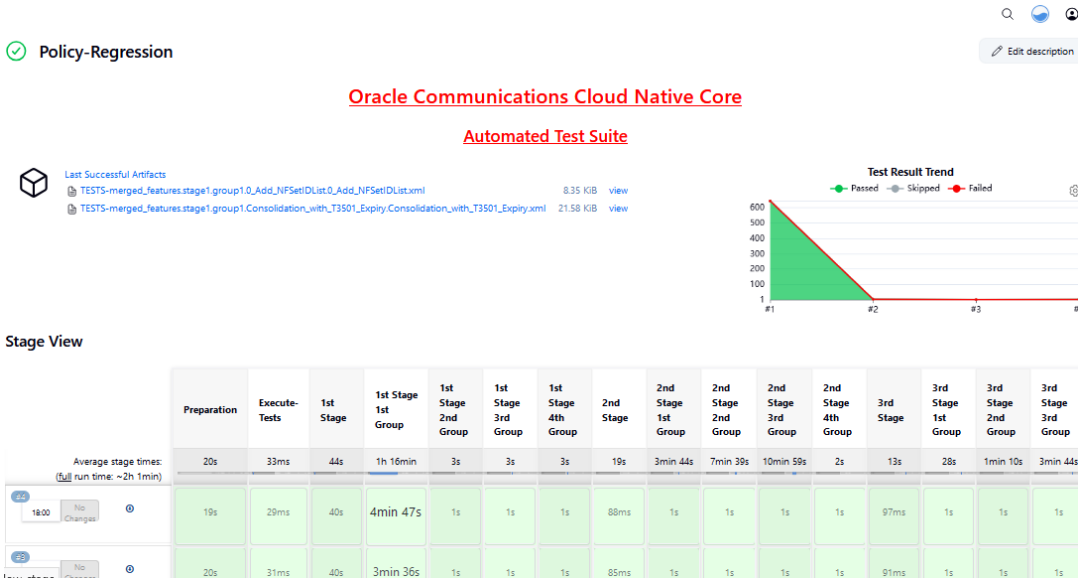
Copyright © 2019, 2025, Oracle and/or its affiliates.

Running Policy Test Cases

To run CNC Policy test cases in regression pipeline, perform the following steps:

1. Click **Policy-Regression** in the **Name** column.
2. Click the **Build with Parameters** link available in the left navigation pane of the **Policy-Regression** Pipeline screen. The following screen appears.

Figure 4-44 Regression Pipeline



> **Note**
>
> ATS display empty list for test cases as it is referring to the **custom** folder for **Policy-Regression** pipeline.

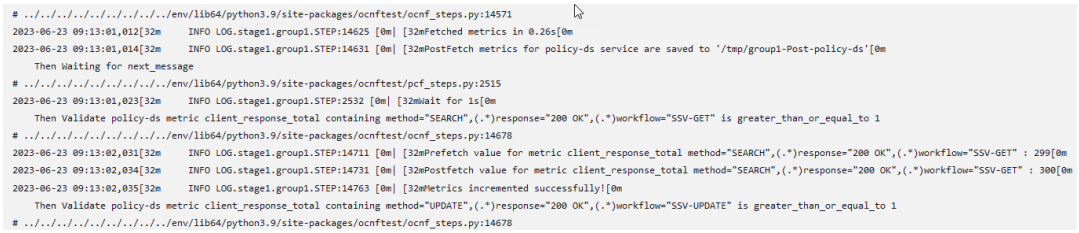
3. Select any of following valid values from the drop-down for **SUT**:
   - PCF
   - CN-PCRF
   - Converged Policy
4. Select any of following valid values from the drop-down for **Configuration_Type**:
   - **Product_Config**: On selecting this option, test cases from product folders are populated on ATS UI and product configuration is applied to them via the key-value pair and yaml files defined or present in the "Product Config" folder.
   - **Custom_Config**: On selecting this option, test cases from custom folders are populated on ATS UI and custom configuration is applied to them via the key-value pair and yaml files defined or present in the "Custom Config" folder.
     To use the parameterization feature, always select the **Custom_Config** option. User can copy, add, or delete the required test cases that are available for the Converged Policy, PCF or PCRF and place them appropriately within the **custom** folder for **Policy-NewFeatures**. Reload the page to view the test cases available in the **custom** NewFeatures folder.

Copyright © 2019, 2025, Oracle and/or its affiliates.

5. For the **Select_Option** field, select any of the following drop-down values:
   - **All:** By default, all the Policy test cases are selected for execution.
   - **Single/MultipleFeatures:** This option allows you to select any number of test cases you want to run from the list of total test cases. Select the check-box for each feature you want to run.

   Based on your selection, related **Test Cases** appear on the page.
6. To collect logs for any given build, select YES from the drop-down menu of **Fetch_Log_Upon_Failure**.
7. Click **Build** and select **Console Output** to view the test results. The following is a sample test result ouput:

Figure 4-45 Policy-Regression Console Output



> **Note**
>
> The Regression pipeline does not have any sanity option. However, users must perform all the steps performed in the **Policy-NewFeatures** pipeline. Ensure that the pipeline script is configured according to the environment variables.

> **Note**
>
> For more details on consolidated test report, see Managing Final Summary Report, Build Color, and Application Log.

4.4.7 Policy-Regression Documentation

This section describes the documentation for **Policy-Regression** pipeline.

To view the documentation for any of the CNC Policy features, on the ATS home page, click PolicyRegression. Then, click **Documentation** in the left navigation pane.

The following screen appears:

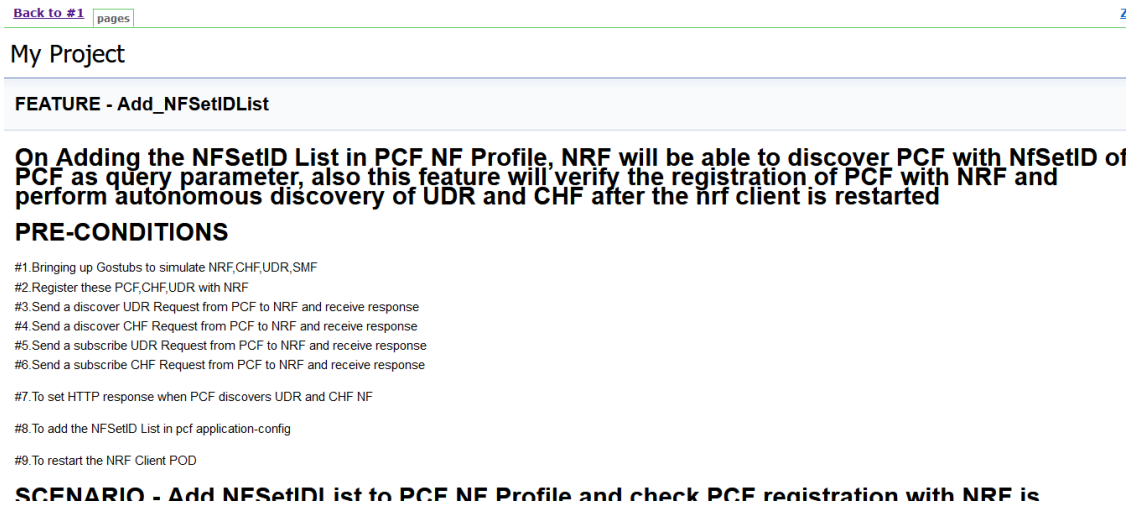
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-46 Policy-Regression Feature List



Example: On clicking **FEATURE - Add_NFSetIDList**, the following screen opens:

Figure 4-47 FEATURE - Add_NFSetIDList



This page shows features of only those test cases that are released in previous releases.

4.5 Running SCP Test Cases using ATS

This section describes how to run SCP test cases using ATS.

Copyright © 2019, 2025, Oracle and/or its affiliates.

4.5.1 Logging in to ATS

Logging in to ATS GUI in Non-OCI Setup

Before logging in to ATS, ensure that ATS and Stubs are deployed using Helm charts. A sample screenshot is provided as follows:

Figure 4-48 Verifying ATS Pod

```
[root@master ocats]# helm status ocats-ocscp -n scpsvc
NAME: ocats-ocscp
LAST DEPLOYED: Thu Aug 31 05:16:22 2023
NAMESPACE: scpsvc
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
# Copyright 2020 (C), Oracle and/or its affiliates. All rights reserved.

Thank you for installing ocats-ocscp.

Your release is named ocats-ocscp , Release Revision: 1.
To learn more about the release, try:

  $ helm status ocats-ocscp
  $ helm get ocats-ocscp
[root@master ocats]# kubectl get pods -n scpsvc | grep ocats
ocats-ocscp-ocats-scp-778f449f9d-lxfwz   1/1     Running   0          2m52s
[root@master ocats]# kubectl get svc -n scpsvc | grep ocats
ocats-ocscp-ocats-scp   LoadBalancer   10.101.29.158   <pending>     8080:30345/TCP   2m56s
```

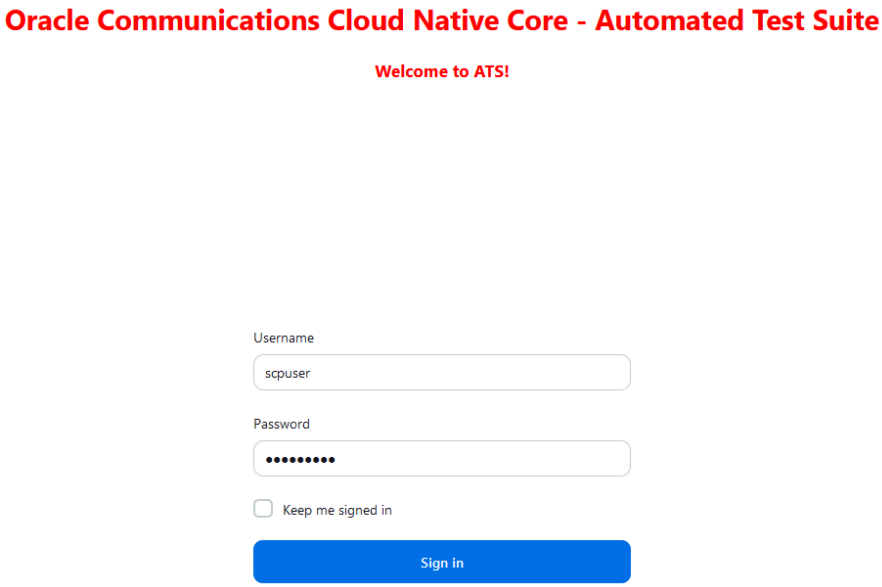
> **Note**
>
> In the Verifying ATS Pod image, the ATS nodeport is highlighted in red as **30345**. For more information about ATS deployment, see Deploying ATS and Stub in the Kubernetes Cluster. Similarly, verify Stub deployments.

To log in to the ATS GUI, open the browser and either provide the external IP address of the worker node and nodeport of the ATS service as `<Worker-Node-IP>:<Node-Port-of-ATS>` or provide the IP address of the load balancer and serviceport of the ATS service as `<ATS-LB-IP>:<Service-Port-of-ATS>`.

The ATS login screen appears.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-49 Logging in to ATS GUI



Running ATS

> **Note**
>
> It is recommended to keep the rerun count to minimum 1 for running ATS testcases.

To run ATS:

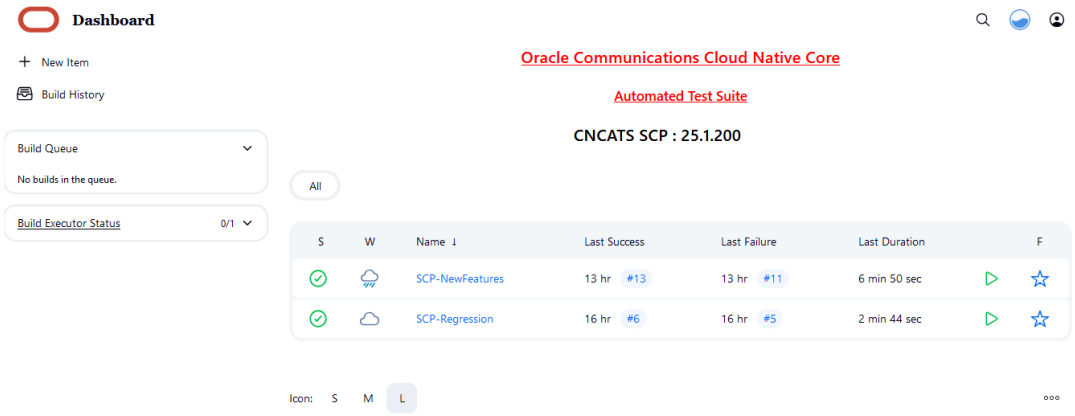
1. Enter the login credentials and click **Sign in**.

   > **Note**
   >
   > To modify default login password, see Modifying Login Password.

   The following screens appear to display preconfigured pipelines for SCP individually.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-50 ATS SCP First Login Screen



The different pre-configured pipelines for SCP are as follows:

- **SCP-NewFeatures:** This pipeline has all the test cases delivered as part of this release.
- **SCP-Regression:** This pipeline covers all the test cases from the previous releases.

Logging in to ATS GUI in OCI Setup

To access the ATS GUI in OCI, see the Accessing ATS GUI in OCI section.

Logging in to ATS GUI with HTTPS Enabled

To access the ATS GUI, see the Enable ATS GUI with HTTPS section.

> **Note**
>
> For more information, see the Support for Transport Layer Security.

4.5.2 SCP-NewFeatures Pipeline

This section describes how to configure new feature pipeline parameters, run SCP test cases, and view functionalities.

4.5.2.1 Configuring New Feature Pipelines

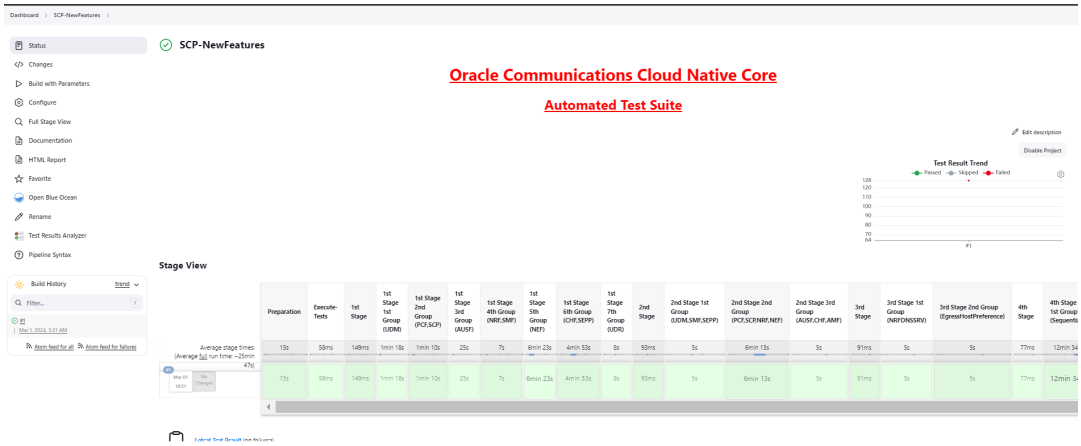
This is a pre-configured pipeline where all the SCP test cases are run.

To configure its parameters, do the following:

1. Click **SCP-NewFeatures** in the **Name** column.
   The SCP-NewFeatures screen appears.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-51 SCP-NewFeatures



According to the installation guide, the configuration parameters can be modified in the `ocats_ocscp_values.yaml` file before deployment. If you are running SCP test cases for the first time without enabling the parameters in the deployment file, then you have to set the input parameters before running any test case. There is no need to set these parameters again unless there is a change in the configuration.

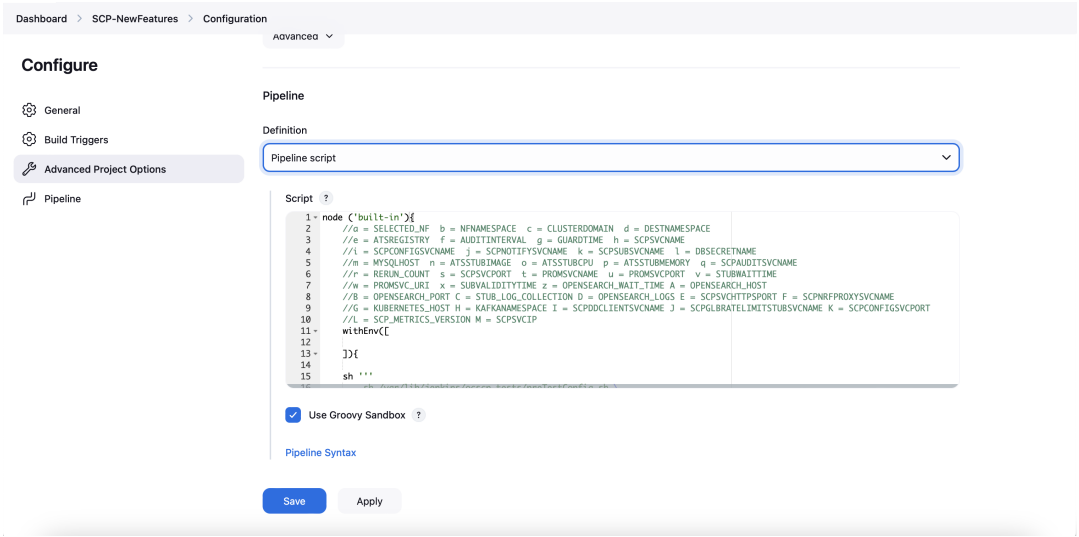
2. In the left navigation pane, click **Configure** to provide input parameters and scroll down to the pipeline script as shown in the following image:
The General tab appears.

> **Note**
>
> Ensure that the following screen loads completely before you perform any action on it. Also, do not modify any other configuration than what is specified in the subsequent steps.

The control moves from the **General** tab to the **Pipeline** tab, as shown in the following screenshot:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-52 Pipeline Tab



3. Modify the values of the pipeline script as required, and then click **Save**.
Parameter "-l" must be changed as part of "Pipeline script". The content of the pipeline script is as follows:

```
node ('built-in'){
    //a = SELECTED_NF  b = NFNAMESPACE  c = CLUSTERDOMAIN  d =
DESTNAMESPACE
    //e = ATSREGISTRY  f = AUDITINTERVAL  g = GUARDTIME  h = SCPSVCNAME
    //i = SCPCONFIGSVCNAME  j = SCPNOTIFYSVCNAME  k = SCPSUBSVCNAME  l =
DBSECRETNAME
    //m = MYSQLHOST  n = ATSSTUBIMAGE  o = ATSSTUBCPU  p = ATSSTUBMEMORY
q = SCPAUDITSVCNAME
    //r = RERUN_COUNT  s = SCPSVCPORT  t = PROMSVCNAME  u = PROMSVCPORT  v
= STUBWAITTIME
    //w = PROMSVC_URI  x = SUBVALIDITYTIME z = OPENSEARCH_WAIT_TIME A =
OPENSEARCH_HOST
    //B = OPENSEARCH_PORT C = STUB_LOG_COLLECTION D = OPENSEARCH_LOGS E =
SCPSVCHTTPSPORT F = SCPNRFPROXYSVCNAME
    //G = KUBERNETES_HOST H = KAFKANAMESPACE I = SCPDDCLIENTSVCNAME J =
SCPGLBRATELIMITSTUBSVCNAME K = SCPCONFIGSVCPORT
    //L = SCP_METRICS_VERSION M = SCPSVCIP
    withEnv([

    ]){

    sh '''
        sh /var/lib/jenkins/ocscp_tests/preTestConfig.sh \    -a SCP \
        -b scp-73999881-dev \
        -c cluster.local \
        -d scp-73999881-dev \
        -e cgbu-ocscp-dev-docker.dockerhub-phx.oci.oraclecorp.com/ocats \
        -f 60 \
        -g 10 \
        -h cicdocscp-scp-worker \
        -i cicdocscp-scpc-configuration \
        -j cicdocscp-scpc-notification \
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
            -k cicdocscp-scpc-subscription \
            -l appuser-secret \
            -m mysql.scp-73999881-dev.svc.cluster.local \
            -n ocats-pystub:25.1.100-beta.35 \
            -o 0.5 \
            -p 0.5G \
            -q cicdocscp-scpc-audit \
            -r 0 \
            -s 8000 \
            -t occne-prometheus-73999881-server.scp-73999881-dev-infra \
            -u 80 \
            -v 300 \
            -w /api/ \
            -x 120 \
            -z 0 \
            -A occne-opensearch-cluster-master.scp-73999881-dev-infra \
            -B 9200 \
            -C yes \
            -D opensearch \
            -E 9443 \
            -F cicdocscp-scp-nrfproxy \
            -G kubernetes.default \
            -H scp-73999881-devkafka \
            -I ocats-ocscp-ocats-ddclientstub \
            -J ocats-ocscp-ocats-scpglbratelimitstub \
            -K 8081 \
            -L v1 \
            -M scpSignallingFqdn \

        '''
        if(env.Include_Regression && "${Include_Regression}" == "YES"){
            sh '''sh /var/lib/jenkins/common_scripts/merge_jenkinsfile.sh'''
            load "/var/lib/jenkins/ocscp_tests/jenkinsData/Jenkinsfile-Merged"
        }
        else{
            load "/var/lib/jenkins/ocscp_tests/jenkinsData/Jenkinsfile-
NewFeatures"
        }

        }
}
```

The description of these parameters is as follows:

> **Note**
>
> Do not modify anything other than these parameters.

- **a:** Selected NF
- **b:** NameSpace in which SCP is deployed
- **c:** K8s Cluster Domain where SCP is deployed
- **d:** Test Stubs NameSpace - **Must be same as SCP Namespace**

Copyright © 2019, 2025, Oracle and/or its affiliates.

- **e:** Docker registry where test stub image is available
- **f:** Audit Interval provided in SCP Deployment file
- **g:** Guard Time provided SCP Deployment file
- **h:** SCP-Worker microservice name as provided during deployment
- **i:** SCPC-Configuration microservice name as provided during deployment
- **j:** SCPC-Notification microservice name as provided during deployment
- **k:** SCPC-Subscription microservice name as provided during deployment
- **l:** DB Secret name as provided during deployment
- **m:** Mysql Host name as provided during deployment
- **n:** Test Stub Image Name with tag
- **o:** Test Stub CPU requests and limit
- **p:** Test Stub Memory requests and limit
- **q:** SCPC-Audit microservice name as provided during deployment
- **r:** re-run count
- **s:** SCPSVCPORT with which SCP is deployed
- **t:** PROMSVCNAME with which Prometheus server is deployed. **Note**: Prometheus service name is followed by namepsace in which Prometheus service is existing. Prometheus service is in a different namespace related to deployed SCP namespace.
- **u:** PROMSVCPORT with which Prometheus server is deployed
- **v:** Time to wait for Stub Creation. Default value of STUBWAITTIME is 240s.**Note**: A user needs to update the STUBWAITTIME in increments or multiples of 30 seconds. The minimum allowable value for STUBWAITTIME is 30 seconds.
- **w:** API URI of Prometheus (/api/) - For examples, use **/api/**or **/prometheus/api/**
- **x:** Subscription Validity Time to set by ATS in response to subscription request from SCP
- **y:** Application Debug Logs collection on Rerun if set to true. The default value is false
- **z:** Opensearch connection timeout
- **A:** Opensearch host/service name
- **B:** Opensearch port
- **C:** Stub log collection
- **D:** Provide Kubernetes or OpenSearch for ATS for app log collection (Fetch_Log_Upon_Failure should be enabled)
- **E:** SCP HTTPS Signaling Port
- **F:** SCP NRF Proxy service name
- **G**: Kubernetes Host
- **I**: scpddclientstub name as provided during deployment
- **J**: scpglobalratelimitstub name as provided during deployment
- **K**: port of configuration pod in SCP deployment
- **L**: scp metrics version

Copyright © 2019, 2025, Oracle and/or its affiliates.

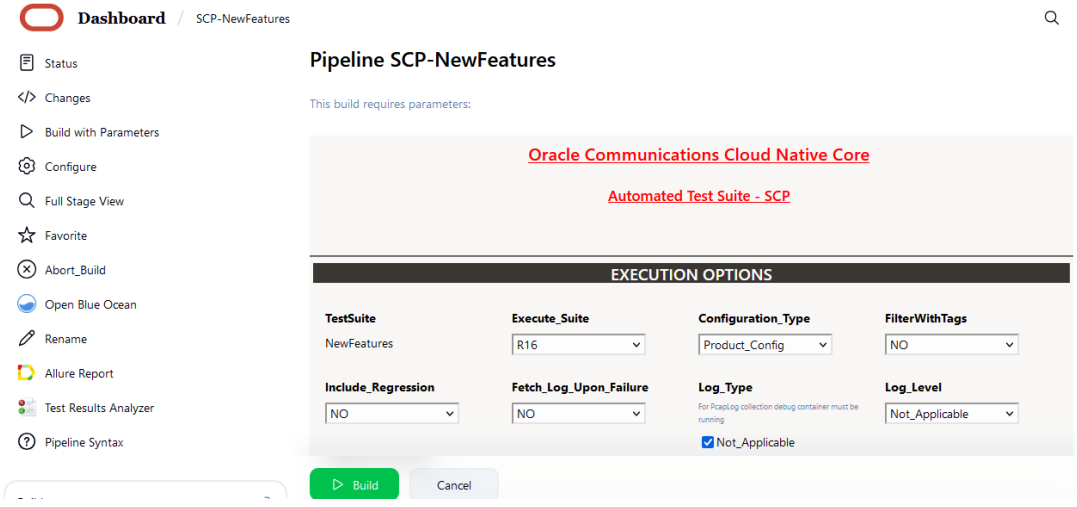
- **M**: scp worker FQDN or cluster IP: The default value is scpSignallingFqdn, which forwards requests to the worker using the worker's FQDN. Alternatively, an IPv4 or IPv6 address (enclosed in square brackets) can also be provided.

4.5.2.2 Running SCP New Feature Pipelines

To run SCP test cases, perform the following steps

1. Click **Build with Parameters** link available in the left navigation pane of the Pipeline SCP-NewFeatures screen.
2. Select the **R16** option from the **Execute_Suite** drop-down menu to run Release 16 test cases for SCP.
3. Select the required option from the **Features** drop-down menu to run the test cases:
   - **All**: To run all the test cases for SCP based on Release15 or Release16 selection.
   - **Sanity**: Enabled for SCP. Sanity cases are run to do a quick sanity check on SCP SUT features.
   - **Select:** This option allows users to separately select specific features for separate execution. Additionally, users can select the run for an entire stage or group instead of running all feature files.
4. Select one of the following configuration types from the **Configuration_Type** drop-down menu:
   - **Product_Config**: To run test cases from the features or regression directory. Test cases in this directory should not be modified.
   - **Custom_Config**: To run test cases from the features or cust_regression directory. Any test case customization required should be done in this directory.
5. Select an appropriate option and click **Build** to perform the test.

Figure 4-53 Build



6. Click **Full Stage View** in the left navigation pane to view the status of the run.
7. Hover over values of **Execute Tests**, and then click **Logs** to check test results and logs.

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> A sample consolidated test report is shown in the following image when rerun is set to 2, which is the default value. All the features passed within the first rerun, therefore, the second rerun stage has not been run in this case.

Figure 4-54 Sample consolidated report

```
Initial Run :-
Features RUN 21, PASS 20, FAIL 1

1st Rerun:-
Features RUN 1, PASS 1, FAIL 0

Final Result:-
Total-Features RUN 21, PASS 21, FAIL 0
```

For more information about consolidated test report, see Managing Final Summary Report, Build Color, and Application Log.

> **Note**
>
> - The New Feature tests in this release have been written according to SCP-ATS parameterization method (Golden Configuration). The individual features are not populated with direct values for the parameters, however, with variable names. While performing the tests, these variables are mapped according to the data provided adjacent to them in the feature specific `<feature_name>.yaml` and `global.yaml` that can be found in the `product_config` directory. This method allows the decoupling of data that can be customer specific from the features. Fore more information about parameterization (Golden Configuration), see Parameterization.

To view the detail view of the respective group, rerun logs, and the consolidated report summary:

a. Click **Open Blue Ocean Homepage** from the left navigation pane.
b. Click the build number in the run column to view the consolidated output of all the feature that are run.
c. Click **Download icon** to download the log files.

4.5.3 SCP-NewFeatures Documentation

This pipeline has an HTML report of all the feature files that you can test as part of the SCP ATS release.

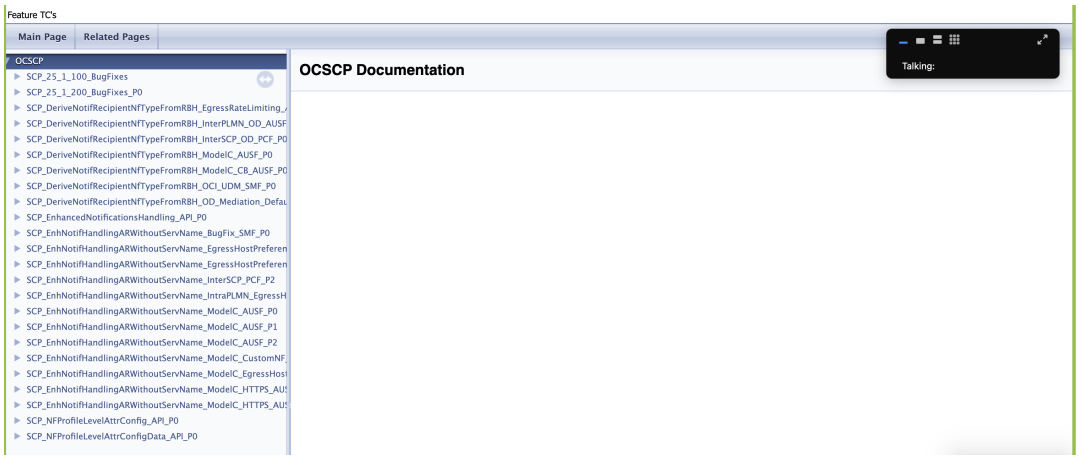
Copyright © 2019, 2025, Oracle and/or its affiliates.

1. To view SCP functionalities, go to the **SCP-NewFeatures** pipeline and click the **Documentation** option in the left navigation pane. It lists all the test cases provided as part of SCP ATS -25.1.202.

> **Note**
>
> The **Documentation** option appears only if the **SCP-NewFeatures** pipeline test cases are run at least once.

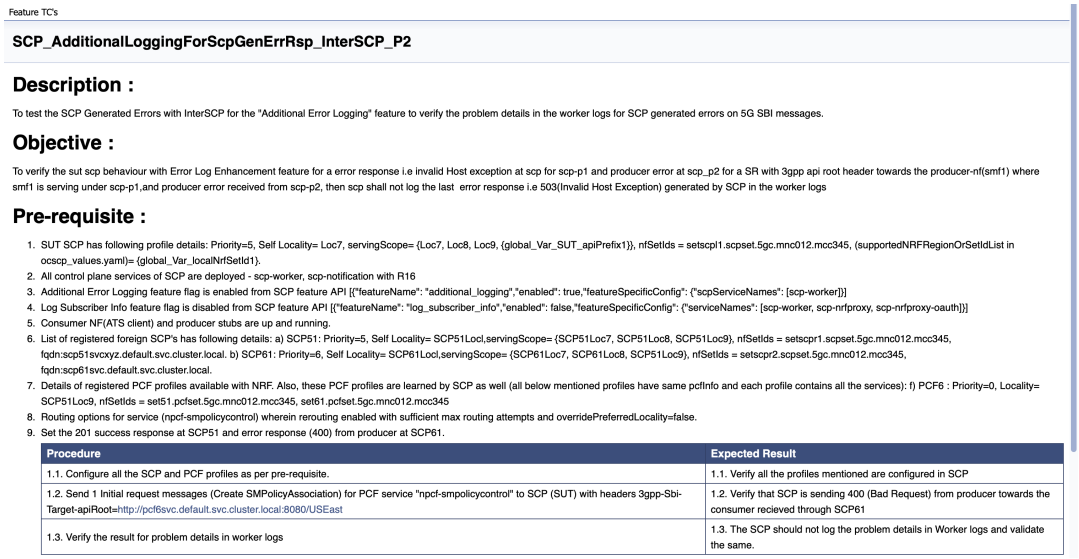
The following screen appears:

Figure 4-55 SCP-NewFeatures Documentation



2. Click any functionality to view its test cases and scenarios for each test case.

Figure 4-56 Sample Scenario



Based on the functionalities covered under Documentation, the screen displays test cases.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3. To navigate back to the Pipeline SCP-NewFeatures screen, click the **Back to SCP-NewFeatures** link available in the upper left corner of the screen.

4.5.4 SCP-Regression Pipeline

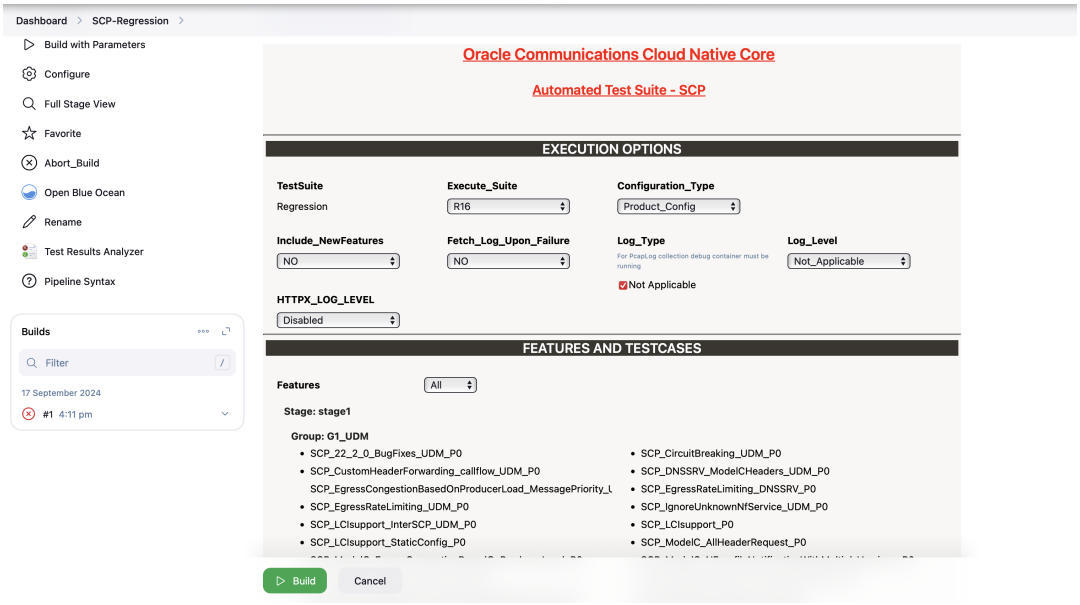
This section describes how to configure regression pipeline parameters, run SCP test cases, and view functionalities.

4.5.4.1 Configuring Regression Pipelines

This pre-configured pipeline has all the test cases from previous releases.

1. Click **SCP-Regression Pipeline**:

Figure 4-57 SCP-Regression Pipeline

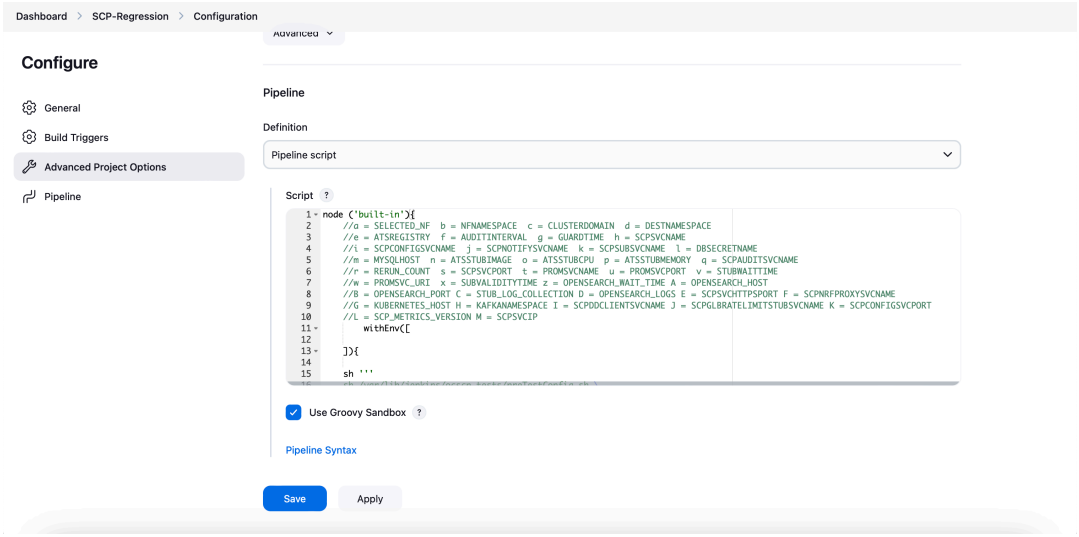


According to the installation guide, the configuration parameters can be modified in the ocats_ocscp_values.yaml file before deployment. If you are running SCP-Regression pipeline test cases for the first time without enabling the parameters in the deployment file, you have to set the input parameters before running the test cases. Subsequent test case run does not require any input unless there is a need to change any configuration.

2. In the left navigation pane, click **Configure** to provide input parameters and scroll-down to the pipeline script as shown in the following image:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-58 Regression - Pipeline Script



3. Modify the values of the Pipeline script as required, and then click **Save**.
Parameter **"-l"** should be changed as part of "Pipeline script". The content of the pipeline script is as follows:

```
node ('built-in'){
    //a = SELECTED_NF  b = NFNAMESPACE  c = CLUSTERDOMAIN  d =
DESTNAMESPACE
    //e = ATSREGISTRY  f = AUDITINTERVAL  g = GUARDTIME  h = SCPSVCNAME
    //i = SCPCONFIGSVCNAME  j = SCPNOTIFYSVCNAME  k = SCPSUBSVCNAME  l =
DBSECRETNAME
    //m = MYSQLHOST  n = ATSSTUBIMAGE  o = ATSSTUBCPU  p = ATSSTUBMEMORY
q = SCPAUDITSVCNAME
    //r = RERUN_COUNT  s = SCPSVCPORT  t = PROMSVCNAME  u = PROMSVCPORT  v
= STUBWAITTIME
    //w = PROMSVC_URI  x = SUBVALIDITYTIME z = OPENSEARCH_WAIT_TIME A =
OPENSEARCH_HOST
    //B = OPENSEARCH_PORT C = STUB_LOG_COLLECTION D = OPENSEARCH_LOGS E =
SCPSVCHTTPSPORT F = SCPNRFPROXYSVCNAME
    //G = KUBERNETES_HOST H = KAFKANAMESPACE I = SCPDDCLIENTSVCNAME J =
SCPGLBRATELIMITSTUBSVCNAME K = SCPCONFIGSVCPORT
    //L = SCP_METRICS_VERSION M = SCPSVCIP
        withEnv([

    ]){

    sh '''
    sh /var/lib/jenkins/ocscp_tests/preTestConfig.sh \
        -a SCP \
        -b scpsvc \
        -c cluster.local \
        -d scpsvc \
        -e cgbu-ocscp-dev-docker.dockerhub-phx.oci.oraclecorp.com/ocats \
        -f 60 \
        -g 10 \
        -h ocscp-scp-worker \
        -i ocscp-scpc-configuration \
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
            -j ocscp-scpc-notification \
            -k ocscp-scpc-subscription \
            -l appuser-secret \
            -m mysql.default.svc.cluster.local \
            -n ocats-pystub:tag \
            -o 0.5 \
            -p 0.5G \
            -q ocscp-scpc-audit \
            -r 0 \
            -s 8000 \
            -t occne-prometheus-server.occne-infra \
            -u 80 \
            -v 300 \
            -w /api/ \
            -x 120 \
            -z 0 \
            -A occne-opensearch-cluster-master.occne-infra \
            -B 9200 \
            -C no \
            -D opensearch \
            -E 9443 \
            -F ocscp-scp-nrfproxy \
            -G kubernetes.default \
            -H scpsvc \
            -I ocats-ocscp-ocats-ddclientstub \
            -J ocats-ocscp-ocats-scpglbratelimitstub \
            -K 8081 \
            -L v1 \
            -M scpSignallingFqdn \

        '''
        if(env.Include_NewFeatures && "${Include_NewFeatures}" == "YES"){
            sh '''sh /var/lib/jenkins/common_scripts/merge_jenkinsfile.sh'''
            load "/var/lib/jenkins/ocscp_tests/jenkinsData/Jenkinsfile-Merged"
        }
        else{
            load "/var/lib/jenkins/ocscp_tests/jenkinsData/Jenkinsfile-
Regression"
        }

        }
}
```

The description of these parameters is as follows:

- **a:** Selected NF
- **b:** NameSpace in which SCP is deployed
- **c:** K8s Cluster Domain where SCP is deployed
- **d:** Test Stubs NameSpace - **Must be same as SCP Namespace**
- **e:** Docker registry where test stub image is available
- **f:** Audit Interval provided in SCP Deployment file
- **g:** Guard Time provided SCP Deployment file
- **h:** SCP-Worker microservice name as provided during deployment

Copyright © 2019, 2025, Oracle and/or its affiliates.

- **i:** SCPC-Configuration microservice name as provided during deployment
- **j:** SCPC-Notification microservice name as provided during deployment
- **k:** SCPC-Subscription microservice name as provided during deployment
- **l:** DB Secret name as provided during deployment
- **m:** Mysql Host name as provided during deployment
- **n:** Test Stub Image Name with tag
- **o:** Test Stub CPU requests and limit
- **p:** Test Stub Memory requests and limit
- **q:** SCPC-Audit microservice name as provided during deployment
- **r:** re-run count
- **s:** SCPSVCPORT with which SCP is deployed
- **t:** PROMSVCNAME with which prometheus server is deployed. **Note**: Prometheus service name is followed by namepsace in which Prometheus service is existing. Prometheus service is in a different namespace related to deployed SCP namespace.
- **u:** PROMSVCPORT with which prometheus server is deployed
- **v:** Time to wait for Stub Creation. Default value of STUBWAITTIME is 240s.**Note**: A user needs to update the STUBWAITTIME in increments or multiples of 30 seconds. The minimum allowable value for STUBWAITTIME is 30 seconds.
- **w:** API URI of Prometheus (/api/) - Use **"/api/"** or **"/prometheus/api/"**
- **x:** Subscription Validity Time to set by ATS in response to subscription request from SCP
- **y:** Application Debug Logs collection on Rerun if set to true. The default value is false
- **z:** Opensearch connection timeout
- **A:** Opensearch host/service name
- **B:** Opensearch port
- **C:** Stub log collection
- **D:** Opensearch API Path Prefix
- **E:** SCP HTTPS Signaling Port
- **F:** SCP NRF Proxy service name
- **G**: Kubernetes Host
- **I**: scpddclientstub name as provided during deployment
- **J**: scpglobalratelimitstub name as provided during deployment
- **K**: port of configuration pod in scp deployment
- **L**: scp metrics version
- **M**: scp worker FQDN or cluster IP. The default value is **scpSignallingFqdn**, which forwards requests to the worker using the worker FQDN. Alternatively, an IPv4 or IPv6 address (enclosed in square brackets) can also be provided.

4.5.4.2 Running Regression Pipelines

This section describes how to run test cases for SCP Regression pipeline.

Copyright © 2019, 2025, Oracle and/or its affiliates.

1. To run the pipeline, click **Build with Parameters** in the left navigation pane.
2. Select the **R16** option from the **Execute_Suite** drop-down menu to run Release 16 test cases for SCP.
3. Select the required regression option from the **Features** drop-down menu to run the test cases:
   - **All**: To run all the test cases for SCP based on Release15 or Release16 selection. This runs all the cases except `SCP_Audit_nnrf_disc`. If SCP is deployed with `nnrf-disc` for audit or registration with NRF is disabled, then the **All** option should not be used and instead Single or MultipleFeatures option can be used to select appropriate cases.
   - **Sanity**: Enabled for SCP. Sanity cases are run to do a quick sanity check on SCP SUT features. Sanity cases are included in the regression suite and are tagged with @sanity.
   - **Select:** This option allows users to separately select specific features for separate test case run. Additionally, users can select the run for an entire stage or group instead of running all feature files.
4. Select one of the following configuration types from the **Configuration_Type** drop-down menu:
   - **Product_Config**: To run test cases from the features or regression directory. Test cases in this directory should not be modified.
   - **Custom_Config**: To run test cases from the features or cust_regression directory. Any test case customization required should be done in this directory.
5. Select an appropriate option and click **Build** to run the test case.
6. Click **Full Stage View** in the left navigation pane to view the status of the run.
7. Hover over the **Execute-Tests** stage of the pipeline, and then click **Logs** to check test case results and logs.
   To view the detail view of the respective group, rerun logs, and the consolidated report summary:
   a. Click **Open Blue Ocean Homepage** from the left navigation pane.
   b. Click the build number in the run column to view the consolidated output of all the feature that are run.
   c. Click **Download icon** to download the log files.

