Oracle® Communications

Cloud Native Core Automated Testing Suite Guide

Release 26.1.201
G59076-01
July 2026

Oracle Communications Cloud Native Core Automated Testing Suite Guide, Release 26.1.201

G59076-01

Copyright © 2019, 2026, 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.

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.

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.

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 2
2.1.1 Generating an API Token for a User 2
2.1.2 Use the RESTful Interfaces 4
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 16
2.1.2.5 Getting Test Suite Artifacts 18
2.2 ATS Custom Abort 21
2.3 ATS Feature Activation and Deactivation 25
2.4 ATS GUI Enhancements 29
2.5 ATS Pod Logs 31
2.6 ATS Health Check 69
2.7 ATS Jenkins Job Queue 71
2.8 Application Log Collection 72
2.8.1 Application Log Collection and Parallel Test Execution Integration 74
2.9 ATS Maintenance Scripts 74
2.9.1 ATS Scripts 75
2.9.2 Updating ats_install.sh 76
2.9.3 Restarting Jenkins without Restarting Pod 76
2.9.4 Updating Stub Scripts 77

Copyright © 2019, 2026, Oracle and/or its affiliates.

2.9.5 Running ATS and Stub Deployment Scripts 78
2.10 ATS System Name and Version Display on the ATS GUI 78
2.11 ATS Tagging Support 79
2.11.1 Combination of Tags and their Results 80
2.12 Custom Folder Implementation 83
2.13 Health Check 84
2.14 Individual Stage Group Selection 94
2.15 Lightweight Performance 95
2.15.1 Configure <NF>-Performance Pipeline 95
2.16 Managing Final Summary Report, Build Color, and Application Log 96
2.17 Modifying Login Password 104
2.18 Multiselection Capability for Features and Scenarios 104
2.18.1 Feature Level Selection 104
2.18.2 Scenario Selection 105
2.19 Parallel Test Execution 106
2.19.1 ATS GUI Page Changes 109
2.19.2 ATS Console Log Changes 109
2.19.3 Downloading or Viewing Individual Group Logs 110
2.20 Parameterization 113
2.20.1 Running Test Cases 114
2.21 PCAP Log Collection 116
2.21.1 Application Log Collection and Parallel Test Execution Integration 116
2.22 Persistent Volume for 5G ATS 117
2.22.1 Processing Flow 117
2.22.2 Deploying Persistent Volume 118
2.22.3 Backward Porting 122
2.23 Single Click Job Creation 122
2.23.1 Configuring Single Click Job 122
2.24 Support for ATS Deployment in OCI 125
2.24.1 Accessing ATS GUI in OCI 125
2.24.2 Adding Ingress and Egress Rules to Access the OCI Console 126
2.25 Support for Transport Layer Security 130
2.25.1 Deploy ATS with TLS Enabled 131
2.25.1.1 Generate JKS File for Jenkins Server 131
2.25.1.2 Enable ATS GUI with HTTPS 136
2.26 Test Results Analyzer 138
2.26.1 Accessing Test Results Analyzer Feature 138
2.27 Support for Test Case Mapping and Count 141
2.27.1 Access Test Case Mapping and Count Feature 141

Copyright © 2019, 2026, Oracle and/or its affiliates.

3 Installing ATS for Different Network Functions

3.1 Installing ATS for NRF 1
3.1.1 Resource Requirements 1
3.1.2 Downloading the ATS Package 2
3.1.3 Pushing the Images to Customer Docker Registry 3
3.1.4 Configuring ATS 4
3.1.4.1 Enabling Static Port 4
3.1.4.2 Enabling Aspen Service Mesh 4
3.1.4.3 Configuring Dual Stack 6
3.1.4.4 Enabling Persistent Volume 7
3.1.4.5 Enabling NF FQDN Authentication 8
3.1.5 Deploying ATS and Stub in Kubernetes Cluster 9
3.1.6 Verifying ATS Deployment 9
3.1.7 Post-Installation Steps (if Persistent Volume is Used) 11
3.2 Installing ATS for NSSF 12
3.2.1 Resource Requirements 12
3.2.2 Locating and Downloading ATS and Simulator Images 14
3.2.3 Deploying ATS in Kubernetes Cluster 16
3.3 Installing ATS for SCP 30
3.3.1 Prerequisites 30
3.3.1.1 Software Requirements 30
3.3.1.2 Environment Setup Requirements 30
3.3.1.3 Resource Requirements 31
3.3.1.4 Downloading the ATS Package 33
3.3.1.5 Pushing the Images to Customer Docker Registry 44
3.3.1.6 Preinstall Preparation of SCP for SCP-ATS 46
3.3.1.7 Preinstallation Preparation for SCP-ATS 51
3.3.2 Configuring SCP-ATS and SCP 55
3.3.2.1 Updating ATS Configuration 56
3.3.3 Deploying ATS and Stub in the Kubernetes Cluster 71
3.3.4 Post Installation and Deployment Steps 72
3.3.4.1 Verifying ATS Deployment 72
3.3.4.2 Modifying the scpc-alternate-resolution Microservice 72
3.3.4.3 ATS Testcase Parametrization on User Input 73
3.3.4.4 ATS Tagging Support for SCP 82
3.3.4.5 Accessing Collected Logs 83
3.3.4.6 Enabling PCAP Logs 84
3.3.4.7 Data Collection after Feature File Failure 84
3.3.5 Appendix 84
3.3.5.1 Creating Custom Service Account 84
3.3.5.2 Adding New Stubs to SCP-ATS 85

Copyright © 2019, 2026, Oracle and/or its affiliates.

3.4 Installing ATS for SEPP 86
3.4.1 Resource Requirements 86
3.4.1.1 SEPP ATS Compatibility Matrix 88
3.4.2 Downloading the ATS Package 89
3.4.3 Pushing the Images to Customer Docker Registry 90
3.4.4 Creating Secrets and Supporting TLS 1.2 and TLS 1.3 91
3.4.4.1 Configuring Root Certificate 91
3.4.4.2 Generating ATS Certificate 94
3.4.4.3 Creating ATS Secret and Configuring Helm chart 98
3.4.5 Generating Hosted SEPP multi-fqdn Certificates and Secrets 100
3.4.6 Create ATS Health Check secret 106
3.4.7 Configuring ATS 107
3.4.7.1 Enabling Aspen Service Mesh 107
3.4.7.2 Configuring NIF 109
3.4.7.3 ATS API 110
3.4.7.4 Enabling Static Port 110
3.4.7.5 Enabling Roaming Hub Mode 111
3.4.7.6 Configuring Hosted SEPP Mode 111
3.4.7.7 Configuring Egress Rate Limiting Feature 115
3.4.7.8 Configuring Ingress Rate Limiting Feature 115
3.4.7.9 Configuring Single Service Account 115
3.4.7.10 Configuring the Cache Refresh Timeout Value 116
3.4.7.11 Configuring the Topology Cache Refresh Timeout Value 116
3.4.7.12 Configuring the Security Counter Measure Cache Refresh Timeout 117
3.4.7.13 Configuring n32cHandshakePlmnIdListValidationEnabled 117
3.4.7.14 Configuring Request Timeout 120
3.4.7.15 Idle Timeout 122
3.4.7.16 Configuring Polling Interval in N32 Egress Gateway 122
3.4.7.17 EvictSanHeaderCacheDelay 122
3.4.7.18 Configuring the Reroute Attempts for Egress Gateway 123
3.4.7.19 Configuring Alternate Routing based on the DNS SRV Record for Home Network Functions 123
3.4.7.20 Configuring Load Sharing among Multiple Remote SEPP Nodes 124
3.4.7.21 Configuring Dual Stack 125
3.4.7.22 ATS Tagging Support 126
3.4.7.23 ATS Helm Enhancement 128
3.4.7.24 Configuring LCI and OCI Header Support 129
3.4.7.25 NRF Selection Mechanisms Using nrf client 130
3.4.7.26 Enabling Persistant Volume Storage 131
3.4.8 Deploying ATS in Kubernetes Cluster 132
3.4.9 Verifying ATS Deployment 132

Copyright © 2019, 2026, Oracle and/or its affiliates.

3.4.10 Post Installation Steps (If persistent volume is used) 133

4 Running NF Test Cases Using ATS

4.1 Running NRF Test Cases using ATS 1
4.1.1 Prerequisites 1
4.1.2 Logging into ATS 11
4.1.3 NRF-NewFeatures Pipeline 12
4.1.4 NRF-NewFeatures Documentation 23
4.1.5 NRF-Regression Pipeline 25
4.1.6 NRF-Regression Documentation 27
4.2 Running NSSF Test Cases using ATS 29
4.2.1 Prerequisites 29
4.2.2 Logging into ATS 29
4.2.3 NSSF-NewFeatures Pipeline 31
4.2.4 NSSF-NewFeatures Documentation 42
4.2.5 NSSF-Regression Pipeline 44
4.2.6 NSSF-Regression Documentation 55
4.3 Running SCP Test Cases using ATS 56
4.3.1 Logging in to ATS 56
4.3.2 SCP-NewFeatures Pipeline 59
4.3.2.1 Configuring New Feature Pipelines 59
4.3.2.2 Running SCP New Feature Pipelines 64
4.3.3 SCP-NewFeatures Documentation 65
4.3.4 SCP-Regression Pipeline 67
4.3.4.1 Configuring Regression Pipelines 67
4.3.4.2 Running Regression Pipelines 71
4.3.5 SCP-Regression Documentation 72
4.3.5.1 Renaming of Regression Feature Files 73
4.4 Running SEPP Test Cases using ATS 81
4.4.1 Prerequisites 81
4.4.2 Logging into ATS 83
4.4.3 SEPP NewFeatures Pipeline 85
4.4.3.1 Single Click Job Creation for SEPP New Features 89
4.4.4 NewFeatures - Documentation 91
4.4.5 SEPP Regression Pipeline 92
4.4.5.1 Single Click Job Creation for SEPP Regression Features 93
4.4.6 SEPP Regression Documentation 95
4.4.7 Running SEPP Health Check Pipeline 96

Copyright © 2019, 2026, Oracle and/or its affiliates.