4.5.5 SCP-Regression Documentation

This pipeline has an HTML report of all the feature files that you can test as part of the SCP ATS release.

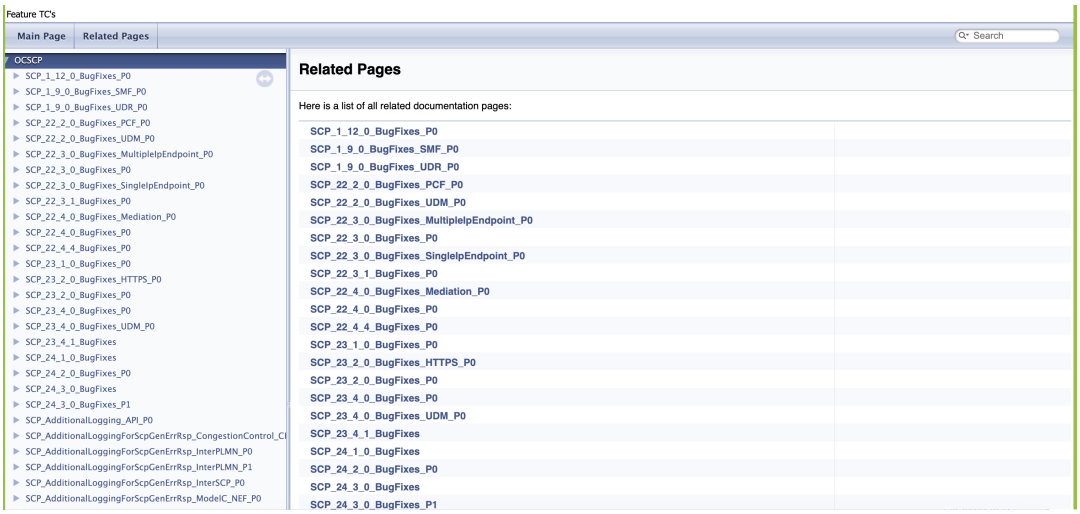
1. To view SCP functionalities, go to the **SCP-Regression** pipeline and click the **Documentation** option in the left navigation pane. It lists all the test cases provided as part of SCP ATS -25.1.202.

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> The **Documentation** option appears only if the **SCP-Regression** pipeline test cases are run at least once.
> In SCP 24.1.0, the feature files are renamed as per the naming convention. For more information, see Renaming of Regression Feature Files

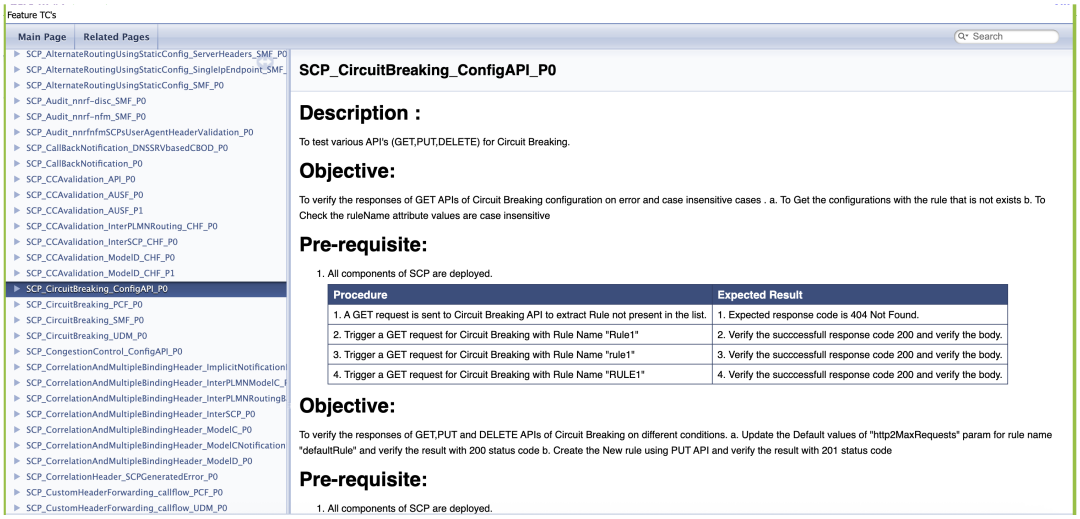
The following screen appears:

Figure 4-59 SCP-Regression Documentation



2. Click any functionality to view its test cases and scenarios for each test case.

Figure 4-60 Sample Scenario



Based on the functionalities covered under Documentation, the screen displays test cases.

Copyright © 2019, 2025, Oracle and/or its affiliates.

3. To navigate back to the Pipeline SCP-Regression screen, click the **Back to SCP-Regression** link available in the upper left corner of the screen.

4.5.5.1 Renaming of Regression Feature Files

The following tables detail the feature file that has been renamed in alignment with the naming convention:

Table 4-3 Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
Route_group_static_config_api_validation	SCP_AlternateRoutingUsingStaticConfig_API_P0
Alternate_Routing_Using_Static_Config_EgressRL_SMF	SCP_AlternateRoutingUsingStaticConfig_EgressRL_SMF_P0
Alternate_Routing_Using_Static_Configuration_Https_SMF	SCP_AlternateRoutingUsingStaticConfig_HTTPS_SMF_P0
Alternate_Routing_Using_Static_Configuration_Notification_CircuitBreaking.	SCP_AlternateRoutingUsingStaticConfig_NotificationMessageCircuitBreaking_P0
Alternate_Routing_Using_Static_Configuration_SMF	SCP_AlternateRoutingUsingStaticConfig_SMF_P0
Alternate_Routing_Using_Static_Config_Overload_Control	SCP_AlternateRoutingUsingStaticConfig_OverloadControl_P0
SCP_Audit_nnrf_disc_SMF	SCP_Audit_nnrf-disc_SMF_P0
SCP_Audit_nnrf_nfm_SMF	SCP_Audit_nnrf-nfm_SMF_P0
SCP_Audit_nnrf_nfm_SMF_user_agent	SCP_Audit_nnrfnfmSCPsUserAgentHeaderValidation_P0
Dns_Srv_Call_Back_Notification_CD_OD_Cases	SCP_CallBackNotification_DNSSRVbasedCBOD_P0
Callback_And_Notification	SCP_CallBackNotification_P0
SCP_CCA_Header_Validation_API_P0	SCP_CCAvalidation_API_P0
CCA_validation_AUSF_P0	SCP_CCAvalidation_AUSF_P0
CCA_validation_AUSF_P1	SCP_CCAvalidation_AUSF_P1
CCA_validation_InterPLMN_CHF_P0	SCP_CCAvalidation_InterPLMNRouting_CHF_P0
CCA_validation_InterSCP_CHF_P0	SCP_CCAvalidation_InterSCP_CHF_P0
CCA_validation_ModelD_CHF_P0	SCP_CCAvalidation_ModelD_CHF_P0
CCA_validation_ModelD_CHF_P1	SCP_CCAvalidation_ModelD_CHF_P1
SCP_Circuit_Breaking_Config_API	SCP_CircuitBreaking_ConfigAPI_P0
Circuit_Breaking_PCF	SCP_CircuitBreaking_PCF_P0
Circuit_Breaking_SMF	SCP_CircuitBreaking_SMF_P0
SCP_Congestion_Control_Config_API	SCP_CongestionControl_ConfigAPI_P0
S_Correlation_Header_interplmn_model_C	SCP_CorrelationHeader_InterPLMNModelC_P0
Correlation_Headersepp_model_D_Inter_PLMN_and_Mediation	SCP_CorrelationHeader_InterPLMNRoutingBasedOnModelDandMediation_P0
Interscp_Correlation_Header_model_C	SCP_CorrelationHeader_InterSCP_P0
Correlation_Header_model_C_cases	SCP_CorrelationHeader_ModelC_P0
Notification_Correlation_Header_model_C	SCP_CorrelationHeader_ModelCNotificationRequest_P0
Correlation_Header_SCP_generated_error	SCP_CorrelationHeader_SCPGeneratedError_P0
DefaultNotificationCallbackInitialRouteAndDiscovery_p0	SCP_DefaultNotificationCallback_InitialRouteAndDiscovery_P0
Dns_Srv_Call_Back_Notification_Https	SCP_DefaultNotificationCallbackUri_DNSSRV_HTTPS_P0
Dns_Srv_Call_Back_Notification	SCP_DefaultNotificationCallbackUri_DNSSRV_P0
DefaultNotificationCallbackUriModelC_Https_p1	SCP_DefaultNotificationCallbackUri_ModelC_HTTPS_P1
DefaultNotificationCallbackUriModelC_p0	SCP_DefaultNotificationCallbackUri_ModelC_P0

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-3 (Cont.) Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
DefaultNotificationCallbackUriModelC_p1	SCP_DefaultNotificationCallbackUri_ModelC_P1
DefaultNotificationCallbackUriModelC_CB_p0	SCP_DefaultNotificationCallbackUri_ModelCCB_P0
DefaultNotificationCallbackUriModelC_SEPP_p1	SCP_DefaultNotificationCallbackUri_ModelCInterPLMNRouting_P1
DefaultNotificationCallbackUriModelC_SEPP_p1	SCP_DefaultNotificationCallbackUri_ModelCInterPLMNRouting_P1
Alternate_resolution_api_validation	SCP_DNSSRV_AlternateResolution_API_P0
Dns_Srv_Based_Routing_For_ModelC_Headers_PCF_cases	SCP_DNSSRV_ModelCHeaders_PCF_P0
Dns_Srv_Based_Routing_For_ModelC_Headers_UDM_cases	SCP_DNSSRV_ModelCHeaders_UDM_P0
Dns_Srv_Producer_Overload_Control	SCP_DNSSRV_ProducerBasedOverloadControl_P0
Alert_egress_congestion_based_on_producer_load_and_message_priority_CHF	SCP_EgressCongestionBasedOnProducerLoad_AlertMessagePriority_CHF_P0
Egress_congestion_based_on_producer_load_and_message_priority_PCF	SCP_EgressCongestionBasedOnProducerLoad_MessagePriority_PCF_P0
Egress_congestion_based_on_producer_load_and_message_priority_UDM	SCP_EgressCongestionBasedOnProducerLoad_MessagePriority_UDM_P0
SCP_Egress_Host_Preference_alternate_routing_SMF_P0_P1	SCP_EgressHostPreference_AlternateRouting_SMF_P1
SCP_Egress_Host_Preference_alternate_routing_withoutHeader_AUSF_P1	SCP_EgressHostPreference_AlternateRoutingWithoutHeader_AUSF_P1
SCP_Egress_Host_Preference_API_P0	SCP_EgressHostPreference_API_P0
SCP_Egress_Host_Preference_CHF_P0	SCP_EgressHostPreference_CHF_P0
SCP_Egress_Host_Preference_CHF_P1	SCP_EgressHostPreference_CHF_P1
SCP_Egress_Host_Preference_InterPLMN_Routing_P0	SCP_EgressHostPreference_InterPLMNRouting_P0
SCP_Egress_Host_Preference_InterPLMN_Routing_P1	SCP_EgressHostPreference_InterPLMNRouting_P1
SCP_Egress_Host_Preference_InterPLMN_Routing_Non_ASM_P0	SCP_EgressHostPreference_InterPLMNRoutingNonASM_P0
SCP_Egress_Host_Preference_InterSCP_Routing_CHF_P0	SCP_EgressHostPreference_InterSCPRouting_CHF_P0
SCP_Egress_Host_Preference_InterSCP_Routing_CHF_P1	SCP_EgressHostPreference_InterSCPRouting_CHF_P1
SCP_Egress_Host_Preference_InterSCP_Routing_CHF_Non_ASM_P0	SCP_EgressHostPreference_InterSCPRoutingNonASM_CHF_P0
SCP_Egress_Host_Preference_ModelD_routing_CHF_NRF_P0	SCP_EgressHostPreference_ModelDRoutingCHF_NRF_P0
SCP_Egress_Host_Preference_ModelD_routing_CHF_NRF_P1	SCP_EgressHostPreference_ModelDRoutingCHF_NRF_P1
SCP_Egress_Host_Preference_ModelD_routing_without_apiRootHdr_CHF_NRF_P1	SCP_EgressHostPreference_ModelDRoutingWithoutAPIRootHdrCHFandNRF_P1
SCP_Egress_Host_Preference_Notification_PCF	SCP_EgressHostPreference_NotificationRequest_PCF_P0
SCP_Egress_Host_Preference_SCP_Generated_Audit_SMF_NRF_P1	SCP_EgressHostPreference_SCPGeneratedAuditSMF_NRF_P1
SCP_Egress_Host_Preference_SMF_P0_P1	SCP_EgressHostPreference_SMF_P1
SCP_Egress_Host_Prefrence_SCP_Generated_Subscription_SCP_NRF_P1	SCP_EgressHostPrefrence_SCPgeneratedSubscription_NRF_P1
Alert_SCPEgressTrafficRateExceededConfiguredLimit_AUSF	SCP_EgressRateLimiting_AlertTrafficRateExceededConfiguredLimit_AUSF_P0

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-3 (Cont.) Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
Alert_SCPEgressTrafficRoutedWithoutRateLimitTreatment_AUSF	SCP_EgressRateLimiting_AlertTrafficRoutedWithoutRateLimitTreatment_AUSF_P0
Egress_Rate_Limiting_Config_API	SCP_EgressRateLimiting_Config_API_P0
EgressRateLimiting_INTERSCP	SCP_EgressRateLimiting_InterSCP_P0
EgressRateLimiting	SCP_EgressRateLimiting_P0
EgressRateLimiting_UDM_Cases	SCP_EgressRateLimiting_UDM_P0
RateLimitingRelease16_AUSF	SCP_EgressRateLimitingRelease16_AUSF_P0
SCP_EgressRateLimiting_DNSSRV.feature	SCP_EgressRateLimiting_DNSSRV_P0.feature
Enhanced_Suspended_State_Routing_API_Validation	SCP_EnhancedSuspendedStateRouting_API_P0
Enhanced_Suspended_State_Routing_Alternate_Routing_Using_Static_Config_DNS_SRV	SCP_EnhancedSuspendedStateRouting_ARusingStaticConfigandDNSSRV_P0
Enhanced_Suspended_State_Routing_Inter_PLMN	SCP_EnhancedSuspendedStateRouting_InterPLMNRouting_P0
Enhanced_Suspended_State_Routing_Mode2_API_Validation_P0	SCP_EnhancedSuspendedStateRouting_Mode2_API_P0
Enhanced_Suspended_State_Routing_Mode2_Alternate_Routing_Using_Static_Config_DNS_SRV_P0	SCP_EnhancedSuspendedStateRouting_Mode2AlternateRoutingStaticConfigDNSSRV_P0
Enhanced_Suspended_State_Routing_Mode2_ModelC_SMF_P1	SCP_EnhancedSuspendedStateRouting_Mode2ModelC_SMF_P1
Enhanced_Suspended_State_Routing_Mode2_ModelC_Inter_PLMN_P0	SCP_EnhancedSuspendedStateRouting_Mode2ModelCInterPLMN_P0
Enhanced_Suspended_State_Routing_Mode2_ModelC_InterSCP_P0	SCP_EnhancedSuspendedStateRouting_Mode2ModelCInterSCP_P0
Enhanced_Suspended_State_Routing_Mode2_ModelC_Overload_Congestion_SMF_P1	SCP_EnhancedSuspendedStateRouting_Mode2ModelCOverloadCongestion_SMF_P1
Enhanced_Suspended_State_Routing_Mode2_ModelD_P0	SCP_EnhancedSuspendedStateRouting_Mode2ModelD_P0
Enhanced_Suspended_State_Routing_Mode2_ModelD_Inter_PLMN_P0	SCP_EnhancedSuspendedStateRouting_Mode2ModelDInterPLMNRoutin_P0
Enhanced_Suspended_State_Routing_Mode2_ModelD_InterSCP_P0	SCP_EnhancedSuspendedStateRouting_Mode2ModelDInterSCP_P0
Enhanced_Suspended_State_Routing_ModelC_SMF	SCP_EnhancedSuspendedStateRouting_ModelC_SMF_P0
Enhanced_Suspended_State_Routing_Parallel_ModelC_SMF_UDM_P0	SCP_EnhancedSuspendedStateRouting_ModelC_SMFandUDM_P0
Enhanced_Suspended_State_Routing_ModelC_UDM	SCP_EnhancedSuspendedStateRouting_ModelC_UDM_P0
Enhanced_Suspended_State_Routing_Model_C_Inter_SCP	SCP_EnhancedSuspendedStateRouting_ModelCInterSCP_P0
Enhanced_Suspended_State_Routing_ModelD	SCP_EnhancedSuspendedStateRouting_ModelD_P0
Enhanced_Suspended_State_Routing_ModelD_InterSCP	SCP_EnhancedSuspendedStateRouting_ModelDInterSCP_P0
Enhanced_NF_Status_Processing_SUSPENDED	SCP_EnhancedSuspendedStateRouting_SuspendedMode2_P0
Enhanced_NF_Status_Processing_SUSPENDED_SMF	SCP_EnhancedSuspendedStateRouting_SuspendedMode2_SMF_P0
Enhanced_NF_Status_Processing_UNDISCOVERABLE	SCP_EnhancedSuspendedStateRouting_UNDISCOVERABLE_P0
SCP_Error_Profile_Config_API_P0	SCP_ErrorProfile_ConfigAPI_P0

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-3 (Cont.) Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
SCPFailureHandling_default_behavior_serverheader_AUSF	SCP_FailureHandling_DefaultBehaviorServerHeader_AUSF_P0
SCPFailureHandling_default_behavior_serverheader	SCP_FailureHandling_DefaultBehaviorServerHeader_P0
SCPFailureHandling_DNSSRV_based_EnhancedServerHeaderV2_NEF_p0	SCP_FailureHandling_DNSSRVbasedEnhancedServerHeaderV2_NEF_P0
SCPFailureHandling_EnhancedServerHeader	SCP_FailureHandling_EnhancedServerHeader_P0
SCPFailureHandling_EnhancedServerHeaderV2_API	SCP_FailureHandling_EnhancedServerHeaderV2_API_P0
SCPFailureHandling_EnhancedServerHeaderV2_p0	SCP_FailureHandling_EnhancedServerHeaderV2_P0
SCPFailureHandling_EnhancedServerHeaderV2_p1	SCP_FailureHandling_EnhancedServerHeaderV2_P1
FailureHandling_InterSCP_EnhancedServerHeader	SCP_FailureHandling_InterSCPEnhancedServerHeader.P0
FormUrlEncoded	SCP_FormUrlEncoded_P0
Global_Egress_Rate_Limiting_Config_API	SCP_GlobalEgressRateLimiting_API_P0
GlobalEgressRateLimiting_P0	SCP_GlobalEgressRateLimiting_P0
SCP_HealthCheck_Support_API_P0	SCP_HealthCheckSupport_API_P0
SCP_HealthCheck_Support_HTTPS_P0	SCP_HealthCheckSupport_HTTPS_P0
SCP_HealthCheckInterSCPModelC_HTTPS_P0	SCP_HealthCheckSupport_InterSCPModelC_HTTPS_P0
SCP_HealthCheckInterSCPModelC_P0	SCP_HealthCheckSupport_InterSCPModelC_P0
SCP_HealthCheckInterSCPModelD_P0	SCP_HealthCheckSupport_InterSCPModelD_P0
Alert_SCPIngressTrafficRateExceededConfiguredLimit	SCP_IngressRateLimiting_AlertTrafficRateExceededConfiguredLimit_P
Alert_SCPIngressTrafficRoutedWithoutRateLimitTreatment	SCP_IngressRateLimiting_AlertTrafficRoutedWithoutRateLimitTreatment_P0
Ingress_Rate_Limiting_Config_API	SCP_IngressRateLimiting_API_P0
Ingress_Rate_Limiting_Config_API_SC1	SCP_IngressRateLimiting_ConfigAPI_P0
Ingress_Rate_Limiting_Config_API_enhancement_for_nfInstanceId	SCP_IngressRateLimiting_ConfigAPIenhancementNFInstanceId_P0
Ingress_Rate_Limiting_enhancement_for_nfInstanceId_to_support_UserAgentHeader	SCP_IngressRateLimiting_NFInstanceIdEnhancementUserAgentHeader_P0
SEPP_Audit_nnrf_nfm	SCP_InterPLMNRouting_Audit_nnrf-nfm_P0
SCP-SEPP_ModelD_Routing	SCP_InterPLMNRouting_ModelDBased_P0
SEPP_Routing_NRF	SCP_InterPLMNRouting_NRF_P0
OutlierDetectionSepp_P0	SCP_InterPLMNRouting_OutlierDetection_P0
SCP-SEPP_Routing	SCP_InterPLMNRouting_P0
SEPP_plmnInfo_DELETE_API	SCP_InterPLMNRouting_PLMNInfoDeleteAPI_P0
SEPP_plmnInfo_GET_API	SCP_InterPLMNRouting_PLMNInfoGetAPI_P0
SEPP_plmnInfo_PATCH_API	SCP_InterPLMNRouting_PLMNInfoPatchAPI_P0
SEPP_plmnInfo_PUT_API	SCP_InterPLMNRouting_PLMNInfoPutAPI_P0
SCP-SEPP_Routing_UDM	SCP_InterPLMNRouting_UDM_P0
Rel16_Inter_SCP_Audit_nnrf_nfm_p0	SCP_InterSCP_Audit_nnrf-nfm_P0
InterSCP_MateInfoListRouting	SCP_InterSCP_MateInfoListRouting_P0
Inter_SCP_Routing_NplusK_p0	SCP_InterSCP_NplusKRouting_P0
Inter_SCP_Subscription_SMF_NRF	SCP_InterSCP_SubscriptionWithForeignNRFforNfTypeSMF_P0
Inter_SCP_Routing	SCP_InterSCPRouting_P0
Jaeger_Api_Validation	SCP_Jaeger_API_P0

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-3 (Cont.) Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
SCP_LCI_Support_Config_API	SCP_LCIsupport_API_P0
Scp_Lci_Support_InterPLMN	SCP_LCIsupport_InterPLMNRouting_P0
SCP_Lci_Support_InterSCP	SCP_LCIsupport_InterSCP_P0
SCP_Lci_Support_InterSCP_PCF_SCP	SCP_LCIsupport_InterSCP_PCF_P0
SCP_Lci_Support_InterSCP_UDM	SCP_LCIsupport_InterSCP_UDM_P0
SCP_Lci_Support_Miscellaneous	SCP_LCIsupport_Miscellaneous_P0
Scp_Lci_Support_ModelD	SCP_LCIsupport_ModelD_P0
Scp_Lci_Support_ModelD_notification_cases	SCP_LCIsupport_ModelDNotificationRequest_P0
SCP_Lci_Support	SCP_LCIsupport_P0
Local_N_Plus_K	SCP_LocalNplusK_P0
Location_hdr_update_for_host_mismatch_API_validation	Location_hdr_update_for_host_mismatch_API_validation
App_Routing_Options_Config_API	SCP_Mediation_AppRoutingOptionsConfig_API_P0
SCP_Mediation_Config_API	SCP_Mediation_ConfigAPI_P0
SCP_Mediation_Worker_side_cases_PCF_Cases	SCP_Mediation_FunctionalCases_PCF_P0
SCP_Mediation_Cases_AUSF	SCP_Mediation_FunctionalCasesPart1_AUSF_P0
SCP_Mediation_Worker_side_cases_AUSF	SCP_Mediation_FunctionalCasesPart2_AUSF_P0
SCP_Mediation_Worker_side_cases_UDM_SMF	SCP_Mediation_FunctionalTestCases_UDMandSMF_P0
SCP_Mediation_Worker_side_cases_UDM_AUSF_SMF_Notification_cases	SCP_Mediation_FunctionalTestCasesForUDMnAUSFnSMFnNotificationRequest_P0
SCP_Mediation_Cases_PCF	SCP_Mediation_PCF_P0
Alert_SCPRoutingFailedForProducer_CHF	SCP_ModelC_AlertSCPRoutingFailedForProducer_CHF_P0
ModelC_allheaderRequest	SCP_ModelC_AllHeaderRequest_P0
PCF_AMPolicy_forwardRoute_Target_apiRoot	SCP_ModelC_AMPolicyTargetAPIRoot_PCF_P0
ModelC_CatchAll_Routing_Configuration_AUSF	SCP_ModelC_CatchAllRoutingConfiguration_AUSF_P0
ModelC_Direct_Routing_Configuration_AUSF	SCP_ModelC_DirectRoutingConfiguration_AUSF_P0
ModelC_ProducerOverloadControl	SCP_ModelC_EgressCongestionBasedOnProducerLoad_P0
Egress_Congestion_based_on_load_PCF_SMPolicy_forwardRoute_Target_apiRoot	SCP_ModelC_EgressCongestionBasedOnProducerLoad_PCF_P0
Https_Routing_For_ModelC_Headers	SCP_ModelC_HTTPS_P0
ModelC_IngressRateLimiting	SCP_ModelC_IngressRateLimiting_P0
Model-C_based_Inter_SCP_Routing	SCP_ModelC_InterSCProuting_P0
ModelC_Inter_SCP_Routing_Options_Notification_Cases	SCP_ModelC_InterSCPRoutingOptionsNotificationCases_P0
InterSCP_VersionSupport	SCP_ModelC_InterSCPVersionSupport_P0
ModelC_NFInstance_AUSF_Cases	SCP_ModelC_NFInstance_AUSF_P0
ModelC_NFInstance_NEF_Cases	SCP_ModelC_NFInstance_NEF_P0
NF_Notification	SCP_ModelC_NFprofileNotificationWithMultipleVersions_P
ModelC_NFSet_NEF_Cases	SCP_ModelC_NFSet_NEF_P0
ModelC_NFSet_PCF_Cases	SCP_ModelC_NFSet_PCF_P0
ModelC_NFSet_SMF_Cases	SCP_ModelC_NFSet_SMF_P0
ModelC_ReverseLookup_CHF_Cases	SCP_ModelC_ReverseLookup_CHF_P0
ModelC_ReverseLookup_UDM_Cases	SCP_ModelC_ReverseLookup_UDM_P0
ModelC_ReverseLookup_UDR_Cases	SCP_ModelC_ReverseLookup_UDR_P0
ModelC_Inter_SCP_Routing_Options	SCP_ModelC_RoutingOptionsInterSCP_P0

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-3 (Cont.) Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
AUSF_AUTH_forwardRoute_Target_apiRoot	SCP_ModelC_TargetAPIRoot_AUSF_P0
CHF_SLC_forwardRoute_Target_apiRoot	SCP_ModelC_TargetAPIRoot_CHF_P0
SMF_forwardRoute_Target_apiRoot	SCP_ModelC_TargetAPIRoot_SMF_P0
UDM_SDM_forwardRoute_Target_apiRoot	SCP_ModelC_TargetAPIRootSDM_UDM_P0
UDM_UECM_forwardRoute_Target_apiRoot	SCP_ModelC_TargetAPIRootUECM_UDM_P0
Version_support	SCP_ModelC_VersionSupport_P0
ModelD_Config_API	SCP_ModelD_API_P0
ModelD_enforceReqSpecificSvcDiscovery	SCP_ModelD_EnforceReqSpecificSvcDiscovery_P0
ModelD_based_Routing_implicit_notification_p0	SCP_ModelD_ImplicitNotificationRouting_P0
ModelD_based_Routing	SCP_ModelD_Routing_P0
Notification_Patch_Profile_Support_SMF	SCP_NotificationRequest_PatchProfileSupport_SMF_P0
SCP_Oauth2_Access_Token_Granularity_Config_API_P0	SCP_Oauth2Support_AccessTokenGranularity_API_P0
SCP_Oauth2_Support_Alternative_Config_ModelD_SMF_NRF_P0	SCP_Oauth2Support_AlternativeConfigModelD_SMFandNRF_P0
SCP_Oauth2_Granularity_config_ModelD_SMF_NRF_P0	SCP_Oauth2Support_GranularityConfigModelD_SMFandNRF_P0
SCP_Oauth2_Local-PLMN-required-config_API_P0	SCP_Oauth2Support_LocalPLMNRequiredConfig_API_P0
SCP_OAuth2_support_ModelC_CHF_P0	SCP_Oauth2Support_ModelC_CHF_P0
Scp_Oauth2_Support_ModelC_SMF_NRF_P0	SCP_Oauth2Support_ModelC_SMFandNRF_P0
SCP_OAuth2_Support_Config_API_P0	SCP_OAuth2Support_ConfigAPI_P0
SCP_Oauth2_Support_ModelC_Error_Handling_CHF_NRF_P1	SCP_Oauth2Support_ModelCErrorHandling_CHFandNRF_P1
SCP_Oauth2_Support_ModelD_SMF_NRF_P0_P1	SCP_Oauth2Support_ModelD_SMFandNRF_P1
SCP_Oauth2_Support_ModelD_interSCP_P0	SCP_Oauth2Support_ModelDinterSCP_P0
SCP_Oauth2_NRF_Configuration_Config_API_P0	SCP_Oauth2Support_NRFConfiguration_API_P0
SCP_Observability_And_Inter_Microservice_Resilience_P0	SCP_ObservabilityAndInterMicroserviceResilience_P0
SCP_OutlierDetection_API	SCP_OutlierDetection_API_P0
OutlierDetectionInterSCP_P0	SCP_OutlierDetection_InterSCP_P0
OutlierDetectionProducer_AUSF_P0	SCP_OutlierDetection_ProducerAUSF_P0
PseudoHeadersModification	SCP_PseudoHeadersModification_P0
PseudoHeadersModificationHttps	SCP_PseudoHeadersModification_HTTPS_P0
NRF_Registration	SCP_RegistrationWithNRF_P0
SCP_Registration_With_PLMNList	SCP_RegistrationWithNRF_PLMNList_P0
Scp_profile_mateScpInfo_update	SCP_RegistrationWithNRF_ProfileMateSCPInfoUpdate_P0
Rel16_Canary_support_PCF_Cases	SCP_Rel16CanarySupport_PCF_P0
Rel16_Canary_support_SMF_Cases	SCP_Rel16CanarySupport_SMF_P0
Rel16_Canary_support_UDM_Cases	SCP_Rel16CanarySupport_UDM_P0
Re_routing_based_on_configurable_http_status_codes_api_validation_p0	SCP_ReroutingBasedonConfigurableHttpStatusCodes_API_P0
Re_routing_based_on_configurable_http_status_codes_p0	SCP_ReRoutingBasedOnConfigurableHttpStatusCodes_P0
SCP_Ro_Enhancments_For_CB_CC_SMF_P0	SCP_RoutingOptionEnhancements_CBAndCongestionControl_SMF_P0
SCP_Ro_Enhancments_For_CB_CC_InterSCP_InterPLMN_SMF_PCF_P0	SCP_RoutingOptionEnhancements_CBAndCongestionControlForInterPLMNandSCPRouting_SMFandPCF_P0

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-3 (Cont.) Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
SCP_Ro_Enahancment_For_CC_CB_Notification_PCF_P0	SCP_RoutingOptionEnhancements_CBAndCongestionControlForNotificationReq_PCF_P0
SCP_Notification_Routing_Option_Config_API	SCP_RoutingOptionForNotificationReq_API_P0
SCP_Routing_Option_Set_API	SCP_RoutingOptionForNotificationReq_RoutingConfigSet_API_P0
Routing_Rules_API_R16	SCP_RoutingRules_R16_API_P0
Sbi_Message_Priority_Config_API	SCP_SBImessagePriority_ConfigAPI_P0
SbiMessagePriority	CP_SBImessagePriority_P0
SCP_Features_API	SCP_SCPFeatures_API_P0
SCP_Generated_Error_Codes_SMF	SCPGeneratedErrorCodes_SMF_P0
SCP_Generated_Error_Codes_UDM	SCP_SCPGeneratedErrorCodes_UDM_P0
NRF_Subscription	SCP_Subscription_SubscriptionWithNRFforNfTypeUDM_P0
Configurataion_Support_For_NEF_services_NEF_P1	SCP_SupportForNEFservices_Configuration_APIs_P1
Support_For_NEF_services_NEF_Enhanced_NF_Status_P0	SCP_SupportForNEFservices_EnhancedSuspendedStateRouting_P0
Support_For_NEF_services_NEF_SEPP_P0	SCP_SupportForNEFservices_InterPLMNRouting_P0
Support_For_NEF_services_NEF_P0	SCP_SupportForNEFservices_NEF_P0
Support_For_NEF_services_NEF_NRF_SCP_p0	SCP_SupportForNEFservices_NRF_P0
Traffic_Feed_REST_API	SCP_TrafficFeed_API_P0
TrafficFeed_CB	SCP_TrafficFeed_CB_P0
TrafficFeed_Notification	SCP_TrafficFeed_NotificationRequest_P0
TrafficFeed_PCF	SCP_TrafficFeed_PCF_P0
TrafficFeed_UDM	SCP_TrafficFeed_UDM_P0
Consumer_Info_API	SCP_UserAgent_ConsumerInfo_API_P0
Scp_User_Agent_Info_API	SCP_UserAgent_Info_API_P0
Worker_Pod_Overload_Action_Policy_Config_API	SCP_WorkerPodOverLoadControl_ActionPolicy_API_P0
CPUOverloadCtrlByNoAction	SCP_WorkerPodOverLoadControl_CPUbasedNoAction_P0
Worker_Pod_Overload_Ctrl_Policy_Config_API	SCP_WorkerPodOverLoadControl_CtrlPolicy_API_P0
Worker_Pod_Overload_Discard_Policy_Config_API	SCP_WorkerPodOverLoadControl_DiscardPolicy_API_P0
Worker_Traffic_Pod_Overload_P0	SCP_WorkerPodOverLoadControl_PendingTransaction_P0
Worker_Traffic_Pod_Overload_P1	SCP_WorkerPodOverLoadControl_PendingTransaction_P1
API_Worker_Pod_Pending_Transaction_Overload_Config_P0.feature	SCP_WorkerPodOverLoadControl_PendingTransactionConfig_API_P0
SCP_1_12_0_Bug_Fixes	SCP_1_12_0_BugFixes_P0
SCP_1.9.0_Bug_Fixes_SMF	SCP_1.9.0_BugFixes_SMF_P0
SCP_1.9.0_Bug_Fixes_UDR	SCP_1.9.0_BugFixes_UDR_P0
SCP_22.2.0_Bug_Fixes_PCF_Cases	SCP_22.2.0_BugFixes_PCF_P0
SCP_22.2.0_Bug_Fixes_UDM_Cases	SCP_22.2.0_BugFixes_UDM_P0
SCP_22.3.0_Bug_Fixes	SCP_22.3.0_BugFixes_P0
SCP_22.3.1_Bug_Fixes	SCP_22.3.1_BugFixes_P0
22.4.0_Bug_Fixes	SCP_22.4.0_BugFixes_P0
22.4.0_Bug_Fixes_Mediation	SCP_22.4.0_Bug_Fixes_Mediation
22.4.4_Bug_Fixes	SCP_22.4.4_BugFixes_PCF_P0
23.1.0_Bug_Fixes	SCP_23.1.0_BugFixes_P0

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-3 (Cont.) Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
23.2.0_Bug_Fixes_HTTPS	SCP_23.2.0_BugFixes_HTTPS_P0
23.2.0_Bug_Fixes	SCP_23.2.0_BugFixes_P0

The following table displays the ATS Regression feature tests that have been split into SingleIpEndpoint and MultipleIpEndpoint categories, and renamed accordingly:

Table 4-4 ATS Regression Feature Tests

Feature Test Name before Spliting	Feature Test Name after Spliting
SCP_SupportForNEFservices_EnhancedSuspendedStateRouting_P0	• SCP_SupportForNEFservices_SingleIpEndpoint_EnhancedSuspendedStateRouting_P0 • SCP_SupportForNEFservices_MultipleIpEndpoint_EnhancedSuspendedStateRouting_P0
SCP_RoutingOptionEnhancments_CBAndCongestionControl_SMF_P0	• SCP_RoutingOptionEnhancments_CBAndCongestionControl_SMF_P0 • SCP_RoutingOptionEnhancments_SingleIpEndpoint_CBAndCongestionControl_SMF_P0 • SCP_RoutingOptionEnhancments_MultipleIpEndpoint_CBAndCongestionControl_SMF_P0
SCP_OutlierDetection_AUSF_P0	• SCP_OutlierDetection_MultipleIpEndpoint_AUSF_P0 • SCP_OutlierDetection_SingleIpEndpoint_AUSF_P0
SCP_Oauth2Support_GranularityConfigModelD_SMFandNRF_P0	• SCP_Oauth2Support_GranularityConfigModelD_SMFandNRF_P0 • SCP_Oauth2Support_SingleIpEndpoint_GranularityConfigModelD_SMFandNRF_P0 • SCP_Oauth2Support_MultipleIpEndpoint_GranularityConfigModelD_SMFandNRF_P0
SCP_ModelC_NFInstance_NEF_P0	• SCP_ModelC_NFInstance_NEF_P0 • SCP_ModelC_SingleIpEndpoint_NFInstance_NEF_P0 • SCP_ModelC_MultipleIpEndpoint_NFInstance_NEF_P0
SCP_LCIsupport_Miscellaneous_P0	• SCP_LCIsupport_Miscellaneous_P0 • SCP_LCIsupport_SingleIpEndpoint_Miscellaneous_P0 • SCP_LCIsupport_MultipleIpEndpoint_Miscellaneous_P0
SCP_EgressHostPreference_InterSCPRoutingNonASM_CHF_P0	• SCP_EgressHostPreference_SingleIpEndpointInterSCPRoutingNonASM_CHF_P0 • SCP_EgressHostPreference_MultipleIpEndpointInterSCPRoutingNonASM_CHF_P0
SCP_DNSSRV_ModelCHeaders_PCF_P0	• SCP_DNSSRV_ModelC_SingleIpEndpoint_PCF_P0 • SCP_DNSSRV_ModelC_MultipleIpEndpoint_PCF_P0
SCP_AlternateRoutingUsingStaticConfig_SMF_P0	• SCP_AlternateRoutingUsingStaticConfig_SMF_P0 • SCP_AlternateRoutingUsingStaticConfig_SingleIpEndpoint_SMF_P0 • SCP_AlternateRoutingUsingStaticConfig_MultipleIpEndpoint_SMF_P0
SCP_22.3.0_BugFixes_P0	• SCP_22.3.0_BugFixes_SingleIpEndpoint_P0 • SCP_22.3.0_BugFixes_MultipleIpEndpoint_P0

Copyright © 2019, 2025, Oracle and/or its affiliates.

4.6 Running SEPP Test Cases using ATS

This section describes how to run SEPP test cases using ATS.

4.6.1 Prerequisites

To run SEPP Test Cases using SEPP ATS 25.1.201, you need to ensure that following prerequisites are fulfilled:

- Debug container should be DISABLED for all microservices during ATS runs.
- To run the ATS testcases, the user should not configure any value for apiPrefix key in `ocsepp-custom-values.yaml` file. It should be empty.
- The number of replicas of all SEPP microservices pods must be set to 1 as ATS is enabled to perform the metric validations for metrics obtained from a single pod.
- Create Kubernetes secret with certificates/keys (public and private) for both PLMN and N32 gateways before deploying SEPP.
- Deploy SEPP 25.1.201 with default helm configurations using helm charts.
- For ATS execution, deploy SEPP microservices with single replica.
- Ensure all microservices of SEPP are up and running.
- Create Kubernetes secret with certificates/keys (public and private) for ATS client and stub server microservices before deploying SEPP ATS.
- Update `ocats_ocsepp_values_25.1.201.yaml` file with the latest secrets and certificates.
- Deploy ATS using helm charts.
- The user must not initiate a job in two different pipelines at the same time.
- User must not abort a running job, this may lead to data corruption in the database.
- The user needs to edit cn32f-svc and pn32f-svc deployment yaml. Set CACHE_REFRESH_TIMEOUT to 1000.
- User need to set `n32cHandshakePlmnIdListValidationEnabled` to true in localProfile section of `ocsepp-custom-values.yaml` file. Or `n32cHandshakePlmnIdListValidationEnabled` Parameter can also be changed after SEPP deployment by editing cn32c-svc and pn32c-svc deployment yaml.
- User needs to update the plmnIdList provided in the ATS installation guide to run the ATS testcases.
- The user needs to edit cn32f-svc and pn32f-svc deployment yaml. Set "TOPOLOGY_HIDING_CACHE_REFRESH_TIMEOUT" to 1000.
- The user needs to edit cn32f-svc and pn32f-svc deployment yaml. Set "SECURITY_CACHE_REFRESH_TIMEOUT" to 1000.
- All the features are set to disable by default. Ensure that the features must be disabled while configuring SEPP ATS.
- The user needs to edit cn32f-svc, pn32f-svc, plmn-egress-gateway, and n32-egress-gateway deployment yaml. Set "REQUEST_TIMEOUT" to 2000.
- The user needs to edit n32-ingress-gateway deployment yaml. Set "REQUEST_TIMEOUT" to 5000.
- The user needs to edit configmap of n32-ingress-gateway. Set "requestTimeout" to 5000.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- The user needs to edit n32-ingress-gateway deployment yaml. SET "IDLE_TIMEOUT" to 5000.
- The user needs to edit pn32f-svc deployment yaml. SET "EVICT_SAN_HEADER_CACHE_DELAY" to 100.
- To run the Cat-3 testcases, user must have pre-installed coherence-svc while SEPP installation.
- For DNS SRV and SOR, plmn-egress-gateway should be deployed in REST mode.
- To run the DNS SRV testcases, user must have pre-installed alternate-route service while SEPP installation.
- To run the Load Sharing among Multiple Remote SEPP Nodes testcases, user must have pre-installed alternate-route service while SEPP installation.
- To run the Egress Rate Limiter testcases user must have enabled the Egress Rate Limiter feature while SEPP deployment. Also set "EgressRateLimiterFlag" to true in `ocats_ocsepp_values_25.1.201.yaml`.
- To run the Ingress Rate Limiter testcases user must have enabled the Ingress Rate Limiter feature while SEPP deployment. Also set "IngressRateLimiterFlag" to true in `ocats_ocsepp_values_25.1.201.yaml`.
- To run the ATS testcases, user should not configure any value for `apiPrefix` key in `ocsepp-custom-values.yaml` file. It should be empty.
- In ASM mode, set `expose_tls_service` parameter to false in stubserver section of `ocats_ocsepp_values_25.1.201.yaml file.`

Custom Folder Implementation

ATS provides custom test case folders (cust_newfeatures, cust_regression and cust_performance) using which you can add new test cases, remove unwanted test cases and modify existing test cases. It does not impact the original product packaged test cases available in the newfeatures, regression and performance folders. For more details, refer to Custom Folder Implementation.

4.6.2 Logging into ATS

Before logging into ATS, you need to ensure that ATS is deployed successfully using Helm charts as shown below:

> **Note**
>
> To modify default log in password, refer to Modifying Login Password.

Figure 4-61 Pods

```
sepp-ats-rel-ats-bddclient-79b9b5ff9f-mtd9r   1/1   Running   0   17m
sepp-ats-rel-stubserver-2-57949d5f8f-j5dkc    1/1   Running   0   117m
sepp-ats-rel-stubserver-3-fcfd757b6-rt2st     1/1   Running   0   117m
sepp-ats-rel-stubserver-6f98bcb75d-z6k6c      1/1   Running   0   117m
```

For more information on verifying ATS deployment, see Verifying ATS Deployment.
There are following ways to login to ATS GUI.

Copyright © 2019, 2025, Oracle and/or its affiliates.

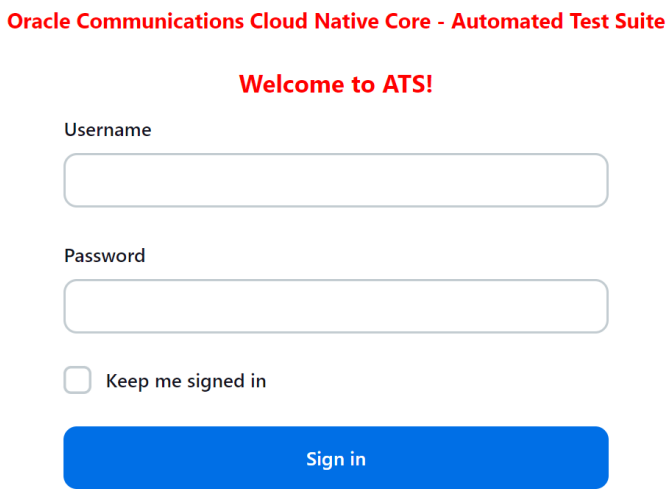
- When an external load balancer (metalLB in case of OCCNE) is available and an external IP Address is provided to the ATS service, login to ATS GUI using <External-IP>:8080.
- When an external IP Address is not provided to the ATS service, open the browser and provide the external IP Address of the worker node and nodeport of the ATS service to login to ATS GUI.
  `<Worker-Node-IP>:<Node-Port-of-ATS>`

> **Note**
>
> In the **Verifying ATS Deployment** screen, ATS nodeport is highlighted in red as 30076. For more details on ATS deployment, refer to **SEPP ATS Installation Procedure.**

- When ATS is installed with TLS enabled:
  - If the external load balancer IP is used in the `ssl.conf` file to open the ATS GUI then the URL will be `"https://<IP>:<https_nodeport>"`.
  - If the external load balancer IP or worker node IP is not present on the setup, then with the common name provided in the `ssl.conf` file, the user can open the ATS GUI. The user needs to do the ssh port forwarding and then need to update the common name (CN) in the `etc/hosts/` file of the local machine against the localhost IP (127.0.0.1).
  - ATS GUI URL format:
    * `https://127.0.0.1:<port-forwarding-port>`
    * `https://<common name>:<port-forwarding-port>`

Open a browser and provide IP Address and port details as <Worker-Node-IP>:<NodePort-of-ATS> (As per the above example: 10.98.101.171:32013). The ATS login screen appears.

Figure 4-62 ATS Login

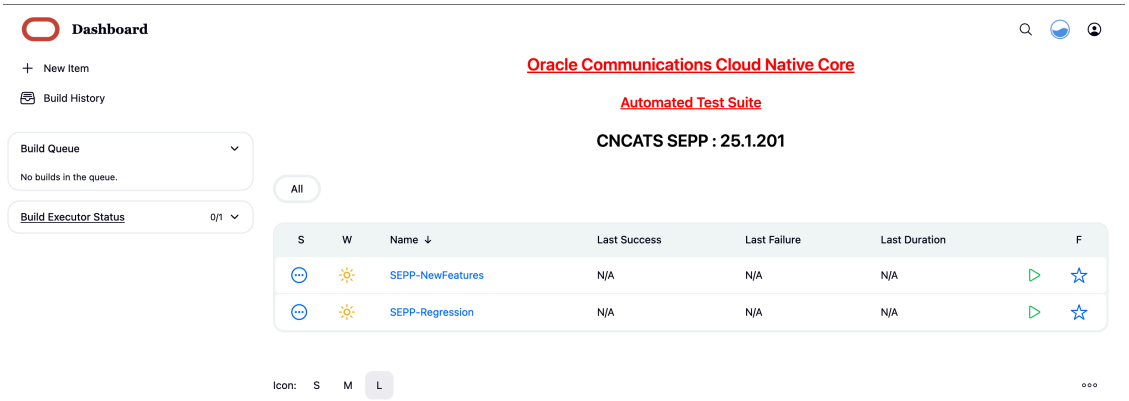


To run ATS:

Copyright © 2019, 2025, Oracle and/or its affiliates.

- Enter the **username** as 'seppuser' and **password** as 'sepppasswd'. Click **Sign in**. The following screen appears showing pre-configured pipelines for SEPP individually (3 Pipelines).

Figure 4-63 Pre-Configured Pipelines



- **SEPP-NewFeatures:** This pipeline has all the test cases that are delivered as part of SEPP ATS.
- **SEPP-Regression:** This pipeline has all the test cases of previous releases.
- **SEPP-HealthCheck:** This pipeline has the utility to run the helm test functionality.

> **Note**
>
> Currently, SEPP does not have any performance pipeline.

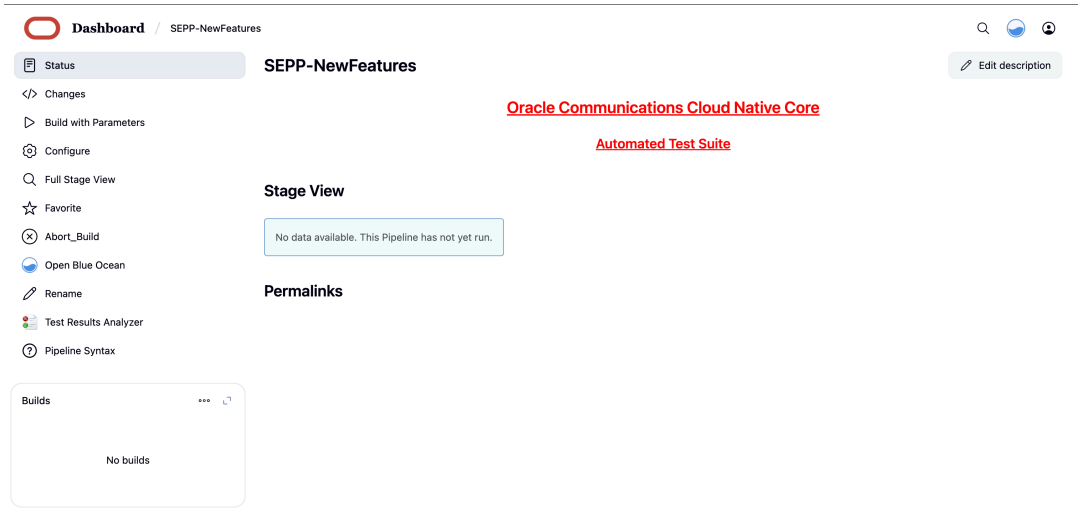
4.6.3 SEPP NewFeatures Pipeline

After identifying the SEPP pipelines, the user needs to do one-time configuration in ATS as per their SUT deployment. In this pipeline, all the new test cases related to SEPP are run. To configure its parameters:

1. Click **SEPP-NewFeatures** in the Name column. The following screen appears:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-64 SEPP-NewFeatures Pipeline



In the above screen:

- Click **Configure** to configure SEPP-New Features.
- Click **Documentation** to view the documented test cases which are part of this SEPP release.
- Click blue dots inside **Build History** box to view the success console logs of the "All" and "Sanity" respectively.
- The **Stage View** shows pipeline that is already run for the customer reference.
- Click **Open Blue Ocean** to view the respective group, re-run logs and the consolidated report summary.
- Click **Abort_Build** button to gracefully stop the currently running ATS build.

2. Click **Configure**. Users must wait for the page to load completely. Once the page loads completely, click the **Pipeline** tab to reach the Pipeline configuration as shown below: The Pipeline section of the configuration page appears as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-65 Pipeline script

```
node ('built-in'){
//a = SELECTED_NF b = SEPPCONFIGSVCNAME c = CSEPPIGWNAME d = PSEPPIGWNAME
//e = SEPPSTUBNAME f = NFINSTANCEID g = PROMSVCIP h = PROMSVCPORT  i= PROMSVCURI j = RERUN_COUNT
//k = ELK_WAIT_TIME l = ELK_HOSTNAME m = ELK_PORT n = STUB_LOGS  o = LOG_TYPE p = GLOBAL_WAIT_TIME
withEnv([

]){

sh '''
sh /var/lib/jenkins/ocsepp_tests/preTestConfig.sh \
-a SEPP \
-b ocsepp-release-config-mgr-svc.csepp \
-c ocsepp-release-plmn-ingress-gateway.csepp \
-d ocsepp-release-n32-ingress-gateway.csepp \
-e sepp-ats-rel-stubserver.csepp \
-f 9faf1bbc-6e4a-4454-a507-aef01a101a06 \
-g prometheus.cne-infra \
-h 9090 \
-i /api/ \
-j 1 \
-k 10 \
-l occne-elastic-elasticsearch-master.occne-infra \
-m 9200 \
-n yes \
-o kubernetes \
-p 5
'''
if(env.Include_Regression && "${Include_Regression}" == "YES"){
   sh '''sh /var/lib/jenkins/common_scripts/merge_jenkinsfile.sh'''
   load "/var/lib/jenkins/ocsepp_tests/jenkinsData/Jenkinsfile-Merged"
}
else{
   load "/var/lib/jenkins/ocsepp_tests/jenkinsData/Jenkinsfile-NewFeatures"
}

}
}
```

In the above screen, you can change the values of the 'Pipeline script'.

- The user must not change any other value apart from line number 12 to line 20 and line number 26.
- User can change only those parameters that are marked as "a" to "h" as per your requirement.
- A new parameter "p" has been introduced to customize the global wait time for the scale up and scale down of pods test cases. Pods scaling time is different on different setups. Sometimes pods take longer than usual for scaling up or scaling down operations. To handle this situation you can increase the global wait time parameter as required. Recommended value is 60.
- Recommended value for rerun count is "3".

```
a: Name of the NF to be tested in capital (SEPP).
b: Config manager service name and namespace
c: Plmn ingress service name and namespace
d: N32 ingress service name and namespace
e: Stub server service name and namespace
f: SEPP Instance ID configured
g: Prometheus hostname and namespace
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
h: Prometheus port
i: Prometheus API endpoint
j: Re run count
k: ELK wait time before logs are fetched
l: ELK hostname and namespace
m: ELK port
n: Stub log collection enable
o: Type of log collection method ( kubernetes/Elasticsearch)
p: GLOBAL WAIT TIME
```

> **Note**
>
> Do not change any value if OCCNE cluster is used and SEPP, ATS and STUB are deployed in ocsepp namespace.

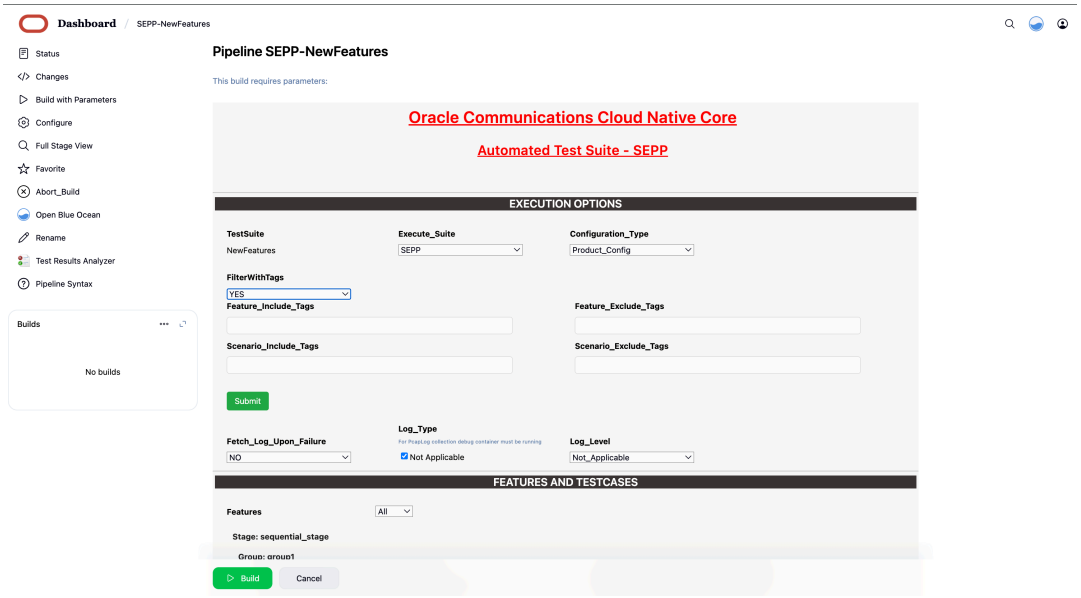
1. Click **Save** after making necessary changes. The **Pipeline SEPP-NewFeatures** screen appears.

Running SEPP Test Cases

To run SEPP test cases:

1. Click the **Schedule a Build with parameters** icon present on the SEPP-NewFeatures screen in the extreme right column corresponding to SEPP-NewFeatures row. The following screen appears:

Figure 4-66 SEPP-NewFeatures



2. In the above screen, there is an option **Execute_Suite** to select SEPP or Roaming_Hub. If you have deployed SUT as SEPP, select **SEPP** option and if you have deployed SUT as Roaming_Hub, select **Roaming_Hub** option.
3. In the above screen, there is an option **Configuration_Type** to select product/custom configuration.

Copyright © 2019, 2025, Oracle and/or its affiliates.

4. In the above screen, there are three **Select_Option(s)**, which are:
   - **All**: This is the default option. It runs all the SEPP test cases. Scroll-down and click **Build** to run all the test cases.
   - **Sanity**: It is recommended to run Sanity before running any test case. It ensures all the deployments are done properly.
   - **Single/Multiple Feature**: This option allows you to select any number of test cases you want to run from total number of test cases available for execution. After selecting the test cases, scroll-down and click **Build** to run the selected SEPP test cases.
5. In the above screen, When **FilterWithTags** option is selected as **Yes**, GUI offers the following four options, select the required tags from the different tags list and click **Submit**.
   a. Feature_Include_Tags: The features that contain either of the tags available in the **Feature_Include_Tags** field are considered for tagging.
   b. Feature_Exclude_Tags: The features that contain neither of the tags available in the **Feature_Exclude_Tags** field are considered for tagging.
   c. Scenario_Include_Tags: The scenarios that contain either of the tags available in the **Scenario_Include_Tags** field are considered.
   d. Scenario_Exclude_Tags: The features that contain neither of the tags available in the **Scenario_Exclude_Tags** field are considered.
6. Click **Abort_Build** button to gracefully stop the currently running ATS build.
7. Click **Build** to execute the ATS run.

In the above screen, the feature files are now shown in the respective stage and group.

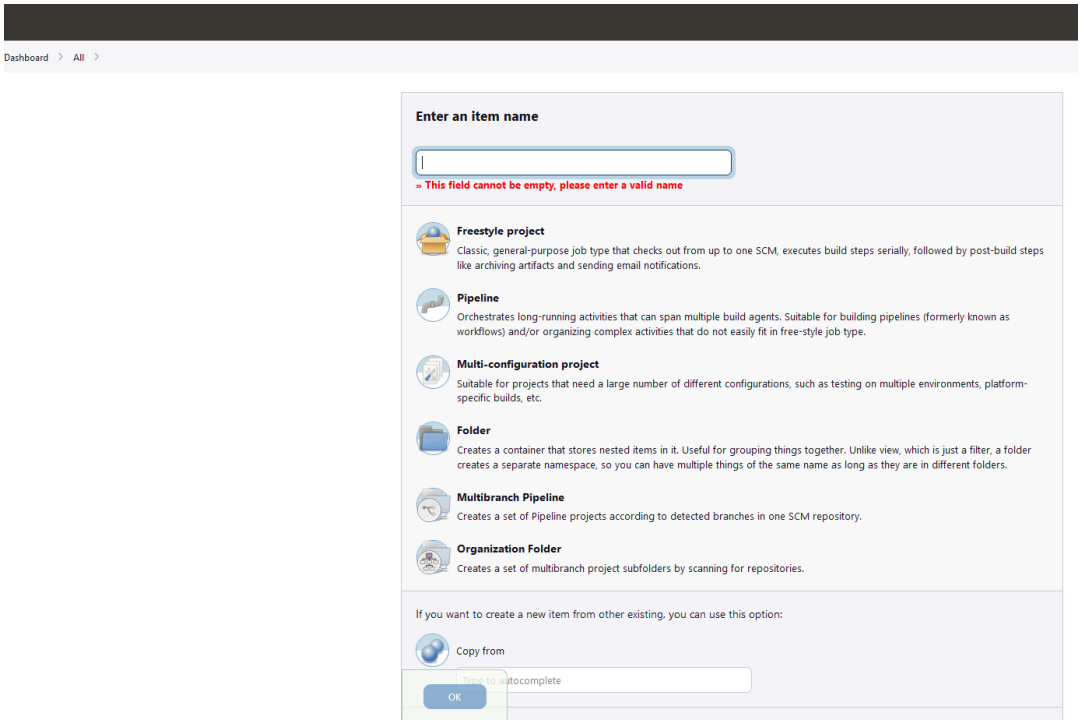
4.6.3.1 Single Click Job Creation for SEPP New Features

Perform the following steps to create the custom pipeline for the new features:

1. Log in to ATS using network function specific log-in credentials.
2. Click **New Item** in the left navigation pane of the ATS application. The following page appears:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-67 New Item Window



3. In the **Enter an item name** text box, enter the job name. Example: <NF-Specific-name>-NewFeatures.
4. In the **Copy from** text box, enter the actual job name for which you need single-click execution functionality. Example: <NF-Specific-name>-NewFeatures.
5. Click **OK**. You are automatically redirected to edit the newly created job's configuration.
6. Under the **General** group, deselect the **This Project is Parameterised** option.
7. Under the **Pipeline** group, make the corresponding changes to remove the 'Active Choice Parameters' dependency.
8. Provide the default values for the **TestSuite**, **SUT**, **Select_Option**, **Configuration_Type**, and other parameters, as required, on the **BuildWithParameters** page.
   Example: Pipeline without Active Choice Parameter Dependency

```
node ('built-in'){
//a = SELECTED_NF b = SEPPCONFIGSVCNAME c = CSEPPIGWNAME d = PSEPPIGWNAME
//e = SEPPSTUBNAME f = NFINSTANCEID g = PROMSVCIP h = PROMSVCPORT  i=
PROMSVCURI j = RERUN_COUNT
//k = ELK_WAIT_TIME l = ELK_HOSTNAME m = ELK_PORT n = STUB_LOGS  o =
LOG_TYPE p = GLOBAL_WAIT_TIME
withEnv([

 'TestSuite=NewFeatures',

 'Execute_Suite=SEPP',

 'FilterWithTags=true,false',

 'Fetch_Log_Upon_Failure=NO',
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
 'Select_Features_Option=All',

 'Configuration_Type=Product_Config'

]){

sh '''
sh /var/lib/jenkins/ocsepp_tests/preTestConfig.sh \
-a SEPP \
-b ocsepp-release-config-mgr-svc.csepp \
-c ocsepp-release-plmn-ingress-gateway.csepp \
-d ocsepp-release-n32-ingress-gateway.csepp \
-e sepp-ats-rel-stubserver.csepp \
-f 9faf1bbc-6e4a-4454-a507-aef01a101a06 \
-g prometheus.cne-infra \
-h 9090 \
-i /api/ \
-j 1 \
-k 10 \
-l occne-elastic-elasticsearch-master.occne-infra \
-m 9200 \
-n yes \
-o kubernetes \
-p 5
'''
if(env.Include_Regression && "${Include_Regression}" == "YES"){
    sh '''sh /var/lib/jenkins/common_scripts/merge_jenkinsfile.sh'''
    load "/var/lib/jenkins/ocsepp_tests/jenkinsData/Jenkinsfile-Merged"
}
else{
    load "/var/lib/jenkins/ocsepp_tests/jenkinsData/Jenkinsfile-
NewFeatures"
}

}
}
```

9. Click **Save**. The ATS application is ready to run TestSuite with 'SingleClick' using the newly created job.
10. Run the custom pipeline job.
11. Click the **Build Now** in the main page for the Custom pipeline job.

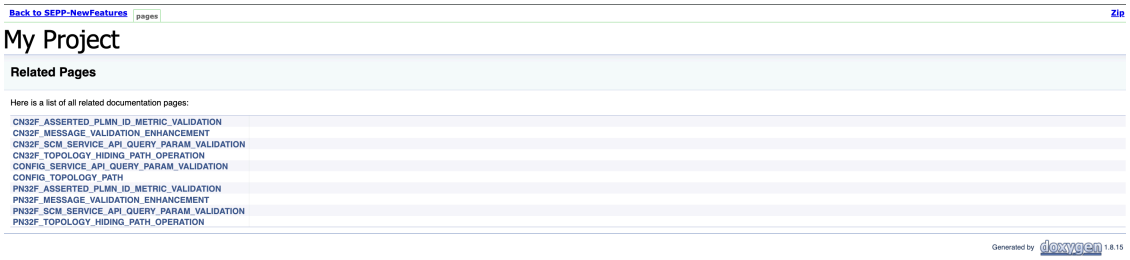
> **Note**
>
> Value of `j (RERUN_COUNT) = 3` is recommended. But it can be changed as per operator environment.

4.6.4 NewFeatures - Documentation

To view SEPP functionalities, go to SEPP-NewFeatures pipeline and click the **Documentation** link in the left navigation pane. The following screen appears:

Copyright © 2019, 2025, Oracle and/or its affiliates.

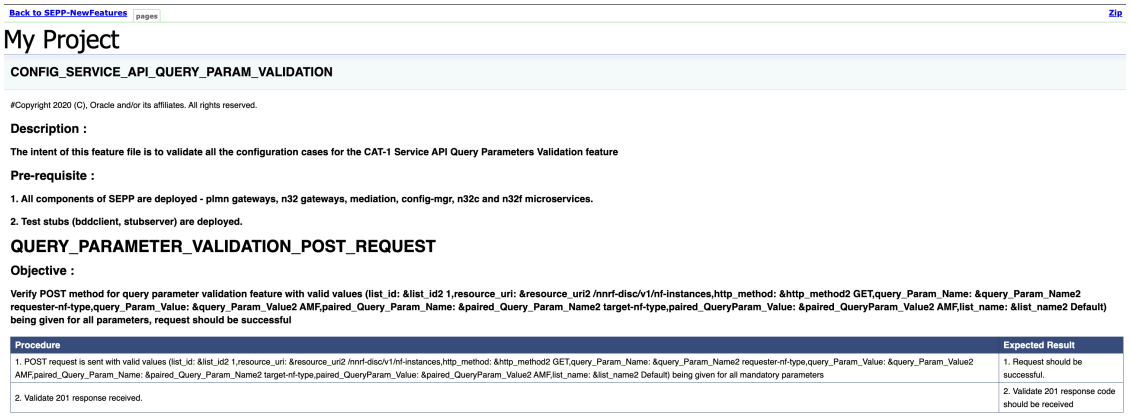
Figure 4-68 Documentation



Click any functionality to view its test cases and scenarios of each test case. A sample screen is given below:

Sample Feature: CONFIG_SERVICE_API_QUERY_PARAM_VALIDATION

Figure 4-69 Test cases and Scenarios



Based on the functionalities covered under the documentation, the build Requires parameter screen displays test cases. To navigate back to the Pipeline SEPP-NewFeatures screen, click the Back to SEPP-NewFeatures link available on top left corner of the screen.

4.6.5 SEPP Regression Pipeline

This section describes how to run test cases for SEPP Regression pipeline.

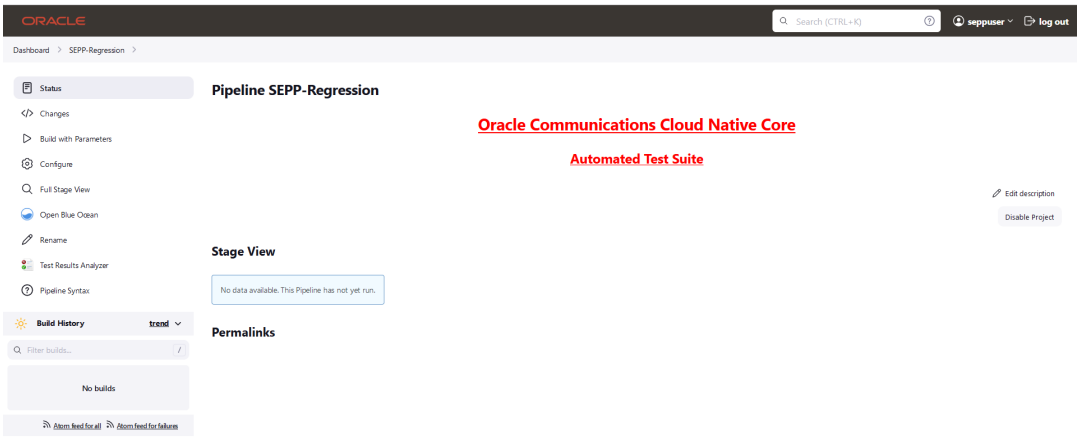
The **SEPP_Regression** pipeline is a pre-configured pipeline where all the test cases of previous releases are available. For example, for SEPP 25.1.2xx, this pipeline has all the test cases released till SEPP 25.1.1xx.

To view **SEPP_Regression** pipeline details, perform the following steps:

1. Click **SEPP_Regression** in the **Name** column. The following screen appears:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-70 SEPP Regression



2. Click **Build with Parameters** in the left navigation pane.
3. Click **Documentation** to view the documented test cases which are part of this SEPP release.
4. Click blue dots in the **Build History** box to view the success console logs of the "All" and "Sanity" respectively.
5. The **Stage View** shows pipeline that is already run for the customer reference.
6. Click **Configure** to configure SEPP Regression. Users must wait for the page to load completely. Once the page loads completely.
7. click the **Pipeline** tab to see the Pipeline configuration.

Copy the required test cases that are available in the SEPP folder and place them appropriately within the **custom** folder for SEPP_Regression. Reload the page to view the test cases available in the **custom** Regression folder.

The configuration method and parameters are same as the **NewFeatures** pipeline.

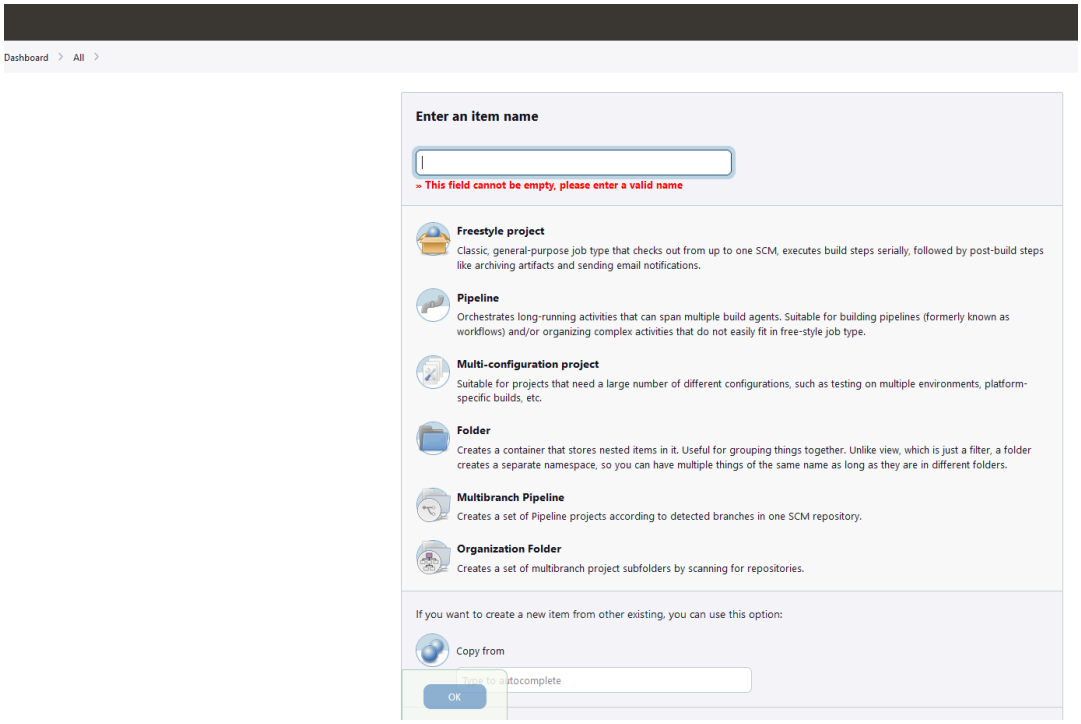
4.6.5.1 Single Click Job Creation for SEPP Regression Features

Perform the following steps to create the custom pipeline for the new features:

1. Log in to ATS using network function specific log-in credentials..
2. Click **New Item** in the left navigation pane of the ATS application. The following page appears:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-71 New Item Window



3. In the **Enter an item name** text box, enter the job name. Example: <NF-Specific-name>-NewFeatures.
4. In the **Copy from** text box, enter the actual job name for which you need single-click execution functionality. Example: <NF-Specific-name>-NewFeatures.
5. Click **OK**. You are automatically redirected to edit the newly created job's configuration.
6. Under the **General** group, deselect the **This Project is Parameterised** option.
7. Under the **Pipeline** group, make the corresponding changes to remove the 'Active Choice Parameters' dependency.
8. Provide the default values for the **TestSuite**, **SUT**, **Select_Option**, **Configuration_Type**, and other parameters, as required, on the **BuildWithParameters** page.
   Example: Pipeline without Active Choice Parameter Dependency

```
node ('built-in'){
//a = SELECTED_NF b = SEPPCONFIGSVCNAME c = CSEPPIGWNAME d = PSEPPIGWNAME
//e = SEPPSTUBNAME f = NFINSTANCEID g = PROMSVCIP h = PROMSVCPORT  i=
PROMSVCURI j = RERUN_COUNT
//k = ELK_WAIT_TIME l = ELK_HOSTNAME m = ELK_PORT n = STUB_LOGS  o =
LOG_TYPE p = GLOBAL_WAIT_TIME
withEnv([

 'TestSuite=Regression',

 'Execute_Suite=SEPP',

 'FilterWithTags=true,false',

 'Fetch_Log_Upon_Failure=NO',
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
 'Select_Features_Option=All',

 'Configuration_Type=Custom_Config'

]){

sh '''
sh /var/lib/jenkins/ocsepp_tests/preTestConfig.sh \
-a SEPP \
-b ocsepp-release-config-mgr-svc.csepp \
-c ocsepp-release-plmn-ingress-gateway.csepp \
-d ocsepp-release-n32-ingress-gateway.csepp \
-e sepp-ats-rel-stubserver.csepp \
-f 9faf1bbc-6e4a-4454-a507-aef01a101a06 \
-g prometheus.cne-infra \
-h 9090 \
-i /api/ \
-j 1 \
-k 10 \
-l occne-elastic-elasticsearch-master.occne-infra \
-m 9200 \
-n yes \
-o kubernetes \
-p 5
'''
if(env.Include_NewFeatures && "${Include_NewFeatures}" == "YES"){
    sh '''sh /var/lib/jenkins/common_scripts/merge_jenkinsfile.sh'''
    load "/var/lib/jenkins/ocsepp_tests/jenkinsData/Jenkinsfile-Merged"
}
else{
    load "/var/lib/jenkins/ocsepp_tests/jenkinsData/Jenkinsfile-Regression"
}

}
}
```

9. Click **Save**. The ATS application is ready to run TestSuite with 'SingleClick' using the newly created job.
10. Run the custom pipeline job.
11. Click the **Build Now** in the main page for the Custom pipeline job.

> **Note**
>
> Value of `j (RERUN_COUNT) = 3` is recommended. But it can be changed as per operator environment.

4.6.6 SEPP Regression Documentation

This section describes the documentation for **SEPP-Regression** pipeline.

To view the documentation for any of the SEPP features, on the ATS home page, click SEPP-Regression. Then, click **Documentation** in the left navigation pane.

This page shows features of only those test cases that are released in previous releases.

Copyright © 2019, 2025, Oracle and/or its affiliates.

The following screen shows all the documentation features:

Figure 4-72 SEPP-Regression documentation



Click any functionality to view its test cases and scenarios of each test case.

The following screen appears if the CN32C_SUCCESS_SECURITY_CAPABILITY_TLS is selected.

Figure 4-73 CN32C_SUCCESS_SECURITY_CAPABILITY_TLS



4.6.7 Running SEPP Health Check Pipeline

To run health check on SEPP:

1. Navigate to **SEPP-HealthCheck** pipeline and click **Configure**.
2. The **General** tab appears. The user must wait for the page to load completely.
3. Click the **Advanced Project Options** tab. Scroll-down to reach the **Pipeline** configuration section as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Important

Ensure that the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

Figure 4-74 SEPP-HealthCheck



Do not change any value other than line numbers 17, 18, and 19. You can change the parameters marked as "a","b",and "c" as per user requirement. The parameters details are provided as comments in line number 7, 8, and 9. The parameters description is as follows:

- a - helm releases [Provide Release Name with Comma Separated if more than 1]
- b - namespace of SUT.
- c - helm version supported

4. Click **Save**. The **Pipeline SEPP-HealthCheck** page appears.
5. Click **Build Now**. This triggers health check for SUT.

Example output:

```
Started by user seppuser
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/.jenkins/workspace/SEPP-HealthCheck
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Helm-Test-Init)
[Pipeline] catchError
[Pipeline] {
[Pipeline] sh
+ set +x
export HelmTestRelease=ocsepp-release-naman
export HelmTestNamespace=sepp2
export HelmCmd=helm
[Pipeline] sh
+ export NF=SEPP
+ NF=SEPP
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
+ /env/bin/python3 /env/lib/python3.9/site-packages/ocnftest_lib/
helmtest_with_secret.py
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Validate-Helm-Test)
[Pipeline] catchError
[Pipeline] {
[Pipeline] echo
2022-08-29 05:14:23,045[32m     INFO LOG.HELM-TEST:108 [0m| [32mhelm test
ocsepp-release-naman -n sepp2[0m
2022-08-29 05:14:53,006[32m     INFO LOG.HELM-TEST:120 [0m| [32m
NAME: ocsepp-release-naman
LAST DEPLOYED: Wed Aug 24 10:08:10 2022
NAMESPACE: sepp2
STATUS: deployed
REVISION: 1
TEST SUITE:     ocsepp-release-naman-test
Last Started:   Mon Aug 29 05:14:36 2022
Last Completed: Mon Aug 29 05:15:00 2022
Phase:          Succeeded[0m

[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

4.7 Running UDR Test Cases using ATS

This section describes how to run UDR test cases using ATS.

4.7.1 Prerequisites

This section describes prerequisites for Subscriber Location Function (SLF) pipelines, Unified Data Repository (UDR) pipelines, and Equipment Identity Register (EIR) pipelines.

- Prerequisites for SLF Pipelines
  - UDR Custom Values File Configuration for SLF Pipelines
  - Provisioning Gateway Custom Values File Configuration for SLF Pipelines
- Prerequisites for UDR Pipelines
  - UDR Custom Values File Configuration for UDR Pipelines
- Prerequisites for EIR Pipelines
  - UDR Custom Values File Configuration for EIR Pipelines

Copyright © 2019, 2025, Oracle and/or its affiliates.

4.7.1.1 Prerequisites for SLF Pipelines

The prerequisites to run SLF pipelines are:

- Select any one of the following System Under Test (SUT) deployment models, depending on your requirements.
  - **Model 1:** To run all the scenarios (SUTSuite: 'All') in the SLF-Regression and SLF-NewFeatures pipeline, deploy UDR with the following configurations:
    * Four UDRs like UDR1, UDR2, UDR3, and UDR4 where UDR1 and UDR2 are in segment1 and UDR3 and UDR4 in segment2.
    * Deploy one Provisioning Gateway along with Provisioning Gateway Auditor service.
    * Deploy two NRF-Stubs
    * Deploy one ATS in same namespace as UDRs, ProvGW, NRF-Stubs

    To deploy four UDRs in same namespace as ProvGw, ATS, and NRF stubs, create four secrets, such as each UDR has unique configuration database (configdb), but UDRs in same segment shares the subscriber database.

    Configure the Provisioning Gateway segment so that the UDR1 and UDR3 are preferred SLFs in their segment 1 and segment 2 respectively. You must also deploy bulk import tool, export tool, and OSO.
  - **Model 2:** To run the SLF-Performance pipeline, deploy UDR with the resources defined in the Resource Requirements section.

    The UDR must be deployed with the following replica counts:

Table 4-5 Replica Count

Service name	Setup type	CPU/Pod	Memory/Pod	Number of pods
ingressgateway-sig	UDR	6	4	2
ingressgateway-prov	UDR	6	4	2
nudr-drservice	UDR	5	4	2
nudr-provservice	UDR	5	4	2

The ATS pod must be installed with the following configuration for 1K TPS:

Table 4-6 ATS Pod Installation for 1K TPS

Service name	Setup type	CPU/Pod	Memory/Pod	Number of pods
ocats-udr	ATS	4	6	1

The ATS pod must be installed with the following configuration for 2K TPS:

Table 4-7 ATS Pod Installation for 2K TPS

Service name	Setup type	CPU/Pod	Memory/Pod	Number of pods
ocats-udr	ATS	8	16	1

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> OSO must also be deployed on the same namespace.

> **Note**
>
> The above configuration is tested for both, constant 1K TPS and varying TPS between 1K and 2K, for a duration of 15 minutes.

– **Model 3:** To run the **Multi_Site** option of the **SUTSuite** field in the SLF Regression pipeline, deploy two UDRs in two different namespaces having two different cnDBTier with database site replication feature enabled between the two. Provisioning Gateway also needs to be deployed on each namespace. Bulk Import tool and export tool needs to be deployed on first namespace. You must also deploy OSO on each namespace.

4.7.1.1.1 UDR Custom Values File Configuration for SLF Pipelines

This section highlights only the necessary changes to be done in the UDR custom-values.yaml file to run ATS.

> **Note**
>
> - Deploy UDR in the SLF mode. For more information about UDR installation, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.
> - To configure ASM with UDR (including setting istio-proxy, CPU, and memory), see the **ASM Specific Configuration** and **UDR Configuration Parameters** sections in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*. The UDR custom-values.yaml file is available with documentation package on MOS.

- Configure the global section of the OCUDR custom values file as follows:

```
global:
  dockerRegistry: <docker-registry>:<docker port>/ocudr
  mysql:
    dbServiceName: <db service to which UDR should be connected>
    port: &dbPortNumber "3306"
    configdbname: &configdbname <config db name>
    dbname: &dbname <subscriber db name>
...
  createNetworkPolicy: true #only if network policy is supported in the
environment. SLF pipeline to validate network policy supports only default
policy
  dbCredSecretName: <udr secret name>
...
  alternateRouteServiceEnable: true
  alternateRouteServiceHost: <udr-helm-release-name>-alternate-route
...
  performanceServiceEnable: true
...
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  udrServices: "nudr-group-id-map"
...
  diamGatewayEnable: false
...
  deploymentMode: &deploymentMode IPv6_IPv4 #If it is intended to run Dual
Stack scenarios, else do not modify the default value
...
  appInfoBaseUrl: "<udr-helm-release-name>-nudr-app-info:5906"
...
  nrfClientNfManagementEnable: true # value is true for UDR1 deployment;
UDR2, UDR3 and UDR4 should be deployed with false
  deploymentNrfClientService:
    envNfNamespace: <namespace where UDR is to be deployed>
...
  enableControlledShutdown: &enableControlledShutDown true
...
  serviceMeshCheck: &serviceMeshFlag true #if UDR is intended to be
deployed with ASM.
```

Note

- If UDR is installed with ASM, then use the annotations defined in the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.
- Database names provided in the SLF custom.values files must match with the names created in the corresponding database secrets.
- The nfInstanceId for each of SLF must be unique for all model.

- Update the **config-server** section as follows:

```
config-server:
  fullnameOverride: <udr-helm-release-name>-config-server
```

- Update the **ingressgateway-sig** section as follows:

```
ingressgateway-sig:
  global:
    istioIngressTlsSupport:
      ingressGateway: true #only if UDR is being deployed with ASM.
...
    lciHeaderConfig:
      enabled: true
      loadThreshold: 30
      localLciHeaderValidity: 10000
...
    svcToSvcInstanceIdMapping:
    - svcName: "<udr-helm-release-name>-nudr-drservice"
      serviceInstanceId: <serviceInstanceId-defined-in-appProfiles-for-
nudr-group-id-map>
...
    xfccHeaderValidation:
      validation:
        enabled: &xfccValidationEnabled true #should be disabled for Model
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
2 and Model 3.
...
  serverHeaderDetails:
    enabled: true
...
  oauthValidatorEnabled: true #should be disabled for model 2 and model 3.
  nrfPublicKeyKubeSecret: <oauth secret name for sig>
  nrfPublicKeyKubeNamespace: <UDR namespace>
  validationType: relaxed
...
  enableControlledShutdown: true
  controlledShutdownConfigMode: REST
...
  userAgentHeaderValidationConfigMode: REST
```

- Update the **ingressgateway-prov** section as follows:

```
ingressgateway-prov:
  global:
   istioIngressTlsSupport:
     ingressGateway: true #only if UDR is being deployed with ASM.
   xfccHeaderValidation:
     validation:
       enabled: &xfccValidationEnabled false
...
  serverHeaderDetails:
    enabled: false
...
  oauthValidatorEnabled: true #should be disabled for model 2 and model 3.
  nrfPublicKeyKubeSecret: <oauth secret name for prov>
  nrfPublicKeyKubeNamespace: <UDR namespace>
  validationType: relaxed
...
  enableControlledShutdown: true
  controlledShutdownConfigMode: REST
...
  checkContentLength: true
  copyContentLengthFromBody: true
```

- Configure targets and ports in the **alternate-route.staticVirtualFqdns** section with the NRF stubs FQDNs and ports.

> **Note**
>
> This is applicable only for Model 1.

```
alternate-route:
  staticVirtualFqdns:
    - name: http://abc.test.com
      alternateFqdns:
       - target: <stub1 fqdn>
         port: <stub1 port>
         priority: 10
       - target: <stub2 fqdn>
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
        port: <stub2 port>
        priority: 20
```

- Update the **appinfo** section as follows:

```
appinfo:
  ...
  dbStatusUri: http://<DB monitoring SVC>:<DB monitoring svc port>/db-tier/
status/local
  realtimeDbStatusUri: http://<DB monitoring SVC>:<DB monitoring svc
port>/db-tier/status/cluster/local/realtime
  replicationUri: http://<DB monitoring SVC>:<DB monitoring svc port>/db-
tier/status/replication
  ...
  prometheusUrl: "http://<Prometheus fqdn or IP>:<prometheus port>/<domain-
name>/prometheus"
  alertmanagerUrl: "http://<alertmanager fqdn or IP>:<prometheus port>/
<domain-name>/alertmanager"
  tagNamespace: kubernetes_namespace # change this field only if it is
being deployed with OSO. Please refer to UDR User Guide for more
information.
  ...
  core_services:
    udr: *slf
  ...
  fullnameOverride: <udr-helm-release-name>-app-info
```

Run the below curl command from ATS pod to find domain name of OSO:

```
curl -i http://<OSO FQDN or IP>:<OSO PORT>/
```

Example:

```
(env) [jenkins@ocats-udr-5c55f59d66-jkzml ~]$ curl -i http://oso-svr:80/
HTTP/1.1 302 Found
Content-Type: text/html; charset=utf-8
Location: /cne-23-1/prometheus
Date: Thu, 20 Apr 2023 06:28:48 GMT
Content-Length: 48

<a href="/cne-23-1/prometheus">Found</a>.
```

Prometheus URL is `http://oso-svr:80/cne-23-1/prometheus`.

- Update the **nrf-client** section as follows:

> **Note**
>
> This is applicable only for Model 1.