Preface

- Documentation Accessibility
- Diversity and Inclusion

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 user supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that user enter.

Copyright © 2019, 2026, 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.

Use My Oracle Support to access support resources, create and manage Service Requests (SRs), and get assistance with product issues.

For help with My Oracle Support registration, Service Request creation, or other support access questions, use the available support options in My Oracle Support, including MOSFS and MOS Chat.

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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Acronyms

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

Table 1 Acronyms

Acronyms	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
CNE	Oracle Communications Cloud Native Core, Cloud Native Environment
cnPCRF	Oracle Communications Cloud Native Core, Policy and Charging Rules Function
CSR	Certificate Signing Request
EIR	Equipment Identity Register
NRF	Oracle Communications Cloud Native Core, Network Repository Function
NSSF	Oracle Communications Cloud Native Core, Network Slice Selection Function
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
Policy	Oracle Communications Cloud Native Core, Converged Policy
PVC	Persistent Volume Claim
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, 2026, Oracle and/or its affiliates.

What's New in This Guide

This section introduces the documentation updates for release 26.1.2xx.

ATS Framework Release 26.1.201 - G59076-01, July 2026

General Updates

- Updated the ATS framework features compliance matrix in Table 2-1.
- Updated the ATS framework versions and their corresponding supporting NF versions in the Table 1-1.
- Added the ATS Pod Logs section with a procedure for setting different log levels.

SEPP Release 26.1.201

- ATS release tag for SEPP is changed to 26.1.201.
- Updated multiple images according to release 26.1.201.
- Updated the SEPP ATS Compatibility Matrix section.
- Updated the Configuring Hosted SEPP section.
- Updated the following subsections of Creating Secrets and Supporting TLS 1.2 and TLS 1.3 section:
  - Configuring Root Certificate
  - Generating ATS Certificate
  - Creating ATS Secret and Configuring Helm chart
- Added Generating Hosted SEPP multi-fqdn Certificates and Secrets section.
- Number of new Test Cases added to pipeline in SEPP mode: 1 feature files with 20 scenarios for SEPP ATS New Feature pipeline.
- Number of Test Cases in Regression Pipeline in SEPP mode: 206 feature files with 1545 scenarios SEPP-Regression pipeline.
- Number of new Test Cases added to pipeline in Roaming Hub mode: 0 feature file with 0 scenarios for ATS New Feature pipeline.
- Number of Test Cases in Regression Pipeline in Roaming Hub mode: 156 feature files with 853 scenarios ATS Regression pipeline.
- Number of new Test Cases added to pipeline in Hosted SEPP mode: 7 feature file with 94 scenarios for ATS New Feature pipeline.
- Number of Test Cases in Regression Pipeline in Hosted SEPP mode: 8 feature files with 53 scenarios ATS Regression pipeline.

NSSF Release 26.1.201

- ATS release tag for NSSF is changed to 26.1.201.
- Updated sample screenshots as per current release.
- Number of test cases in NSSF_NewFeatures pipeline: 53 new feature files with 982 new scenarios.
- Number of test cases in the NSSF_Regression pipeline: 3 feature files with 21 scenarios.
- Added details for Deploying ATS in Kubernetes Cluster.

Copyright © 2019, 2026, Oracle and/or its affiliates.

SCP Release 26.1.201

- ATS release tag for SCP is changed to 26.1.201
- Number of test cases in SCP_NewFeatures pipeline: 38 new feature files with 317 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: 590 feature files for Release 16 and no new feature file with no new scenarios for Release 15 SCP deployments.
- Number of sanity features: 33 sanity features and 49 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 26.1.201 release updates:
  - Added the Accessing Collected Logs section with information related to accessing collected logs.
  - Added the Enabling PCAP Logs section with information to enable PCAP logs.
  - Added the Data Collection after Feature File Failure section with information about collecting dumps after the feature files failure.
  - Made the following changes in Table 3-12:
    - Added the following parameters:
      - `PVStorageClassName`
      - `PVStorage`
      - `retainPVC`
      - `prometheusClusterRole.rules`
      - `ocats-dnsstub.image.repository`
      - `ocats-dnsstub.image.tag`
      - `ocats-dnsstub.image.pullPolicy`
      - `ocats-dnsstub.replicaCount`
      - `ocats-dnsstub.service.customExtension.type`
      - `ocats-dnsstub.service.customExtension.port`
      - `ocats-dnsstub.service.ipFamilyPolicy`
    - Removed the following parameters:
      - `ocats-dnsstub.image.repository`
      - `ocats-dnsstub.image.tag`
      - `ocats-dnsstub.image.pullPolicy`
      - `ocats-dnsstub.replicaCount`
      - `ocats-dnsstub.service.customExtension.type`
      - `ocats-dnsstub.service.customExtension.port`
  - Updated the sample `26.1.201.yaml` file to enable PVC in the Enabling Persistent Volume section.
  - Updated the value of `rateDataReporterStartingOffset` from 35 to 3 in Table 3-12.
  - Updated the resources requirements in the following sections:

Copyright © 2019, 2026, Oracle and/or its affiliates.

    * Resource Requirements
    * Software Requirements

NRF Release 26.1.201

- The ATS release tag for NRF is changed to 26.1.201.
- Updated the following sections with 26.1.201 release updates:
  - Prerequisites
  - Logging into ATS
  - NRF-NewFeatures Pipeline
  - NRF-NewFeatures Documentation
  - NRF-Regression Pipeline
- Updated the memory and CPU value in the Resource Requirements section.
- Updated the value of `SUBSCRIPTION_WAIT_TIME` in `preTestConfig.sh` is increased from 0 to 0.25 in the NRF-NewFeatures Pipeline and Prerequisites sections.
- Added the two new stubs in the `preTestConfig.sh` script for the Prerequisites to Run Testcases for the NRF Growth Features in the Prerequisites section.
- **Number of new Test Cases added to pipeline**:
  - **New Features**: **10** feature files with **25** scenarios.
  - **Regression**: **603** feature files with **1711** scenarios.
    * "2-Site-Geo" cases: 7 feature files with 19 scenarios.
    * "3-Site-Geo" cases: 10 feature files with 15 scenarios.
    * "NF-FQDN-Authentication-Feature" cases: 6 feature files with 28 scenarios.
    * "CCA-Header" cases: 6 feature files with 20 scenarios.
    * "NRF-Growth" cases: 23 feature files with 32 scenarios.
  - **Total regression count**: **655** feature files with **1825** scenarios.

Copyright © 2019, 2026, 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, Network Repository Function (NRF)
- Oracle Communications Cloud Native Core, Network Slice Selection Function (NSSF)
- Oracle Communications Cloud Native Core, Service Communication Proxy (SCP)
- Oracle Communications Cloud Native Core, Security Edge Protection Proxy (SEPP)

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.

ATS Framework Version with Supporting NF Version

Table 1-1 ATS Framework Version with Supporting NF

ATS Framework Version	NRF	NSSF	SCP	SEPP
26.1.200	-	-	-	-

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 1-1 (Cont.) ATS Framework Version with Supporting NF

ATS Framework Version	NRF	NSSF	SCP	SEPP
26.1.201	26.1.201	26.1.201	26.1.201	26.1.200

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

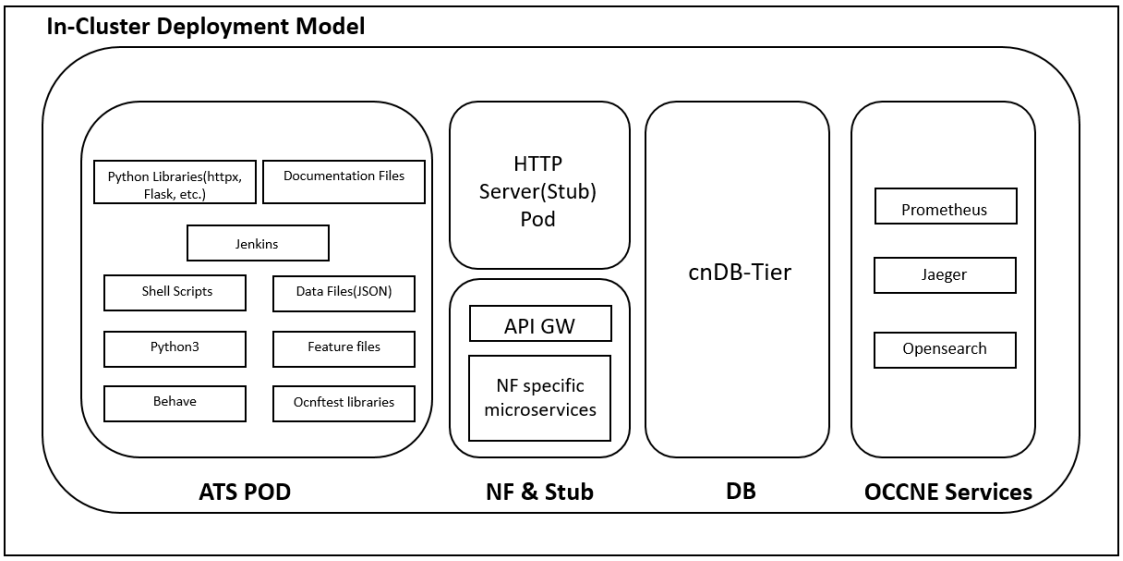
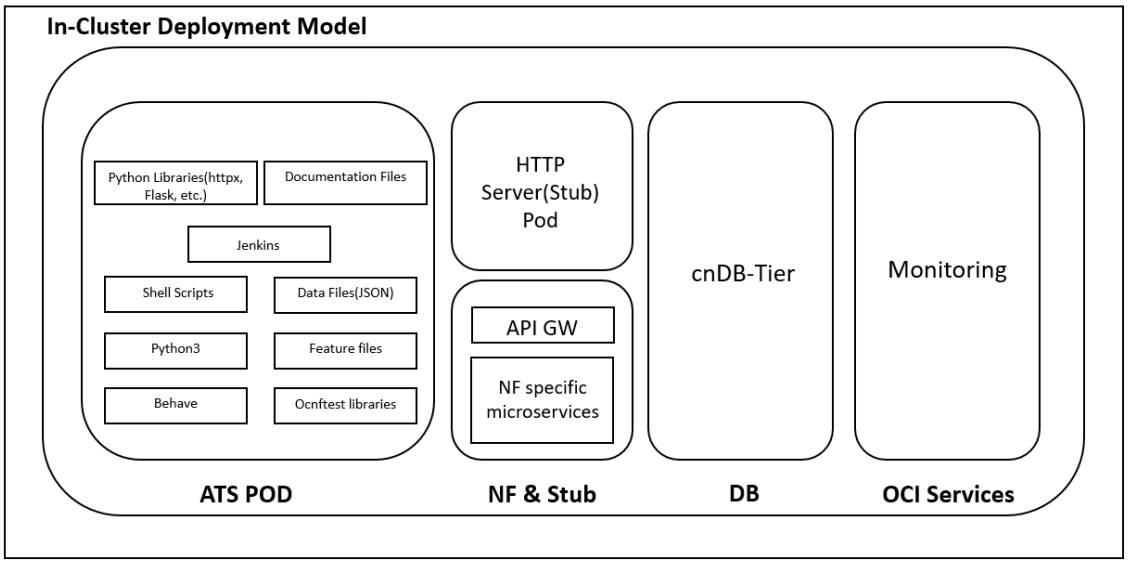