```
nrf-client:
  profile: |
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    #uncomment appProfiles that is dedicated to used for SLF mode (nudr-
group-id-map) and comment the rest of the appProfiles.
    enableVirtualNrfResolution=true
    #virtualNrfFqdn value must be same as that of alternate-
route.staticVirtualFqdns.name[0] without the url scheme as shown below
    virtualNrfFqdn=abc.test.com
  nrf-client-nfmanagement:
    ...
    replicas: 1
    ...
    enablePDBSupport: false
```

- Update the **perf-info** section as follows:

```
perf-info:
  envMysqlDatabase: <configdb name used for UDR>
...
  configmapPerformance:
    prometheus: "http://<Prometheus fqdn or IP>:<prometheus port>/<domain-
name>/prometheus"
  log:
    level:
      perfinfo: "WARN"
```

> **Note**
>
> The values for namespace, container name, and servicename depend on CNE or OSO versions. For information, see *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

- Update the **egressgateway** section as follows:

```
egressgateway:
  userAgentHeaderConfigMode: REST
```

- Configure the following parameters in the bulk-import tool yaml file:

> **Note**
>
> To test the bulk import tool, it is important to install the bulk import tool as a separate helm deployment in the same namespace as that of UDR. For information about the bulk import tool installation and sample bulk import yaml file, see *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*. This is applicable for Model 1 only.

  – Ensure the secret is used to install subscriber bulk import tool is same as UDR1. The Bulk Import tool should communicate with UDR1.
    bulk import deployment

```
global:
  mysql:
    dbServiceName: "<DB Endpoint provided for UDR deployment>"
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    port: "<PORT Used provided UDR deployment>"
  ...
  dbCredSecretName: '<UDR1 Secret>'
...
xmlToCsv:
  enabled: false
...
customExtension:
    allResources:
      labels: {}
      annotations:
        oracle.com/cnc: "true"
...
serviceMeshCheck: true #if exprt tool is intended to be deployed with
ASM ...
...
ocudrReleaseName: '<UDR1 Helm name>'
```

> **Note**
>
> * To run the bulk import scenarios, disable **xmlToCsv**.
> * Subscriber bulk import tool must always be clean installed.

Subscriber Export tool Configuration

Configure the following parameters in the nudr-export-tool yaml file:

> **Note**
>
> To test the subscriber export tool, it is important to install the subscriber export tool as a separate helm deployment in the same namespace as that of UDR. For information about the subscriber export tool, see Subscriber Export Tool section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

- Ensure the secret is used to install the subscriber export tool is same as UDR1. The subscriber export tool should communicate with UDR1.
  Update the **nudr-export-tool deployment** section as follows

```
global:
  dockerRegistry: <docker-registry>:<docker port>/ocudr
  mysql:
    dbServiceName: "<DB Endpoint provided for UDR deployment>"
    port: "<PORT Used provided UDR deployment>"
  ...
  dbCredSecretName: '<UDR1 Secret>'
...
sftpDetails:
  ...
  sftpExportEnabled: false
  ...
...
serviceMeshCheck: true #if export tool is intended to be deployed with ASM
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
...
ocudrReleaseName: '<UDR1 Helm name>'
...
batch:
  corePoolSize: 1
  maxPoolSize: 1
exportMode: SLF_EXPORT
```

> **Note**
>
> Subscriber export tool must always be clean installed.

Other configuration details

- Configure Prometheus with UDR and Provisioning Gateway default alerts. For more information about UDR alerts, see *Oracle Communications Cloud Native Core, Unified Data Repository User Guide* and for Provisioning Gateway alerts, see *Oracle Communications Cloud Native Core, Provisioning Gateway Installation Guide*.
- Certificates used for Provisioning Gateway TLS should be provided in `/var/lib/jenkins/ocslf_tests/certs` in pod.
- Create Envoy Filter for both, UDR1 setup and ATS as described in Enabling Service Mesh for XFCC header validation scenarios and server header scenarios if installed with ASM.
- Create Peer Authentication as described in Enabling Service Mesh and *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*, if installed with ASM.

4.7.1.1.2 Provisioning Gateway Custom Values File Configuration for SLF Pipelines

> **Note**
>
> For more information about Provisioning Gateway installation, see *Oracle Communications Cloud Native Core Provisioning Gateway Installation Guide*. The ProvGW custom-values.yaml file is available with documentation package on MOS.

- Update the global section of the ProvGW custom-values.yaml file as follows:

```
global:
  dockerRegistry: <docker-registry>:<docker port>/provgw
  mysql:
    dbServiceName: <db service to which UDR should be connected>
    port: &dbPortNumber "3306"
    configdbname: &configdbname <ProvGw DB name>
    dbname: &dbname <ProvGw DB name>
...
  deploymentMode: &deploymentMode IPv6_IPv4 #If it is intended to run Dual
Stack scenarios, else do not modify the default value
...
  createNetworkPolicy: true #only if network policy is supported in the
environment. SLF pipeline to validate network policy supports only default
policy
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
...
  serviceMeshCheck: &serviceMeshFlag true #if UDR is intended to be
deployed with ASM.
...
  usageMode: "SLF"
  udr:
    httpsEnabled: false
    segDetails:
      - name: SEG-1
        fqdnValues: <UDR1 fqdn>,<UDR2 fqdn>
        preferred: <UDR1 fqdn>
      - name: SEG-2
        fqdnValues: <UDR3 fqdn>,<UDR4 fqdn>
        preferred: <UDR3 fqdn>
```

> **Note**
>
> Database names provided in the Provisioning Gateway custom values files must match with the names created in the corresponding database secret.

- Database monitoring fqdn and port must provided under provgw-service:
  provgw-service

  ```
  provgw-service:
  ...
    probes:
      dbStatusCheckUrl: "http://<DB Monitoring svc>:<DB monitoring port>/db-
  tier/status/cluster/local/realtime"
  ```

- Enable the **auditor-service** parameter as follows:

  ```
  auditor-service:
    enabled: true
  ...
    sftpDetails:
      secrets:
        privatekey:
          name: <private key secret created for SFTP from export tool>
        publickey:
          name: <public key secret created for SFTP from export tool>
  ...
    usePersistentVolume: true
  ...
    probes:
       dbStatusCheckUrl: "http://<DB Monitoring svc>:<DB monitoring port>/db-
  tier/status/cluster/local/realtime"
  ```

- Configure **provgw-config** as below:

  ```
  provgw-config:
  ...
    probes:
      dbStatusCheckUrl: "http://<DB Monitoring svc>:<DB monitoring port>/db-
  tier/status/cluster/local/realtime"
  ```


Copyright © 2019, 2025, Oracle and/or its affiliates.

- Under the **prov-ingressgateway annotations** section, set the **traffic.sidecar.istio.io/excludeInboundPorts** as '8443'.
  Ensure the `ssl` section has secret names and namespace that are created as part of Transport Layer Security (TLS) support. Set the **initssl** and **enableIncomingHttps** parameters as 'true'.

```
prov-ingressgateway:
 deployment:
    customExtension:
      labels: {}
      annotations:
        traffic.sidecar.istio.io/excludeInboundPorts: "8443"
    ssl:
      tlsVersion: TLSv1.2,TLSv1.3
      privateKey:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <ProvGw Namespace>
        rsa:
          fileName: <RSA private key pem file>
        ecdsa:
          fileName: <ECDSA private key pem file>

      certificate:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <ProvGw Namespace>
        rsa:
          fileName: <RSA certificate>
        ecdsa:
          fileName: <ECDSA certificate>

      caBundle:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <ProvGw namespace>
        fileName: <caBundle file name>

      keyStorePassword:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <ProvGw namespace>
        fileName: <key txt file>

      trustStorePassword:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <ProvGw namespace>
        fileName: <trust txt file>
...
  initssl: true
...
  enableIncomingHttps: true
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> For information about Provisioning Gateway installation and ASM configuration for Provisioning Gateway, see *Oracle Communications Cloud Native Core Provisioning Gateway Installation Guide*. This section highlights the necessary changes that you are required to do in the Provisioning Gateway custom-values.yaml file to run ATS.

4.7.1.2 Prerequisites for UDR Pipelines

The prerequisites to run UDR pipelines are:

- Select any one of the following SUT deployment models depending on your requirements.
  - **Model 1: Single Site Deployment**
    To run UDR-NewFeatures, select **SUTSuite:** as 'All' and for UDR-Regresssion, select **SUTSuite:** 'ALL', Features: All (Except UDR_Multisite). For this, you must deploy one UDR, one Provisioning Gateway in UDR mode, two diameter stubs, one Fourg stub, one NOTIFY Stub, two NRF-stubs, one SCP stub, and one ATS in same namespace.
  - **Model 2: Multiple Site Deployment:** To run UDR-Regression pipeline where **SUTSuite:** is 'MULTISITE', you must deploy:
    * Two UDRs in two different namespaces that has two different cnDBTier with database site replication enabled between the two.
    * ATS in any one namespace.
    * Namespace-1 with one UDR, two diameter stubs, one Fourg stub, one NOTIFY stub, two NRF-Stubs, one SCP stub, and one ATS.
    * Namespace-2 with one UDR.

4.7.1.2.1 UDR Custom Values File Configuration for UDR Pipelines

This section highlights only the necessary changes that you are required to do in the UDR custom-values.yaml file to run ATS.

> **Note**
>
> For information about UDR installation and ASM configuration for UDR, see the *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide* available on My Oracle Support.

- Update the `global section` as follows:

```
global:
  dockerRegistry: <docker-registry>:<docker port>/ocudr
  mysql:
    dbServiceName: <db service to which UDR should be connected>
    port: &dbPortNumber "3306"
    configdbname: &configdbname <config db name>
    dbname: &dbname <subscriber db name>
...
  createNetworkPolicy: true #only if network policy is supported in the
environment. SLF pipeline to validate network policy supports only default
policy
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
...
  dbCredSecretName: <udr secret name>
...
  alternateRouteServiceEnable: true
  alternateRouteServiceHost: <udr-helm-release-name>-alternate-route
  performanceServiceEnable: true
...
 autoCreateSubscriber: false
  FourGPolicyConfiguration:
    ondemandMigration:
      enabled:  &onDemandMigrationEnabled true
...
  s13InterfaceEnable: false
...
  deploymentMode: &deploymentMode IPv6_IPv4 #If it is intended to run Dual
Stack scenarios, else do not modify the default value
...
  appInfoBaseUrl: "<udr-helm-release-name>-nudr-app-info:5906"
...
  deploymentNrfClientService:
    envNfNamespace: <UDR namespace>
...
  egressHttpsEnabled: &egressHttpsFlag true
```

> **Note**
>
> Database names provided in the UDR custom values files must match with the names created in the corresponding database secret.

- Update the `ingressgateway-sig` section as follows:

```
ingressgateway-sig:
  global:
...
    lciHeaderConfig:
      enabled: true
      loadThreshold: 30
      localLciHeaderValidity: 10000
...
    svcToSvcInstanceIdMapping:
    - svcName: "<udr-helm-release-name>-nudr-drservice"
...
    xfccHeaderValidation:
     validation:
       enabled: &xfccValidationEnabled false
...
  service:
    ...
    ssl:
      tlsVersion: TLSv1.2,TLSv1.3
      privateKey:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR Namespace>
        rsa:
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
            fileName: <RSA private key pem file>
          ecdsa:
            fileName: <ECDSA private key pem file>

      certificate:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR Namespace>
        rsa:
          fileName: <RSA certificate>
        ecdsa:
          fileName: <ECDSA certificate>

      caBundle:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR namespace>
        fileName: <caBundle file name>

      keyStorePassword:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR namespace>
        fileName: <key txt file>

      trustStorePassword:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR namespace>
        fileName: <trust txt file>
...
  serverHeaderDetails:
    enabled: true
...
  initssl: true
...
  oauthValidatorEnabled: &oauthEnabled true
  ...
  nrfPublicKeyKubeSecret: <oauth secret name>
  nrfPublicKeyKubeNamespace: <UDR namespace>
  validationType: relaxed
  ...
...
  enableIncomingHttps: true
...
  userAgentHeaderValidationConfigMode: REST
```

- Update the `ingressgateway-prov` section as follows:

```
ingressgateway-prov:
  global:
   xfccHeaderValidation:
     validation:
       enabled: &xfccValidationEnabled false
...
  service:
    ...
    ssl:
      tlsVersion: TLSv1.2,TLSv1.3
      privateKey:
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR Namespace>
        rsa:
          fileName: <RSA private key pem file>
        ecdsa:
          fileName: <ECDSA private key pem file>

      certificate:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR Namespace>
        rsa:
          fileName: <RSA certificate>
        ecdsa:
          fileName: <ECDSA certificate>

      caBundle:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR namespace>
        fileName: <caBundle file name>

      keyStorePassword:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR namespace>
        fileName: <key txt file>

      trustStorePassword:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR namespace>
        fileName: <trust txt file>
...
  serverHeaderDetails:
    enabled: false
...
  initssl: true
...
  oauthValidatorEnabled: &oauthEnabled true
  ...
  nrfPublicKeyKubeSecret: <oauth secret name>
  nrfPublicKeyKubeNamespace: <UDR namespace>
  validationType: relaxed
  ...
...
  enableIncomingHttps: true
...
  checkContentLength: true
  copyContentLengthFromBody: true
```

- Update the `alternate-route` section as follows:

```
alternate-route:
  staticVirtualFqdns:
    - name: http://abc.test.com
      alternateFqdns:
       - target: <nrf stub1 fqdn>
         port: <nrf stub1 port>
         priority: 10
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
      - target: <nrf stub2 fqdn>
        port: <nrf stub2 port>
        priority: 20
```

- Update the `nrf-client` section as follows:

```
nrf-client:
  configmapApplicationConfig:
    ...
    profile:
      ...
      enableVirtualNrfResolution=true
      #virtualNrfFqdn value must be same as that of alternate-
route.staticVirtualFqdns.name[0] without the url scheme as shown below
      virtualNrfFqdn=abc.test.com
      ...

nrfRetryConfig=[{"serviceRequestType":"ALL_REQUESTS","primaryNRFRetryCount"
:1,"nonPrimaryNRFRetryCount":1,"alternateNRFRetryCount":1,"errorReasonsForF
ailure":
["503","504","500","SocketTimeoutException","JsonProcessingException","Unkn
ownHostException","NoRouteToHostException","IOException"],"gatewayErrorCode
s":["503"],"requestTimeout":10},
{"serviceRequestType":"AUTONOMOUS_NFREGISTER","primaryNRFRetryCount":1,"non
PrimaryNRFRetryCount":1,"alternateNRFRetryCount":-1,"errorReasonsForFailure
":
["503","504","500","SocketTimeoutException","JsonProcessingException","Unkn
ownHostException","NoRouteToHostException","IOException"],"gatewayErrorCode
s":["503"],"requestTimeout":10}]
  ...
  nrf-client-nfmanagement:
    enablePDBSupport: false
```

- Update the `egressgateway` section to run SCP routing and TLS notification scenarios as follows:

```
egressgateway:
...
  global:
    lciHeaderConfig:
      enabled: true
      localLciHeaderValidity: 1000
...
    svcToSvcInstanceIdMapping:
    - svcName: "<udr-helm-release-name>-nudr-drservice"
...
  sbiRouting:
    peerConfiguration:
    - id: peer1
      host: '<scp stub svc and this should match with scp stub
expectedAuthorityHeader in scp stub custom values>'
      port: '<scp stub port>'
      apiPrefix: "/"
    peerSetConfiguration:
    - id: set0
      httpConfiguration:
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
      - priority: 1
        peerIdentifier: peer1
...
  routesConfig:
    ...
  - id: scp_route
    ...
    path: /scp/**
    order: 2
    metadata:
       ...
       sbiRoutingEnabled: true
...
  initssl: true
  enableOutgoingHttps: *egressHttpsFlag
...
  service:
  ...
   ssl:
       tlsVersion: TLSv1.2,TLSv1.3
       privateKey:
         k8SecretName: <TLS support secret name>
         k8NameSpace: <UDR Namespace>
         rsa:
           fileName: <RSA private key pem file>
         ecdsa:
           fileName: <ECDSA private key pem file>

       certificate:
         k8SecretName: <TLS support secret name>
         k8NameSpace: <UDR Namespace>
         rsa:
           fileName: <RSA certificate>
         ecdsa:
           fileName: <ECDSA certificate>

       caBundle:
         k8SecretName: <TLS support secret name>
         k8NameSpace: <UDR namespace>
         fileName: <caBundle file name>

       keyStorePassword:
         k8SecretName: <TLS support secret name>
         k8NameSpace: <UDR namespace>
         fileName: <key txt file>

       trustStorePassword:
         k8SecretName: <TLS support secret name>
         k8NameSpace: <UDR namespace>
         fileName: <trust txt file>
...
  userAgentHeaderConfigMode: REST
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

- Update the `perf-info` section as follows:

```
perf-info:
  envMysqlDatabase: <configdb name used for UDR>
...
  configmapPerformance:
    prometheus: "http://<Prometheus fqdn or IP>:<prometheus port>/<domain-
name>/prometheus"
  log:
    level:
      perfinfo: "WARN"
```

Formation of Prometheus URL is as below:

```
(env) [jenkins@ocats-udr-5c55f59d66-jkzml ~]$ curl -i http://oso-svr:80/
HTTP/1.1 302 Found
Content-Type: text/html; charset=utf-8
Location: /cne-23-1/prometheus
Date: Thu, 20 Apr 2023 06:28:48 GMT
Content-Length: 48

<a href="/cne-23-1/prometheus">Found</a>.

Prometheus url would be: http://oso-svr:80/cne-23-1/prometheus
```

> **Note**
>
> The values for namespace, container name and service name depend on CNE versions. For more information, see *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

- Update the `app-info` section as follows:

```
appinfo:
  ...
  dbStatusUri: http://<DB monitoring SVC>:<DB monitoring svc port>/db-tier/
status/local
  realtimeDbStatusUri: http://<DB monitoring SVC>:<DB monitoring svc
port>/db-tier/status/cluster/local/realtime
  replicationUri: http://<DB monitoring SVC>:<DB monitoring svc port>/db-
tier/status/replication
  ...
  prometheusUrl: "http://<Prometheus fqdn or IP>:<prometheus port>/<domain-
name>/prometheus"
  alertmanagerUrl: "http://<alertmanager fqdn or IP>:<prometheus port>/
<domain-name>/alertmanager"
  ...
  fullnameOverride: <udr-helm-release-name>-app-info
```

- Configure the following parameters in the `bulk-import tool yaml file`:

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> To test the bulk import tool, it is important to install the bulk import tool as part of UDR. For information about Bulk Import tool installation and sample bulk import yaml file, see *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

- Ensure the secret used to install bulk import tool is same as UDR. Configure the bulk import deployment values.yaml file as follows:

```
global:
  mysql:
    dbServiceName: "<DB Endpoint provided for UDR deployment>"
    port: "<PORT Used provided UDR deployment>"
  ...
  dbCredSecretName: '<UDR Secret>'
...
batchConfig:
  asynchronous: false
...
xmlToCsv:
  enabled: true
...
ocudrReleaseName: '<UDR Helm name>'
```

- Do not change the containers name. It should be 'nudr-bulk-import' and 'nudr-xmltocsv'.

> **Note**
>
> Subscriber bulk import tool must always be clean installed.

- Configure the following parameters in the `ocudr_nudr_migration_tool_custom_values.yaml` file:

> **Note**
>
> Migration tool must be installed as a separate helm deployment. For information about Migration tool installation, see "Migration Tool" section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

- Ensure the secret used to install the migration tool is same as UDR. Configure the `ocudr_nudr_migration_tool_custom_values.yaml` file as follows:

```
global:
  mysql:
    dbServiceName: "<DB Endpoint provided for UDR deployment>"
    port: "<PORT Used provided UDR deployment>"
  ...
  dbCredSecretName: '<UDR Secret>'
...
  preInstall:
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    ...
      config:
        additionalErrorLogging: ENABLED
        logSubscriberInfo: ENABLED
...
fourG:
  realm: "oracle.com"
  identity: "tekelec.oracle.com"
  nodePort: 3000
  nodeHost: "fourgstub-stub-service"
  maxConnectionRetry: 100
...
hikari:
  queryTimeout: "-1"
...
ocudrReleaseName: '<UDR Helm name>'
...
sourceEndPointURL: 'http://10.75.229.43:3868'
udrServiceBaseURL: '<UDR Helm name>-ingressgateway-prov:80'
...
gracefulShutdown:
  gracePeriod: 30s
```

- Configure the following parameters in the `ocudr_nudr_export_tool_custom_values.yaml` file:

  > **Note**
  >
  > The subscriber export tool must be installed as a separate helm deployment. For information about Migration tool installation, see "Subscriber Export Tool " section in *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.

  – Ensure the secret used to install the subscriber export tool is same as UDR. Configure the `ocudr_nudr_export_tool_custom_values.yaml` file as follows:

```
global:
  dockerRegistry: <docker-registry>:<docker port>/ocudr
  mysql:
    dbServiceName: "<DB Endpoint provided for UDR deployment>"
    port: "<PORT Used provided UDR deployment>"
  ...
  dbCredSecretName: '<UDR Secret>'
...
createExportToolPVC:
  ...
  fileExtension: ".exml"
  ...
...
sftpDetails:
  ...
  sftpExportEnabled: false
  ...
...
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
ocudrReleaseName: '<UDR Helm name>'
...
batch:
  corePoolSize: 1
  maxPoolSize: 1
exportMode: EXML_EXPORT
```

> **Note**
>
> Subscriber export tool must always be clean installed.

- Other configuration details:
  - Configure Prometheus with UDR default alerts. For more information about UDR alerts, see *Oracle Communications Cloud Native Core, Unified Data Repository User Guide*.
  - Certificates used for UDR-TLS should be provided in `/var/lib/jenkins/ocudr_tests/certs` in ATS pod as defined for severheaderFlag in values.yaml file (must be true during UDR installation). For more information, see Installing ATS for UDR.

4.7.1.2.2 Provisioning Gateway Custom Values File Configuration for UDR Pipelines

> **Note**
>
> For more information about Provisioning Gateway installation and ASM installation for Provisioning Gateway, see *Oracle Communications Cloud Native Core Provisioning Gateway Installation Guide*. The ProvGW custom-values.yaml file is available with documentation package on MOS. This section highlights only the necessary changes that are required in the ProvGW custom-values.yaml file to run ATS.

- Update the global section of the ProvGW custom-values.yaml file as follows:

```
global:
  dockerRegistry: <docker-registry>:<docker port>/provgw
  mysql:
    dbServiceName: <db service to which UDR should be connected>
    port: &dbPortNumber "3306"
    configdbname: &configdbname <ProvGw DB name>
    dbname: &dbname <ProvGw DB name>
...
  deploymentMode: &deploymentMode IPv6_IPv4 #If it is intended to run Dual
Stack scenarios, else do not modify the default value
...
  createNetworkPolicy: true #only if network policy is supported in the
environment. SLF pipeline to validate network policy supports only default
policy
...
  usageMode: "UDR"
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> Database names provided in the Provisioning Gateway custom values files must match with the names created in the corresponding database secret.

- Enable the provgw-service parameter as follows:

```
provgw-service:
...
  soapService:
    udrIp: <UDR-Prov fqdn>
    udrSignallingIp: <UDR-Sig Fqdn>
    convertToSec: false
    secEntity: |
      - name: QuotaEntity
      elementString: usage
      innerElemenString: quota
      outerFields:
        - version
        - quota
        - SequenceNumber
      innerFields:
        - name
        - cid
        - time
        - totalVolume
        - inputVolume
        - outputVolume
        - serviceSpecific
        - nextResetTime
        - Type
        - grantedTotalVolume
        - grantedInputVolume
        - grantedOutputVolume
        - grantedTime
        - grantedServiceSpecific
        - QuotaState
        - RefInstanceId
        - custom1
      - name: DynamicQuotaEntity
      elementString: definition
      innerElemenString: DynamicQuota
      outerFields:
        - version
        - DynamicQuota
        - SequenceNumber
      innerFields:
        - Type
        - name
        - InstanceId
        - Priority
        - InitialTime
        - InitialTotalVolume
        - InitialInputVolume
        - InitialOutputVolume
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
      - InitialServiceSpecific
      - activationdatetime
      - expirationdatetime
      - purchaseddatetime
      - Duration
      - InterimReportingInterval
      - custom1
    - name: Subscriber
    elementString: subscriber
    outerFields:
      - IMSI
      - MSISDN
      - NAI
      - EXTID
      - ACCOUNTID
      - BillingDay
      - Entitlement
      - Tier
      - SequenceNumber
      - custom1
      - custom2
      - custom3
      - custom4
      - custom5
      - custom6
      - custom7
      - custom8
      - custom9
      - custom10
      - custom11
      - custom12
      - custom13
      - custom14
      - custom15
    - name: StateEntity
    elementString: state
    innerElemenString: property
    outerFields:
      - version
      - property
      - SequenceNumber
    innerFields:
      - name
      - value
    - name: CustomEntity
    elementString: customusage
    innerElemenString: customent
    outerFields:
      - version
      - customent
      - SequenceNumber
    innerFields:
      - name
      - custcid
      - custtotal
...
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  probes:
    dbStatusCheckUrl: "http://<DB Monitoring svc>:<DB monitoring port>/db-
tier/status/cluster/local/realtime"
```

- Disable auditor-service and configure auditor-service section as follows:

```
auditor-service:
  enabled: false
```

- Enable the **provgw-config** section as below:

```
provgw-config:
...
  probes:
     dbStatusCheckUrl: "http://<DB Monitoring svc>:<DB monitoring port>/db-
tier/status/cluster/local/realtime"
```

- Under the **prov-ingressgateway annotations** section, set the **routesConfig** for provgw-service port as '62001'. Ensure the `ssl` section has secret names and namespace that are created as part of Transport Layer Security (TLS) support. Set the **initssl** and **enableIncomingHttps** parameters as 'true'.

```
prov-ingressgateway:
 deployment:
    ...
    ssl:
      tlsVersion: TLSv1.2,TLSv1.3
      privateKey:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR Namespace>
        rsa:
          fileName: <RSA private key pem file>
        ecdsa:
          fileName: <ECDSA private key pem file>

      certificate:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR Namespace>
        rsa:
          fileName: <RSA certificate>
        ecdsa:
          fileName: <ECDSA certificate>

      caBundle:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR namespace>
        fileName: <caBundle file name>

      keyStorePassword:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR namespace>
        fileName: <key txt file>

      trustStorePassword:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR namespace>
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
        fileName: <trust txt file>
...
  initssl: true
...
  enableIncomingHttps: true
...
  routesConfig:
  - id: traffic_mapping_rest_group_prov
    uri: http://{{ .Release.Name }}-provgw-service:62001
```

> **Note**
>
> For information about Provisioning Gateway installation and ASM configuration for Provisioning Gateway, see *Oracle Communications Cloud Native Core Provisioning Gateway Installation Guide*. This section highlights the necessary changes that you are required to do in the Provisioning Gateway custom-values.yaml file to run ATS.

4.7.1.3 Prerequisites for EIR Pipelines

To run EIR Pipelines, you must deploy one EIR, two NRF-stubs, one ATS, one diam-stub deployed in n5g-eir-eic mode, subscriber bulk import tool, and subscriber export tool in same namespace.

4.7.1.3.1 UDR Custom Values File Configuration for EIR Pipelines

> **Note**
>
> Deploy UDR in the EIR mode. For more information about UDR installation, see *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*. This section highlights only the necessary changes that you are required to do in the UDR custom-values.yaml file to run ATS.

- Update the global section as follows:

```
global:
  dockerRegistry: <docker-registry>:<docker port>/ocudr
  mysql:
    dbServiceName: <db service to which UDR should be connected>
    port: &dbPortNumber "3306"
    configdbname: &configdbname <config db name>
    dbname: &dbname <subscriber db name>
...
  dbCredSecretName: <udr secret name>
...
  alternateRouteServiceEnable: true
  alternateRouteServiceHost: <udr-helm-release-name>-alternate-route
...
  performanceServiceEnable: true
...
  udrServices: "n5g-eir-eic"
...
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  diamGatewayEnable: true
...
  deploymentMode: &deploymentMode IPv6_IPv4 #If it is intended to run Dual
Stack scenarios, else do not modify the default value
...
   appInfoBaseUrl: "<udr-helm-release-name>-nudr-app-info:5906"
...
  nrfClientNfManagementEnable: true # value is true for UDR1 deployment;
UDR2, UDR3 and UDR4 should be deployed with false
  deploymentNrfClientService:
    envNfNamespace: <namespace where UDR is to be deployed>
```

> **Note**
>
> Database names provided in the Provisioning Gateway custom values files must match with the names created in the corresponding database secret.

- Update the config-server as follows:

```
config-server:
  fullnameOverride: <udr-helm-release-name>-config-server
```

- Update the ingressgateway-sig as follows:

```
ingressgateway-sig:
  global:
...
    lciHeaderConfig:
      enabled: true
      loadThreshold: 30
      localLciHeaderValidity: 10000
...
    svcToSvcInstanceIdMapping:
    - svcName: "<udr-helm-release-name>-nudr-drservice"
...
    xfccHeaderValidation:
     validation:
       enabled: &xfccValidationEnabled true
...
  service:
    ...
    ssl:
      tlsVersion: TLSv1.2,TLSv1.3
      privateKey:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR Namespace>
        rsa:
          fileName: <RSA private key pem file>
        ecdsa:
          fileName: <ECDSA private key pem file>

      certificate:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR Namespace>
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
        rsa:
          fileName: <RSA certificate>
        ecdsa:
          fileName: <ECDSA certificate>

      caBundle:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR namespace>
        fileName: <caBundle file name>

      keyStorePassword:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR namespace>
        fileName: <key txt file>

      trustStorePassword:
        k8SecretName: <TLS support secret name>
        k8NameSpace: <UDR namespace>
        fileName: <trust txt file>
...
  serverHeaderDetails:
    enabled: true
...
  initssl: true
...
  oauthValidatorEnabled: &oauthEnabled true
  nfType: 5G_EIR
  producerScope: n5g-eir-eic
  ...
  nrfPublicKeyKubeSecret: <oauth secret name>
  nrfPublicKeyKubeNamespace: <UDR namespace>
  validationType: relaxed
  ...
...
  enableIncomingHttps: true
...
  userAgentHeaderValidationConfigMode: REST
```

- Update the ingressgateway-prov as follows:

```
ingressgateway-prov:
  global:
   xfccHeaderValidation:
     validation:
       enabled: &xfccValidationEnabled false
...
  serverHeaderDetails:
    enabled: false
...
  initssl: true
...
  oauthValidatorEnabled: &oauthEnabled true
  nfType: 5G_EIR
  producerScope: n5g-eir-prov
  ...
  nrfPublicKeyKubeSecret: <oauth secret name>
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  nrfPublicKeyKubeNamespace: <UDR namespace>
  validationType: relaxed
  ...
...
   enableIncomingHttps: true
...
  checkContentLength: true
  copyContentLengthFromBody: true
```

- Configure targets and ports in the alternate-route.staticVirtualFqdns section with the NRF stubs FQDNs and ports.

```
alternate-route:
  staticVirtualFqdns:
    - name: http://abc.test.com
      alternateFqdns:
       - target: <nrf stub1 fqdn>
         port: <nrf stub1 port>
         priority: 10
       - target: <nrf stub2 fqdn>
         port: <nrf stub2 port>
         priority: 20
```

- Update the appinfo section as follows:

```
appinfo:
...
  dbStatusUri: http://<DB monitoring SVC>:<DB monitoring svc port>/db-tier/
status/local
  realtimeDbStatusUri: http://<DB monitoring SVC>:<DB monitoring svc
port>/db-tier/status/cluster/local/realtime
  replicationUri: http://<DB monitoring SVC>:<DB monitoring svc port>/db-
tier/status/replication
...
  prometheusUrl: "http://<Prometheus fqdn or IP>:<prometheus port>/<domain-
name>/prometheus"
  alertmanagerUrl: "http://<alertmanager fqdn or IP>:<prometheus port>/
<domain-name>/alertmanager"
...
  core_services:
    5g_eir: *eir
  fullnameOverride: <udr-helm-release-name>-app-info
```

- Update the nrf-client section as follows:

```
nrf-client:
  profile: |
    #uncomment appProfiles that is dedicated to used for EIR mode (n5g-eir-
eic) and comment the rest of the appProfiles.
    enableVirtualNrfResolution=true
    #virtualNrfFqdn value must be same as that of alternate-
route.staticVirtualFqdns.name[0] without the url scheme as shown below
    virtualNrfFqdn=abc.test.com
  nrf-client-nfmanagement:
    ...
    replicas: 1
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
    ...
    enablePDBSupport: false
```

- Update the perf-info section as follows:

```
pperf-info:
  envMysqlDatabase: <configdb name used for UDR>
...
  configmapPerformance:
    prometheus: "http://<Prometheus fqdn or IP>:<prometheus port>/<domain-
name>/prometheus"
  log:
    level:
      perfinfo: "WARN"
```

> **Note**
>
> The values for namespace, container name, and servicename depend on CNE or OSO versions. For information, see *Oracle Communications Cloud Native Core, Unified Data Repository User Guide.*.

- Update the egressgateway section as follows:

```
egressgateway:
  userAgentHeaderConfigMode: REST
```

- Update the nudr-diameterproxy to verify diameter S13 scenarios as follow:

```
nudr-diameterproxy:
  enabled: true
```

- Configure the following parameters in the bulk-import tool yaml file:

> **Note**
>
> To test the subscriber bulk import tool, it is important to install the bulk import tool as a separate helm deployment in the same namespace as that of EIR. For information about the bulk import tool installation and sample bulk import yaml file, see *Oracle Communications Cloud Native Core, Unified Data Repository User Guide.*.

  – Ensure the secret is used to install subscriber bulk import tool is same as EIR. Configure the Bulk Import deployment values.yaml file as follows:

```
global:
  mysql:
    dbServiceName: "<DB Endpoint provided for UDR deployment>"
    port: "<PORT Used provided UDR deployment>"
  ...
  dbCredSecretName: '<EIR Secret>'
...
batchConfig:
```


Copyright © 2019, 2025, Oracle and/or its affiliates.

```
  asynchronous: false
...
xmlToCsv:
  enabled: false
...
pdbiToCsv:
  enabled: true
...
ocudrReleaseName: '<EIRHelm name>'
```

> **Note**
>
> Subscriber bulk import tool must always be clean installed.

- Configure the following parameters in the nudr-export-tool deployment yaml file:

> **Note**
>
> To test the subscriber export tool, it is important to install the subscriber export tool as a separate helm deployment in the same namespace as that of UDR. For information about the subscriber export tool installation, see *Oracle Communications Cloud Native Core, Unified Data Repository User Guide.*.

  – Ensure the secret is used to install subscriber export tool is same as EIR. Configure the subscriber export tool deployment values.yaml file as follows:

```
global:
  dockerRegistry: <docker-registry>:<docker port>/ocudr
  mysql:
    dbServiceName: "<DB Endpoint provided for EIR deployment>"
    port: "<PORT Used provided UDR deployment>"
  ...
  dbCredSecretName: '<EIR Secret>'
...
sftpDetails:
  ...
  sftpExportEnabled: false
  ...
...
ocudrReleaseName: '<EIR Helm name>'
...
exportMode: EIR_EXPORT
```

> **Note**
>
> Subscriber export tool must always be clean installed.

- Other configuration details:
  – Configure Prometheus with EIR default alerts. For more information about EIR alerts, see *Oracle Communications Cloud Native Core, Unified Data Repository User Guide.*

Copyright © 2019, 2025, Oracle and/or its affiliates.

4.7.2 UDR Golden Configuration Support

The ATS golden configuration feature will facilitate the provisioning of customer specific configuration to run test cases. UDR ATS has been integrated with the ATS golden configuration feature. This feature enables the customer to provide their own configuration and save the configuration once the ATS runs are completed.

When running a test, ATS detects the existence of a <feature_name>.yaml configuration file within the <conf_type>_config directory. ATS will populate the parameterized steps and data files with the values present in the global and feature-specific configuration files. The two configuration types are:

- Product configuration
- Custom configuration

You must use the custom_config directory files for provisioning the golden configuration data.

To Configure ATS Golden Configuration

- Modify the parameter values in the /ocslf_tests/custom_config/global.yaml and/or feature config file (/ocslf_tests/custom_config/<feature>.yaml).
- Select the custom_config in the jenkins pipeline GUI to run the test cases with data provided in the custom_config directory.

ATS Golden Configuration Directory

ATS golden configuration uses the below directories located within the /var/lib/jenkins/ocslf_tests directory:

- Parameterized features
  - features/
    - cust_newfeatures/
    - cust_regression/Singlesite/
- Configuration files
  - custom_config/
    - yaml
    - <feature_name>.yaml
- Test data
  - cust_data/

Parameterized features

New and regression feature configuration files use the same custom_config directory.

ATS Golden Configuration Feature Configuration files

This section describes the ATS golden configuration files:

- The global.yaml configuration file contains Key-Value pair parameters. The Key-Value pair parameters applies to more than one parameterized feature file.
- The Key-Value pair parameters in the <feature>.yaml configuration file includes the following fields:

Copyright © 2019, 2025, Oracle and/or its affiliates.

  - subscriber profile data information
  - cust_data payload file values
  - data folder name
  - other feature specific fields
- The <feature>.yaml file is divided into scenario specific sections, containing Key-Value pair parameters. Each parameter in the parameterized feature file takes values only from its own scenario File_Parameters dictionary. You can use the same parameter name in more than one scenario with different scenario specific values.
- When goldenConfig.py is running, each feature config yaml contains the global section at the top between #START_GLOBAL and #END_GLOBAL strings.
- Each SLF <feature>.yaml file allows to make the following changes, if applicable:
  - Use global.yaml anchors to copy the values to scenario specific sections that uses the yalm alias. The data folder name is taken from the global section and is used when a behave step is needed to open a json payload file.
  - Provisioning the subscriber profile data.
  - Overwrite data_folder payload file values. The payload file values will be overwritten in the specified path. The golden configuration feature will overwrite data string with cust_data even if each path is written with the data directory.
  - Provisioning custom values to feature specific fields.

Test data

The test data files includes the json payload files used by each feature. A new cust_data directory, which is a copy of the original data directory, is created to maintain the data directory with product configuration unchanged because the golden configuration feature allows to overwrite data in these json files.

ATS Golden Configuration backup for SLF and Provisioning Gateway

The SLF and Provisioning Gateway backup process is implemented in the before_all and after_all behave hooks to allow ATS pipeline runs to implement feature specific configuration and revert them to the customer specific golden configuration at the end.

ATS Golden Configuration backup Considerations for SLF:

- Backup will be automatically executed for SLF and Provisioning Gateway deployments.
- If failure to export SLF or Provisioning Gateway golden configuration, pipeline execution will be aborted and feature test will not be performed.
- If you fail to import SLF or Provisioning Gateway golden configuration then you are required to manually backup by using 'slfgoldenconfig.zip' and 'provgwgoldenconfig.zip' that is located in ATS pod in the `/var/lib/jenkins/ocslf_tests/` directory.

Selective Run of Features and Scenarios

For SLF ATS golden configuration support, test execution of certain features is depended on the values set for fields in the ocslf_tests/custom_config/global.yaml file.

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> The values provided in the `global.yaml` must match with the SUT configurations (`SLF values.yaml`).

```
#START_GLOBAL
global:
  data_folder: &data_folder cust_data
  # To select an appropriate retries value take from SLF values.yaml the
localLciHeaderValidity value (value in
  #   milliseconds) and perform the following formula:
(localLciHeaderValidity/5000)+1.
  #   It is recommended to use 6 as the minimum retries value for igw-sig-
lciHeaderRequestRetries.
  igw-sig-lciHeaderRequestRetries: &igw-sig-lciHeaderRequestRetries '6'
  # Values provided in this global.yaml should match with the SUT
configurations (SLF values.yaml).
  alternateRouteServiceEnable: true #true/false
  performanceServiceEnable: true #true/false
  igw-sig-lciheader-enabled: true #true/false
  igw-sig-oauth-enabled: true #true/false
  igw-sig-userAgentHeaderValidationConfigMode: REST #HELM/REST
  igw-sig-userAgentHeaderValidation-enabled: true #true/false
  igw-sig-userAgentHeader-validationType: &igw-sig-userAgentHeader-
validationType relaxed #relaxed/strict
  igw-prov-oauth-enabled: true #true/false
  nrf-client-enableVirtualNrfResolution: true #true/false
  egw-userAgentHeaderConfigMode: REST #HELM/REST
  egw-userAgentHeader-enabled: true #true/false
  egw-addFqdnToHeader: true #true/false
  egressgateway_nfFqdn: &egressgateway_nfFqdn udr.oracle.com
  error_logging_enabled: true
#END_GLOBAL
```

List of Parameterized Features

The below tables provide the list of parameterized features for EIR, SLF, and UDR:

Table 4-8 Parameterized Features for EIR

EIR Feature
Prometheus_Scenarios.feature
EIR_XFCC.feature
EIR_User_Agent_Header.feature
EIR_User_AgentHeader_Nrf.feature
EIR_TLS_Signalling.feature
EIR_TLS_Provisioning.feature
EIR_Subscriber_Prov.feature
EIR_Signalling.feature
EIR_Server_Header.feature
EIR_Provisioning_Err.feature

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-8 (Cont.) Parameterized Features for EIR

EIR Feature
EIR_PDBI_Import.feature
EIR_OAuth2_Signalling.feature
EIR_OAuth2_Provisioning.feature
EIR_MGM_API.feature
EIR_Default_Response.feature
EIR_CSV_Import.feature
EIR_AccessLog.feature
EIR_IMEISV.feature
EIR_IMEISV_Default_Response.feature
EIR_IMEISV_AccessLog.feature
EIR_PDBI_IMEISV_Import.feature
EIR_Subscriber_Export_Tool.feature
EIR_22_4_x_Patch_PDBI_Import.feature
EIR_Validation_Additional_Scenarios.feature
EIR_PDBI_Synchronous_Scenarios.feature
EIR_Subscriber_Export_Tool_New.feature

Table 4-9 Parameterized Features for SLF

SLF Feature
SLF_BulkImport.feature
ProvGw_Config_ImportExport.feature
SLF_CHFGroupId.feature
SLF_DefaultGroupId.feature
SLF_User_Agent_Header.feature
SLF_LCI_Header.feature
SLF_Prov_OAuth2_Validation.feature
SLF_User_AgentHeader_Nrf.feature
SLF_LCI_Via_Header.feature
SLF_Subscriber_Activity_Logging.feature
SLF_Subscriber_Export_Tool.feature
SLF_Singlesite_NFCS_BulkImport.feature
SLF_Singlesite_NFCS_Nrf_Client_Scenarios.feature
SLF_Singlesite_NFCS_Prometheus_Scenarios.feature
SLF_Singlesite_NFCS_Scenarios.feature
SLF_Singlesite_NFCS_Subscriber_Export_Tool.feature
ProvGw_Auditor_Status_API.feature
SLF_NF_Scoring_Multisite.feature
SLF_Network_Policy_Validation.feature
SLF_Subscriber_Export_File_Cleanup.feature

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-10 Parameterized Features for UDR

UDR Feature
UDR_LCI_Header_Via.feature
UDR_Subscriber_Activity_Logging.feature
UDR_VSA_PCF_Validation.feature
UDR_PCF_ETag_Validation.feature
UDR_PCF_ETag_Notify_Validation.feature
UDR_AUTO_Enroll_policy_Data.feature
UDR_22.4.x_PCF_VSA_GET_DELETE.feature
ProvGw_SOAP_Additional_scenarios.feature
23_1_1_ProvGw_fixes.feature
UDR_Mgm_Attribute_Mapping.feature
UDR_PCF_Auto_Enrollment_New.feature
UDR_ProvGw_Prov_Headers.feature
UDR_Quota_Migration_Policy_Sh_Soap.feature
UDR_Network_Policy_Validation.feature
UDR_VSA_Delete_All_Validation.feature
UDR_Quota_Migration_Sh_interface.feature
UDR_Quota_Migration_OnDemand_VSA.feature
UDR_Quota_Migration_OnDemand_Sh.feature
UDR_Quota_Migration_OnDemand_SOAP_ProvPolicy.feature
UDR_Quota_Migration_OnDemand_NudrPolicy.feature
UDR_Quota_Migration_OnDemand_Misc.feature
UDR_Network_Policy_Validation.feature
UDR_Bug_Fix_25_1_0.feature
UDR_Default_Headers.feature

4.7.3 UDR Application Log Collection

Application log collection enables ATS to collect logs from System Under Test (SUT) when there are scenario failures. User can set log level to one of values mentioned below:

- WARN
- INFO
- DEBUG
- ERROR
- TRACE

Application log collection is enabled or disabled from the Jenkins Graphical User Interface (GUI). By default application log collection is disabled. When the application log collection is enabled you must follow the ATS Persistent Volume Claim (PVC) guidelines to deploy the ATS with appropriate PVC size. Depending on the log type, the PVC size for the ATS will vary. For storing PCAP logs ATS requires more PVC size than the application logs. Logs can be collected for application logs or PCAP logs or both based on the selection provided through the Jenkins GUI. For more information, see Application Log Collection and PCAP Log Collection.

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> - If the application log collection feature is enabled, it will increase the execution time of the pipeline because some of the steps are executed before and after each scenario. It is recommended to enable the application log collection only if there are consistent failures and pod logs are required to further debug the issues.
> - Do not provide any values for Values.k8sResource.container.prefix and Values.k8sResource.container.suffix in the UDR custom values file and do not modify debug tools container name when PCAP logs are collected.

Configuration

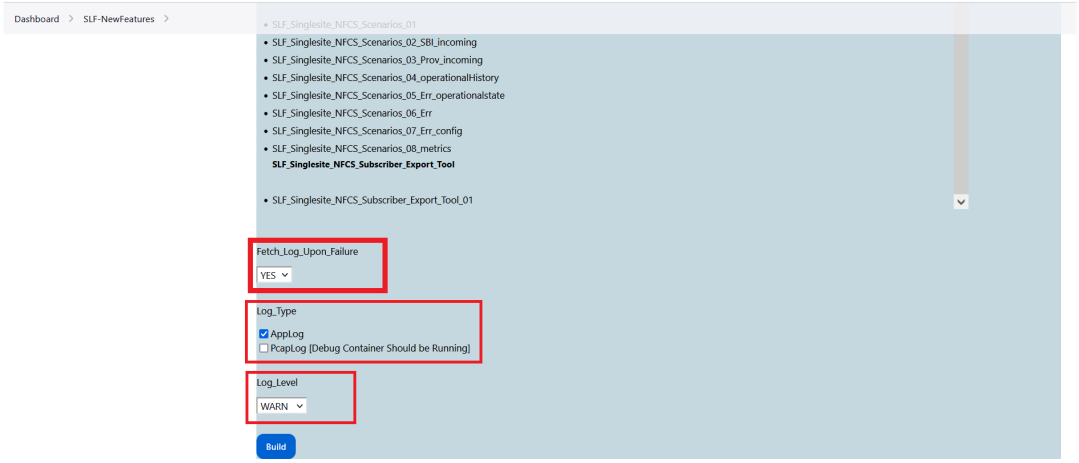
To configure application log collection before installing ATS follow the steps below:

1. ATS must be deployed with PVC enabled. For more information on PVC configurations. For more information, see Enabling Persistent Volume.
2. For PVC size refer to ATS PVC Size Calculation.
3. SUT must be deployed with debug tools to capture PCAP logs on SLF, UDR, and EIR. For more information, see Pod Security Policy, Role, and Role Binding Creation section in *Oracle Communications Cloud Native Core, Unified Data Repository Installation, Upgrade, and Fault Recovery Guide*.
4. To enable a debug tool containers for UDR and Provision Gateway edit the global section of the custom-values.yaml file of UDR and Provision Gateway. For more information, see Using Debug Tools in *Oracle Communications Cloud Native Core, Unified Data Repository Troubleshooting Guide* and *Oracle Communications Cloud Native Core, Provisioning Gateway Guide*.

To configure application log collection after installing ATS follow the steps below:

1. When the ATS and SUT are installed enable the log collection from the ATS Jenkins GUI. This is available on each pipelines on the BuildWithParameters screen of the UDR ATS. This is currently provided only for NewFeatures and Regression Pipelines.

Figure 4-75 SLF-NewFeatures BuildWithParameters



2. To enable the application log collection follow the below steps
   a. From the **Fetch_Log_Upon_Failure** select **YES** from the dropdown list.

Copyright © 2019, 2025, Oracle and/or its affiliates.

b. From the **Log_Type** select the required options, **AppLog** or **PcapLog** or both. Follow the below steps depending on the selection and requirement:

   i. ATS must be deployed with the corresponding PVC size if **AppLog** is selected from the **Log_Type**.

   ii. SUT must be deployed with debug tools container and ATS must be deployed with corresponding PVC size if **PcapLog** is selected from the **Log_Type** or if both **AppLog** and **PcapLog** are selected.

3. Application log collection provides **Log_Level** values from the dropdown. The log level can be set for all the microservices to capture the SUT log for failure. The possible values for **Log_Level** are as follows:
   - WARN
   - INFO
   - DEBUG
   - ERROR
   - TRACE

The captured logs for AppLogs, PcapLogs, or both are stored on the PVC and can be accessed from the ATS builds path as follows:

- Run the following command to log in to the ATS pod:

  ```
  $kubectl exec -it <ATS_POD> -n <NAMESPACE> bash
  ```

- Select the builds path to capture AppLogs and PcapLogs :

  ```
  (env) [jenkins@slfats-ocats-udr-56f76cd668-jzj4d ~]$cd /var/lib/
  jenkins/.jenkins/jobs/<JOB-NAME>/builds/<BUILD-NUM>/


  (env) [jenkins@slfats-ocats-udr-56f76cd668-jzj4d ~]$ls -lrt
  ```

ATS PVC Guidelines

ATS must be deployed with required PVC storage to store the application logs collected during the ATS pipeline runs for the failed scenarios. The PVC size can vary depending on the following selections:

- ATS_Type, Pipeline_Type
- Log_Type
- Log_Level
- Number_of_pipeline_runs

The PVC requirement size per scenario for ATS_Type (SLF-ATS, UDR-ATS, and EIR-ATS) and Log_Type (AppLogs,PcapLogs, or both), and the minimum memory requirements to store the logs on the PVC are given below:

1. SLF-ATS:
   - AppLogs only: Per failure scenario: 72MB
   - PcapLogs only: Per failure scenario: 554MB
   - AppLogs and PcapLogs: Per failure scenario: 626MB

Copyright © 2019, 2025, Oracle and/or its affiliates.

2. UDR-ATS:
   - AppLogs only: Per failure scenario: 1MB
   - PcapLogs only: Per failure scenario: 12GB
   - AppLogs and PcapLogs: Per failure scenario: 13GB
3. EIR-ATS:
   - AppLogs only: Per failure scenario: 500KB
   - PcapLogs only: Per failure scenario: 10GB
   - AppLogs and PcapLogs: Per failure scenario: 11GB

ATS PVC Size Calculation

The maximum PVC size for a given ATS_Type, Pipeline_Type, and Log_Type are calculated using the below formula:

> **Note**
>
> The maximum PVC size must be able to store the logs even if all the scenarios fail in all the runs.

PVC_size = Number of scenarios in Pipeline Type * PVC requirement per scenario for given type [ATS_Type or Log_Type] * Total runs to be executed.

Example 1: To capture the application logs for SLF-ATS-NewFeatures and the Log_Type is set as AppLogs only for 10 pipeline runs, the PVC for ATS is calculated as:

The given SLF-ATS-NewFeatures has 19 scenarios. The memory requirement per scenario is 72MB.

PVC_size = 19*72MB*10 = 13GB. The maximum PVC size required for ATS is 13 GB.

Example 2: To capture the application logs for EIR-ATS-Regression and the Log_Type is set as PcapLogs only for 1 pipeline run, the PVC for ATS is calculated as:

The given EIR-ATS-Regression has 242 scenarios. The memory requirement per scenario is 10GB.

PVC_size = 242*10GB*1 = 2420GB. The maximum PVC size required for ATS is 2420 GB.

4.7.4 Parallel Test Execution

Parallel test execution allows you to perform multiple logically grouped tests simultaneously on the same System Under Test (SUT) to reduce the overall execution time of ATS.

ATS currently runs all its tests in a sequential manner, which is time-consuming. With parallel test execution, tests can be run concurrently rather than sequentially or one at a time. Test cases or feature files are now separated into different folders, such as stages and groups, for concurrent test execution. Different stages, such as stage 1, stage 2, and stage 3, run the test cases in a sequential order, and each stage has its own set of groups. For more information about Parallel Test Execution, see Parallel Test Execution.

SLF ATS Stages and Groups

Following table provides different stages and groups for SLF ATS:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-11 SLF ATS Stages and Groups

Single Site Stages and Groups	Multiple Site Stages and Groups
Singlesite/ ├── Stage_1 │ ├── bulk_import_basics │ │ ├── behave.ini │ │ └── SLF_BulkImport.feature │ ├── export_tool_basics │ │ ├── behave.ini │ │ └── SLF_Subscriber_Export_Tool.feature │ ├── independent_features │ │ ├── behave.ini │ │ ├── SLF_Network_Policy_Validation.feature │ │ └── SLF_Sbi_Correlation_Header.feature │ ├── ingressgatway_features │ │ ├── behave.ini │ │ ├── SLF_Ingress_XFCC_Header.feature │ │ ├── SLF_LCI_Via_Header.feature │ │ ├── SLF_OAuth2_Validation.feature │ │ ├── SLF_Prov_OAuth2_Validation.feature │ │ └── SLF_Server_Header.feature │ ├── nrf-client │ │ ├── behave.ini	Multisite/ └── Multi_Site └── two_site ├── behave.ini ├── SLF_Multisite.feature ├── SLF_Multisite_NFCS_Scenarios.feature └── SLF_NF_Scoring_Multisite.feature

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-11 (Cont.) SLF ATS Stages and Groups

Single Site Stages and Groups	Multiple Site Stages and Groups
│ │ ├── Prometheus_Scenarios_NrfClient.feature │ │ ├── SLF_Dns_Srv.feature │ │ ├── SLF_Nrf_Client_Scenarios.feature │ │ └── SLF_User_AgentHeader_Nrf.feature │ ├── provgw │ │ ├── behave.ini │ │ ├── Prometheus_Scenarios_ProvGw.feature │ │ ├── ProvGw_Basic_Func_Error.feature │ │ ├── ProvGw_Basic_Func_Success.feature │ │ ├── ProvGw_CHFGroupId.feature │ │ ├── ProvGw_Data_Mismatch.feature │ │ └── ProvGw_SSL_Enabled.feature │ └── slf_basics │ ├── behave.ini │ ├── SLF_CHFGroupId.feature │ └── SLF_Prov_and_Lookup_NonMetric.feature ├── Stage_2 │ ├── provgw_config │ │ ├── behave.ini	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-11 (Cont.) SLF ATS Stages and Groups

Single Site Stages and Groups	Multiple Site Stages and Groups

<pre>│ │ └── Provgw_Config.feature
│ └── slf_config
│ ├── behave.ini
│ ├── SLF_DefaultGroupId.feature
│ ├── SLF_Mgm_Apis.feature
│ └── SLF_Subscriber_Activity_Logging.feature
└── Stage_3
└── sequential_execution
├── behave.ini
├── Prometheus_Scenarios.feature
├── ProvGw_Auditor.feature
├── ProvGw_Auditor_Status_API.feature
├── ProvGw_Config_ImportExport.feature
├── ProvGw_Loglevel.feature
├── ProvGw_Segdown.feature
├── SLF_Config_Import_Export.feature
├── SLF_LCI_Header.feature
├── SLF_LogLevels.feature
├── SLF_NF_Scoring.feature
├── SLF_Prov_and_Lookup_Metric.feature</pre>	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-11 (Cont.) SLF ATS Stages and Groups

Single Site Stages and Groups	Multiple Site Stages and Groups
├──── SLF_Singlesite_NFCS_BulkImport.feature ├──── SLF_Singlesite_NFCS_Nrf_Client_Scenarios.feature ├──── SLF_Singlesite_NFCS_Prometheus_Scenarios.feature ├──── SLF_Singlesite_NFCS_Scenarios.feature ├──── SLF_Singlesite_NFCS_Subscriber_Export_Tool.feature ├──── SLF_Subs_Activity_Logging_Loglevels.feature └──── SLF_User_Agent_Header.feature	

UDR ATS Stages and Groups

Following table provides different stages and groups for UDR ATS:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
Singlesite/ ├── All │ └── Stage_1 │ └── sequential_execution │ ├── behave.ini │ ├── UDR_Bulk_Import_PNR_Validation.feature │ ├── UDR_PUR_Scenarios.feature │ ├── UDR_Quota_Dq_Additional_Scenarios.feature │ ├── UDR_Sm_Data_Dnn_Schema_Validation.feature │ └── UDR_Subscriber_Export_Tool.feature └── MISCELLANEOUS └── Stage_1 └── sequential_execution ├── behave.ini ├── UDR_Bulk_Import_PNR_Validation.feature ├── UDR_PUR_Scenarios.feature	Singlesite/ ├── All │ ├── Stage_1 │ │ ├── bulk_import_metrics_autocreate_true │ │ │ ├── behave.ini │ │ │ ├── UDR_Bulk_Import.feature │ │ │ ├── UDR_Bulk_Import_ixml_part1.feature │ │ │ ├── UDR_Bulk_Import_PolicyAndVsa.feature │ │ │ └── UDR_ProvGw_Prov_Headers.feature │ │ ├── diameter_only_metrics │ │ │ ├── behave.ini │ │ │ ├── Diameter_All_Entities.feature │ │ │ ├── Diameter_Config.feature │ │ │ ├── Diameter_Misc.feature │ │ │ ├── Diameter_User_Data.feature │ │ │ ├── UDR_22_4_1_Diameter.feature │ │ │ ├── UDR_23_1_0_Diameter.feature │ │ │ ├── UDR_23_2_0_Diameter.feature │ │ │ ├── UDR_All_Migration_State_Entity_part1.feature │ │ │ ├── UDR_Migration_tool_Additional_scenarios.feature │ │ │ └──	Multisite/ └── Multi_Site └── two_site ├── behave.ini └── UDR_Multisite.feature

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
├── UDR_Quota_Dq_Additional_Scenarios.feature ├── UDR_Sm_Data_Dnn_Schema_Validation.feature └── UDR_Subscriber_Export_Tool.feature	UDR_Migration_tool_part1.feature ├── nrf_only_metrics │ ├── behave.ini │ ├── UDR_NRFclient_DNS_Srv.feature │ ├── UDR_NRFclient_Scenarios.feature │ └── UDR_User_AgentHeader_Nrf.feature ├── oauth2_only_metrics │ ├── behave.ini │ ├── UDR_OAuth2_Validation.feature │ ├── UDR_PCF_Prov_OAuth2_Validation.feature │ └── UDR_UDM_Prov_OAuth2_Validation.feature ├── pcf_subscription_no_metrics │ ├── behave.ini │ ├── UDR_PCF_DELETE.feature │ ├── UDR_PCF_SMData_Patch.feature │ ├── UDR_Subscriber_Policy_Data.feature │ ├── UDR_UDM_API_SDM_Subscription.feature │ └── UDR_UDM_Subscription_Data_Subscription.feature └── vsa_no_metrics	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	│ │ ├── behave.ini │ │ ├── UDR_22.4.x_PCF_VSA_GET_DELETE.feature │ │ ├── UDR_VSA_PATCH_Fix.feature │ │ └── UDR_VSA_PCF_Validation.feature │ ├── Stage_2 │ │ ├── bulk_import_autocreate_false │ │ │ ├── behave.ini │ │ │ └── UDR_Bulk_Import_ixml_part2.feature │ │ └── no_igw_dr_metrics │ │ ├── behave.ini │ │ ├── UDR_PCF_ETag_Notify_Validation.feature │ │ ├── UDR_PCF_Notification_New_API.feature │ │ ├── UDR_PCF_User_Agent_Header.feature │ │ ├── UDR_Suppressed_Notifications.feature │ │ ├── UDR_UDM_Notification.feature │ │ ├── UDR_UDM_UserAgent_Header.feature │ │ └── UDR_User_AgentHeader_Notification.feature │ ├── Stage_3 │ │ ├── provgw_no_igw_dr_metrics │ │ │ ├──	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	23_1_1_ProvGw_fixes.feature behave.ini ProvGw_SOAP_Additional_scenarios.feature ProvGw_SOAP_CNCESUDR82_misc.feature ProvGw_SOAP_ConvertToSec_strict.feature ProvGw_SOAP_Custom_Entities.feature ProvGw_SOAP_Custom_Fields.feature ProvGw_SOAP_Format_Tags.feature ProvGw_SOAP_GetSubscription.feature ProvGw_SOAP_Misc.feature ProvGw_SOAP_Reset_Quota.feature ProvGw_SOAP_SSL.feature ProvGw_SOAP_Subscriber_Entities.feature ProvGw_SOAP_Subscriber_Entity_Fields.feature ProvGw_SOAP_Subscriber_Fields.feature ProvGw_SOAP_Subscriber_Profile.feature UDM_SUBS_Enhancements_Non_Metrics.feature udr_no_igw_dr_metrics	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	│ │ ├── behave.ini │ │ ├── UDR_EGW_LCI_Header.featu re │ │ ├── UDR_LCI_Header_Via.featu re │ │ ├── UDR_SCP_Routing.feature │ │ ├── UDR_SM_Data_Custom_Data_ Enhancements.feature │ │ └── UDR_VSA_Notifications.fe ature │ └── Stage_4 │ └── sequential_execution │ ├── behave.ini │ ├── ProvGw_SOAP_AndUDR_VSA_N ew.feature │ ├── ProvGw_SOAP_Bug_Fixes.fe ature │ ├── UDM_SUBS_Enhancements_Me trics.feature │ ├── UDR_23_2_0_Bug_fixes.fea ture │ ├── UDR_All_Migration_State_ Entity_rest_metric.featu re │ ├── UDR_AUTO_Enroll_policy_D ata.feature │ ├── UDR_Bulk_Import_All_metr ics.feature │ ├── UDR_Bulk_Import_Configur ation_validation.feature │ ├── UDR_Bulk_Import_ixml_bug _fixes.feature	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	│ ├── UDR_Bulk_Import_VSA.feat ure │ ├── UDR_Config_Import_Export .feature │ ├── UDR_Config_Service.featu re │ ├── UDR_Custom_Fields_Profil e.feature │ ├── UDR_Diameter_CER_Changes .feature │ ├── UDR_Diameter_Ondemand_Mi g.feature │ ├── UDR_Egw_TLS.feature │ ├── UDR_IGW_LCI_Header.featu re │ ├── UDR_Log_Level.feature │ ├── UDR_Mgm_Attribute_Mappin g.feature │ ├── UDR_Migration_tool_part2 .feature │ ├── UDR_Misc_Metrics_Alerts. feature │ ├── UDR_NEF_AM_Data.feature │ ├── UDR_NEF_MGM_APIs.feature │ ├── UDR_NEF_Misc.feature │ ├── UDR_NEF_Notification.fea ture │ ├── UDR_NEF_Prov.feature │ ├── UDR_NEF_Prov_OAuth2_Vali dation.feature │ ├──	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	UDR_NEF_SM_Data.feature UDR_NEF_Subscritption.feature UDR_Network_Policy_Validation.feature UDR_New_Log_Level.feature UDR_PCF_Auto_Enrollment_New.feature UDR_PCF_ETag_Validation.feature UDR_PCF_MGM_APIs.feature UDR_PCF_New_API_AllData.feature UDR_PCF_New_API_AM.feature UDR_PCF_New_API_SM.feature UDR_PCF_New_API_Subscription.feature UDR_PCF_New_API_UeData.feature UDR_PCF_New_API_UMD.feature UDR_PCF_New_Server_Header.feature UDR_PCF_Ondemand.feature UDR_PCF_Sbi_Correlation_Header.feature UDR_PCF_User_Agent_Header_New.feature UDR_PCF_VSA_Diameter.fea	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	ture │ ├── UDR_PCF_VSA_ETag_New.fea ture │ ├── UDR_Profile_Data_New.fea ture │ ├── UDR_Prov_Policy_Ondemand .feature │ ├── UDR_Quota_Migration_OnDe mand_Misc.feature │ ├── UDR_Quota_Migration_OnDe mand_NudrPolicy.feature │ ├── UDR_Quota_Migration_OnDe mand_Sh.feature │ ├── UDR_Quota_Migration_OnDe mand_SOAP_ProvPolicy.fea ture │ ├── UDR_Quota_Migration_OnDe mand_VSA.feature │ ├── UDR_Quota_Migration_Poli cy_Sh_Soap.feature │ ├── UDR_Quota_Migration_Sh_i nterface.feature │ ├── UDR_SMData_Custom_Fields .feature │ ├── UDR_Subscriber_Activity_ Logging.feature │ ├── UDR_TLS_Enable.feature │ ├── UDR_UDM_API_Additional_E rr.feature │ ├── UDR_UDM_API_Authenticati on_Data.feature │ ├── UDR_UDM_API_Context_Data .feature	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	│ ├── UDR_UDM_API_EE_Profile_Data.feature │ ├── UDR_UDM_API_Identity_Data.feature │ ├── UDR_UDM_API_Operator_Determined_Barring_Data.feature │ ├── UDR_UDM_API_Operator_Specific_Data.feature │ ├── UDR_UDM_API_Provisioned_Data.feature │ ├── UDR_UDM_API_Provisioned_paramenter_Data.feature │ ├── UDR_UDM_API_UE_Update_Confirmation_Data.feature │ ├── UDR_UDM_Mgm_Apis.feature │ ├── UDR_UDM_Prov_Authentication_Data.feature │ ├── UDR_UDM_Prov_Context_Data.feature │ ├── UDR_UDM_Prov_EE_Profile_Data.feature │ ├── UDR_UDM_Prov_Identity_Data.feature │ ├── UDR_UDM_Prov_Operator_Determined_Barring_Data.feature │ ├── UDR_UDM_Prov_Operator_Specific_Data.feature │ ├── UDR_UDM_Prov_Provisioned_Data.feature │ ├── UDR_UDM_Prov_Provisioned_Paramenter_Data.feature	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	│ ├── UDR_UDM_Prov_UE_Update_C onfirmation_Data.feature │ ├── UDR_UDM_Sbi_Correlation_ Header.feature │ ├── UDR_VSA_Delete_All_Valid ation.feature │ ├── UDR_Vsa_Ondemand.feature │ ├── UDR_XFCC.feature │ └── VSA_SMPOLICY.feature ├── BULK-IMPORT │ ├── Stage_1 │ │ └── bulk_import_metrics_auto create_true │ │ ├── behave.ini │ │ ├── UDR_Bulk_Import.feature │ │ ├── UDR_Bulk_Import_ixml_par t1.feature │ │ └── UDR_Bulk_Import_PolicyAn dVsa.feature │ ├── Stage_2 │ │ └── bulk_import_autocreate_f alse │ │ ├── behave.ini │ │ └── UDR_Bulk_Import_ixml_par t2.feature │ └── Stage_4 │ └── sequential_execution │ ├── behave.ini │ ├── UDR_Bulk_Import_All_metr ics.feature │ ├── UDR_Bulk_Import_ixml_bug	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	_fixes.feature └──── UDR_Bulk_Import_VSA.feature ├──── CONFIG └──── Stage_4 └──── sequential_execution ├──── behave.ini ├──── UDR_Bulk_Import_Configuration_validation.feature ├──── UDR_Config_Import_Export.feature ├──── UDR_Config_Service.feature └──── UDR_New_Log_Level.feature ├──── DIAMETER └──── Stage_1 └──── diameter_only_metrics ├──── behave.ini ├──── Diameter_All_Entities.feature ├──── Diameter_Config.feature ├──── Diameter_Misc.feature ├──── Diameter_User_Data.feature ├──── UDR_22_4_1_Diameter.feature ├──── UDR_23_1_0_Diameter.feature └──── UDR_23_2_0_Diameter.feature ├──── MISCELLANEOUS ├──── Stage_1	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	│ │ ├── bulk_import_metrics_auto create_true │ │ │ ├── behave.ini │ │ │ └── UDR_ProvGw_Prov_Headers. feature │ │ ├── diameter_only_metrics │ │ │ ├── behave.ini │ │ │ ├── UDR_All_Migration_State_ Entity_part1.feature │ │ │ ├── UDR_Migration_tool_Addit ional_scenarios.feature │ │ │ └── UDR_Migration_tool_part1 .feature │ │ ├── nrf_only_metrics │ │ │ ├── behave.ini │ │ │ └── UDR_User_AgentHeader_Nrf .feature │ │ └── oauth2_only_metrics │ │ ├── behave.ini │ │ └── UDR_OAuth2_Validation.fe ature │ ├── Stage_2 │ │ └── no_igw_dr_metrics │ │ ├── behave.ini │ │ ├── UDR_Suppressed_Notificat ions.feature │ │ └── UDR_User_AgentHeader_Not ification.feature │ ├── Stage_3 │ │ ├── provgw_no_igw_dr_metrics	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	│ │ │ ├── 23_1_1_ProvGw_fixes.feat ure │ │ │ ├── behave.ini │ │ │ ├── ProvGw_SOAP_Additional_s cenarios.feature │ │ │ ├── ProvGw_SOAP_CNCESUDR82_m isc.feature │ │ │ ├── ProvGw_SOAP_ConvertToSec _strict.feature │ │ │ ├── ProvGw_SOAP_Custom_Entit ies.feature │ │ │ ├── ProvGw_SOAP_Custom_Field s.feature │ │ │ ├── ProvGw_SOAP_Format_Tags. feature │ │ │ ├── ProvGw_SOAP_GetSubscript ion.feature │ │ │ ├── ProvGw_SOAP_Misc.feature │ │ │ ├── ProvGw_SOAP_Reset_Quota. feature │ │ │ ├── ProvGw_SOAP_SSL.feature │ │ │ ├── ProvGw_SOAP_Subscriber_E ntities.feature │ │ │ ├── ProvGw_SOAP_Subscriber_E ntity_Fields.feature │ │ │ ├── ProvGw_SOAP_Subscriber_F ields.feature │ │ │ └── ProvGw_SOAP_Subscriber_P rofile.feature │ │ └── udr_no_igw_dr_metrics │ │ ├── behave.ini	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	│ │ ├──── UDR_EGW_LCI_Header.feature │ │ ├──── UDR_LCI_Header_Via.feature │ │ └──── UDR_SCP_Routing.feature │ └──── Stage_4 │ └──── sequential_execution │ ├──── behave.ini │ ├──── ProvGw_SOAP_AndUDR_VSA_New.feature │ ├──── ProvGw_SOAP_Bug_Fixes.feature │ ├──── UDR_23_2_0_Bug_fixes.feature │ ├──── UDR_All_Migration_State_Entity_rest_metric.feature │ ├──── UDR_Custom_Fields_Profile.feature │ ├──── UDR_Diameter_CER_Changes.feature │ ├──── UDR_Egw_TLS.feature │ ├──── UDR_IGW_LCI_Header.feature │ ├──── UDR_Log_Level.feature │ ├──── UDR_Migration_tool_part2.feature │ ├──── UDR_Misc_Metrics_Alerts.feature │ ├──── UDR_Network_Policy_Validation.feature │ ├────	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	UDR_PCF_VSA_Diameter.feature ├── UDR_PCF_VSA_ETag_New.feature ├── UDR_Quota_Migration_OnDemand_Misc.feature ├── UDR_Quota_Migration_OnDemand_NudrPolicy.feature ├── UDR_Quota_Migration_OnDemand_Sh.feature ├── UDR_Quota_Migration_OnDemand_SOAP_ProvPolicy.feature ├── UDR_Quota_Migration_OnDemand_VSA.feature ├── UDR_Quota_Migration_Policy_Sh_Soap.feature ├── UDR_Quota_Migration_Sh_interface.feature ├── UDR_Subscriber_Activity_Logging.feature ├── UDR_UDM_Mgm_Apis.feature ├── UDR_VSA_Delete_All_Validation.feature └── UDR_XFCC.feature ├── NEF │ └── Stage_4 │ └── sequential_execution ├── behave.ini ├── UDR_NEF_AM_Data.feature ├── UDR_NEF_MGM_APIs.feature ├── UDR_NEF_Misc.feature	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	│ ├── UDR_NEF_Notification.fea ture │ ├── UDR_NEF_Prov.feature │ ├── UDR_NEF_Prov_OAuth2_Vali dation.feature │ ├── UDR_NEF_SM_Data.feature │ └── UDR_NEF_Subscritption.fe ature ├── NRF-CLIENT │ └── Stage_1 │ └── nrf_only_metrics │ ├── behave.ini │ ├── UDR_NRFclient_DNS_Srv.fe ature │ └── UDR_NRFclient_Scenarios. feature ├── ONDEMAND │ └── Stage_4 │ └── sequential_execution │ ├── behave.ini │ ├── UDR_Diameter_Ondemand_Mi g.feature │ ├── UDR_PCF_Ondemand.feature │ ├── UDR_Prov_Policy_Ondemand .feature │ └── UDR_Vsa_Ondemand.feature ├── PCF │ ├── Stage_1 │ │ ├── oauth2_only_metrics │ │ │ ├── behave.ini │ │ │ └── UDR_PCF_Prov_OAuth2_Vali	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	dation.feature │ │ └── pcf_subscription_no_metr ics │ │ ├── behave.ini │ │ ├── UDR_PCF_DELETE.feature │ │ ├── UDR_PCF_SMData_Patch.fea ture │ │ └── UDR_Subscriber_Policy_Da ta.feature │ ├── Stage_2 │ │ └── no_igw_dr_metrics │ │ ├── behave.ini │ │ ├── UDR_PCF_ETag_Notify_Vali dation.feature │ │ ├── UDR_PCF_Notification_New _API.feature │ │ └── UDR_PCF_User_Agent_Heade r.feature │ ├── Stage_3 │ │ └── udr_no_igw_dr_metrics │ │ ├── behave.ini │ │ └── UDR_SM_Data_Custom_Data_ Enhancements.feature │ └── Stage_4 │ └── sequential_execution │ ├── behave.ini │ ├── UDR_AUTO_Enroll_policy_D ata.feature │ ├── UDR_Mgm_Attribute_Mappin g.feature │ ├── UDR_PCF_Auto_Enrollment_	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	New.feature ├── UDR_PCF_ETag_Validation. feature ├── UDR_PCF_MGM_APIs.feature ├── UDR_PCF_New_API_AllData. feature ├── UDR_PCF_New_API_AM.featu re ├── UDR_PCF_New_API_SM.featu re ├── UDR_PCF_New_API_Subscrip tion.feature ├── UDR_PCF_New_API_UeData.f eature ├── UDR_PCF_New_API_UMD.feat ure ├── UDR_PCF_New_Server_Heade r.feature ├── UDR_PCF_Sbi_Correlation_ Header.feature ├── UDR_PCF_User_Agent_Heade r_New.feature ├── UDR_Profile_Data_New.fea ture ├── UDR_SMData_Custom_Fields .feature └── UDR_TLS_Enable.feature ├── UDM ├── Stage_1 ├── oauth2_only_metrics ├── behave.ini └── UDR_UDM_Prov_OAuth2_Vali	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	dation.feature │ │ └──── pcf_subscription_no_metrics │ │ ├──── behave.ini │ │ ├──── UDR_UDM_API_SDM_Subscription.feature │ │ └──── UDR_UDM_Subscription_Data_Subscription.feature │ ├──── Stage_2 │ │ └──── no_igw_dr_metrics │ │ ├──── behave.ini │ │ ├──── UDR_UDM_Notification.feature │ │ └──── UDR_UDM_UserAgent_Header.feature │ ├──── Stage_3 │ │ └──── provgw_no_igw_dr_metrics │ │ ├──── behave.ini │ │ └──── UDM_SUBS_Enhancements_Non_Metrics.feature │ └──── Stage_4 │ └──── sequential_execution │ ├──── behave.ini │ ├──── UDM_SUBS_Enhancements_Metrics.feature │ ├──── UDR_UDM_API_Additional_Err.feature │ ├──── UDR_UDM_API_Authentication_Data.feature │ ├──── UDR_UDM_API_Context_Data.feature │ ├────	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	UDR_UDM_API_EE_Profile_Data.feature UDR_UDM_API_Identity_Data.feature UDR_UDM_API_Operator_Determined_Barring_Data.feature UDR_UDM_API_Operator_Specific_Data.feature UDR_UDM_API_Provisioned_Data.feature UDR_UDM_API_Provisioned_paramenter_Data.feature UDR_UDM_API_UE_Update_Confirmation_Data.feature UDR_UDM_Prov_Authentication_Data.feature UDR_UDM_Prov_Context_Data.feature UDR_UDM_Prov_EE_Profile_Data.feature UDR_UDM_Prov_Identity_Data.feature UDR_UDM_Prov_Operator_Determined_Barring_Data.feature UDR_UDM_Prov_Operator_Specific_Data.feature UDR_UDM_Prov_Provisioned_Data.feature UDR_UDM_Prov_Provisioned_Paramenter_Data.feature UDR_UDM_Prov_UE_Update_Confirmation_Data.feature	

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-12 (Cont.) UDR ATS Stages and Groups

UDR New Feature	UDR Regression (Single Site)	UDR Regression (Multiple Site)
	│ └──── UDR_UDM_Sbi_Correlation_Header.feature └──── VSA ├──── Stage_1 │ └──── vsa_no_metrics │ ├──── behave.ini │ ├──── UDR_22.4.x_PCF_VSA_GET_DELETE.feature │ ├──── UDR_VSA_PATCH_Fix.feature │ └──── UDR_VSA_PCF_Validation.feature ├──── Stage_3 │ └──── udr_no_igw_dr_metrics │ ├──── behave.ini │ └──── UDR_VSA_Notifications.feature └──── Stage_4 └──── sequential_execution ├──── behave.ini └──── VSA_SMPOLICY.feature	

EIR ATS Stages and Groups

Following table provides different stages and groups for EIR ATS:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-13 EIR ATS Stages and Groups

EIR NewFeatures	EIR Regression
cust_newfeatures/ └── Stage_1 └── eir_basics ├── behave.ini └── EIR_ErrorResponseDetail.feature	cust_regression/ ├── Stage_1 │ ├── bulk_import_basics │ │ ├── behave.ini │ │ ├── EIR_22_4_x_Patch_PDBI_Import.feature │ │ ├── EIR_CSV_Import.feature │ │ ├── EIR_PDBI_IMEISV_Import.feature │ │ ├── EIR_PDBI_Import.feature │ │ └── Prometheus_bulk_import.feature │ ├── eir_basics │ │ ├── behave.ini │ │ ├── EIR_Provisioning_Err.feature │ │ ├── EIR_TLS_Provisioning.feature │ │ ├── EIR_TLS_Signalling.feature │ │ └── EIR_Validation_Additional_Scenarios.feature │ ├── export_tool_basics │ │ ├── behave.ini │ │ ├── EIR_Subscriber_Export_Tool.feature │ │ └── EIR_Subscriber_Export_Tool_New.feature

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-13 (Cont.) EIR ATS Stages and Groups

EIR NewFeatures	EIR Regression
	│ ├── ingressgatway_features │ │ ├── behave.ini │ │ ├── EIR_OAuth2_Provisioning.feature │ │ ├── EIR_OAuth2_Signalling.feature │ │ ├── EIR_Server_Header.feature │ │ └── EIR_XFCC.feature │ └── nrf-client │ ├── behave.ini │ ├── EIR_Dns_Srv.feature │ ├── EIR_Nrf_Client_Scenarios.feature │ ├── EIR_User_AgentHeader_Nrf.feature │ └── Prometheus_nrfclient.feature └── Stage_2 └── sequential_execution ├── behave.ini ├── EIR_22_4_x_Patch_PDBI_Import_part2.feature ├── EIR_AccessLog.feature ├── EIR_Config_Import_Export.feature ├── EIR_CSV_Import_part2.feature

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-13 (Cont.) EIR ATS Stages and Groups

EIR NewFeatures	EIR Regression
	├──── EIR_DefaultIMEINotFound_Response.feature ├──── EIR_Default_Response.feature ├──── EIR_Export_Configuration_validation.feature ├──── EIR_IMEISV_AccessLog.feature ├──── EIR_IMEISV_Default_Response.feature ├──── EIR_IMEISV.feature ├──── EIR_LogLevels.feature ├──── EIR_MGM_API.feature ├──── EIR_PDBI_IMEISV_Import_part2.feature ├──── EIR_PDBI_Import_part2.feature ├──── EIR_PDBI_Synchronous_Scenarios.feature ├──── EIR_Provisioning_Err_part2.feature ├──── EIR_Signalling.feature ├──── EIR_Subscriber_Prov.feature ├──── EIR_User_Agent_Header.feature ├──── EIR_Validation_Additional_Scenarios_part2.feature

Copyright © 2019, 2025, Oracle and/or its affiliates.

Table 4-13 (Cont.) EIR ATS Stages and Groups

EIR NewFeatures	EIR Regression
	└──── Prometheus_Scenarios.feature

Viewing Console Logs for Specific Group

When parallel test execution feature is enabled and ATS runs are triggered, you can view the progress of each stage and group on the corresponding pipeline dashboard.

Figure 4-76 Sample SLF Regression

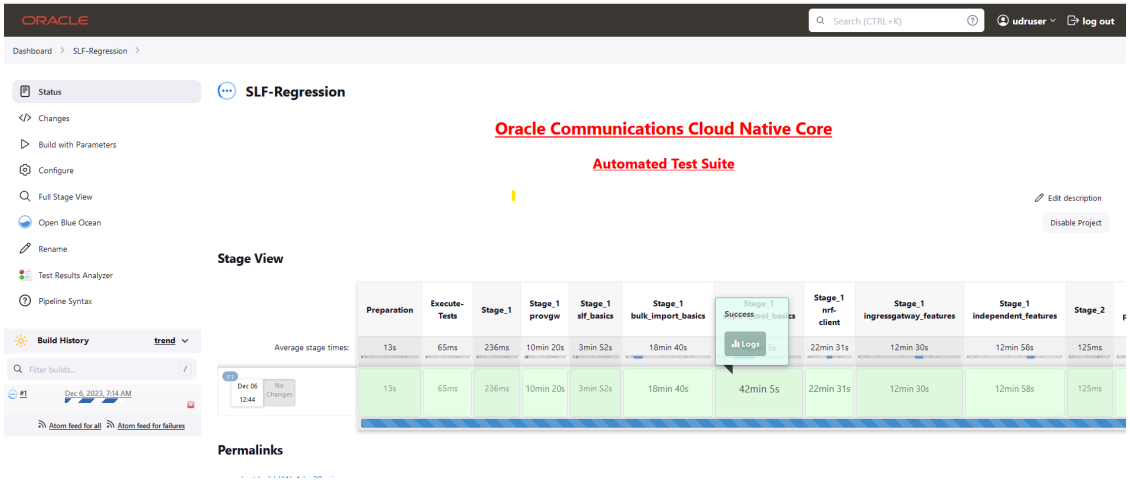
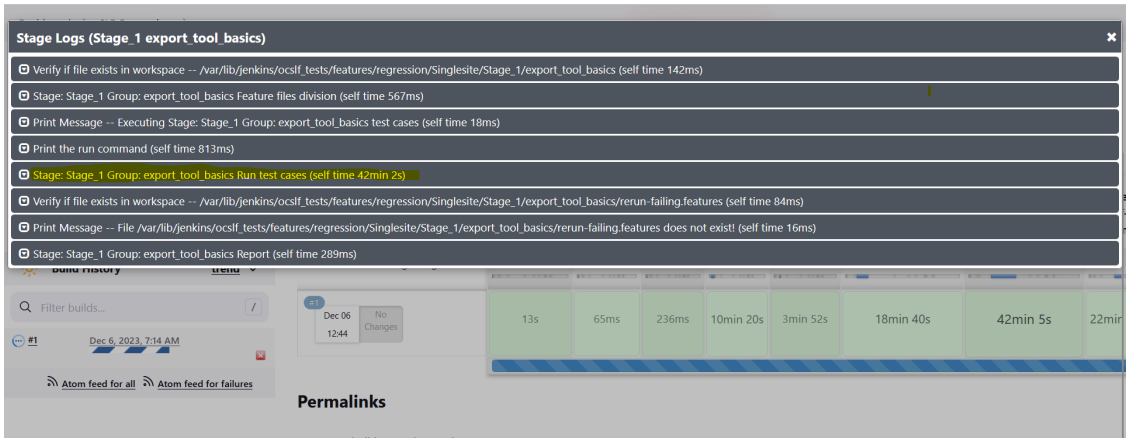


Figure 4-77 Sample Stage Logs



Copyright © 2019, 2025, Oracle and/or its affiliates.

4.7.5 Logging into ATS

Before logging into ATS, it is important to know the nodeport of the "-ocats-UDR" service. To get the nodeport detail, run the following command:

```
kubectl get svc -n <slf_namespace>
```

Example: `kubectl get svc -n ocudr ocats-udr-ga`

Figure 4-78 UDR Nodeport

```
ats-ocats-udr   LoadBalancer   10.233.22.95   <pending>     8080:30156/TCP,5001:31175/TCP   57s
```

In the previous image, 31175 is the nodeport.

Run the following command to get the worker node IP address:

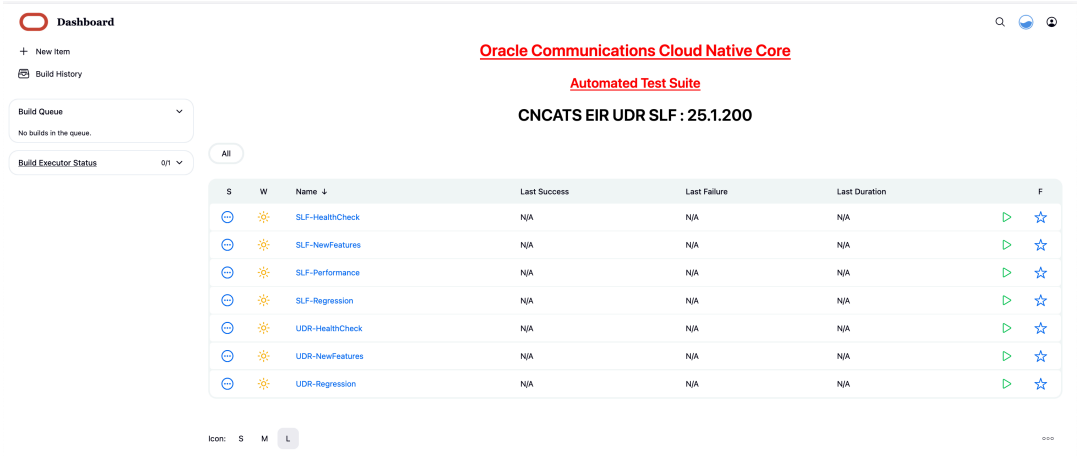
```
kubectl get nodes -o wide
```

To log in to ATS:

1. In the web browser, type **http://<Worker IP>:nodeport** and press **Enter**.
   Example: http://10.75.229.87:30156

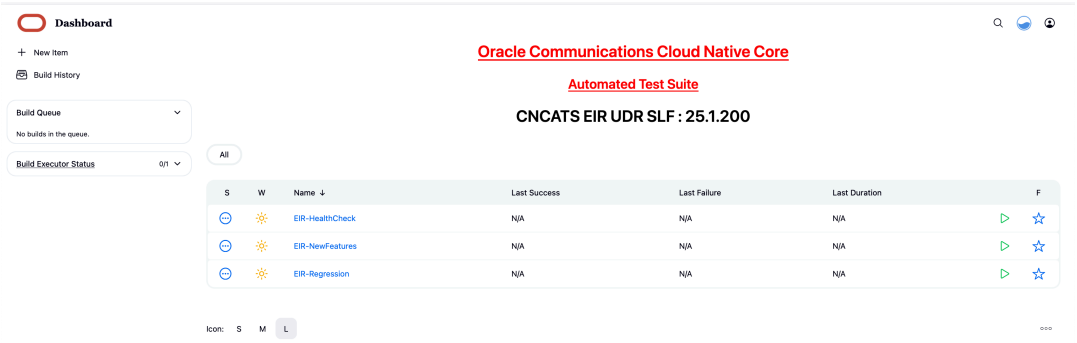
   The log in page appears.
2. Enter the username and password. Click **Sign in**. A page with pre-configured pipelines appears.

Figure 4-79 SLF, EIR, and UDR Pre-configured Pipelines



Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-80 EIR ATS Pipeline



> **Note**
>
> To modify the default log in password, see Modifying Login Password.

- **SLF-HealthCheck:** This pipeline does a test probe with SUT - trigger Helm test and provide the results in Console Logs.
- **SLF-NewFeatures:** This pipeline has all the test cases delivered as part of SLF ATS - 25.1.200 release.
- **SLF-Performance:** This pipeline provides a means to run 1K TPS of lightweight traffic.
- **SLF-Regression:** This pipeline has all the test cases of previous releases.
- **UDR-HealthCheck:** This pipeline does a test probe with SUT - trigger helm test and provide the results in Console Logs.
- **UDR-NewFeatures:** This pipeline has all the test cases delivered as part of UDR ATS - 25.1.200 release.
- **UDR-Regression:** This pipeline has all the UDR test cases of previous releases.
- **EIR-HealthCheck:** This pipeline does a test probe with SUT - trigger Helm test and provide the results in Console Logs.
- **EIR-NewFeatures:** This pipeline has all the test cases delivered as part of EIR ATS - 25.1.200 release.
- **EIR-Regression:** This pipeline has all the EIR test cases of previous releases.

4.7.6 Running SLF Pipelines

This section describes how to run SLF related pipelines. It includes:

- Running SLF-NewFeatures Pipeline
- Running SLF Regression Pipeline - Single Site Deployment
- Running SLF Regression Pipeline - Multiple Site Deployment
- Running SLF HealthCheck Pipeline
- Running SLF-Performance Pipeline: Model 2: Single Site Deployment
- SLF NewFeatures Documentation - Single Site Scenarios

Copyright © 2019, 2025, Oracle and/or its affiliates.