Figure 1-2 In-Cluster Deployment Model in OCI



Copyright © 2019, 2026, Oracle and/or its affiliates.

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, 2026, 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	NRF	NSSF	SCP	SEPP
ATS API	Yes	No	Yes	Yes
ATS Custom Abort	Yes	Yes	Yes	Yes
ATS Feature Activation and Deactivation	No	No	No	Yes
ATS GUI Enhancements	Yes	Yes	Yes	Yes
ATS Health Check	No	No	Yes	Yes
ATS Jenkins Job Queue	No	No	Yes	Yes
Application Log Collection	No	No	Yes	Yes
ATS Maintenance Scripts	Yes	Yes	Yes	Yes
ATS System Name and Version Display on Jenkins GUI	Yes	Yes	Yes	Yes
ATS Tagging Support	No	No	Yes	Yes
Custom Folder Implementation	Yes	Yes	Yes	Yes
Health Check	No	Yes	No	Yes
Individual Stage Group Selection	No	No	Yes	Yes
Lightweight Performance	No	No	No	No
Managing Final Summary Report, Build Color, and Application Log	Partially compliant (Application Log is not supported.)	Yes	Yes	Yes
Modifying Login Password	Yes	Yes	Yes	Yes
Multiselection Capability for Features and Scenarios	Partially compliant (Feature level selection is supported)	No	Yes	Yes
Parallel Test Execution	No	Partially compliant (Parallel Test Execution Framework integrated, but only supports sequential execution)	Yes	Partially compliant (Parallel Test Execution Framework integrated, but ATS test cases need to be organized to utilize the parallel execution)

Copyright © 2019, 2026, Oracle and/or its affiliates.

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

Features	NRF	NSSF	SCP	SEPP
Parameterization	Partially compliant (Supports only new features)	No	Yes	Yes
PCAP Log Collection	No	No	Yes	Yes
Persistent Volume	Yes	Yes	Yes	Yes
Single Click Job Creation	Yes	Yes	Yes	Yes
Support for ATS Deployment in OCI	No	No	Yes	Yes
Support for Transport Layer Security	No	No	Yes	Yes
Test Result Analyzer	Yes	Yes	Yes	Yes
Test Case mapping and Count	No	No	Yes	Yes

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.

> **Note**
>
> You can configure the `deploymentMode` Helm parameter, which supports the dual stack mechanism, to establish connections with ATS GUI and APIs. For more information, see Table 2-5.

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

2.1.1 Generating an API Token for a User

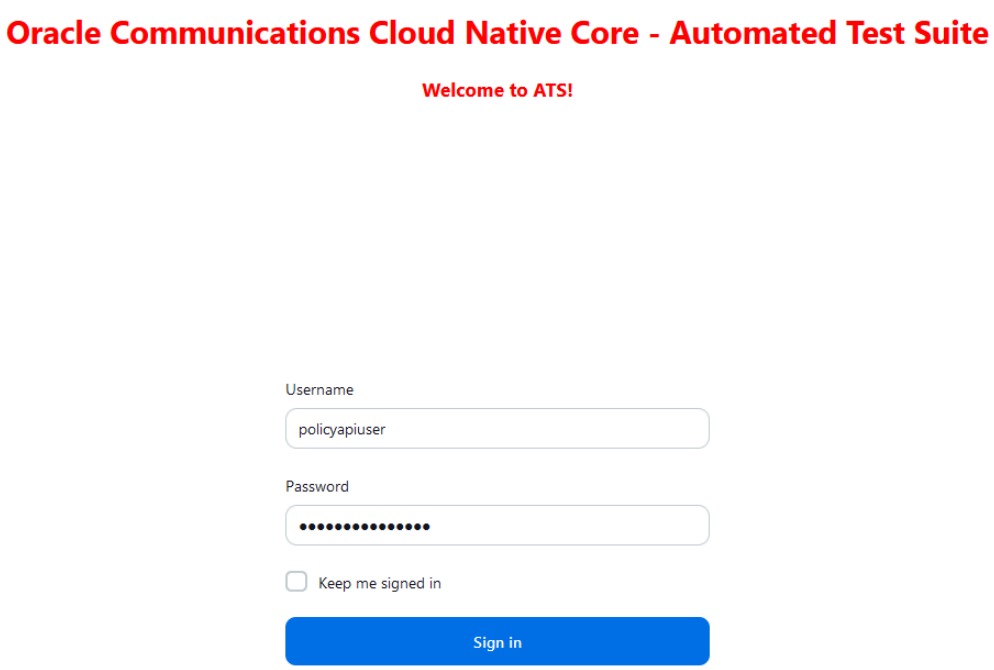
An API token that has to be generated for the user to perform routine ATS tasks using the Restful Interfaces API. Any API call requires the use of an API token for authentication. You can generate the API token, and it works until it is revoked or deleted.

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.

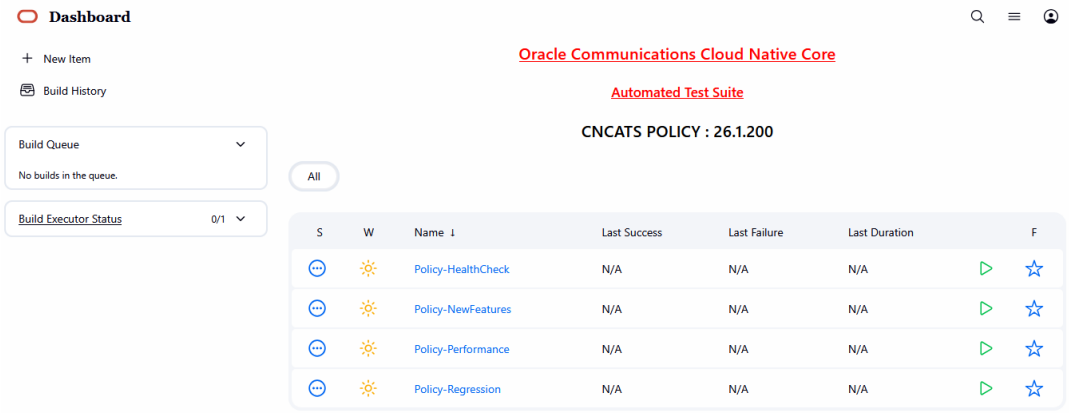
Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-1 ATS Login Page



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

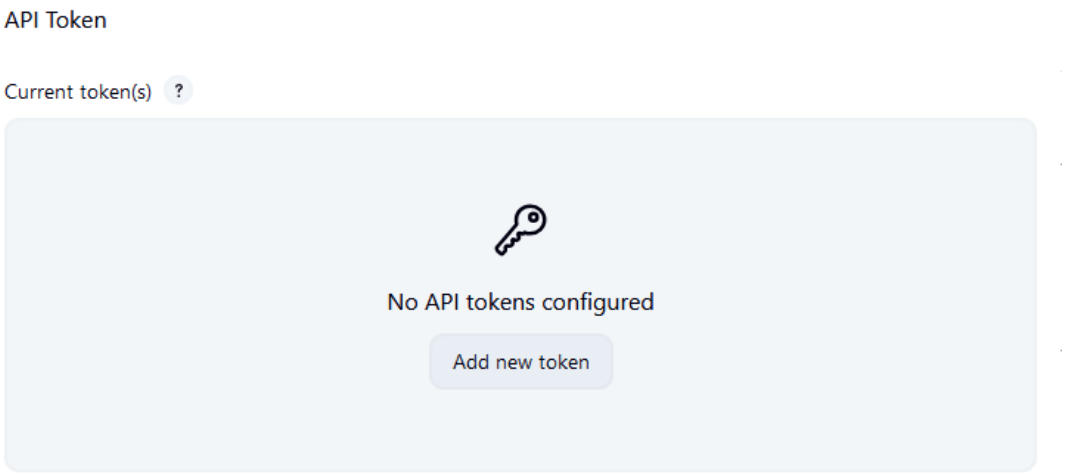
Figure 2-2 Add Token



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

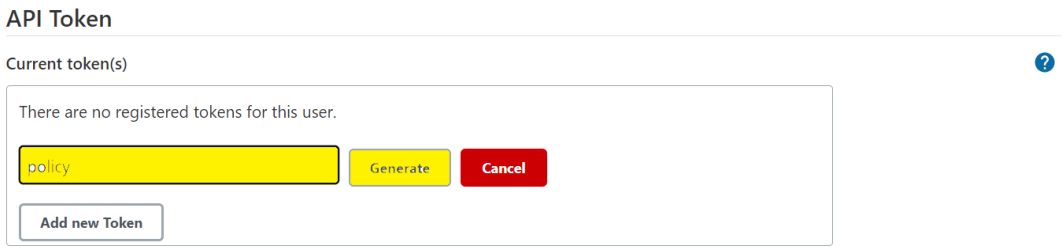
Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-3 Add New Token



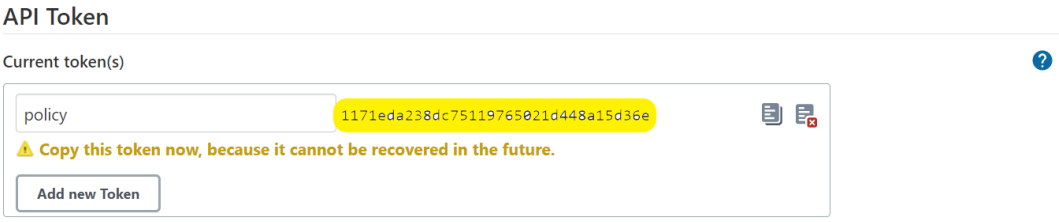
4. Enter a suitable name for the token, such as policy, and then click **Generate**.

Figure 2-4 Generate Token



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

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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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

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

Copyright © 2019, 2026, Oracle and/or its affiliates.

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
```

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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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"}'
    --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)

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 2-2 (Cont.) API Parameters

Parameters	Mandatory	Default Value	Description
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?`.
   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 `&`.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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
```

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": {
    }
  }' \
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
--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:

Table 2-3 otherBuildParameters Details

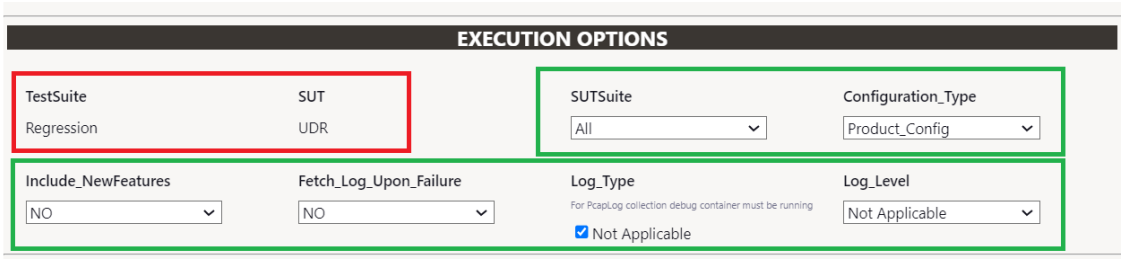
Parameter	Mandatory/Optional	Default Value	Description
otherBuildParameters	Optional	NA	"otherBuildParameters" is a dictionary of build parameters. You can add key-value pairs to customize the build process. For example:`"otherBuildParameters" : { "Features" : "YamlSchema_Import_Export,Custom_Jsons", "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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-6 Execution Option



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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

Starting a job with otherBuildParameters

Run 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>" }
```

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
```

Run 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

Copyright © 2019, 2026, Oracle and/or its affiliates.

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

Run 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
```

Run Groups

Copyright © 2019, 2026, Oracle and/or its affiliates.

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

Run with Tags

In the API request, you 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 you, and if any feature or scenario is provided, they are not considered. Only tags and stages or groups are provided as part of the request.

Tags can be specified in the request within the "otherBuildParameters" dictionary as follows:

```
"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?FilterWithTags=Yes",
    "otherBuildParameters": {
      "Feature_Include_Tags":"cne-common,cm-service",
"Scenario_Include_Tags":"cleanup,sanity"
    }
  }' \
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
--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 run 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 run.

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:

- 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
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
<username>:<API_token> --verbose
      --cacert<path_to_root_certificate>
```

For example,

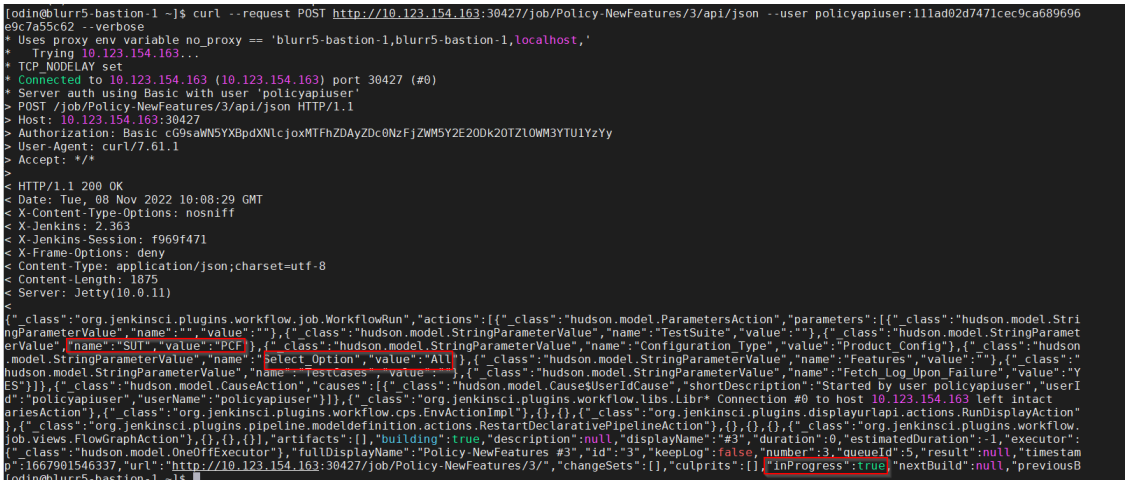
```
curl --request POST http://10.123.154.163:30427/job/Policy-
NewFeatures/3/api/json--user
policyapiuser:111ad02d7471cec9ca689696e9c7a55c62
          --verbose
```

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
```

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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

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
```

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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

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
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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

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
```

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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

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/
      *.<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,

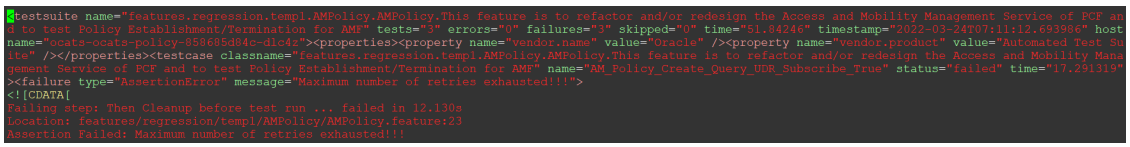
```
curl --request POST http://
      10.75.217.4:31881/job/Policy-NewFeatures/21/artifact/test-results/
reports/*.goldenfeature.xml,*.AMPolicy.xml/*zip*/test-results.zip
      –user policyapiuser:11c3344996c4fda01ded2124bec4f9aa17 --verbose  --
output test-results.zip
```

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

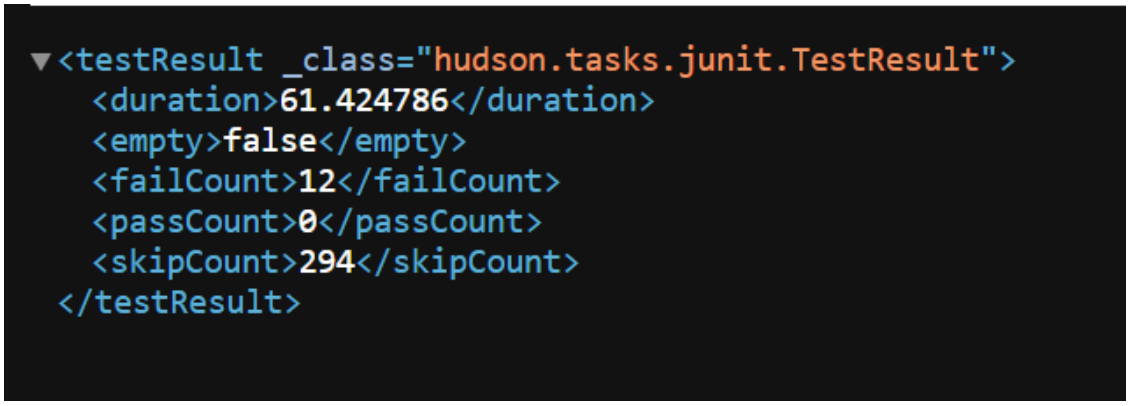


In the API call, specify other selected features in comma-separated form as `/*<Feature1_name>.xml,*<Feature2_name>.xml,*<Feature3_name>.xml,*<Feature4_name>.xml/` for Select_Option = Single/MultipleFeatures.

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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-9 Sample Output



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.

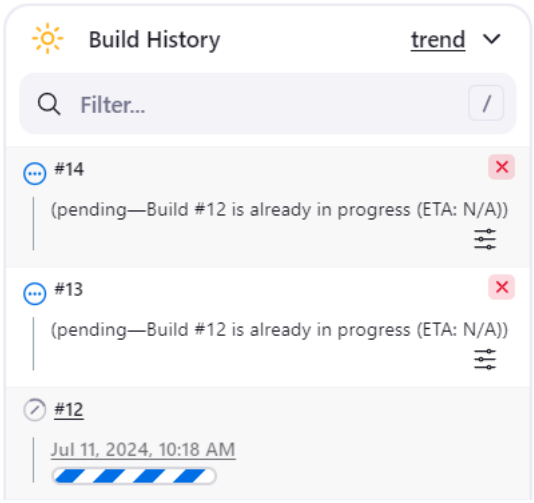
To address the issues with manual abort, the release 24.3.0 introduces an **Abort_Build** menu in all new features and regression pipelines, ensuring a more controlled and graceful termination of builds.