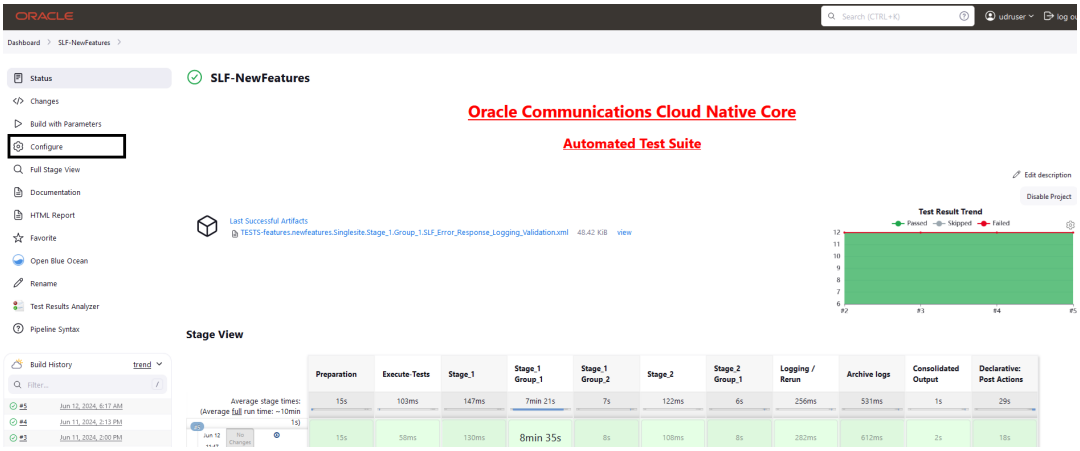
- SLF Regression Documentation - Single Site Scenarios
- SLF Regression Documentation - Multiple Site Scenarios

4.7.6.1 Running SLF-NewFeatures Pipeline

To run the test cases:

1. Click **SLF-NewFeatures**. The following page appears:

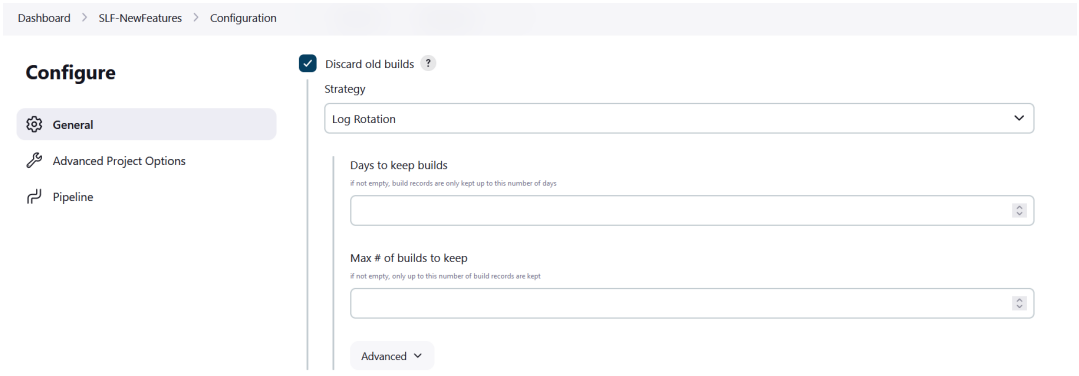
Figure 4-81 SLF-NewFeatures Configure



2. Click **Configure** in the left navigation pane. The **General** tab appears. You must wait for the page to load completely.
3. Select the **Discard old builds**. This option allows you to configure the number of builds you want to retain in the persistent volume.

> **Note**
>
> It is recommended to configure this option. If there is large number of builds, the Persistent Volume may be utilized completely.

Figure 4-82 Discard Old Builds

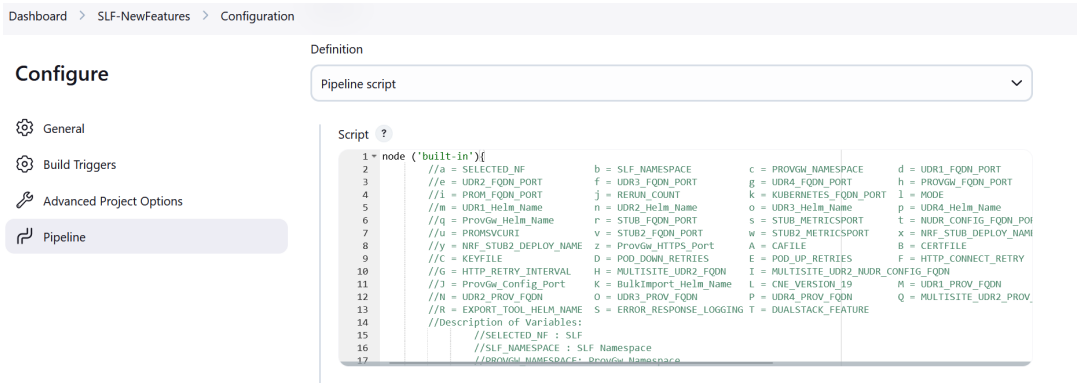


4. Click the **Advanced Project Options** tab. Scroll down to reach the **Pipeline** configuration section as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Important**
>
> Make sure the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

Figure 4-83 SLF Configuration page



In the pipeline script, do not change any value other than from line number **68 to line 112**. You can change the parameters marked as **"b" - to - "T"** as per user requirement. The parameters details are provided as comments in line number **15 to 60**. The parameters description is as follows:

- a - Selected NF
- b - NameSpace in which UDR is Deployed (Mandatory)
- c - NameSpace in which ProvGw is Deployed (Mandatory)
- d - Preferred UDR1 FQDN and PORT of segment 1 in the format FQDN:PORT
- e - UDR2 FQDN and PORT of segment 1 in the format FQDN:PORT
- f - Preferred UDR3 FQDN and PORT of segment 2 in the format FQDN:PORT
- g - UDR4 FQDN and PORT of segment 2 in the format FQDN:PORT
- h - ProvGw FQDN and PORT in the format FQDN:PORT
- i - prometheus_ip and port of prometheus server in the format FQDN:PORT
- j - re-run count
- k - Name and Port of Kubernetes Host server in the format FQDN:PORT
- l - Mode of Communication (Can be either IP or fqdn)
- m - Helm Name for UDR1
- n - Helm Name for UDR2
- o - Helm Name for UDR3
- p - Helm Name for UDR4
- q - Helm Name for ProvGw
- r - FQDN:PORT format for primary nrf-stub
- s - primary nrf-stub metrics port

Copyright © 2019, 2025, Oracle and/or its affiliates.

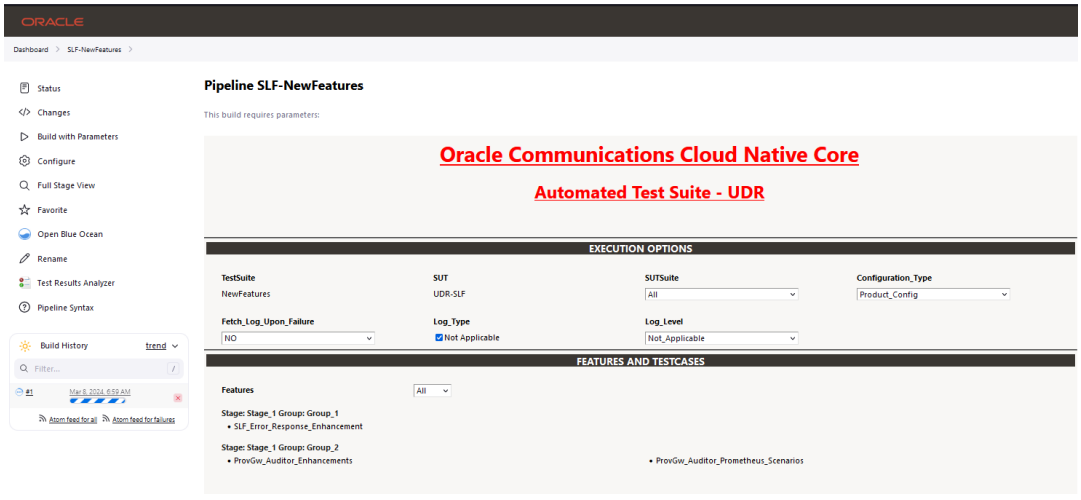
- t - nudr-config FQDN:PORT format
- u - prometheus uri apiName and apiVersion
- v - FQDN:PORT format for secondary nrf-stub
- w - secondary nrf-stub metrics port
- x - primary nrf-stub deployment name
- y - secondary nrf-stub deployment name
- z - ProvGw HTTPS port
- A - root certificate authority
- B - Server certificate signed with root CA private key
- C - private key in .pem format
- D - Number of retries to check if pod down is successful
- E - Number of retries to check if pod up is successful
- F - Number of retries when ATS detects tcp connection failures
- G - Time interval in seconds between tcp connection retries
- H - UDR2 FQDN:PORT for multi-site scenarios
- I - UDR2 nudr-config FQDN:PORT for multi-site scenarios
- J - ProvGateway Config service host and port in the format FQDN:PORT
- K - Bulk import helm name
- L - true if CNE VERSION is 1.9 or above or OSO version is 1.10.x or above, else false.
- M - Preferred UDR1 PROV FQDN and PORT of segment 1 in the format FQDN:PORT
- N - UDR2 PROV FQDN and PORT of segment 1 in the format FQDN:PORT
- O - Preferred UDR3 PROV FQDN and PORT of segment 2 in the format FQDN:PORT
- P - UDR4 PROV FQDN and PORT of segment 2 in the format FQDN:PORT
- Q - MultiSite UDR2 PROV FQDN and PORT in the format FQDN:PORT
- R - Helm Release name of SLF export tool (slfexport)
- S - Enable Error Response Logging Validation (true)
- T - true if underlying platform supports dual stack, else false. All dual stack scenarios will be marked as skipped if false (Default is false)

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> Setting rerun count, number of retries to check pod status and segment details:
>
> - Rerun count can be set to any desired value by changing option 'j' in the pipeline script.
> - Number of retries to check if pod is down can be set by providing option 'D' in the pipeline script.
> - Number of retries to check if pod is up can be set by providing option 'E' in the pipeline script.
> - If ProvGw is deployed with segment details having UDR fqdn/IP:port, provide option 'l' as "IP". If ProvGw is deployed with segment details having only UDR fqdn/IP, provide option 'l' as "fqdn". If any value other than "IP" and "fqdn" is provided, ATS assumes that the segment details in ProvGw is in "IP" mode.
> - Option 'u' shall take value '/<domain name>/prometheus/api/v1 for CNE version 1.9 or above or OSO version is 1.10.x or above. If domain name is not known, then domain name can be found after executing the command from ATS pod:
>
> ```
> curl http://<prometheus fqdn>:<port>/
> ```
>
> Example:
>
> ```
> curl http://occne-kube-prom-stack-kube-prometheus.occne-infra/
> <a href="/udr-cne/prometheus">Found</a>
> ```

5. Click **Save**. The SLF-NewFeatures page appears.
6. Click **Build with Parameters**. The following page displays all the newly added features and total count of the features, counts of scenarios in each feature, name of each feature and scenarios in each feature:

Figure 4-84 SLF Build with Parameters



In the previous image,

Copyright © 2019, 2025, Oracle and/or its affiliates.

- Select the **Configuration_Type**. For information about this feature, see Parameterization.
- There are two **Select_Option**(s), which are:
  - **All:** By default, it selects all the SLF test cases to run. Scroll-down and click **Build** to run all the test cases.
  - **Single/MultipleFeatures:** This option allows you to select the test cases you want to run. After selecting the test cases, scroll-down and click **Build**.

  If the **Select_Option** is selected as 'All', the **TestCases** details are mapped to each feature and if the **Select_Option** is selected as 'Single/MultipleFeatures', only the TestCases details corresponding to those test cases appear. For more information, see Support for Test Case Mapping and Count.

To view consolidated and detailed stack trace results in case of any failures, click **Test Results Analyzer** in the left navigation pane. The test results analyzer report appears. For more information, see Test Results Analyzer.

A sample consolidated test report, when rerun is set to 0, is as follows:

Figure 4-85 Consolidated Test Report - Rerun is 0

```
**********************************Stage: Stage_1 Group: Group_1 Result**************************************
Final Result:-
Total-Features RUN 1, PASS 1, FAIL 0

[Pipeline] }
[Pipeline] // catchError
[Pipeline] catchError
[Pipeline] {
[Pipeline] sh (Stage: Stage_1  Group: Group_2 Result)
+ set +x
**********************************Stage: Stage_1 Group: Group_2 Result**************************************
Final Result:-
Total-Features RUN 2, PASS 2, FAIL 0

[Pipeline] }
[Pipeline] // catchError
[Pipeline] catchError
[Pipeline] {
[Pipeline] sh (Stage: Stage_2  Group: Group_1 Result)
+ set +x
**********************************Stage: Stage_2 Group: Group_1 Result**************************************
Final Result:-
Total-Features RUN 0, PASS 0, FAIL 0

[Pipeline] }
[Pipeline] // catchError
[Pipeline] sh (Overall Result)
+ set +x
**********************************Overall Result**************************************

Took 1h21m21s

Final Result:-
Total-Features RUN 3, PASS 3, FAIL 0
```

For more details on consolidated test report, see Final Summary Report, Build Color and Application Logs.

Copyright © 2019, 2025, Oracle and/or its affiliates.

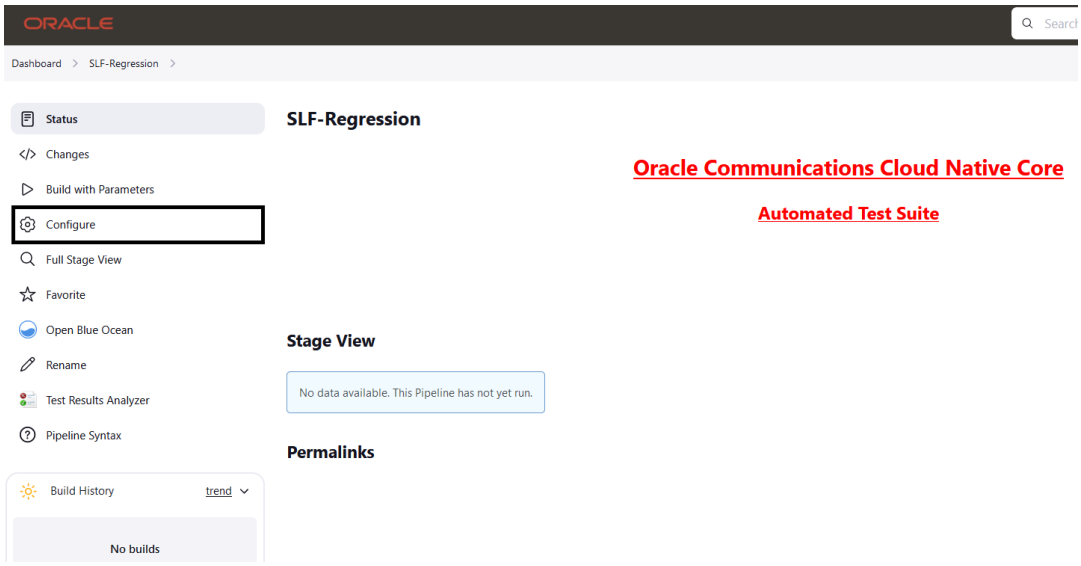
4.7.6.2 Running SLF Regression Pipeline - Single Site Deployment

This pre-configured pipeline consists of all the test cases delivered till SLF ATS 25.1.200. However, some test cases are updated as per new implementation of SLF. To run the single site scenarios:

> **Note**
>
> Model 1 deployment is required for this pipeline. For more information, see Prerequisites for SLF Pipelines.

1. Click **SLF-Regression** and then, click **Configure**.

Figure 4-86 SLF-Regression Pipeline



2. The **General** tab appears. The user must wait for the page to load completely.
3. Select the **Discard old Builds**. This option allows you to configure the number of builds you want to retain in the persistent volume.

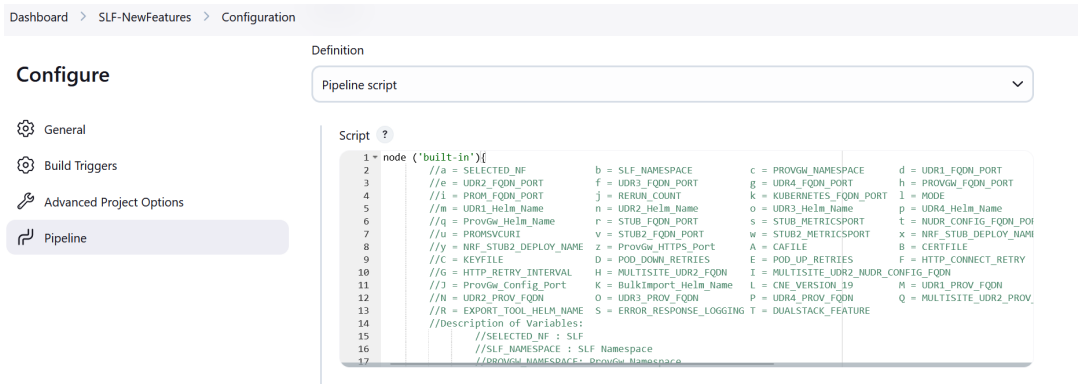
> **Note**
>
> It is recommended to configure this option. If there is a large number of builds, the Persistent Volume may be utilized completely.

4. Click the **Advanced Project Options** tab. Scroll-down to reach the **Pipeline** configuration section as follows:

> **Important**
>
> Make sure the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-87 SLF Configuration page



Do not change any value other than from line number **68 to 112**. You can change the parameters marked as **"b" - to - "T"** as per user requirement. The parameters details are provided as comments in line number **15 to 60**. The parameters description is as follows:

- a - Selected NF
- b - NameSpace in which UDR is Deployed (Mandatory)
- c - NameSpace in which ProvGw is Deployed (Mandatory)
- d - Preferred UDR1 FQDN and PORT of segment 1 in the format FQDN:PORT
- e - UDR2 FQDN and PORT of segment 1 in the format FQDN:PORT
- f - Preferred UDR3 FQDN and PORT of segment 2 in the format FQDN:PORT
- g - UDR4 FQDN and PORT of segment 2 in the format FQDN:PORT
- h - ProvGw FQDN and PORT in the format FQDN:PORT
- i - prometheus_ip and port of prometheus server in the format FQDN:PORT
- j - re-run count
- k - Name and Port of Kubernetes Host server in the format FQDN:PORT
- l - Mode of Communication (Can be either IP or fqdn)
- m - Helm Name for UDR1
- n - Helm Name for UDR2
- o - Helm Name for UDR3
- p - Helm Name for UDR4
- q - Helm Name for ProvGw
- r - FQDN:PORT format for primary nrf-stub
- s - primary nrf-stub metrics port
- t - nudr-config FQDN:PORT format
- u - prometheus uri apiName and apiVersion
- v - FQDN:PORT format for secondary nrf-stub
- w - secondary nrf-stub metrics port
- x - primary nrf-stub deployment name

Copyright © 2019, 2025, Oracle and/or its affiliates.

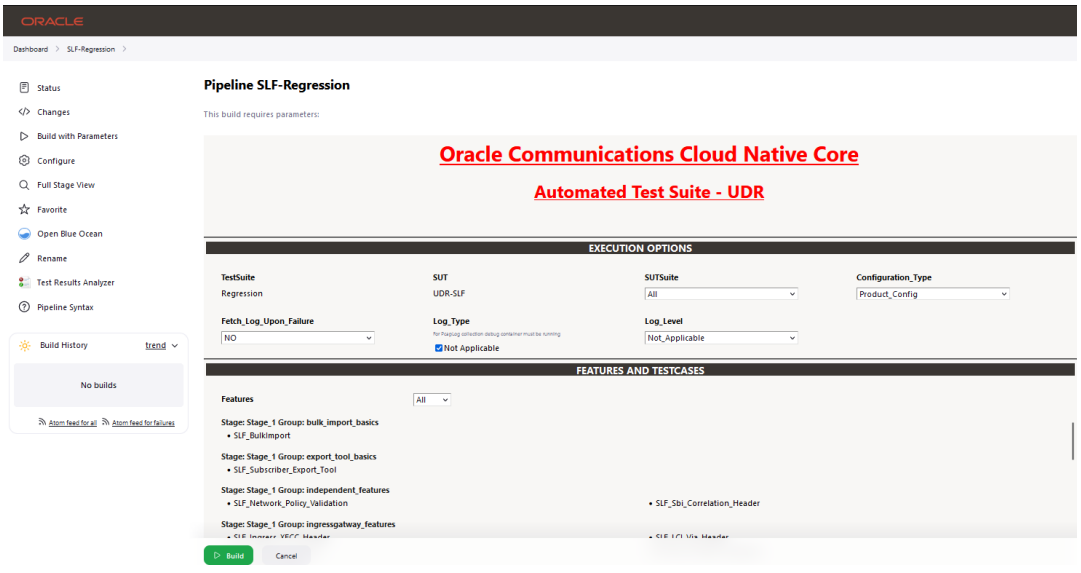
- y - secondary nrf-stub deployment name
- z - ProvGw HTTPS port
- A - root certificate authority
- B - Server certificate signed with root CA private key
- C - private key in .pem format
- D - Number of retries to check if pod down is successful
- E - Number of retries to check if pod up is successful
- F - Number of retries when ATS detects tcp connection failures
- G - Time interval in seconds between tcp connection retries
- H - UDR2 FQDN:PORT for multi-site scenarios
- I - UDR2 nudr-config FQDN:PORT for multi-site scenarios
- J - ProvGateway Config service host and port in the format FQDN:PORT
- K - Bulk import helm name
- L - true if CNE VERSION is 1.9 or above or OSO version is 1.10.x or above, else false.
- M - Preferred UDR1 PROV FQDN and PORT of segment 1 in the format FQDN:PORT
- N - UDR2 PROV FQDN and PORT of segment 1 in the format FQDN:PORT
- O - Preferred UDR3 PROV FQDN and PORT of segment 2 in the format FQDN:PORT
- P - UDR4 PROV FQDN and PORT of segment 2 in the format FQDN:PORT
- Q - MultiSite UDR2 PROV FQDN and PORT in the format FQDN:PORT
- R - Helm Release name of SLF export tool (slfexport)
- S - Enable Error Response Logging Validation (true)
- T - true if underlying platform supports duals tack, else false. All dual stack scenarios will be marked as skipped if false (Default is false)

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> Setting rerun count, number of retries to check pod status and segment details:
>
> - Rerun count can be set to any desired value by changing option 'j' in the pipeline script.
> - Number of retries to check if pod is down can be set by providing option 'D' in the pipeline script.
> - Number of retries to check if pod is up can be set by providing option 'E' in the pipeline script.
> - If ProvGw is deployed with segment details having UDR fqdn/IP:port, provide option 'l' as "IP". If ProvGw is deployed with segment details having only UDR fqdn/IP, provide option 'l' as "fqdn". If any value other than "IP" and "fqdn" is provided, ATS assumes that the segment details in ProvGw is in "IP" mode.
> - Option 'u' shall take value '/<domain name>/prometheus/api/v1 for CNE version 1.9 or above or OSO version is 1.10.x or above. If domain name is not known, then domain name can be found after executing the command from ATS pod:
>
>   ```
>   curl http://<prometheus fqdn>:<port>/
>   ```
>
>   Example:
>
>   ```
>   curl http://occne-kube-prom-stack-kube-prometheus.occne-infra/
>   <a href="/udr-cne-190/prometheus">Found</a>
>   ```

5. Click **Save**. The **Pipeline SLF-Regression** page appears.
6. Click **Build with Parameters**.

Figure 4-88 Build with Parameters



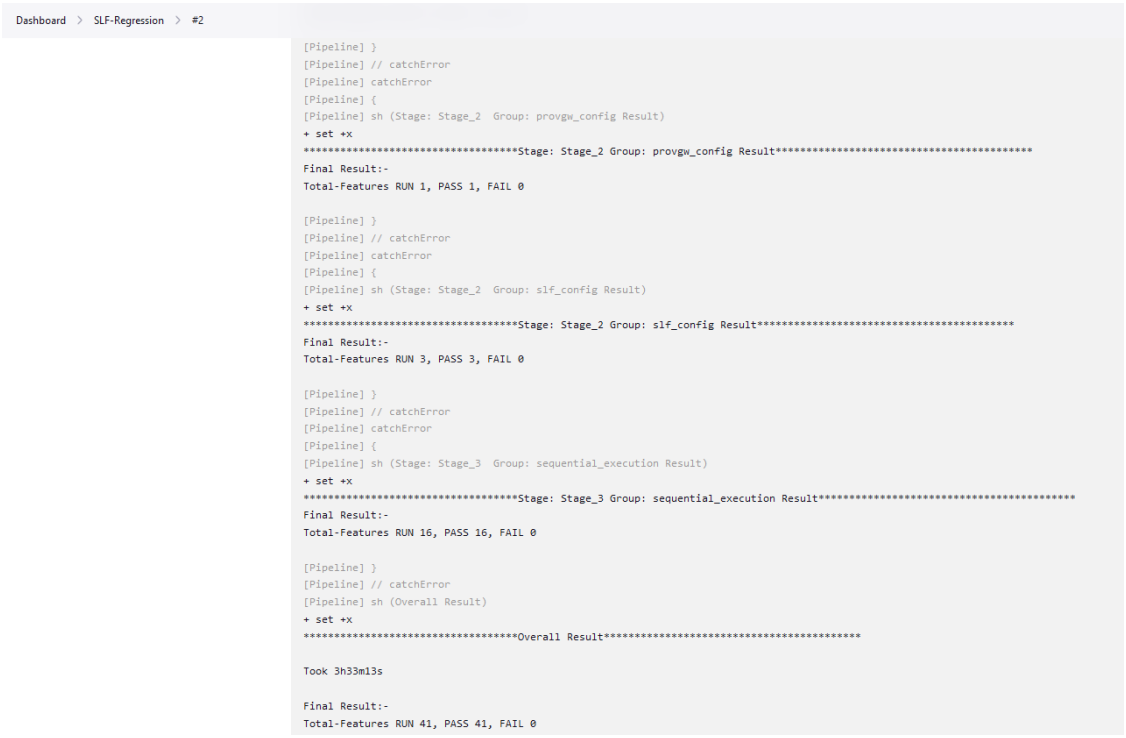
7. Select the Configuration_Type. For more information about this feature, see Parameterization.

Copyright © 2019, 2025, Oracle and/or its affiliates.

8. Select **SUTSuite** as 'All'. This lists all regression features and displays the total count of the features, counts of scenarios in each feature, name of each feature, and scenarios in each feature.
9. Click **Build**.
   For more details on consolidated test report, see Final Summary Report, Build Color and Application Logs.

A sample consolidated test report, when rerun is set to 0, is as follows:

Figure 4-89 Consolidated Test Report Rerun



4.7.6.3 Running SLF Regression Pipeline - Multiple Site Deployment

This pre-configured pipeline consists of all the test cases delivered till SLF ATS 25.1.200. However, some test cases are updated as per new implementation of SLF. To run the multiple site scenarios:

> **Note**
>
> Model 3 deployment is required for this pipeline. For more information, see Prerequisites for SLF Pipelines.

1. Click **SLF-Regression** and then, click **Configure**.
2. The **General** tab appears. The user must wait for the page to load completely.
3. Select the **Discard old Builds**. This option allows you to configure the number of builds you want to retain in the persistent volume.

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> It is recommended to configure this option. If there are large number of builds, the Persistent Volume may be utilized completely.

4. Click the **Advanced Project Options** tab. Scroll-down to reach the **Pipeline** configuration section as follows:

> **Important**
>
> Make sure the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

Do not change any value other than from **line number 68 to line 112**. You can change the parameters marked as **"b" - to - "T"** as per user requirement. The parameters details are provided as comments in **line number 15 to 60**. The parameters description is as follows:

- a - Selected NF
- b - NameSpace in which UDR is Deployed (Mandatory)
- c - NameSpace in which ProvGw is Deployed (Mandatory)
- d - Preferred UDR1 FQDN and PORT of segment 1 in the format FQDN:PORT
- e - UDR2 FQDN and PORT of segment 1 in the format FQDN:PORT
- f - Preferred UDR3 FQDN and PORT of segment 2 in the format FQDN:PORT
- g - UDR4 FQDN and PORT of segment 2 in the format FQDN:PORT
- h - ProvGw FQDN and PORT in the format FQDN:PORT
- i - prometheus_ip and port of prometheus server in the format FQDN:PORT
- j - re-run count
- k - Name and Port of Kubernetes Host server in the format FQDN:PORT
- l - Mode of Communication (Can be either IP or fqdn)
- m - Helm Name for UDR1
- n - Helm Name for UDR2
- o - Helm Name for UDR3
- p - Helm Name for UDR4
- q - Helm Name for ProvGw
- r - FQDN:PORT format for primary nrf-stub
- s - primary nrf-stub metrics port
- t - nudr-config FQDN:PORT format
- u - prometheus uri apiName and apiVersion
- v - FQDN:PORT format for secondary nrf-stub
- w - secondary nrf-stub metrics port
- x - primary nrf-stub deployment name
- y - secondary nrf-stub deployment name

Copyright © 2019, 2025, Oracle and/or its affiliates.

- z - ProvGw HTTPS port
- A - root certificate authority
- B - Server certificate signed with root CA private key
- C - private key in .pem format
- D - Number of retries to check if pod down is successful
- E - Number of retries to check if pod up is successful
- F - Number of retries when ATS detects tcp connection failures
- G - Time interval in seconds between tcp connection retries
- H - UDR2 FQDN:PORT for multi-site scenarios
- I - UDR2 nudr-config FQDN:PORT for multi-site scenarios
- J - ProvGateway Config service host and port in the format FQDN:PORT
- K - Bulk import helm name
- L - true if CNE VERSION is 1.9 or above or OSO version is 1.10.x or above, else false.
- M - Preferred UDR1 PROV FQDN and PORT of segment 1 in the format FQDN:PORT
- N - UDR2 PROV FQDN and PORT of segment 1 in the format FQDN:PORT
- O - Preferred UDR3 PROV FQDN and PORT of segment 2 in the format FQDN:PORT
- P - UDR4 PROV FQDN and PORT of segment 2 in the format FQDN:PORT
- Q - MultiSite UDR2 PROV FQDN and PORT in the format FQDN:PORT
- R - Helm Release name of SLF export tool (slfexport)
- S - Enable Error Response Logging Validation (true)
- T - true if underlying platform supports dual stack, else false. All dual stack scenarios will be marked as skipped if false (Default is false)

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> Setting rerun count, number of retries to check pod status, and segment details:
>
> - Rerun count can be set to any desired value by changing option 'j' in the pipeline script.
> - Number of retries to check if pod is down can be set by providing option 'D' in the pipeline script.
> - Number of retries to check if pod is up can be set by providing option 'E' in the pipeline script.
> - If ProvGw is deployed with segment details having UDR fqdn/IP:port, provide option 'l' as "IP". If ProvGw is deployed with segment details having only UDR fqdn/IP, provide option 'l' as "fqdn". If any value other than "IP" and "fqdn" is provided, ATS assumes that the segment details in ProvGw is in "IP" mode.
> - Option 'u' shall take value '/<domain name>/prometheus/api/v1 for CNE version 1.9 or above or OSO version is 1.10.x or above. If domain name is not known, then domain name can be found after executing the command from ATS pod:
>
> ```
> curl http://<prometheus fqdn>:<port>/
>
> curl http://occne-kube-prom-stack-kube-prometheus.occne-infra/
> <a href="/udr-cne-190/prometheus">Found</a>
> ```

5. Click **Save**. The **Pipeline SLF-Regression** page appears.
6. Click **Build with Parameters**. Select the **Configuration_Type**.

> **Note**
>
> For information about the Configuration_Type, see Parameterization.

7. Select **SUTSuite** as 'Multi_Site'. This lists all the regression features and displays the total count of the features, counts of scenarios in each feature, name of each feature, and scenarios in each feature.
8. Click **Build**.
   A sample consolidated test report, when rerun is set to 0 is as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-90 SLF- Regression to run Multiple Site Scenarios - Consolidated Test Report - Rerun is 0

```
3 features passed, 0 failed, 0 skipped
21 scenarios passed, 0 failed, 0 skipped
794 steps passed, 0 failed, 0 skipped, 0 undefined
Took 17m58.722s
[Pipeline] sh
+ cd /var/lib/jenkins/ocslf_tests
++ cat /var/lib/jenkins/ocslf_tests/environ.sh
++ grep RERUN
++ cut -d= -f2
++ cut '-d;' -f1
+ rerun=0
+ [[ 0 -ne 0 ]]
+ sh re-run.sh 0
1
Scenario passed
```

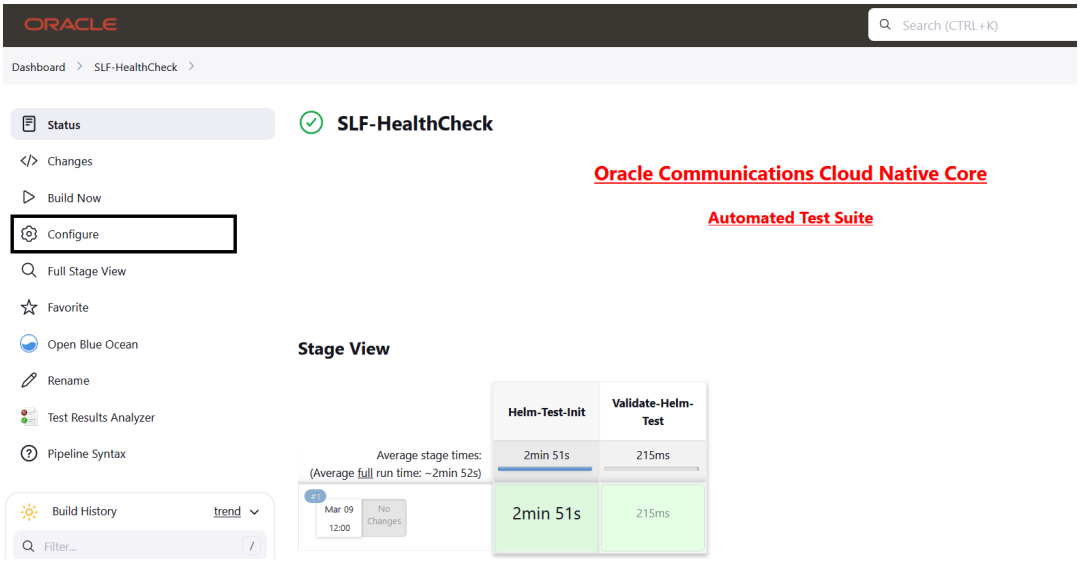
For more details on consolidated test report, see Final Summary Report, Build Color and Application Logs.

4.7.6.4 Running SLF HealthCheck Pipeline

To run health check on SLF or Provisioning Gateway:

1. Navigate to **SLF-HealthCheck** pipeline and click **Configure**.

Figure 4-91 UDR - SLF-HealthCheck Pipeline



2. The **General** tab appears. The user must wait for the page to load completely.
3. Click the **Advanced Project Options** tab. Scroll-down to reach the **Pipeline** configuration section as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Important**
>
> Make sure the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

Do not change any value other than **line numbers 16 and 17**. You can change the parameters marked as **"a" "b"** and **"c"** as per user requirement. The parameters details are provided as comments in **line number 7 and 9**. The parameters description is as follows:

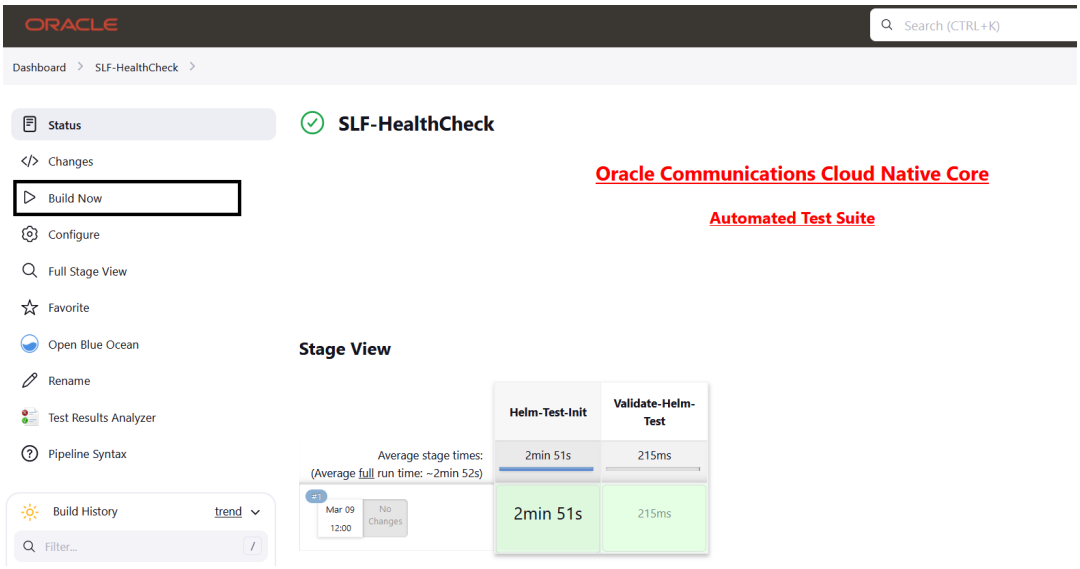
- a - helm releases [Provide Release Name with Comma Separated if more than 1]
- b - namespace of SUT.
- c - helm command alias that is being used on the CNE (helm or helm2 or helm3)

Figure 4-92 SLF-HealthCheck Pipeline configuration



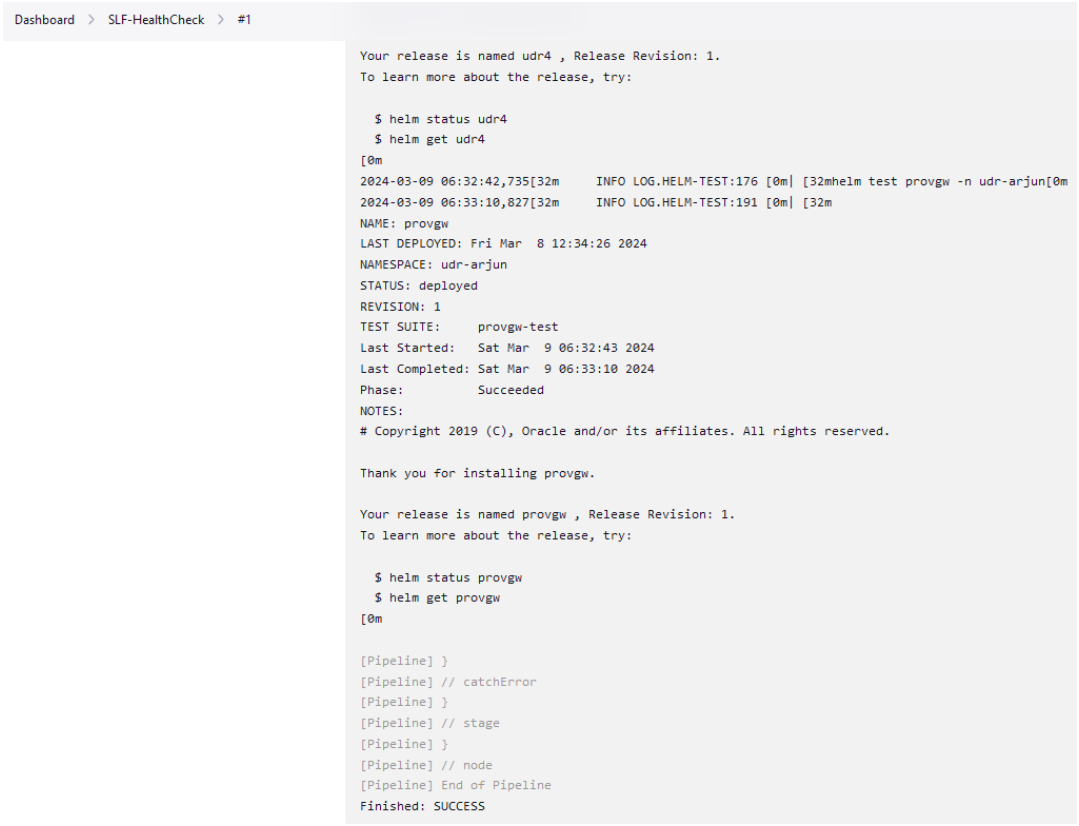
4. Click **Save**. The **Pipeline SLF-HealthCheck** page appears.
5. Click **Build Now**. This triggers health check for SUT.

Figure 4-93 SLF-HealthCheck Pipeline Build



Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-94 Sample Output - SLF Health Check



4.7.6.5 Running SLF-Performance Pipeline: Model 2: Single Site Deployment

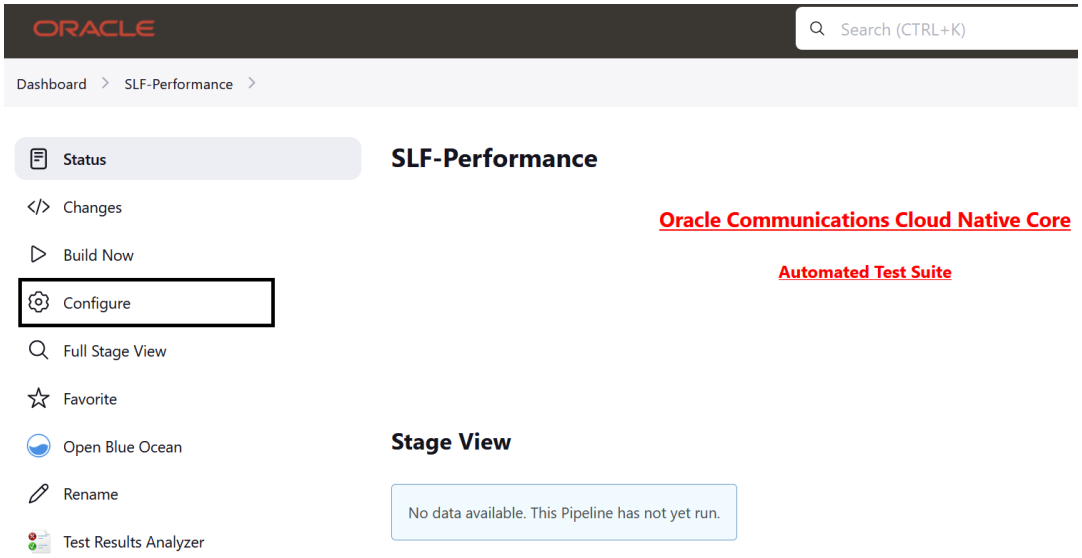
To run light weight performance scenarios:

> **Note**
>
> Model 2 deployment is required for this pipeline. For more information, see Prerequisites for SLF Pipelines.

1. Navigate to the **SLF-Performance** pipeline and click **Configure**.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-95 SLF Single Site Performance Model 2



2. The **General** tab appears. The user must wait for the page to load completely.
3. Click the **Advanced Project Options** tab. Scroll-down to reach the **Pipeline** configuration section as follows:

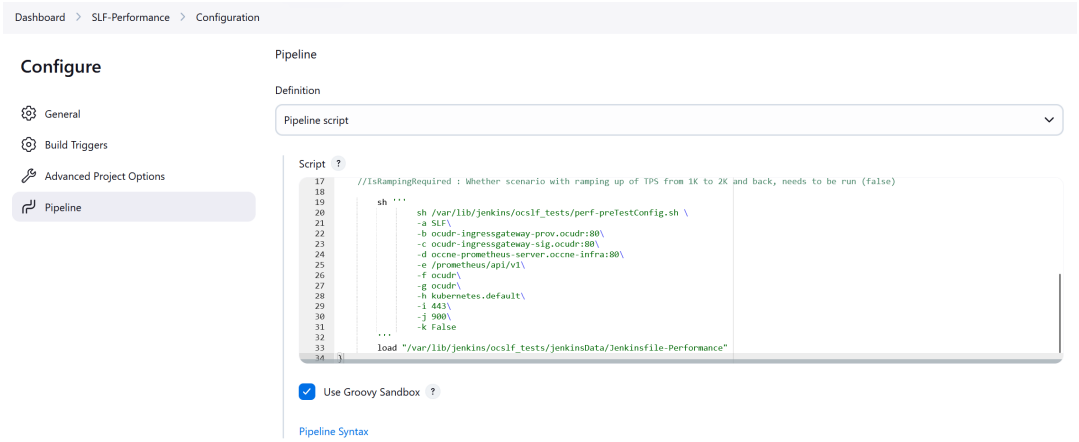
> **Important**
>
> Make sure the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

Do not change any value other than line number **22 and line number 31**. You can change the parameters marked as "b" and "k" as per user requirement. The parameters details are provided as comments in line number 7 and 21. The parameters description is as follows:

- a - Selected NF (should always be SLF)
- b - UDR ingressgateway provisioning FQDN and port for performance testing (ocudr-ingressgateway-prov.ocudr:80)
- c - UDR ingressgateway signalling FQDN and port for performance testing (ocudr-ingressgateway-sig.ocudr:80)
- d - Prometheus Server FQDN and port (occne-prometheus-server.occne-infra:80)
- e - Prometheus URI (/prometheus/api/v1)
- f - Namespace in which UDR is deployed for performance testing (ocudr)
- g - Helm Name of UDR (ocudr)
- h - Kubernetes host FQDN (kubernetes.default)
- i - Kubernetes host port number (443)
- j - Duration for which performance needs to be tested (in seconds) (900)
- k - Whether scenario with ramping up of TPS from 1K to 2K and back, needs to be run (false)

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-96 SLF-Performance Pipeline Configuration



4. Click **Save**. The **Pipeline SLF-Performance** page appears.
5. Click **Build Now**. This triggers lightweight traffic on UDR-SLF.