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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-10 Stopping Builds



Abort_Build

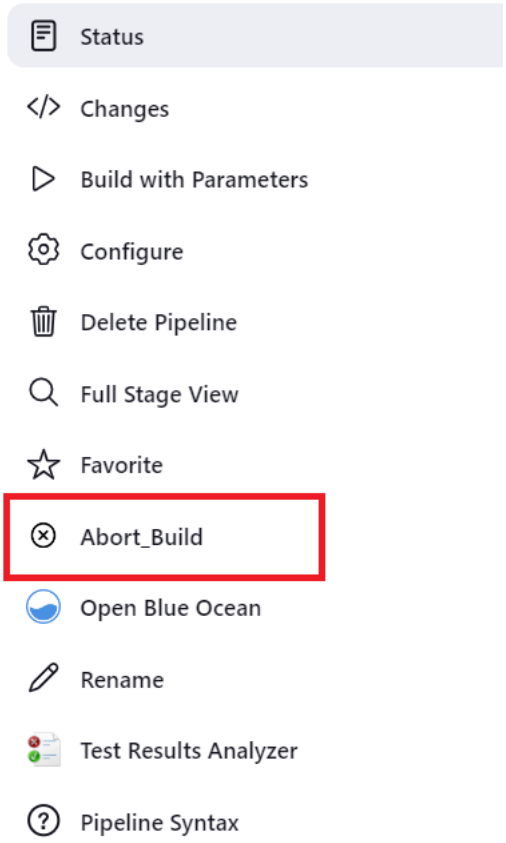
The **Abort_Build** is available on every regression and new features pipeline. It supports the graceful termination of builds. This **Abort** button available in the GUI triggers a stop API request to the ATS API server. This request ensures that the execution is stopped gracefully, allowing all currently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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-11 Abort Build

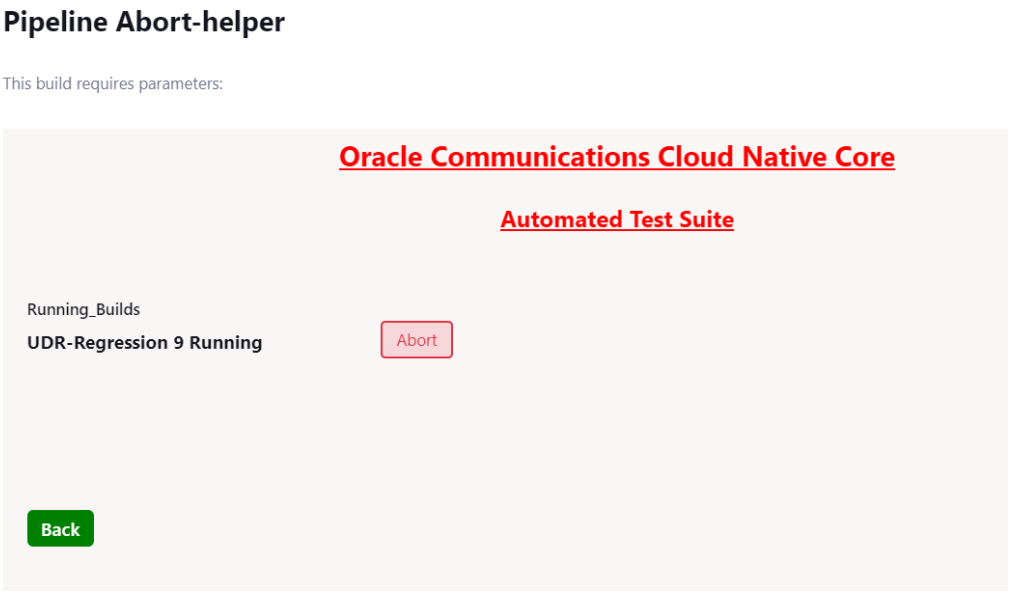


You will be redirected to the Pipeline Abort-helper page.

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.

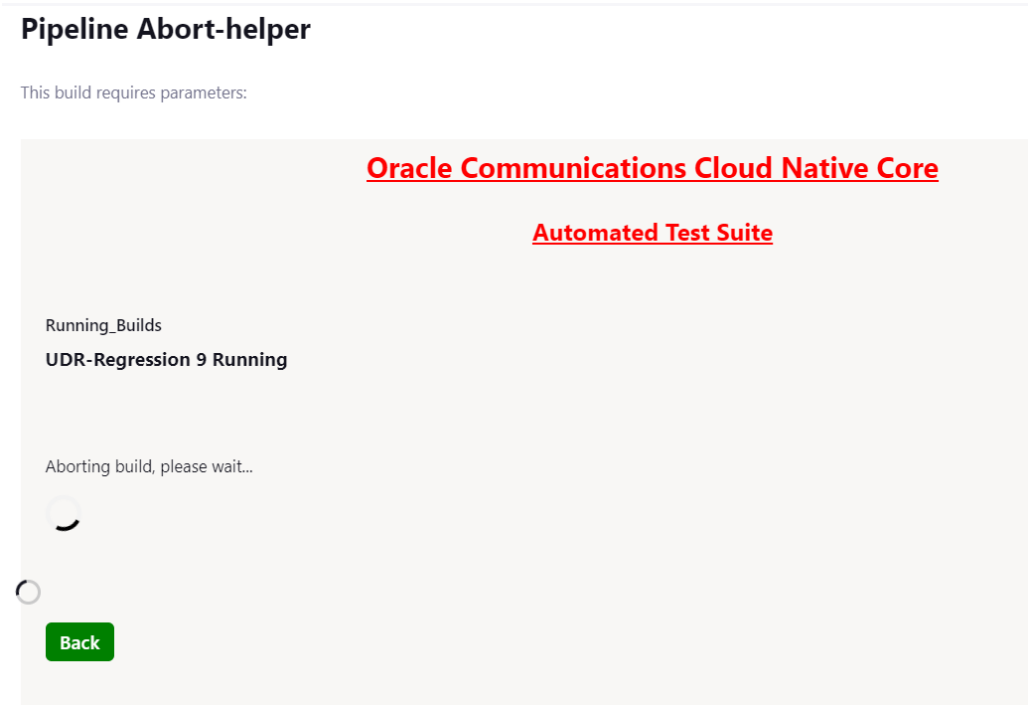
Copyright © 2019, 2026, Oracle and/or its affiliates.

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, 2026, 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**
>
> When 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, 2026, 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, 2026, 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.
Prometheus Requires Token-based Authentication ATS connects to Prometheus to retrieve metrics and alerts for validating traffic and configurations during automated testing. Prometheus continuously collects metrics from NFs' services as a central platform for accessing metrics and alerts. You can configure ATS to enable or disable authentication and TLS for communication with Prometheus. Authentication can be enabled if Prometheus requires Role-Based Access Control (RBAC), utilizing a service account token for authorization. ATS uses the token located at `/var/run/secrets/kubernetes.io/serviceaccount/token`, which is included as a bearer token in each request to Prometheus. Kubernetes manages the token, ensuring that any expired tokens are automatically updated within the pod.	`PrometheusAuthEnabled`	Set this parameter to `true` when Prometheus requests require a token in the header for authentication.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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

Features	Parameter	Description
	PrometheusTLSEnabled	Set this parameter to true when Prometheus has HTTPS deployment. **Note:** When the connection is established using HTTPS, it remains insecure because no certificates are maintained with ATS for this connection. Sample configuration: #The below section is related to the deployments where prometheus needs authenticated token to request for metrics/alerts PrometheusAuthEnabled: false #Set this to true only when prometheus reqeusts requrire a token in the header for authentication PrometheusTLSEnabled: false #Set this to true only when prometheus is a https deployment. #PrometheusScrapeInterval: "60" #Uncomment this line only when scrapeInterval is any other value than 60 seconds.(ATS will assume 60s as default if no response comes from prometheus config.) # Optionally override ClusterRole rules used when PrometheusAuthEnabled is true. prometheusClusterRole: # If rules is empty or this section is omitted, the default rule is used: # - apiGroups: [""] # resources: ["namespaces"] # verbs: ["get"] # - apiGroups: ["monitoring.coreos.com"] # resources: ["prometheuses/api"] # resourceNames: ["k8s"]

Copyright © 2019, 2026, Oracle and/or its affiliates.

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

Features	Parameter	Description

		`# verbs: ["get", "create", "update"]`
		`# - nonResourceURLs: ["/*"]`
		`# verbs: ["get"]`
		`rules: []`
		`# Example sample rules:`
		`# rules:`
		`# - apiGroups: [""]`
		`# resources: ["pods", "services"]`
		`# verbs: ["get", "list"]`
		`# - nonResourceURLs:`
		`# - "/*"`
		`# verbs:`
		`# - get`

Support for Dual Stack	`deploymentMode`	Set this parameter to one of the possible values to enable the dual stack mechanism. The dual stack mechanism enables ATS and stubs to establish connections with REST APIs and ATS GUI in a Kubernetes cluster using IPv4 or IPv6 or both simultaneously. Default value: ClusterPreferred Possible values: IPv4, IPv6, IPv4_IPv6, IPv6_IPv4, and ClusterPreferred. **Note:** This parameter is supported only on Kubernetes 1.23.0 and later.

Note

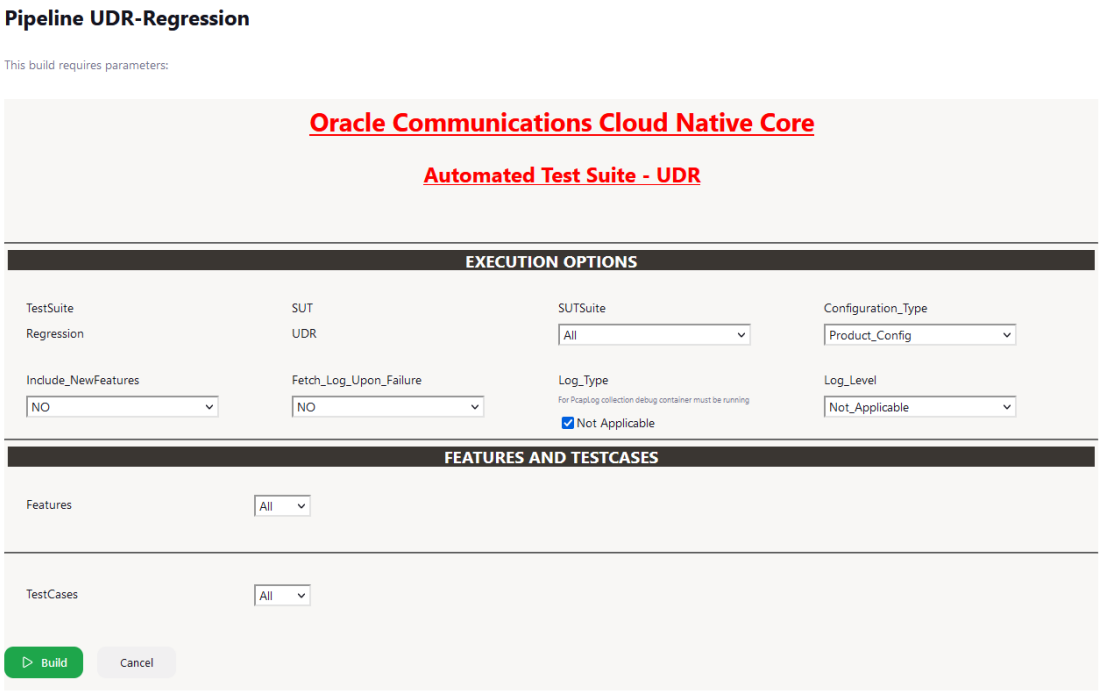
- You can edit the parameters relating to the features that NF supports. Keep the default value for the remaining parameters.
- For the current release, mergedExecution, individualStageGroupSelection, and parallelTestCaseExecution parameters value should not be modified.

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.

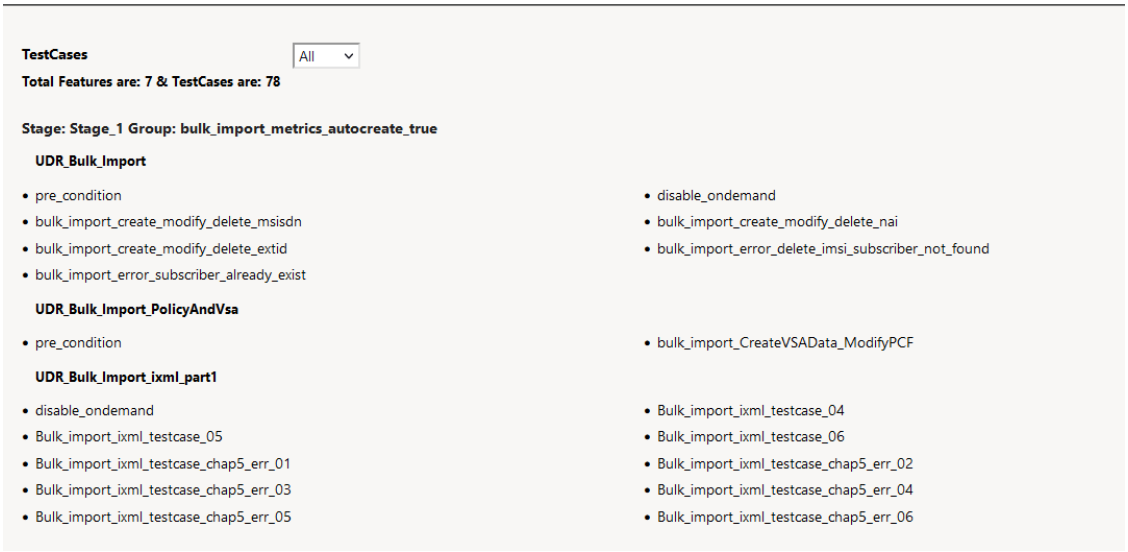
Copyright © 2019, 2026, Oracle and/or its affiliates.

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



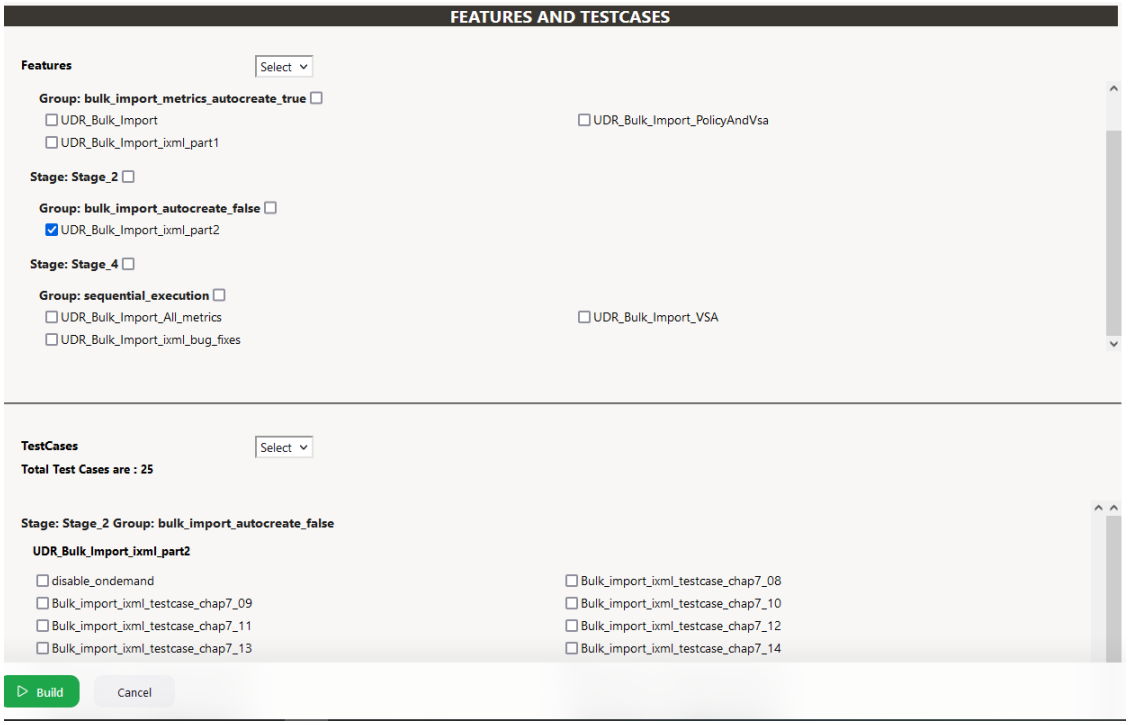
Tooltips have been introduced to allow user to have quick insights into the grouping and staging of individual test cases, thereby improving overall understanding and navigation within the ATS environment.

Copyright © 2019, 2026, 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 Pod Logs

ATS pod logs are enhanced with different log categories to provide detailed and structured logging to improve the debugging and analysis process.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> To view the logs, run the following command:
>
> ```
> kubectl logs <ATS_POD_NAME> -n <NAMESPACE>
> ```

Perform the following procedure to change the log levels.

1. Log in to the ATS GUI using the login credentials.
2. On the ATS home page, from the **More actions** menu in the upper right area, select **Change ATS Log Level**.
3. From the **ATS_LogLevel** drop-down list, select the required log level.

   The different log levels are as follows:

   - **INFO**: Enables default ATS audit and application logs for high-level debugging, including user actions, build triggers or cancellations, build status, and success or failure summaries. This is the default log level.
   - **DEBUG**: Enables detailed Jenkins and ATS API logs, including INFO logs, job URLs, internal Jenkins API calls, request flow, authentication, endpoint handling, and detailed execution decisions.

     > **Note**
     >
     > The INFO level logs provides sufficient information about activities in ATS. DEBUG logs are used in certain scenarios, such as reporting an issue in ATS.

   - **WARN**: At Jenkins level, WARN falls back to DEBUG because Jenkins-specific WARN audit logs are not separately supported. At the ATS API level, WARN displays only warning and higher-severity messages such as WARNING, ERROR, and CRITICAL.

   > **Note**
   >
   > You cannot switch between log levels while any build is running in ATS. Switching works only after all currently running builds are complete.

   Sample INFO Logs:

   ```
   2026-02-13 11:01:55:277 - JENKINS-INFO - /login by NA from 10.233.101.0
   2026-02-13 11:02:03:612 - JENKINS-INFO - /adjuncts/8a9821e0/hudson/views/
   BuildButtonColumn/icon.js by guiadmin from 10.233.101.0
   2026-02-13 11:02:48:530 - JENKINS-INFO - /job/Policy-Regression/
   configSubmit by guiadmin from 10.233.101.0
   2026-02-13 11:02:48.572+0000 [id=26]    INFO
   h.p.sidebar_link.LinkAction#<init>: Created link 'Abort_Build': url='/job/
   Abort-helper/build?delay=0sec', icon='/userContent/layout/abort.svg'
   2026-02-13 11:04:39.819 - FLASK-INFO - route /: default route called
   method=GET
   2026-02-13 11:04:39.821 - FLASK-INFO - 127.0.0.1 - - [13/Feb/2026
   11:04:39] "GET / HTTP/1.1" 200 -
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-13 11:10:15.464 - FLASK-INFO - 127.0.0.1 - - [13/Feb/2026
11:10:15] "GET /job/Policy-Regression/1/stop HTTP/1.1" 405 -
2026-02-13 11:10:29.458 - FLASK-INFO - route /job/Policy-Regression/1/
stop: called
2026-02-13 11:10:29.545 - FLASK-INFO - valid_pipeline: SUCCESS
pipeline_name=Policy-Regression
2026-02-13 11:10:29.599 - FLASK-WARNING - is_build_running: build not
found pipeline=Policy-Regression job_id=1
2026-02-13 11:10:29.600 - FLASK-INFO - stop: completed pipeline=Policy-
Regression job_id=1 status=404 response_len=20
2026-02-13 11:10:29.600 - FLASK-INFO - 127.0.0.1 - - [13/Feb/2026
11:10:29] "POST /job/Policy-Regression/1/stop HTTP/1.1" 404 -
2026-02-13 11:11:17:976 - JENKINS-INFO - job/Policy-Regression/ #1 Started
by user guiadmin, Parameters:[: {}, : {}, TestSuite: {}, SUT: {PCF},
Configuration_Type: {Product_Config}, Include_NewFeatures: {NO},
FilterWithTags: {true,false,}, shell_executor: {}, Feature_Include_Tags:
{}, Feature_Exclude_Tags: {}, Scenario_Include_Tags: {},
Scenario_Exclude_Tags: {}, json_writer: {}, Submit_Tags: {},
Fetch_Log_Upon_Failure: {YES}, Log_Validation: {NO}, : {},
Select_Features_Option: {Select}, Features:
{false,false,true,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,},  : {}, Select_Testcases_Option: {All}, TestCases: {}]
2026-02-13 11:11:28.531 - FLASK-INFO - route /job/Policy-Regression/1/
stop: called
2026-02-13 11:11:28.583 - FLASK-INFO - valid_pipeline: SUCCESS
pipeline_name=Policy-Regression
2026-02-13 11:11:28.605 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-13 11:11:28.615 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-13 11:11:28.632 - FLASK-INFO - send_abort_request: pipeline=Policy-
Regression job_id=1 immediate=None
2026-02-13 11:11:28:641 - JENKINS-INFO - /job/Policy-Regression/1/stop by
guiadmin from 127.0.0.1
2026-02-13 11:11:28.944 - FLASK-INFO - send_abort_request: stop request
status=200
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-13 11:11:38.963 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-13 11:11:38.986 - FLASK-INFO - send_abort_request: pipeline=Policy-
Regression job_id=1 immediate=None
2026-02-13 11:11:38:995 - JENKINS-INFO - /job/Policy-Regression/1/stop by
guiadmin from 127.0.0.1
2026-02-13 11:11:39.070 - FLASK-INFO - send_abort_request: stop request
status=200
2026-02-13 11:11:43.646+0000 [id=58]    WARNING
o.j.p.w.s.s.i.POSTHyperlinkNote#<init>: You need to define the root URL of
Jenkins
2026-02-13 11:11:49.083 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-13 11:11:49.123 - FLASK-INFO - send_abort_request: pipeline=Policy-
Regression job_id=1 immediate=None
2026-02-13 11:11:49:143 - JENKINS-INFO - /job/Policy-Regression/1/stop by
guiadmin from 127.0.0.1
2026-02-13 11:11:49.210 - FLASK-INFO - send_abort_request: stop request
status=200
2026-02-13 11:11:53.997+0000 [id=56]    WARNING
o.j.p.w.s.s.i.POSTHyperlinkNote#<init>: You need to define the root URL of
Jenkins
2026-02-13 11:11:59.294 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-13 11:12:04.144+0000 [id=55]    WARNING
o.j.p.w.s.s.i.POSTHyperlinkNote#<init>: You need to define the root URL of
Jenkins
2026-02-13 11:12:04.342 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-13 11:12:07:989 - JENKINS-INFO - Policy-Regression #1 Started by
user guiadmin, Parameters:[: {}, : {}, TestSuite: {}, SUT: {PCF},
Configuration_Type: {Product_Config}, Include_NewFeatures: {NO},
FilterWithTags: {true,false,}, shell_executor: {}, Feature_Include_Tags:
{}, Feature_Exclude_Tags: {}, Scenario_Include_Tags: {},
Scenario_Exclude_Tags: {}, json_writer: {}, Submit_Tags: {},
Fetch_Log_Upon_Failure: {YES}, Log_Validation: {NO}, : {},
Select_Features_Option: {Select}, Features:
{false,false,true,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,}, : {}, Select_Testcases_Option: {All}, TestCases: {}] on
Built-In Node started at 2026-02-13T11:11:17Z completed in 49984ms
completed: ABORTED
2026-02-13 11:12:09.375 - FLASK-INFO - is_build_running: build not running
with values, pipeline=Policy-Regression job_id=1
2026-02-13 11:12:09.376 - FLASK-INFO - stop: completed pipeline=Policy-
Regression job_id=1 status=200 response_len=15
2026-02-13 11:12:09.376 - FLASK-INFO - 127.0.0.1 - - [13/Feb/2026
11:12:09] "POST /job/Policy-Regression/1/stop HTTP/1.1" 200 -
```