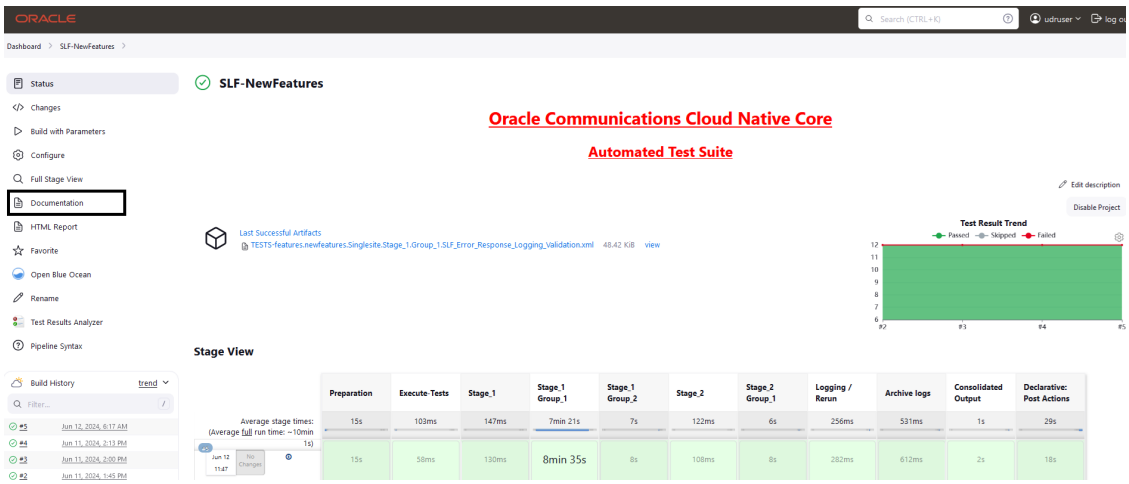
Figure 4-97 Sample Output - SLF Performance

```
1 feature passed, 0 failed, 0 skipped
1 scenario passed, 0 failed, 1 skipped
30 steps passed, 0 failed, 29 skipped, 0 undefined
Took 21m47.747s
```

4.7.6.6 SLF NewFeatures Documentation - Single Site Scenarios

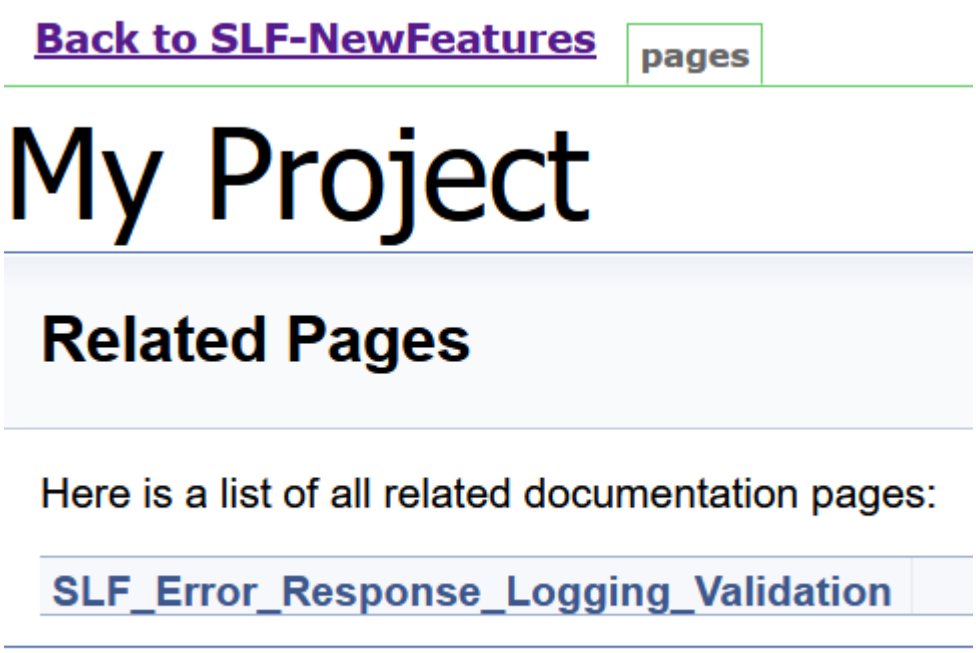
To view the documentation of SLF-NewFeatures, click the **Documentation** link in the left navigation pane (present inside the build), as follows:

Figure 4-98 SLF-New Features Documentation



The page lists all the features. Click any feature to view description of that test case.

Copyright © 2019, 2025, Oracle and/or its affiliates.



The SLF new features list along with its description is as follows:

Table 4-14 SLF New Features

Feature	Description
SLF_Subscriber_Export_File_Cleanup.feature	This feature file has scenario to validate the clean up of the older SLF export dumps.

> **Note**
>
> The Documentation option appears only if you have run the SLF-NewFeatures pipeline test cases atleast once.

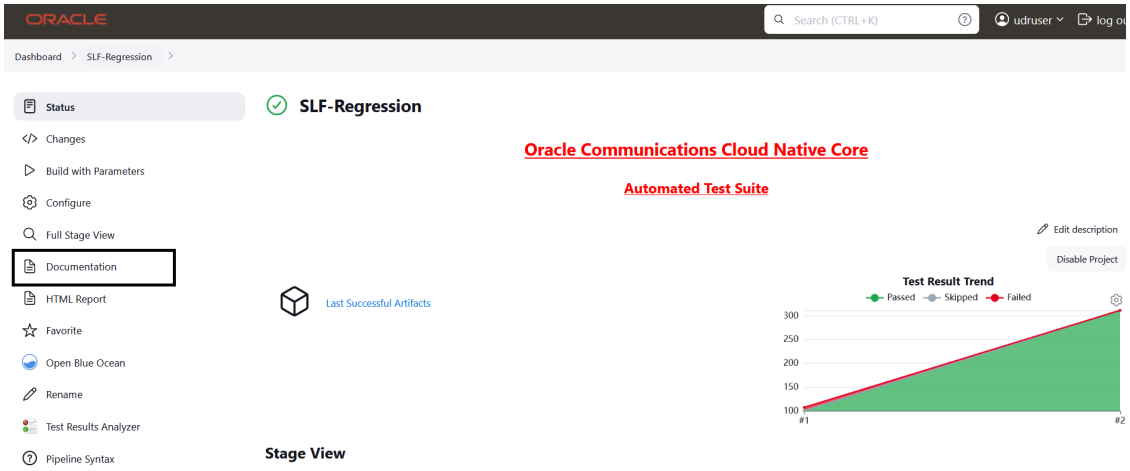
Based on the functionalities covered under Documentation, the **Build Requires Parameters** page displays test cases. To navigate back to the Pipeline SLF-NewFeatures page, click **Back to SLF-NewFeatures** link available on top left corner of the page.

4.7.6.7 SLF Regression Documentation - Single Site Scenarios

Click **Documentation** in the left navigation pane of the SLF-Regression pipeline to view all the test cases.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-99 SLF-Regression Documentation

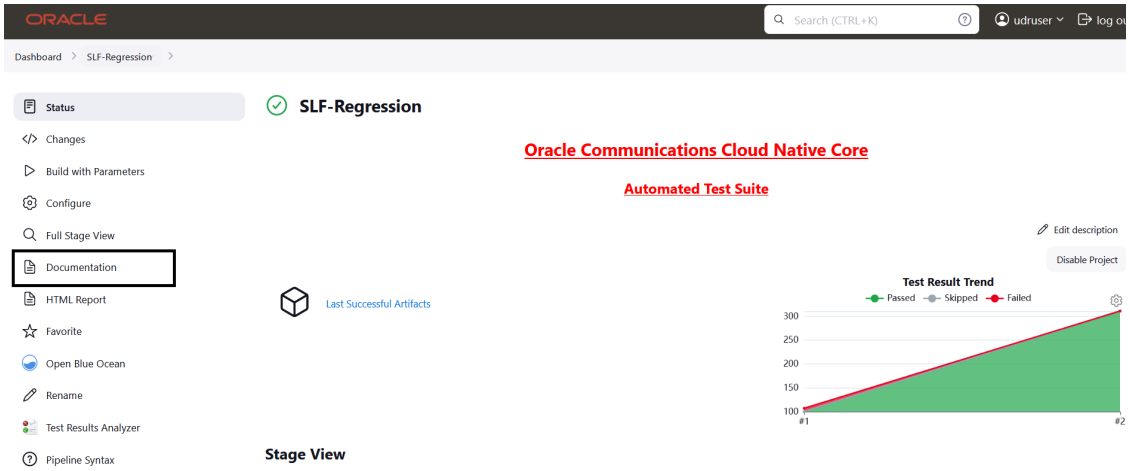


Click any feature to see the description of that test case.

4.7.6.8 SLF Regression Documentation - Multiple Site Scenarios

Click **Documentation** in the left navigation pane of the SLF-Regression pipeline to view all the test cases provided.

Figure 4-100 SLF-Regression Documentation



Click any feature to view its details.

4.7.7 Running UDR Pipelines

This section describes how to run UDR related pipelines. It includes:

- Running UDR NewFeatures Pipeline
- Running UDR Regression Pipeline

Copyright © 2019, 2025, Oracle and/or its affiliates.

- Running UDR HealthCheck Pipeline
- UDR NewFeatures Documentation
- UDR Regression Documentation

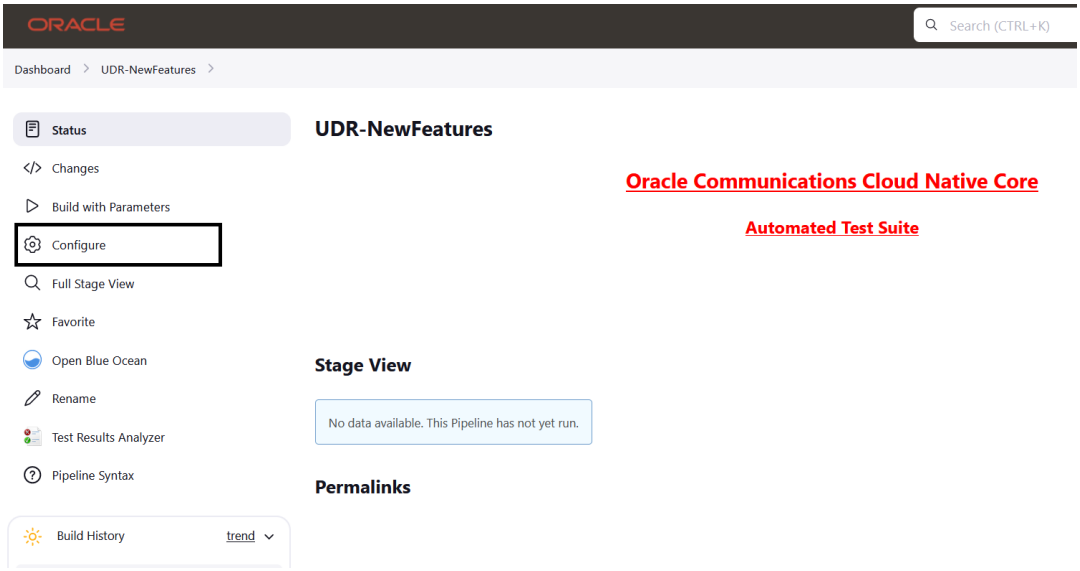
4.7.7.1 Running UDR NewFeatures Pipeline

To run the UDR-NewFeatures (Model 1) pipeline test cases:

Running UDR-NewFeatures (Model 1) pipeline test cases

1. Click **UDR-NewFeatures**. Click **Configure** in the left navigation pane. The **General** tab appears. The user must wait for the page to load completely.

Figure 4-101 UDR NewFeatures Pipeline



2. Once the page loads completely, select the **Discard old Builds**. This option allows you to configure the number of builds you want to retain in the persistent volume.

> **Note**
>
> It is recommended to configure this option. If there is large number of build, the Persistent Volume may be utilized completely.

3. Click the **Advanced Project Options** tab. Scroll-down to reach the **Pipeline** configuration as follows:

> **Important**
>
> Make sure the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

You should not change any other value apart from **line number 68 to 112**. It means you can change the parameters marked as **"b"** - to - **"T"** as per user requirement. The

Copyright © 2019, 2025, Oracle and/or its affiliates.

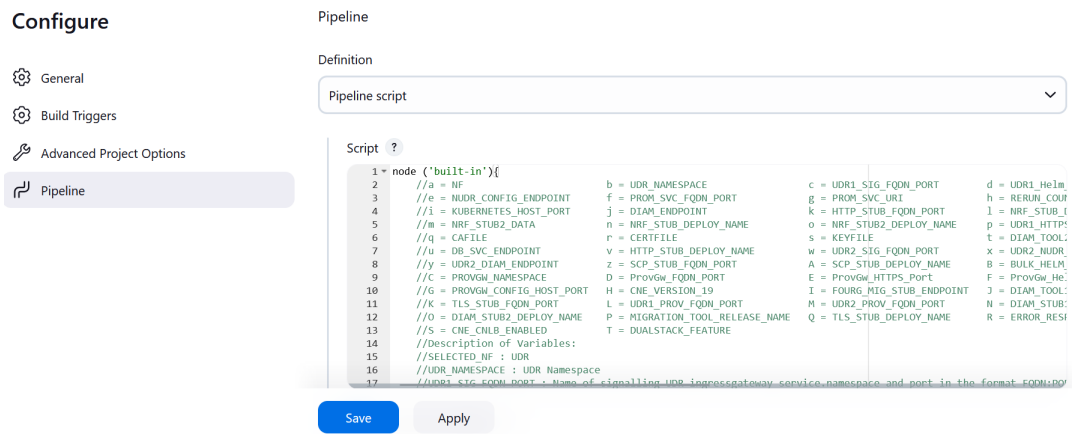
parameters details are provided as comments in **line number 15 to 60**. The parameters description is as follows:

- a - Name of the NF to be tested in capital (UDR).
- b - Namespace in which the udr is deployed.
- c - Name of signalling UDR1_ingressgateway_service.namespace and port (ocudr-ingressgateway-sig.ocudr:80).
- d - Helm release name of UDR (ocudr)
- e - FQDN and Port UDR's nudr-config in the format fqdn:port (ocudr-nudr-config.ocudr:5001)
- f - Name_of_Prometheus_service.namespace and port(occne-prometheus-server.occne-infra:80).
- g - If this option is not provided, Prometheus URI shall be /api/v1 (Default: /prometheus/api/v1) (if CNE 1.9+ /<cne-cluster-name>/prometheus/api/v1)
- h - Number of times the re-run of failed case is allowed (default as 2).
- i - Name of Kubernetes Host server and port (kubernetes.default:443)
- j - diameter endpoint fqdn and port in the form fqdn:port (ocudr-nudr-diam-gateway.ocudr:3868)
- k - fqdn or IP of http2-server in the format <ip/fqdn> (fqdn) (notify-stub-service.ocudr:80)
- l - fqdn or IP for primary nrf stub and port in the format <ip/fqdn>:<port> (nrf-stub-server.ocudr:8080)
- m - fqdn or IP for secondary nrf stub and port in the format <ip/fqdn>:<port> (nrf-stub-server2.ocudr:8080)
- n - primary stub deployment name (nrfstub-ocstub-py)
- o - secondary stub deployment name (nrfstub2-ocstub-py)
- p - HTTPS port for UDR (443)
- q - root certificate authority (CA) (caroot.cer)
- r - Server certificate signed with root CA private key (apigatewayrsa.cer)
- s - private key to .pem format (rsa_private_key_pkcs1.pem)
- t - fqdn and port for 2nd diameter tool endpoint
- u - MYSQL SERVICE ENDPOINT
- v - NOTIFY STUB deployment name
- w - Name of signaling in the format fqdn:port (ocudr-s2-ingressgateway-sig.ocudr-s2:80)
- x - fqdn and port for UDR2 nudr-config (ocudr2-nudr-config.ocudr:5001)
- y - fqdn and port for UDR2 DIAMETER_ENDPOINT (fqdn:port) (ocudr-s2-nudr-diam-gateway.ocudr-s2:3868)
- z - FQDN and port for SCP stub (scp-stub-service:8080)
- A - Deployment name for SCP stub (scpstub-notify-ocstub-py)
- B - Bulk import helm name
- C - ProvGw namespace (ocudr)

Copyright © 2019, 2025, Oracle and/or its affiliates.

- D - ProvGw endpoint in format fqdn:port (provgw-prov-ingressgateway.ocudr:80)
- E - ProvGw endpoint https port(443)
- F - ProvGw helm name
- G - ProvGw config host in the format fqdn:port(provgw-provgw-config.ocudr:5001)
- H - true if OCCNE is 1.9.x and above, else false (true)
- I - fqdn and port for fourg stub in the form (fqdn:port)
- J - fqdn and port for 1st diameter tool endpoint
- K - fqdn and port for notify stub for TLS scenarios (tlsnotify-stub-service.ocudr:8443)
- L - Service name of UDR1 provisioning ingressgateway in the format FQDN:PORT(ocudr-ingressgateway-prov.ocudr:80)
- M - Service name of site2 UDR provisioning ingressgateway in the format FQDN:PORT(ocudr2-ingressgateway-prov.ocudr-s2:80)
- N - Diameter Stub Tool1 Deploy Name (diamtoola-diam-stub)
- O - Diameter Stub Tool2 Deploy Name (diamtoolb-diam-stub)
- P - Migration tool Helm release name (migration-tool)
- Q - Deployment name of the TLS notify stub (tlsnotifystub-ocstub-py)
- R - Enable error response logging validation (true)
- S - true (when underling CNE has support for CNLB Multus) to validate Traffic segregation feature else false
- T - true if underlying platform supports dual stack, else false. All dual stack scenarios will be marked as skipped if false (Default: false)

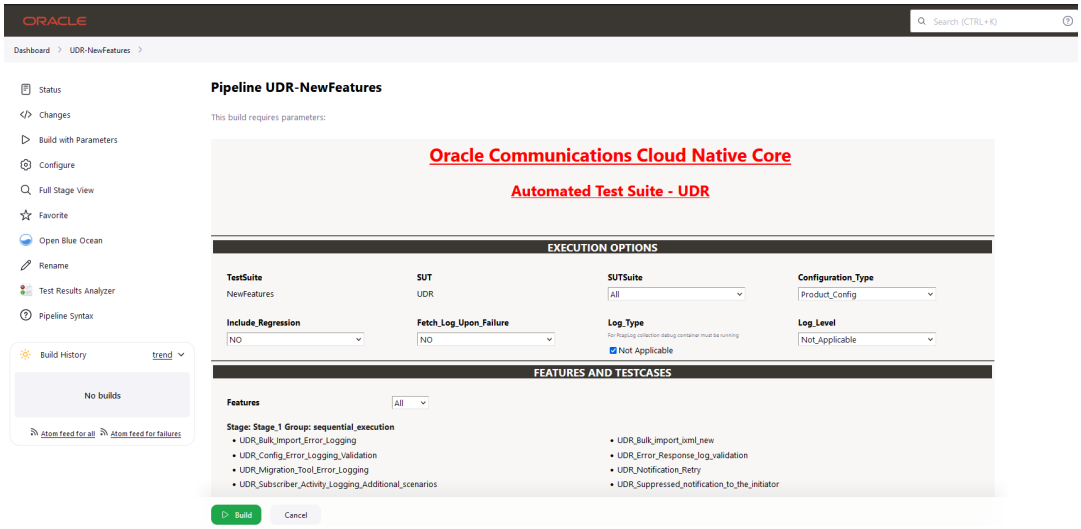
Figure 4-102 UDR NewFeatures Pipeline Configuration



4. Click **Save** after making necessary changes. The **Pipeline UDR-NewFeatures** page appears.
5. Click **Build with Parameters**. The page lists all the newly added features and displays the total number of features and the scenarios in each feature, name of each feature, and scenarios in each feature.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-103 UDR NewFeatures Pipeline Build with Parameter



- Select the Configuration_Type. For more information, see Parameterization.
- Choose the **Select_Option**(s), which are:
  - **All:** By default, it selects all the UDR test cases to run. Scroll-down and click **Build** to run all the test cases.
  - **Single/MultipleFeatures:** This option allows you to select any number of test cases you want to run from the list of total test cases available for execution.

6. After selecting the test cases, scroll-down and click **Build** to run the selected UDR test cases.

Sample output with expected failures are as follows:

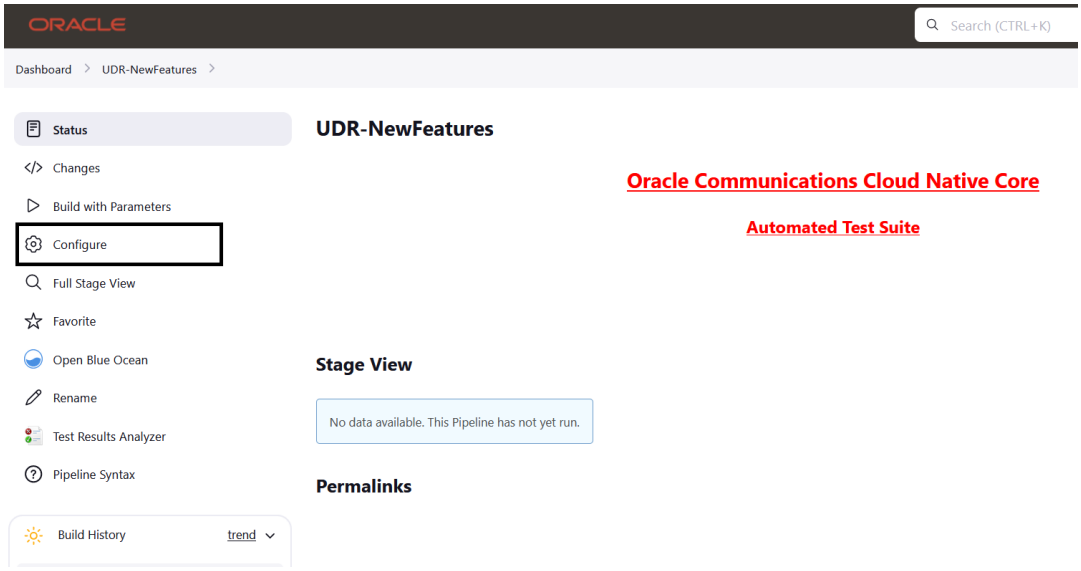
```
Stage:Stage_2 Group:no_igw_dr_metrics
  UDR_IGW_LCI_Header.feature:81  UDR_LCI_03_UserAgent_Err
  UDR_PCF_User_Agent_Header_New.feature:74  UDR_USER_AGENT_HEADER_02
  UDR_PCF_User_Agent_Header_New.feature:140  UDR_USER_AGENT_HEADER_03
```

Running UDR-NewFeatures (Model 2) pipeline test cases

1. Click **UDR-NewFeatures**. Click **Configure** in the left navigation pane. The **General** tab appears. The user must wait for the page to load completely.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-104 UDR NewFeatures Pipeline



2. Once the page loads completely, select the **Discard old Builds**. This option allows you to configure the number of builds you want to retain in the persistent volume.

> **Note**
>
> It is recommended to configure this option. If there is large number of build, the Persistent Volume may be utilized completely.

3. Click the **Advanced Project Options** tab. Scroll-down to reach the **Pipeline** configuration as follows:

> **Important**
>
> Make sure the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

You should not change any other value apart from **line number 65 to 107**. It means you can change the parameters marked as **"b"** - to - **"R"** as per user requirement. The parameters details are provided as comments in **line number 14 to 57**. The parameters description is as follows:

- a - Name of the NF to be tested in capital (UDR).
- b - Namespace in which the udr is deployed.
- c - Name of signalling UDR1_ingressgateway_service.namespace and port (ocudr-ingressgateway-sig.ocudr:80).
- d - Helm release name of UDR (ocudr)
- e - FQDN and Port UDR's nudr-config in the format fqdn:port (ocudr-nudr-config.ocudr:5001)
- f - Name_of_Prometheus_service.namespace and port(occne-prometheus-server.occne-infra:80).

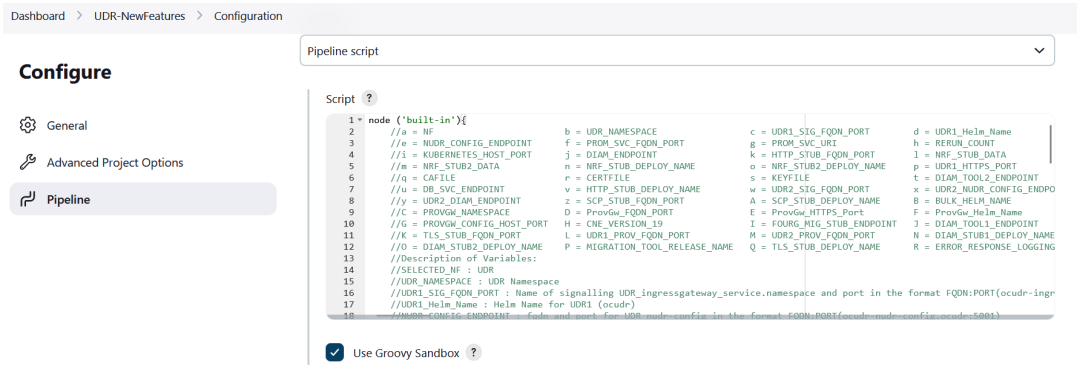
Copyright © 2019, 2025, Oracle and/or its affiliates.

- g - If this option is not provided, Prometheus URI shall be /api/v1 (Default: /prometheus/api/v1) (if CNE 1.9+ /<cne-cluster-name>/prometheus/api/v1)
- h - Number of times the re-run of failed case is allowed (default as 2).
- i - Name of Kubernetes Host server and port (kubernetes.default:443)
- j - diameter endpoint fqdn and port in the form fqdn:port (ocudr-nudr-diam-gateway.ocudr:3868)
- k - fqdn or IP of http2-server in the format <ip/fqdn> (fqdn) (notify-stub-service.ocudr:80)
- l - fqdn or IP for primary nrf stub and port in the format <ip/fqdn>:<port> (nrf-stub-server.ocudr:8080)
- m - fqdn or IP for secondary nrf stub and port in the format <ip/fqdn>:<port> (nrf-stub-server2.ocudr:8080)
- n - primary stub deployment name (nrfstub-ocstub-py)
- o - secondary stub deployment name (nrfstub2-ocstub-py)
- p - HTTPS port for UDR (443)
- q - root certificate authority (CA) (caroot.cer)
- r - Server certificate signed with root CA private key (apigatewayrsa.cer)
- s - private key to .pem format (rsa_private_key_pkcs1.pem)
- t - fqdn and port for 2nd diameter tool endpoint
- u - MYSQL SERVICE ENDPOINT
- v - NOTIFY STUB deployment name
- w - Name of signaling in the format fqdn:port (ocudr-s2-ingressgateway-sig.ocudr-s2:80)
- x - fqdn and port for UDR2 nudr-config (ocudr2-nudr-config.ocudr:5001)
- y - fqdn and port for UDR2 DIAMETER_ENDPOINT (fqdn:port) (ocudr-s2-nudr-diam-gateway.ocudr-s2:3868)
- z - FQDN and port for SCP stub (scp-stub-service:8080)
- A - Deployment name for SCP stub (scpstub-notify-ocstub-py)
- B - Bulk import helm name
- C - ProvGw namespace (ocudr)
- D - ProvGw endpoint in format fqdn:port (provgw-prov-ingressgateway.ocudr:80)
- E - ProvGw endpoint https port(443)
- F - ProvGw helm name
- G - ProvGw config host in the format fqdn:port(provgw-provgw-config.ocudr:5001)
- H - true if OCCNE is 1.9.x and above, else false (true)
- I - fqdn and port for fourg stub in the form (fqdn:port)
- J - fqdn and port for 1st diameter tool endpoint
- K - fqdn and port for notify stub for TLS scenarios (tlsnotify-stub-service.ocudr:8443)
- L - Service name of UDR1 provisioning ingressgateway in the format FQDN:PORT(ocudr-ingressgateway-prov.ocudr:80)

Copyright © 2019, 2025, Oracle and/or its affiliates.

- M - Service name of site2 UDR provisioning ingressgateway in the format FQDN:PORT(ocudr2-ingressgateway-prov.ocudr-s2:80)
- N - Diameter Stub Tool1 Deploy Name (diamtoola-diam-stub)
- O - Diameter Stub Tool2 Deploy Name (diamtoolb-diam-stub)
- P - Migration tool Helm release name (migration-tool)
- Q - Deployment name of the TLS notify stub (tlsnotifystub-ocstub-py)
- R - Enable error response logging validation (true)

Figure 4-105 UDR NewFeatures Pipeline Configuration



4. Click **Save** after making necessary changes. The **Pipeline UDR-NewFeatures** page appears.
5. Click **Build with Parameters**. The page lists all the newly added features and displays the total number of features and the scenarios in each feature, name of each feature, and scenarios in each feature.
   - Select the Configuration_Type. For more information, see Parameterization.
   - Choose the **Select_Option**(s), which are:
     - **All:** By default, it selects all the UDR test cases to run. Scroll-down and click **Build** to run all the test cases.
     - **Single/MultipleFeatures:** This option allows you to select any number of test cases you want to run from the list of total test cases available for execution.
6. After selecting the test cases, scroll-down and click **Build** to run the selected UDR test cases.

4.7.7.2 Running UDR Regression Pipeline

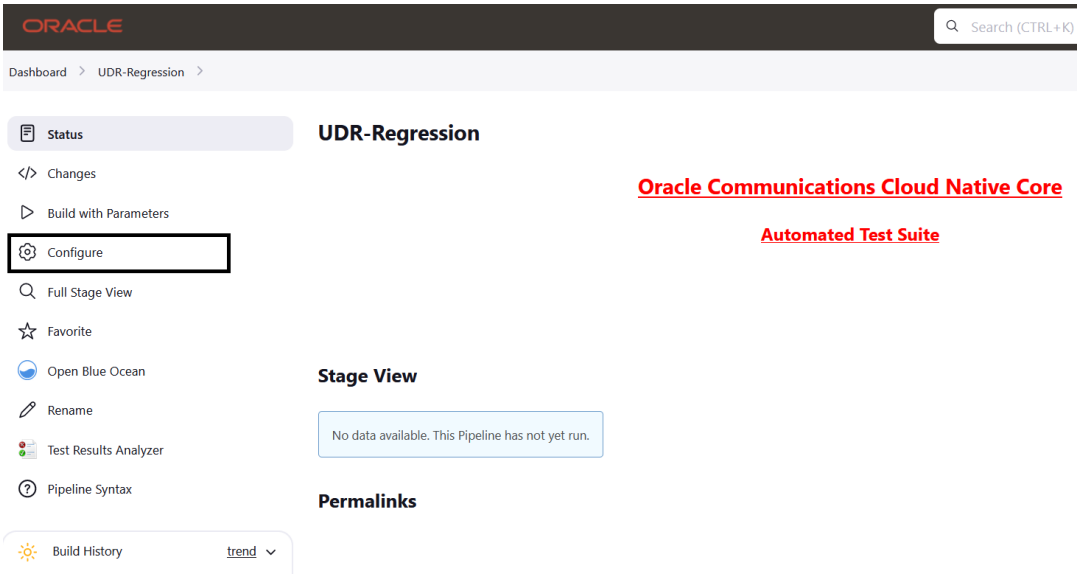
To run the UDR-Regression pipeline test cases:

Running UDR Regression Pipeline: Model 1: Single Site Deployment

1. Click **UDR-Regression** and then, click **Configure**.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-106 UDR Regression Pipeline



2. The **General** tab appears. The user must wait for the page to load completely.
3. Select the **Discard old Builds**. This option allows you to configure the number of builds you want to retain in the persistent volume.

> **Note**
>
> It is recommended to configure this option. If there are large number of builds, the Persistent Volume may be utilized completely.

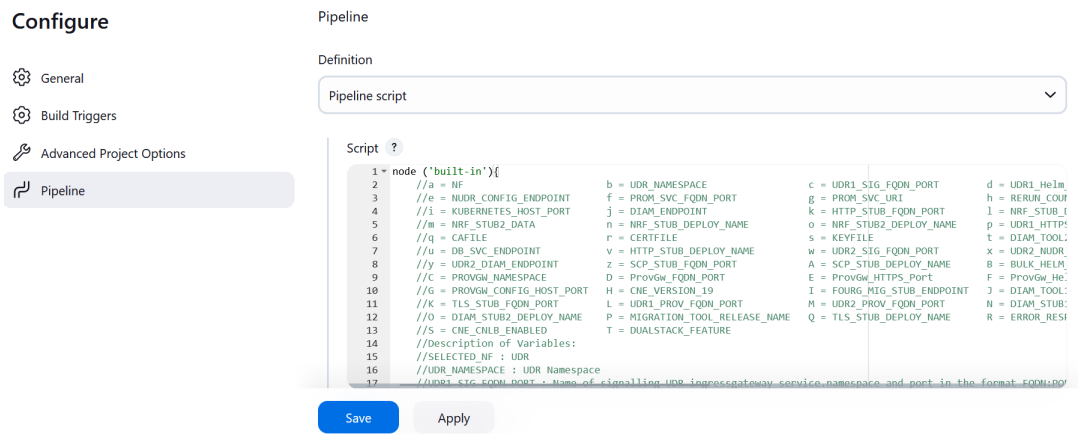
4. Click the **Advanced Project Options** tab. Scroll-down to reach the **Pipeline** configuration section as follows:

> **Important**
>
> Make sure the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

Do not change any value other than from **line number 68 to 112**. You can change the parameters marked as **"b"** - to - **"T"** as per user requirement. The parameters details are provided as comments in **line number 15 to 60**. The parameters description is as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-107 UDR Regression Pipeline Configuration



- a - Name of the NF to be tested in capital (UDR).
- b - Namespace in which the udr is deployed.
- c - Name of signalling UDR1_ingressgateway_service.namespace and port (ocudr-ingressgateway-sig.ocudr:80).
- d - Helm release name of UDR (ocudr)
- e - FQDN and Port UDR's nudr-config in the format fqdn:port (ocudr-nudr-config.ocudr:5001)
- f - Name_of_Prometheus_service.namespace and port(occne-prometheus-server.occne-infra:80).
- g - If this option is not provided, Prometheus URI shall be /api/v1 (Default: /prometheus/api/v1) (if CNE 1.9+ /<cne-cluster-name>/prometheus/api/v1)
- h - Number of times the re-run of failed case is allowed (default as 1).
- i - Name and port of Kubernetes Host server in the format FQDN:PORT(kubernetes.default:443).
- j - fqdn and port for DIAMETER_ENDPOINT in the format FQDN:PORT (ocudr-nudr-diam-gateway:3868)
- k - fqdn or IP of http2-server in the format <ip/fqdn> (fqdn) (notify-stub-service.ocudr:80)
- l - fqdn or IP for primary nrf stub and port in the format <ip/fqdn>:<port> (nrf-stub-server.ocudr:8080)
- m - fqdn or IP for secondary nrf stub and port in the format <ip/fqdn>:<port> (nrf-stub-server2.ocudr:8080)
- n - primary stub deployment name (nrfstub-ocstub-py)
- o - secondary stub deployment name (nrfstub2-ocstub-py)
- p - HTTPS port for UDR (443)
- q - root certificate authority (CA) (caroot.cer)
- r - Server certificate signed with root CA private key (apigatewayrsa.cer)
- s - private key to .pem format (rsa_private_key_pkcs1.pem)

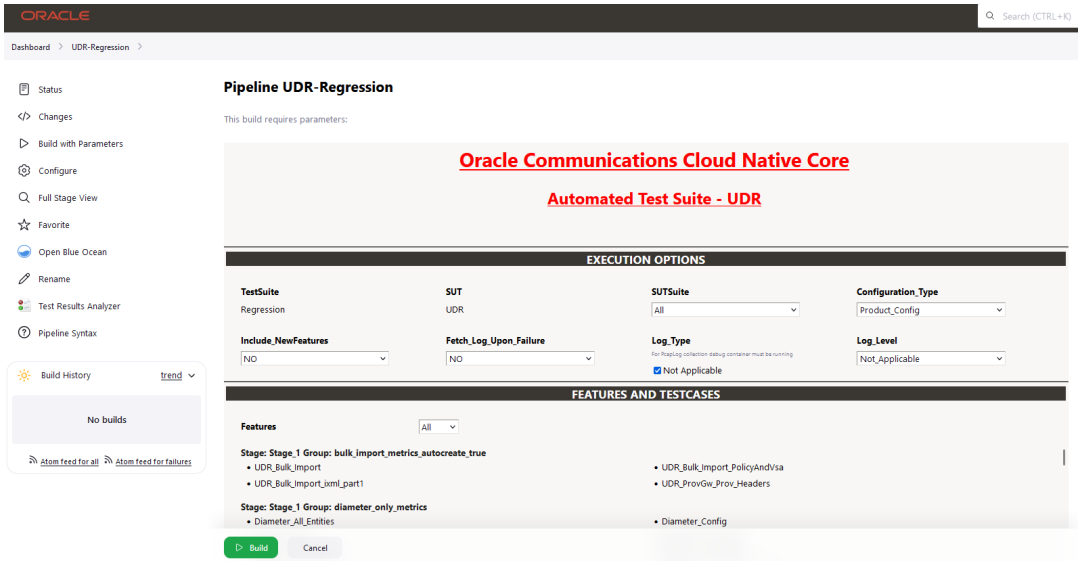
Copyright © 2019, 2025, Oracle and/or its affiliates.

- t - fqdn and port for 2nd diameter tool endpoint
- u - MYSQL SERVICE ENDPOINT
- v - NOTIFY STUB deployment name
- w - Name of signaling in the format fqdn:port (ocudr-s2-ingressgateway-sig.ocudr-s2:80)
- x - fqdn and port for UDR2 nudr-config (ocudr2-nudr-config.ocudr:5001)
- y - fqdn and port for UDR2 DIAMETER_ENDPOINT (fqdn:port) (ocudr-s2-nudr-diam-gateway.ocudr-s2:3868)
- z - FQDN and port for SCP stub (scp-stub-service:8080)
- A - Deployment name for SCP stub (scpstub-notify-ocstub-py)
- B - Bulk import helm name
- C - ProvGw namespace (ocudr)
- D - ProvGw endpoint in format fqdn:port (provgw-prov-ingressgateway.ocudr:80)
- E - ProvGw endpoint https port(443)
- F - ProvGw helm name
- G - ProvGw config host in the format fqdn:port(provgw-provgw-config.ocudr:5001)
- H - true if OCCNE is 1.9.x and above, else false (true)
- I - fqdn and port for fourg stub in the form (fqdn:port)
- J - fqdn and port for 1st diameter tool endpoint
- K - port for notify stub for TLS scenarios (8443)
- L - Service name of UDR1 provisioning ingressgateway in the format FQDN:PORT(ocudr-ingressgateway-prov.ocudr:80)
- M - Service name of site2 UDR provisioning ingressgateway in the format FQDN:PORT(ocudr2-ingressgateway-prov.ocudr-s2:80)
- N - Diameter Stub Tool1 Deploy Name (diamtoola-diam-stub)
- O - Diameter Stub Tool2 Deploy Name (diamtoolb-diam-stub)
- P - Migration tool helm release name (migration-tool)
- Q - Deployment name of the TLS notify stub (tlsnotifystub-ocstub-py)
- R - Enable error response logging validation (true)
- S - true (when underling CNE has support for CNLB Multus) to validate Traffic segregation feature else false
- T - true if underlying platform supports dual stack, else false. All dual stack scenarios will be marked as skipped if false (Default: false)

5. Click **Save**. The **Pipeline UDR-Regression** page appears.
6. Click **Build with Parameters**. A page listing all the regression features and displays the total count of the features and displays the total count of the features, count of scenarios in each feature, name of each feature, and scenarios in each feature.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-108 UDR Regression Pipeline Site Deployment



7. Click **Build** to run the test cases.

Figure 4-109 Consolidated Test Report

```
[Pipeline] catchError
[Pipeline] {
[Pipeline] sh (Stage: Stage_3  Group: provgw_no_igw_dr_metrics Result)
+ set +x
**********************************Stage: Stage_3 Group: provgw_no_igw_dr_metrics Result**********************************
Final Result:-
Total-Features RUN 15, PASS 15, FAIL 0

[Pipeline] }
[Pipeline] // catchError
[Pipeline] catchError
[Pipeline] {
[Pipeline] sh (Stage: Stage_3  Group: udr_no_igw_dr_metrics Result)
+ set +x
**********************************Stage: Stage_3 Group: udr_no_igw_dr_metrics Result**********************************
Final Result:-
Total-Features RUN 5, PASS 5, FAIL 0

[Pipeline] }
[Pipeline] // catchError
[Pipeline] catchError
[Pipeline] {
[Pipeline] sh (Stage: Stage_4  Group: sequential_execution Result)
+ set +x
**********************************Stage: Stage_4 Group: sequential_execution Result**********************************
```

> **Note**
>
> - For information about Configuration_Type, see Parameterization.
> - The **SUITSuite** option selected as 'ALL' will run all suites except multiple site scenarios.

Copyright © 2019, 2025, Oracle and/or its affiliates.

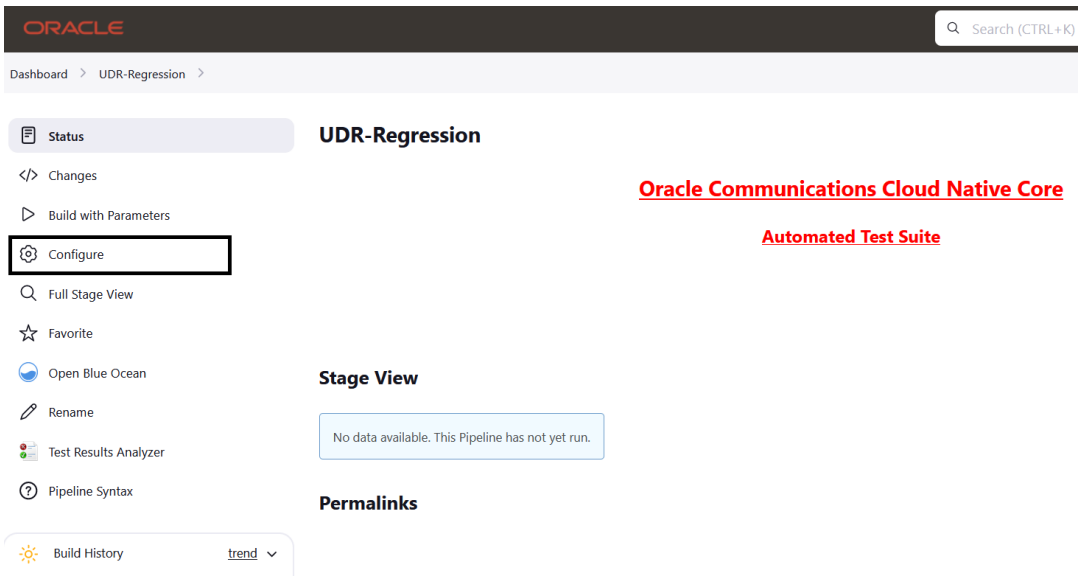
Sample output with expected failures are as follows:

```
Stage:Stage_2 Group:no_igw_dr_metrics
  UDR_Error_Response_log_validation.feature:299
UDR_Error_Response_log_Ondemand_05
  UDR_Suppressed_notification_to_the_initiator.feature:255
UDR_Suppressed_notification_to_the_initiator_05
```

Running UDR Regression Pipeline: Model 2: Multiple Site Deployment

1. Click **UDR-Regression** and then, click **Configure**.

Figure 4-110 UDR Regression Pipeline



2. The **General** tab appears. The user must wait for the page to load completely.
3. Select the **Discard old Builds**. This option allows you to configure the number of builds you want to retain in the persistent volume.

> **Note**
>
> It is recommended to configure this option. If there are large number of builds, the Persistent Volume may be utilized completely.

4. Click the **Advanced Project Options** tab. Scroll-down to reach the **Pipeline** configuration section as follows:

> **Important**
>
> Make sure the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

Do not change any value other than from **line number 68 to 112**. You can change the parameters marked as **"b"** - to - **"T"** as per user requirement. The parameters details are provided as comments in line number **15 to 60**. The parameters description is as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

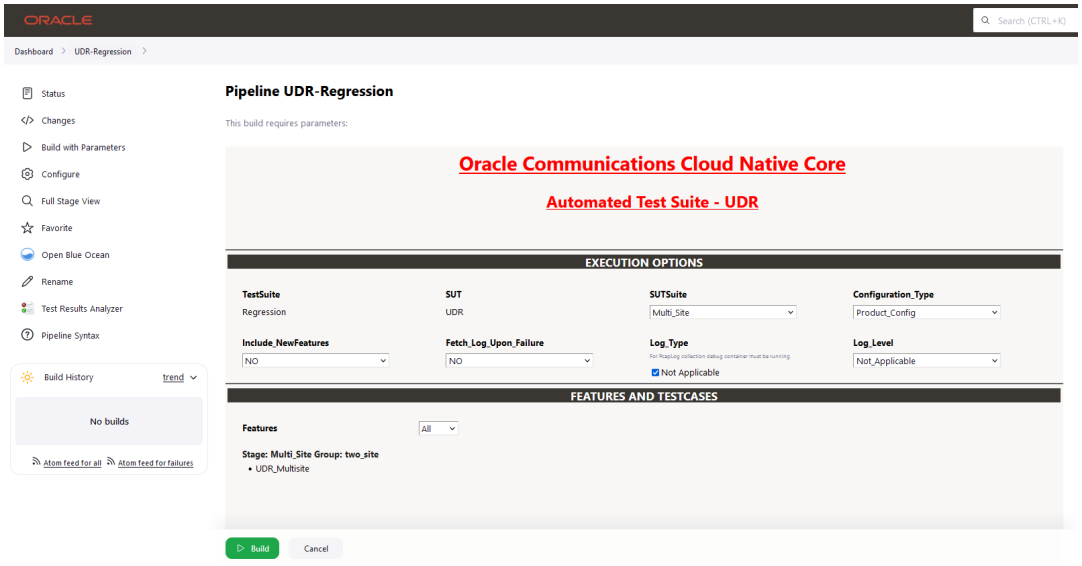
- a - Name of the NF to be tested in capital (UDR).
- b - Namespace in which the udr is deployed.
- c - Name of signalling UDR1_ingressgateway_service.namespace and port (ocudr-ingressgateway-sig.ocudr:80).
- d - Helm release name of UDR (ocudr)
- e - FQDN and Port UDR's nudr-config in the format fqdn:port (ocudr-nudr-config.ocudr:5001)
- f - Name_of_Prometheus_service.namespace and port(occne-prometheus-server.occne-infra:80).
- g - If this option is not provided, Prometheus URI shall be /api/v1 (Default: /prometheus/api/v1) (if CNE 1.9+ /<cne-cluster-name>/prometheus/api/v1)
- h - Number of times the re-run of failed case is allowed (default as 1).
- i - Name and port of Kubernetes Host server in the format FQDN:PORT(kubernetes.default:443).
- j - fqdn and port for DIAMETER_ENDPOINT in the format FQDN:PORT (ocudr-nudr-diam-gateway:3868)
- k - fqdn or IP of http2-server in the format <ip/fqdn> (fqdn) (notify-stub-service.ocudr:80)
- l - fqdn or IP for primary nrf stub and port in the format <ip/fqdn>:<port> (nrf-stub-server.ocudr:8080)
- m - fqdn or IP for secondary nrf stub and port in the format <ip/fqdn>:<port> (nrf-stub-server2.ocudr:8080)
- n - primary stub deployment name (nrfstub-ocstub-py)
- o - secondary stub deployment name (nrfstub2-ocstub-py)
- p - HTTPS port for UDR (443)
- q - root certificate authority (CA) (caroot.cer)
- r - Server certificate signed with root CA private key (apigatewayrsa.cer)
- s - private key to .pem format (rsa_private_key_pkcs1.pem)
- t - fqdn and port for 2nd diameter tool endpoint
- u - MYSQL SERVICE ENDPOINT
- v - NOTIFY STUB deployment name
- w - Name of signaling in the format fqdn:port (ocudr-s2-ingressgateway-sig.ocudr-s2:80)
- x - fqdn and port for UDR2 nudr-config (ocudr2-nudr-config.ocudr:5001)
- y - fqdn and port for UDR2 DIAMETER_ENDPOINT (fqdn:port) (ocudr-s2-nudr-diam-gateway.ocudr-s2:3868)
- z - FQDN and port for SCP stub (scp-stub-service:8080)
- A - Deployment name for SCP stub (scpstub-notify-ocstub-py)
- B - Bulk import helm name
- C - ProvGw namespace (ocudr)
- D - ProvGw endpoint in format fqdn:port (provgw-prov-ingressgateway.ocudr:80)

Copyright © 2019, 2025, Oracle and/or its affiliates.

- E - ProvGw endpoint https port(443)
- F - ProvGw helm name
- G - ProvGw config host in the format fqdn:port(provgw-provgw-config.ocudr:5001)
- H - true if OCCNE is 1.9.x and above, else false (true)
- I - fqdn and port for fourg stub in the form (fqdn:port)
- J - fqdn and port for 1st diameter tool endpoint
- K - port for notify stub for TLS scenarios (8443)
- L - Service name of UDR1 provisioning ingressgateway in the format FQDN:PORT(ocudr-ingressgateway-prov.ocudr:80)
- M - Service name of site2 UDR provisioning ingressgateway in the format FQDN:PORT(ocudr2-ingressgateway-prov.ocudr-s2:80)
- N - Diameter Stub Tool1 Deploy Name (diamtoola-diam-stub)
- O - Diameter Stub Tool2 Deploy Name (diamtoolb-diam-stub)
- P - Migration tool helm release name (migration-tool)
- Q - Deployment name of the TLS notify stub (tlsnotifystub-ocstub-py)
- R - Enable error response logging validation (true)
- S - true (when underling CNE has support for CNLB Multus) to validate Traffic segregation feature else false
- T - true if underlying platform supports dual stack, else false. All dual stack scenarios will be marked as skipped if false (Default: false)

5. Click **Save**. The **Pipeline UDR-Regression** page appears.
6. Click **Build with Parameters**. A page listing all the regression features and displays the total count of the features and displays the total count of the features, count of scenarios in each feature, name of each feature, and scenarios in each feature.

Figure 4-111 UDR Regression Pipeline Multiple Site Deployment



7. Click **Build** to run the test cases.

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> - For information about Configuration_Type, see Parameterization.
> - The **SUITSuite** option selected as 'MULTISITE' will run multiple site scenarios only.

Figure 4-112 Consolidated Test Report

```
**********************************Overall Result*****************************************

Took 0h7m14s

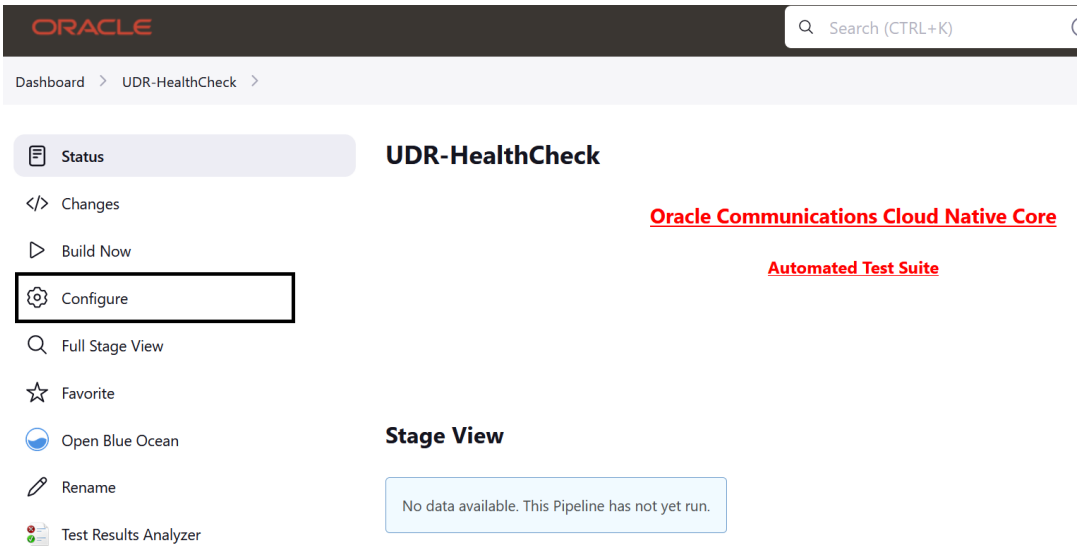
Final Result:-
Total-Features RUN 1, PASS 1, FAIL 0
```

4.7.7.3 Running UDR HealthCheck Pipeline

To run health check on UDR:

1. Navigate to **UDR-HealthCheck** pipeline and click **Configure**.

Figure 4-113 UDR-HealthCheck Pipeline



2. The **General** tab appears. The user must wait for the page to load completely.
3. Click the **Advanced Project Options** tab. Scroll-down to reach the **Pipeline** configuration section as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-114 UDR-HealthCheck Configuration



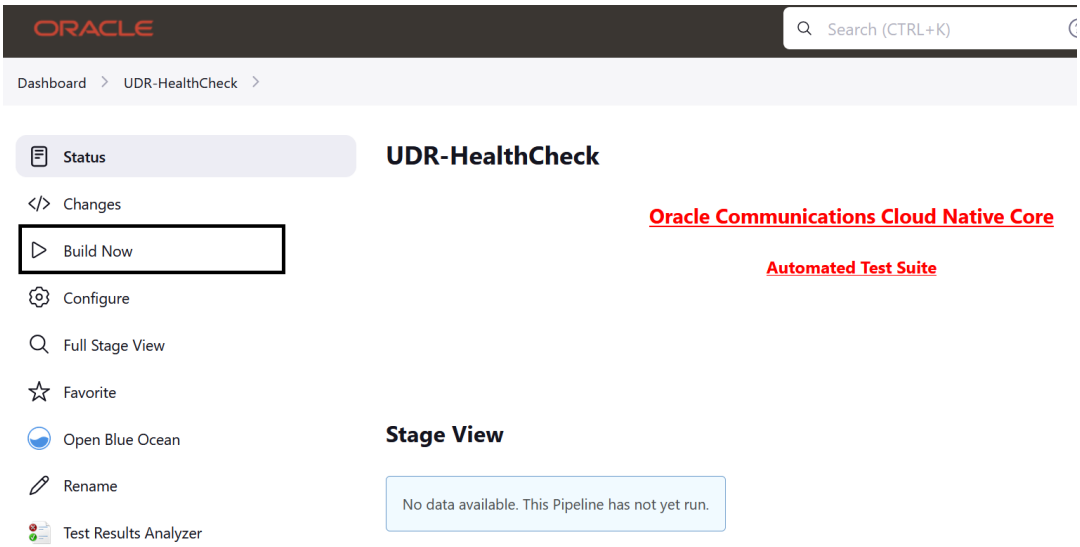
> **Important**
>
> Make sure the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

Do not change any value other than **line numbers 17 and 18**. You can change the parameters marked as **"a", "b"** and **"c"**as per user requirement. The parameters details are provided as comments in **line number 7 and 9**. The parameters description is as follows:

- a - helm releases [Provide Release Name with Comma Separated if more than 1]
- b - namespace of SUT
- c - -helm command alias (helm or helm2 or helm3)

4. Click **Save**. The **Pipeline UDR-HealthCheck** page appears.
5. Click **Build Now**. This triggers health check for SUT.

Figure 4-115 UDR-HealthCheck Build Now



Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-116 Sample Output

```
NAME: ocudr-ravi
LAST DEPLOYED: Sun Oct 15 19:15:05 2023
NAMESPACE: ats-ravi
STATUS: deployed
REVISION: 1
TEST SUITE:     ocudr-ravi-test
Last Started:   Wed Oct 18 17:47:13 2023
Last Completed: Wed Oct 18 17:47:38 2023
Phase:          Succeeded
NOTES:
# Copyright 2019 (C), Oracle and/or its affiliates. All rights reserved.

Thank you for installing ocudr.

Your release is named ocudr-ravi , Release Revision: 1.
To learn more about the release, try:

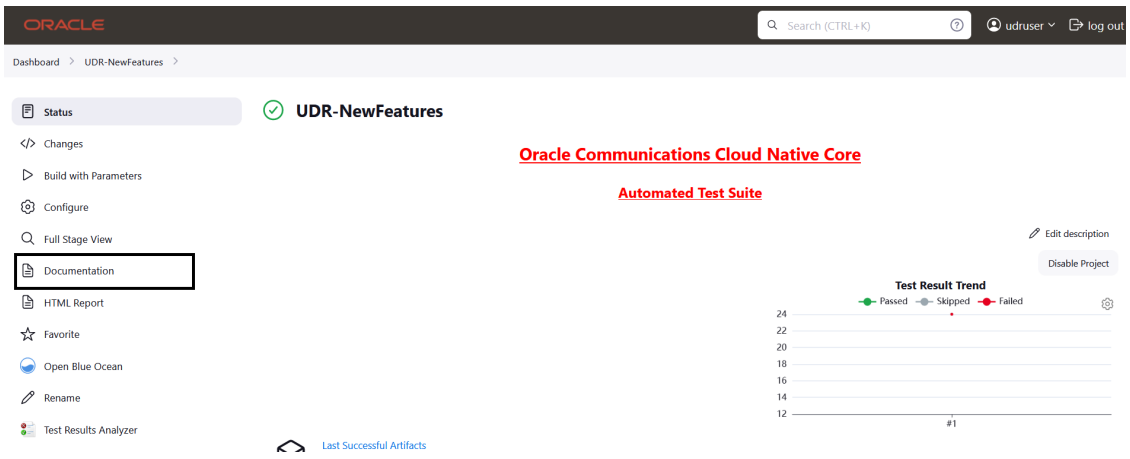
  $ helm status ocudr-ravi
  $ helm get ocudr-ravi
[0m

[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

4.7.7.4 UDR NewFeatures Documentation

To view the documentation of UDR-NewFeatures, click **Documentation** in the left navigation pane of the UDR-NewFeatures pipeline to view all the UDR new features test cases.

Figure 4-117 UDR-NewFeatures Documentation



Copyright © 2019, 2025, Oracle and/or its affiliates.

UDR new features list along with its description is as follows:

Table 4-15 UDR New Features List

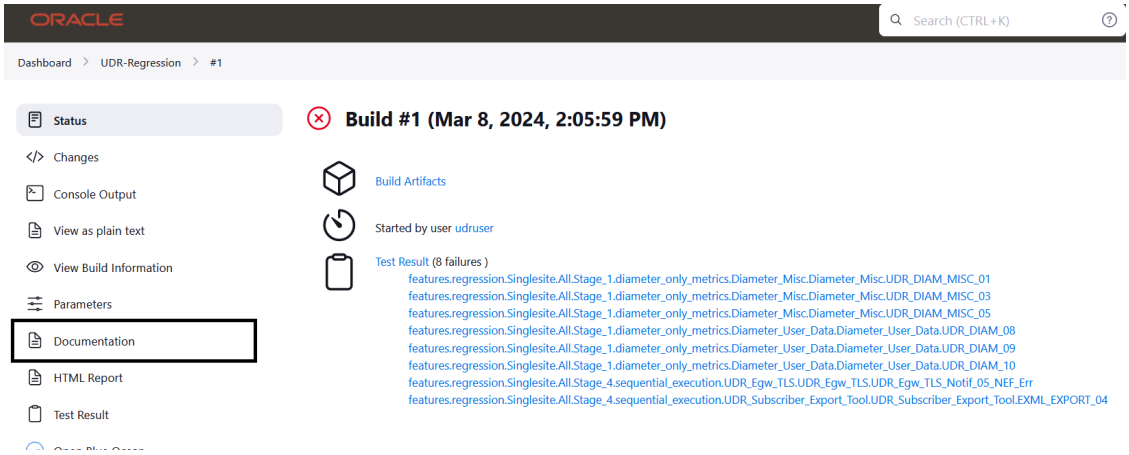
Feature	Description
UDR_Subscriber_Policy_Data_Export.feature	This feature file validates the policy subscriber export feature in CSV format.
UDR_PNR_Multisite.feature	This feature file validates the PNR scenarios on the Multisite.
UDR_Bug_Fix_25_1_100.feature	This feature file validates the bugs raised for 25.1.100 release.
UDR_Default_Headers.feature	This feature file validates multiple default headers received on the ingress gateway signaling and provisioning.

Click any feature to view its description.

4.7.7.5 UDR Regression Documentation

Click **Documentation** in the left navigation pane of the UDR-Regression pipeline to view all the UDR regression test cases.

Figure 4-118 UDR-Regression Documentation



Click any feature to view its description.

4.7.8 Running EIR Pipelines

This section describes how to run EIR related pipelines. It includes:

- Running EIR NewFeatures Pipeline
- Running EIR HealthCheck Pipeline
- EIR NewFeatures Documentation

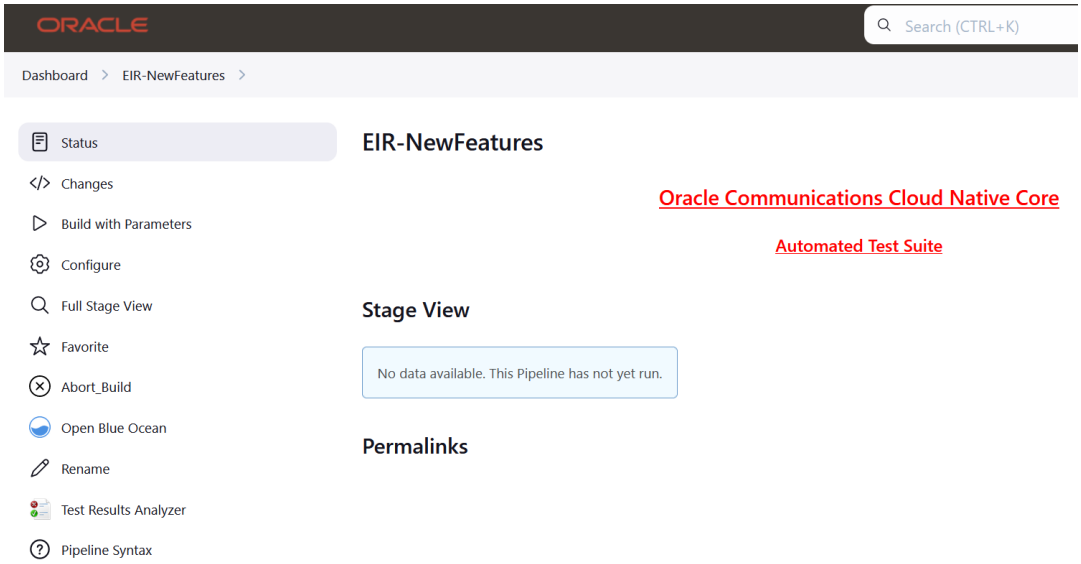
4.7.8.1 Running EIR NewFeatures Pipeline

To run the EIR-NewFeatures pipeline test cases:

Copyright © 2019, 2025, Oracle and/or its affiliates.

1. Click **EIR-NewFeatures**. Click **Configure** in the left navigation pane. The **General** tab appears. The user must wait for the page to load completely.

Figure 4-119 EIR NewFeatures Pipeline



2. Once the page loads completely, select the **Discard old Builds**. This option allows you to configure the number of builds you want to retain in the persistent volume.

> **Note**
>
> It is recommended to configure this option. If there is large number of build, the Persistent Volume may be utilized completely.

3. Click the **Advanced Project Options** tab. Scroll-down to reach the **Pipeline** configuration as follows:

> **Important**
>
> Make sure the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

You should not change any other value apart from **line number 54 to 87**. It means you can change the parameters marked as **"b"** - to - **"l"** as per user requirement. The parameters details are provided as comments in **line number 12 to 46**. The parameters description is as follows:

- a - Selected NF.
- b - NameSpace in which UDR is Deployed.
- c - EIR signaling FQDN and PORT in the format FQDN:POR.
- d - EIR provisioning FQDN and PORT in the format FQDN:PORT.
- e - EIR config service FQDN and PORT in the format FQDN:PORT.
- f - FQDN:PORT format for primary nrf-stub.

Copyright © 2019, 2025, Oracle and/or its affiliates.

- g - FQDN:PORT format for secondary nrf-stub.
- h - FQDN and PORT of Kubernetes Host server in the format FQDN:PORT.
- i - Helm name of EIR.
- j - primary nrf-stub deployment name.
- k - secondary nrf-stub deployment name.
- l - primary nrf-stub metric port
- m - secondary nrf-stub metric port
- n - bulk import Helm name
- o - Number of retries to check if pod down is successful.
- p - Number of retries to check if pod up is successful.
- q - Number of retries when ATS detects tcp connection failures.
- r - Time interval in seconds between tcp connection retries.
- s - re-run count.
- t - FQDN and PORT of Prometheus server in the format FQDN:PORT.
- u - prometheus uri apiName and apiVersion.
- v - EIR HTTPS port.
- w - root certificate authority for signalling ingressgateway.
- x - Server certificate signed with root CA private key for signalling ingressgateway.
- y - private key in .pem format for signalling ingressgateway.
- z - root certificate authority for provisioning ingressgateway.
- A - Server certificate signed with root CA private key for provisioning ingressgateway.
- B - Private key in .pem format for provisioning ingressgateway.
- C - True if HA prometheus else false (default: true).
- D - Enable error response logging validation (true).
- E - fqdn and port for diameter gateway in the format FQDN:PORT (oceir-nudr-diam-gateway:3868).
- F - fqdn and port for diam-stub endpoint (diam-stub-service:3000).
- G - Diameter Stub Tool1 Deploy Name (diamtoola-diam-stub).
- H - true (when underling CNE has support for CNLB Multus) to validate traffic segregation feature else false (Default: false)
- I - true if underlying platform supports dual stack, else false. All dual stack scenarios will be marked as skipped if false (Default: false)

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-120 EIR NewFeatures Pipeline Configuration

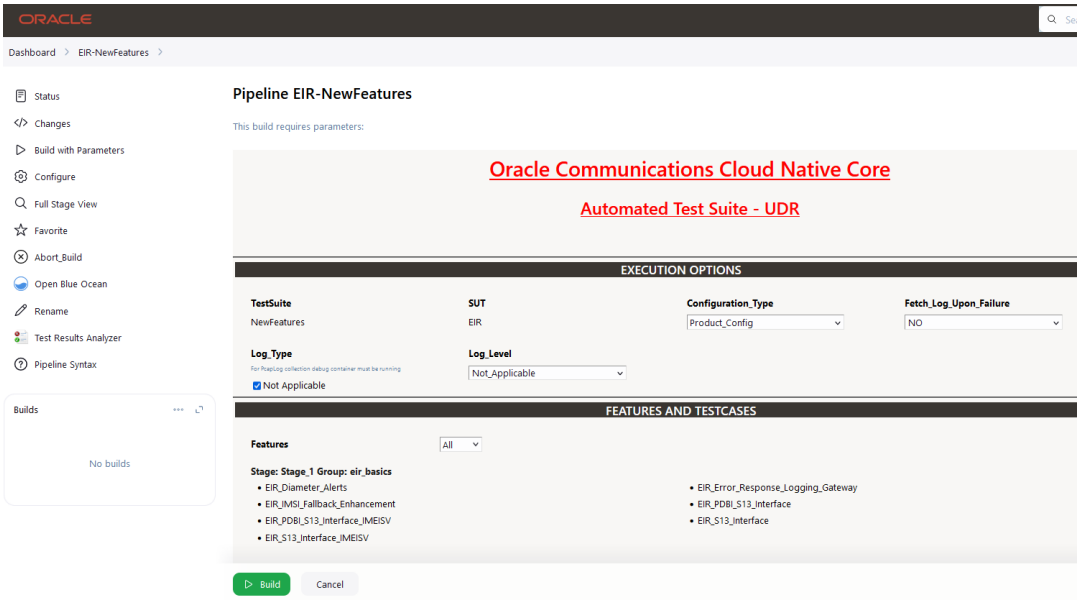


> **Note**
>
> Setting rerun count, number of retries to check pod status and segment details:
>
> - Rerun count can be set to any desired value by changing option 's' in the pipeline script.
> - Number of retries to check if pod is down can be set by providing option 'o' in the pipeline script.
> - Number of retries to check if pod is up can be set by providing option 'p' in the pipeline script.
> - If ProvGw is deployed with segment details having UDR fqdn and port, provide option 'l' as port.
> - Option 'u' will take the value '/<domain name>/prometheus/api/v1 for CNE version 1.9 and above or OSO version 1.10.x and above. Run the below command, If domain name is not known. The domain name can be found after executing the command from ATS pod:
>
>   ```
>   curl http://<prometheus fqdn>:<port>/
>   ```
>
>   Example:
>
>   ```
>   curl http://occne-kube-prom-stack-kube-prometheus.occne-infra/
>   <a href="/udr-cne/prometheus">Found</a>
>   ```

4. Click **Save** after making necessary changes. The **Pipeline EIR-NewFeatures** page appears.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-121 EIR Build with Parameters



5. Click **Build with Parameters**. The page lists all the newly added features and displays the total count of features, counts of scenarios in each feature, name of each feature, and scenarios in each feature.
6. To view consolidated and detailed stack trace results in case of any failures, click Test Results Analyzer in the left navigation pane. The test results analyzer report appears. For more information, see Test Results Analyzer.

Figure 4-122 EIR-NewFeatures Pipeline Consolidated Automation Suite Test Report

```
The below are the consolidated output of all the features that are run
[Pipeline] catchError
[Pipeline] { (hide)
[Pipeline] sh (Stage: Stage_1  Group: eir_basics Result)
+ set +x
**********************************Stage: Stage_1 Group: eir_basics Result****************************************
Final Result:-
Total-Features RUN 1, PASS 1, FAIL 0

[Pipeline] }
[Pipeline] // catchError
[Pipeline] sh (Overall Result)
+ set +x
**********************************Overall Result*****************************************

Took 0h1m3s

Final Result:-
Total-Features RUN 1, PASS 1, FAIL 0
```

For more details on consolidated test report, see Final Summary Report, Build Color and Application Logs.

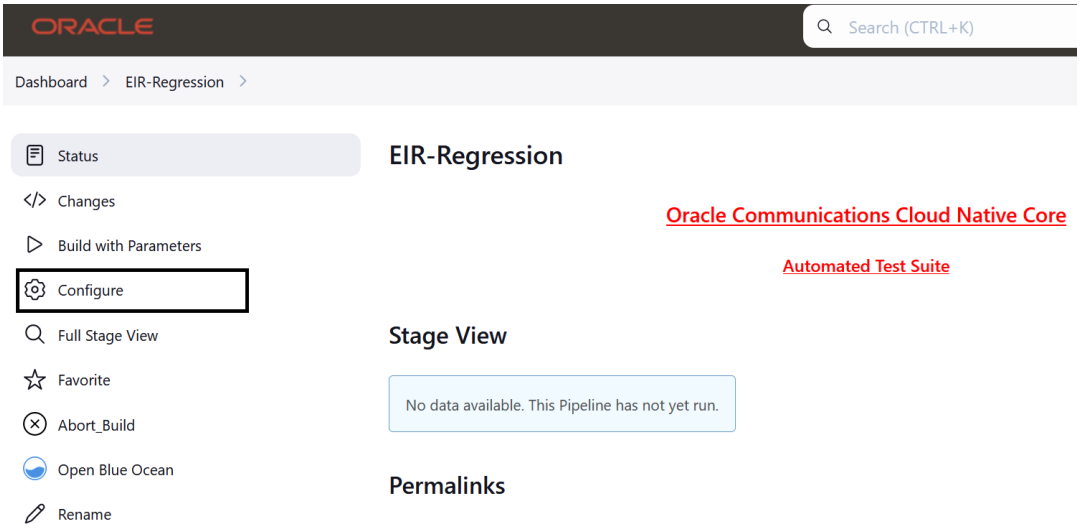
Copyright © 2019, 2025, Oracle and/or its affiliates.

4.7.8.2 Running EIR Regression Pipeline

Perform the below steps to run the EIR-Regression pipeline test cases:

1. Click **EIR-Regression** and then, click **Configure**.

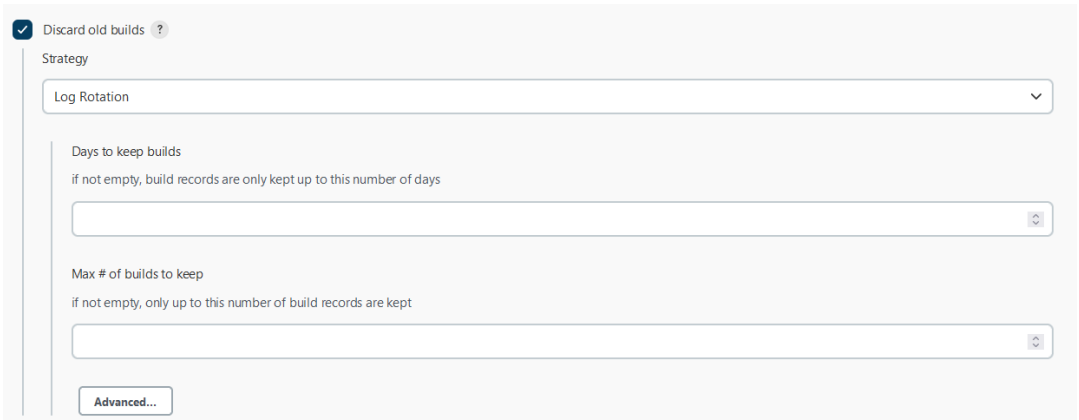
Figure 4-123 EIR Regression Pipeline



2. The **General** tab appears. The user must wait for the page to load completely.
3. Select the **Discard old Builds**. This option allows you to configure the number of builds you want to retain in the persistent volume.

> **Note**
>
> It is recommended to configure this option. If there are large number of builds, the Persistent Volume may be utilized completely.

Figure 4-124 Discard Builds



4. Click the **Advanced Project Options** tab. Scroll-down to reach the **Pipeline** configuration section as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Important**
>
> Make sure the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

Do not change any value other than from **line number 48 to 76**. You can change the parameters marked as **"b" - to - "D"** as per user requirement. The parameters details are provided as comments in **line number 11 to 40**. The parameters description is as follows:

- a - Selected NF
- b - NameSpace in which UDR is Deployed
- c - EIR signaling FQDN and PORT in the format FQDN:PORT
- d - EIR provisioning FQDN and PORT in the format FQDN:PORT
- e - EIR config service FQDN and PORT in the format FQDN:PORT
- f - FQDN:PORT format for primary nrf-stub
- g - FQDN:PORT format for secondary nrf-stub
- h - FQDN and PORT of Kubernetes Host server in the format FQDN:PORT
- i - helm name of EIR
- j - primary nrf-stub deployment name
- k - secondary nrf-stub deployment name
- l - primary nrf-stub metric port
- m - secondary nrf-stub metric port
- n - bulk import Helm name
- o - Number of retries to check if pod down is successful
- p - Number of retries to check if pod up is successful
- q - Number of retries when ATS detects tcp connection failures
- r - Time interval in seconds between tcp connection retries
- s - re-run count
- t - FQDN and PORT of Prometheus server in the format FQDN:PORT
- u - prometheus uri apiName and apiVersion
- v - EIR HTTPS port
- w - root certificate authority for signalling ingressgateway
- x - Server certificate signed with root CA private key for signalling ingressgateway
- y - private key in .pem format for signalling ingressgateway
- z - root certificate authority for provisioning ingressgateway
- A - Server certificate signed with root CA private key for provisioning ingressgateway
- B - Private key in .pem format for provisioning ingressgateway
- C - True if HA prometheus else false(default: true)
- D - Enable error response logging validation (true)

Copyright © 2019, 2025, Oracle and/or its affiliates.

- E - FQDN and port for diameter gateway in the format FQDN:PORT (oceir-nudr-diam-gateway:3868)
- F - FQDN and port for diam-stub endpoint (diam-stub-service:3000)
- G - Diameter Stub Tool1 deploy name (diamtoola-diam-stub)
- H - true (when underling CNE has support for CNLB Multus) to validate traffic segregation feature else false (Default: false)
- I - true if underlying platform supports dual stack, else false. All dual stack scenarios will be marked as skipped if false (Default: false)

5. Click **Save**. The **Pipeline EIR-Regression** page appears.

Figure 4-125 EIR Regression Configuration

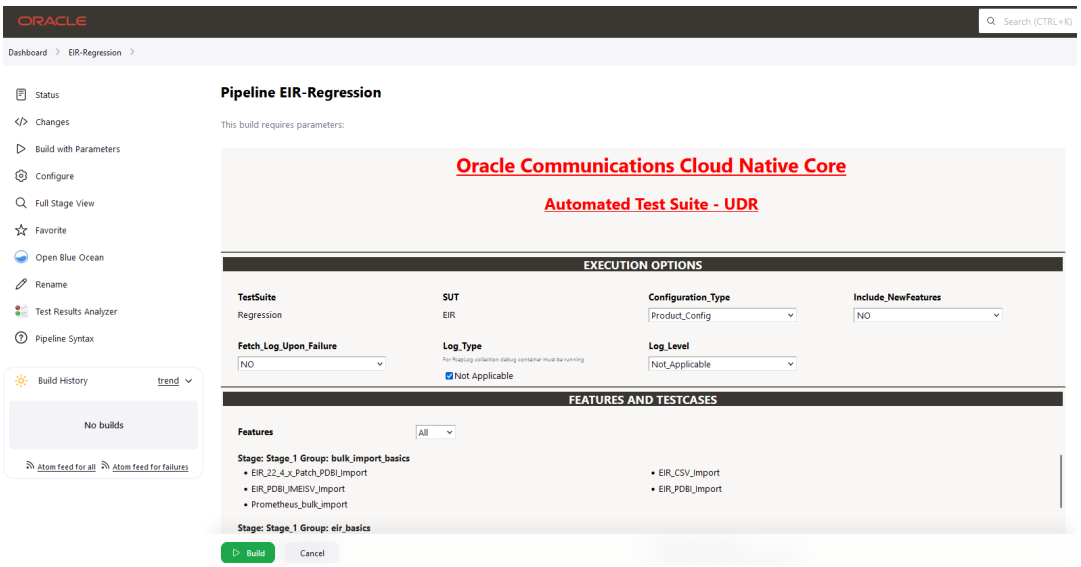


Copyright © 2019, 2025, Oracle and/or its affiliates.

> **Note**
>
> Setting rerun count and the number of retries to check the pod status and segment details is given below:
>
> - Rerun count can be set to any desired value by changing option 's' in the pipeline script.
> - Number of retries to check if pod is down can be set by providing option 'o' in the pipeline script.
> - Number of retries to check if pod is up can be set by providing option 'p' in the pipeline script.
> - If ProvGw is deployed with segment details having UDR fqdn/IP:port, provide option 'l' as "IP". If ProvGw is deployed with segment details having only UDR fqdn/IP, provide option 'l' as "fqdn". If any value other than "IP" and "fqdn" is provided, ATS assumes that the segment details in ProvGw is in "IP" mode.
> - Option 'u' shall take value '/<domain name>/prometheus/api/v1 for CNE version 1.9 or above or OSO version is 1.10.x or above. Run the below command from ATS pod to find the domain name, if the domain name is not known:
>
>   ```
>   curl http://<prometheus fqdn>:<port>/
>   ```
>
>   Example:
>
>   ```
>   curl http://occne-kube-prom-stack-kube-prometheus.occne-infra/
>   <a href="/udr-cne/prometheus">Found</a>
>   ```

6. Click **Save**. The **Pipeline EIR-Regression** page appears.
7. Click **Build with Parameters**. A page listing all the regression features and displays the total count of the features and displays the total count of the features, count of scenarios in each feature, name of each feature, and scenarios in each feature.

Figure 4-126 EIR Regression Build with Parameters



Copyright © 2019, 2025, Oracle and/or its affiliates.

8. A sample consolidated automation suite test report, when re-run is set to 0 is as follows

Figure 4-127 EIR-Regression Consolidated Automation Suite Test Report

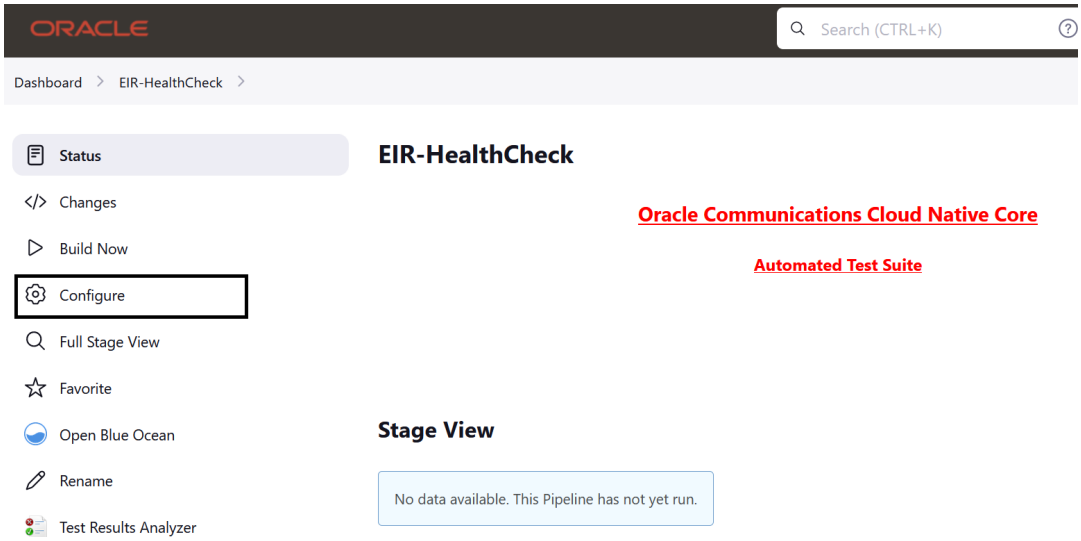
```
31 features passed, 0 failed, 0 skipped
262 scenarios passed, 0 failed, 0 skipped
6690 steps passed, 0 failed, 0 skipped, 0 undefined
Took 232m49.448s
```

4.7.8.3 Running EIR HealthCheck Pipeline

To run health check on EIR:

1. Navigate to **EIR-HealthCheck** pipeline and click **Configure**.

Figure 4-128 EIR HealthCheck Pipeline



2. The **General** tab appears. The user must wait for the page to load completely.
3. Click the **Advanced Project Options** tab. Scroll-down to reach the **Pipeline** configuration section as follows:

> **Important**
>
> Make sure the page loads completely before you perform any action on it. Do not modify any configuration except the one discussed below.

Do not change any value other than **line numbers 16 and 17**. You can change the parameters marked as "a" "b" and "c" as per user requirement. The parameters details are provided as comments in **line number 7 and 9**. The parameters description is as follows:

- a - helm releases [Provide Release Name with Comma Separated if more than 1]
- b - namespace of SUT.
- c - helm command alias (helm, helm2, or helm3)

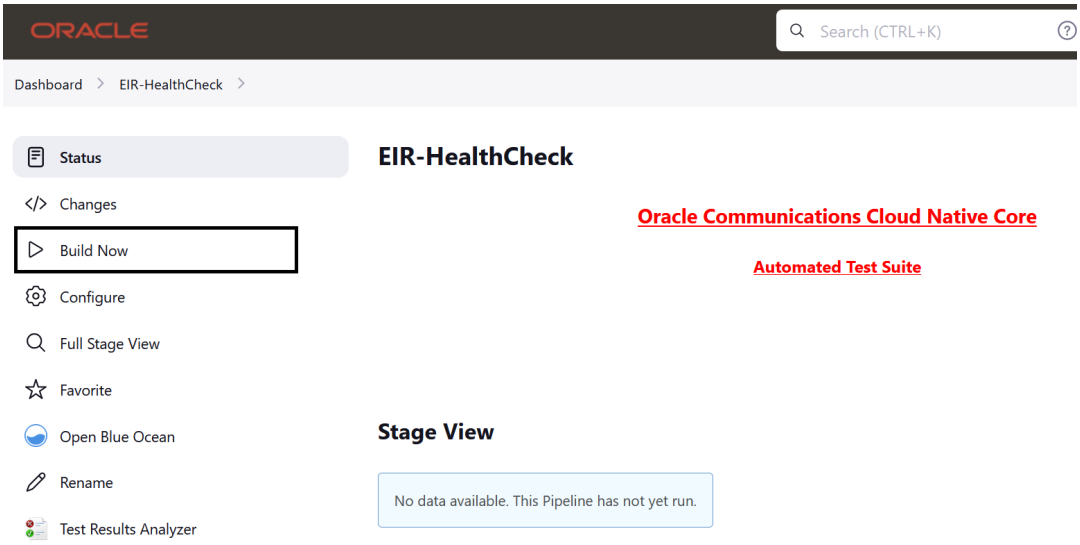
Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-129 EIR-HealthCheck Pipeline configuration



4. Click **Save**. The **Pipeline EIR-HealthCheck** page appears.
5. Click **Build Now**. This triggers health check for SUT.

Figure 4-130 EIR-HealthCheck Build Now

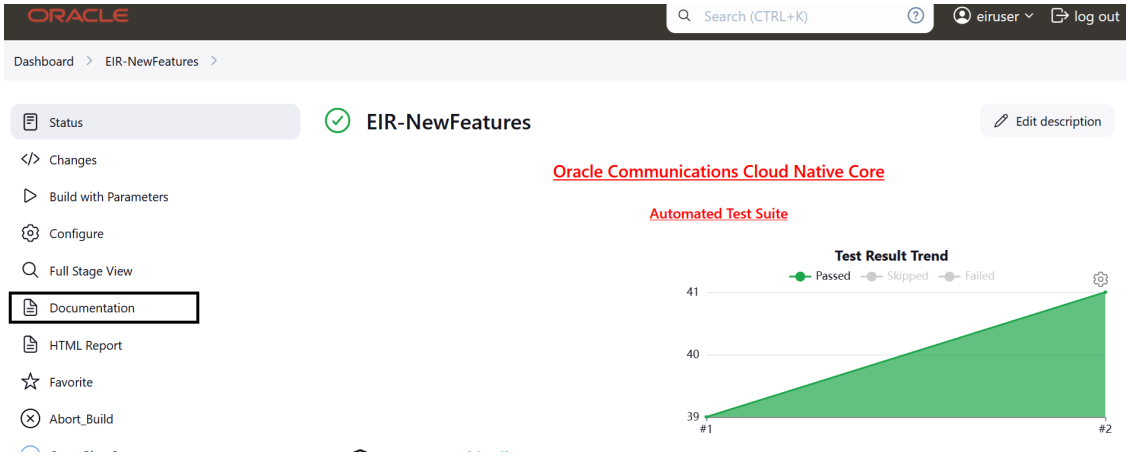


4.7.8.4 EIR NewFeatures Documentation

To view the documentation of EIR-NewFeatures, click the **Documentation** link in the left navigation pane (present inside the build), as follows:

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-131 EIR-New Features Documentation



The page lists all the features. Click any feature to view description of that test case.

EIR new features list along with its description is as follows:

Table 4-16 EIR New Features List

Feature	Description
NA	NA

> **Note**
>
> The Documentation option appears only if you have run the EIR-NewFeatures pipeline test cases atleast once.

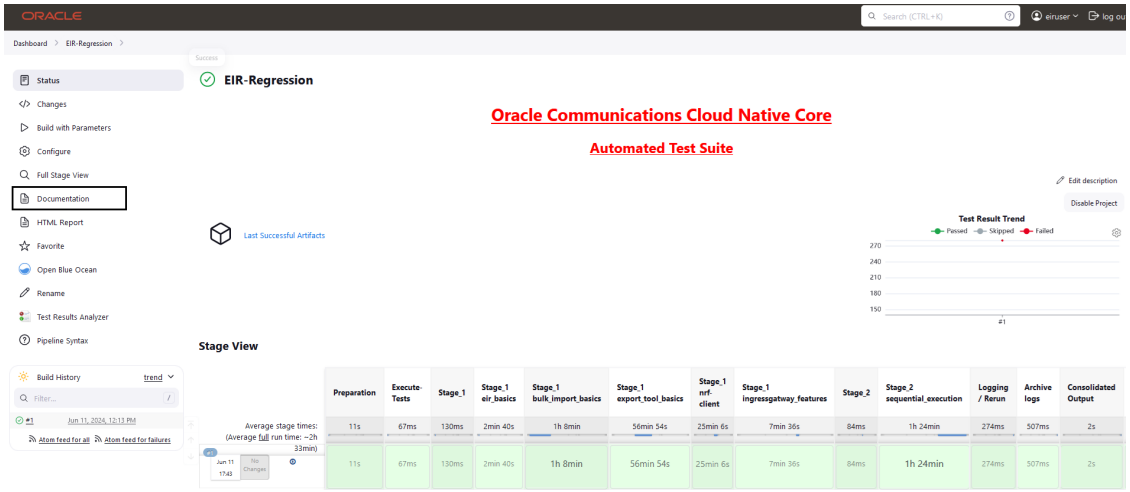
Based on the functionalities covered under **Documentation**, the **Build Requires Parameters** page displays test cases. To navigate back to the Pipeline EIR-NewFeatures page, click **Back to EIR-NewFeatures** link available on top left corner of the page.

4.7.8.5 EIR Regression Documentation

Click **Documentation** in the left navigation pane of the EIR Regression pipeline to view all the test cases provided.

Copyright © 2019, 2025, Oracle and/or its affiliates.

Figure 4-132 EIR Regression Documentation



Click any feature to see the description of that test case.

Copyright © 2019, 2025, Oracle and/or its affiliates.