Sample DEBUG Logs:

```
2026-02-16 05:57:24:099 - JENKINS-DEBUG - /login by NA from 10.233.101.0
2026-02-16 05:57:28:353 - JENKINS-DEBUG - /job/Policy-Regression/ by
guiadmin from 10.233.101.0
2026-02-16 05:57:32:165 - JENKINS-DEBUG - /static/aa795ad3/plugin/
blueocean-rest-impl/images/48x48/blueocean.png by guiadmin from
10.233.101.0
2026-02-16 05:57:34:137 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:57:39:430 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:57:40:430 - JENKINS-DEBUG - /job/Policy-Regression/configure
by guiadmin from 10.233.101.0
2026-02-16 05:57:49:627 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/hudson.security.AuthorizationMatrixProperty/checkName by
guiadmin from 10.233.101.0
2026-02-16 05:57:49:824 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/jenkins.branch.RateLimitBranchProperty$JobPropertyImpl/
fillDurationNameItems by guiadmin from 10.233.101.0
2026-02-16 05:57:50:468 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/hudson.security.AuthorizationMatrixProperty/checkName by
guiadmin from 10.233.101.0
2026-02-16 05:57:50:695 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.pipeline.modeldefinition.properties.PreserveStashesJo
bProperty/checkBuildCount by guiadmin from 10.233.101.0
2026-02-16 05:57:51:607 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/jenkins.branch.RateLimitBranchProperty$JobPropertyImpl/
checkCount by guiadmin from 10.233.101.0
2026-02-16 05:57:51:832 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/jenkins.triggers.ReverseBuildTrigger/
checkUpstreamProjects by guiadmin from 10.233.101.0
2026-02-16 05:57:52:087 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/org.jenkinsci.plugins.workflow.job.WorkflowJob/
checkDisplayNameOrNull by guiadmin from 10.233.101.0
2026-02-16 05:57:52:587 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:52:867 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:53:095 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:53:324 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:53:547 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:53:776 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:54:003 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:54:231 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:54:460 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:54:688 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:54:912 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:55:139 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:55:364 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:55:600 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:55:823 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:56:310 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:56:534 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:56:838 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:57:062 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:57:320 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:57:544 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:57:796 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:58:048 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:58:344 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:58:580 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:59:037 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:59:264 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:59:490 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:59:715 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:57:59:943 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:00:167 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:00:394 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:00:947 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:01:171 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:01:394 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:01:619 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:01:843 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:02:161 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:02:398 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:02:626 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:02:918 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:03:147 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:03:373 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:03:706 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SecureGroovyScript/
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:03:941 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/hudson.triggers.TimerTrigger/checkSpec by guiadmin from
10.233.101.0
2026-02-16 05:58:04:166 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/hudson.triggers.SCMTrigger/checkScmpoll_spec by guiadmin
from 10.233.101.0
2026-02-16 05:58:04:618 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/org.jenkinsci.plugins.workflow.cps.CpsFlowDefinition/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:38:546 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/org.jenkinsci.plugins.workflow.cps.CpsFlowDefinition/
checkScriptCompile by guiadmin from 10.233.101.0
2026-02-16 05:58:39:424 - JENKINS-DEBUG - /job/Policy-Regression/
descriptorByName/org.jenkinsci.plugins.workflow.cps.CpsFlowDefinition/
checkScript by guiadmin from 10.233.101.0
2026-02-16 05:58:39:835 - JENKINS-DEBUG - /job/Policy-Regression/
configSubmit by guiadmin from 10.233.101.0
2026-02-16 05:58:39.883+0000 [id=86]    INFO
h.p.sidebar_link.LinkAction#<init>: Created link 'Abort_Build': url='/job/
Abort-helper/build?delay=0sec', icon='/userContent/layout/abort.svg'
2026-02-16 05:58:40:253 - JENKINS-DEBUG - /job/Policy-Regression/ by
guiadmin from 10.233.101.0
2026-02-16 05:58:41:574 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:58:46:823 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:58:52:590 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:58:58:598 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:59:04.079 - FLASK-DEBUG - Default route: called method=GET
2026-02-16 05:59:04.079 - FLASK-INFO - route /: default route called
method=GET
2026-02-16 05:59:04.080 - FLASK-INFO - 127.0.0.1 - - [16/Feb/2026
05:59:04] "GET / HTTP/1.1" 200 -
2026-02-16 05:59:04:582 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:59:10:586 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:59:16:587 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:59:22:595 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:59:28:598 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:59:34:591 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:59:40:597 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:59:46:584 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:59:52:679 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 05:59:55.652 - FLASK-INFO - 127.0.0.1 - - [16/Feb/2026
05:59:55] "GET /job/Policy-Regression/1/stop HTTP/1.1" 405 -
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-16 05:59:58:599 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:00:04:594 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:00:10:598 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:00:11.533 - FLASK-DEBUG - stop: called pipeline=Policy-
Regression job_id=1 method=POST
2026-02-16 06:00:11.533 - FLASK-INFO - route /job/Policy-Regression/1/
stop: called
2026-02-16 06:00:11.533 - FLASK-DEBUG - stop: immediate=None
2026-02-16 06:00:11.534 - FLASK-DEBUG - authenticate_request: called
2026-02-16 06:00:11.534 - FLASK-DEBUG - authenticate_request: received
authorization username=guiadmin
2026-02-16 06:00:11.534 - FLASK-DEBUG - create_session: called
user=guiadmin tls=True
2026-02-16 06:00:11.534 - FLASK-DEBUG - create_session:
session.verify=/tmp/caroot.cer
2026-02-16 06:00:11.535 - FLASK-DEBUG - authenticate_request: validating
creds via https to 127.0.0.1:8443
2026-02-16 06:00:11.535 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name] has_data=False
2026-02-16 06:00:11.535 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name] timeout=(10, 27)
delete_session=False
2026-02-16 06:00:11.536 - FLASK-DEBUG - Starting new HTTPS connection (1):
127.0.0.1:8443
2026-02-16 06:00:11.662 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /api/
json?tree=jobs%5Bname%5D HTTP/1.1" 200 368
2026-02-16 06:00:11.663 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:00:11.663 - FLASK-DEBUG - authenticate_request: auth
validation status_code=200
2026-02-16 06:00:11.664 - FLASK-DEBUG - valid_job_id: called job_id=1
2026-02-16 06:00:11.664 - FLASK-DEBUG - valid_pipeline: called
pipeline_name=Policy-Regression
2026-02-16 06:00:11.665 - FLASK-DEBUG - valid_pipeline: checking existence
pipeline_name=Policy-Regression endpoint=https://127.0.0.1:8443/api/json?
tree=jobs[name]
2026-02-16 06:00:11.665 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name]
has_data=False
2026-02-16 06:00:11.665 - FLASK-DEBUG - jenkins_request: creating session
from request auth for endpoint=https://127.0.0.1:8443/api/json?
tree=jobs[name]
2026-02-16 06:00:11.665 - FLASK-DEBUG - create_session: called
user=guiadmin tls=True
2026-02-16 06:00:11.665 - FLASK-DEBUG - create_session:
session.verify=/tmp/caroot.cer
2026-02-16 06:00:11.665 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name] timeout=(10, 27)
delete_session=True
2026-02-16 06:00:11.667 - FLASK-DEBUG - Starting new HTTPS connection (1):
127.0.0.1:8443
2026-02-16 06:00:11.690 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /api/
json?tree=jobs%5Bname%5D HTTP/1.1" 200 368
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-16 06:00:11.691 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:00:11.691 - FLASK-DEBUG - close_session: closing session
2026-02-16 06:00:11.691 - FLASK-INFO - valid_pipeline: SUCCESS
pipeline_name=Policy-Regression
2026-02-16 06:00:11.692 - FLASK-DEBUG - valid_job_id: called job_id=1
2026-02-16 06:00:11.692 - FLASK-DEBUG - create_session: called
user=guiadmin tls=True
2026-02-16 06:00:11.692 - FLASK-DEBUG - create_session:
session.verify=/tmp/caroot.cer
2026-02-16 06:00:11.692 - FLASK-DEBUG - is_build_running: called
pipeline=Policy-Regression job_id=1
2026-02-16 06:00:11.692 - FLASK-DEBUG - is_build_running: endpoint=https://
127.0.0.1:8443/job/Policy-Regression/1/api/json?tree=building
2026-02-16 06:00:11.692 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/
json?tree=building has_data=False
2026-02-16 06:00:11.692 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:11.694 - FLASK-DEBUG - Starting new HTTPS connection (1):
127.0.0.1:8443
2026-02-16 06:00:11:715 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:00:11.748 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:11.753 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:11.753 - FLASK-WARNING - is_build_running: build not
found pipeline=Policy-Regression job_id=1
2026-02-16 06:00:11.754 - FLASK-DEBUG - close_session: closing session
2026-02-16 06:00:11.754 - FLASK-INFO - stop: completed pipeline=Policy-
Regression job_id=1 status=404 response_len=20
2026-02-16 06:00:11.754 - FLASK-DEBUG - response_generator: called
status=404 message_type=str
2026-02-16 06:00:11.754 - FLASK-INFO - 127.0.0.1 - - [16/Feb/2026
06:00:11] "POST /job/Policy-Regression/1/stop HTTP/1.1" 404 -
2026-02-16 06:00:16:584 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:00:22:587 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:00:28:603 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:00:34:593 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:00:40:589 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:00:46:582 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:00:52:589 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:00:55.322 - FLASK-DEBUG - triggerBuild: called method=POST
2026-02-16 06:00:55.322 - FLASK-INFO - route /build: called
2026-02-16 06:00:55.322 - FLASK-DEBUG - authenticate_request: called
2026-02-16 06:00:55.322 - FLASK-DEBUG - authenticate_request: received
authorization username=guiadmin
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-16 06:00:55.322 - FLASK-DEBUG - create_session: called
user=guiadmin tls=True
2026-02-16 06:00:55.322 - FLASK-DEBUG - create_session:
session.verify=/tmp/caroot.cer
2026-02-16 06:00:55.322 - FLASK-DEBUG - authenticate_request: validating
creds via https to 127.0.0.1:8443
2026-02-16 06:00:55.323 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name] has_data=False
2026-02-16 06:00:55.323 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name] timeout=(10, 27)
delete_session=False
2026-02-16 06:00:55.324 - FLASK-DEBUG - Starting new HTTPS connection (1):
127.0.0.1:8443
2026-02-16 06:00:55.355 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /api/
json?tree=jobs%5Bname%5D HTTP/1.1" 200 368
2026-02-16 06:00:55.355 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:00:55.355 - FLASK-DEBUG - authenticate_request: auth
validation status_code=200
2026-02-16 06:00:55.355 - FLASK-DEBUG - valid_pipeline: called
pipeline_name=Policy-Regression
2026-02-16 06:00:55.356 - FLASK-DEBUG - valid_pipeline: checking existence
pipeline_name=Policy-Regression endpoint=https://127.0.0.1:8443/api/json?
tree=jobs[name]
2026-02-16 06:00:55.356 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name]
has_data=False
2026-02-16 06:00:55.356 - FLASK-DEBUG - jenkins_request: creating session
from request auth for endpoint=https://127.0.0.1:8443/api/json?
tree=jobs[name]
2026-02-16 06:00:55.356 - FLASK-DEBUG - create_session: called
user=guiadmin tls=True
2026-02-16 06:00:55.356 - FLASK-DEBUG - create_session:
session.verify=/tmp/caroot.cer
2026-02-16 06:00:55.356 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name] timeout=(10, 27)
delete_session=True
2026-02-16 06:00:55.358 - FLASK-DEBUG - Starting new HTTPS connection (1):
127.0.0.1:8443
2026-02-16 06:00:55.380 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /api/
json?tree=jobs%5Bname%5D HTTP/1.1" 200 368
2026-02-16 06:00:55.380 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:00:55.380 - FLASK-DEBUG - close_session: closing session
2026-02-16 06:00:55.381 - FLASK-INFO - valid_pipeline: SUCCESS
pipeline_name=Policy-Regression
2026-02-16 06:00:55.381 - FLASK-DEBUG - FormulatepageAndQuery: called
2026-02-16 06:00:55.381 - FLASK-INFO - FormulatepageAndQuery: start
pipeline=Policy-Regression pageAndQuery=buildWithParameters?
Configuration_Type=Custom_Config&FilterWithTags=Yes
2026-02-16 06:00:55.381 - FLASK-DEBUG - retrieve_execution_options (SUT,
Execute_Suite..): called
2026-02-16 06:00:55.381 - FLASK-DEBUG - retrieve_execution_options: param=
value=
2026-02-16 06:00:55.381 - FLASK-DEBUG - find_jenkinsData_path: called
2026-02-16 06:00:55.554 - FLASK-DEBUG - find_jenkinsData_path: resolved
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
path=/var/lib/jenkins/ocpcf_tests/jenkinsData
2026-02-16 06:00:55.554 - FLASK-INFO - FormulatepageAndQuery: tagging
enabled; executing shell for tags
2026-02-16 06:00:56.027 - FLASK-DEBUG - create_feature_tags_string (List
of all unique feature tags): called
2026-02-16 06:00:56.027 - FLASK-DEBUG - create_feature_tags_string:
json_path=/var/lib/jenkins/ocpcf_tests/jenkinsData/Cust-Policy-Regression-
PCF.json
2026-02-16 06:00:56.030 - FLASK-DEBUG - create_feature_tags_string:
unique_feature_tags_count=38
2026-02-16 06:00:56.030 - FLASK-DEBUG - create_scenario_tags_string (List
of all unique scenario tags): called
2026-02-16 06:00:56.030 - FLASK-DEBUG - create_scenario_tags_string:
json_path=/var/lib/jenkins/ocpcf_tests/jenkinsData/Cust-Policy-Regression-
PCF.json
2026-02-16 06:00:56.033 - FLASK-DEBUG - create_scenario_tags_string:
unique_scenario_tags_count=31
2026-02-16 06:00:56.033 - FLASK-DEBUG - write_tags_in_json: called
2026-02-16 06:00:56.033 - FLASK-INFO - write_tags_in_json:
json_path=/var/lib/jenkins/ocpcf_tests/jenkinsData/Cust-Policy-Regression-
PCF.json
2026-02-16 06:00:56.034 - FLASK-DEBUG - process_tags: called
2026-02-16 06:00:56.034 - FLASK-DEBUG - process_tags: input_count=2
output=cne-common,cm-service
2026-02-16 06:00:56.034 - FLASK-DEBUG - process_tags: called
2026-02-16 06:00:56.034 - FLASK-DEBUG - process_tags: input_count=1 output=
2026-02-16 06:00:56.034 - FLASK-DEBUG - process_tags: called
2026-02-16 06:00:56.034 - FLASK-DEBUG - process_tags: input_count=2
output=sanity
2026-02-16 06:00:56.034 - FLASK-DEBUG - process_tags: called
2026-02-16 06:00:56.035 - FLASK-DEBUG - process_tags: input_count=1 output=
2026-02-16 06:00:56.041 - FLASK-INFO - write_tags_in_json:
Execute_Tagging=True
2026-02-16 06:00:56.042 - FLASK-DEBUG - create_features_string: called
2026-02-16 06:00:56.042 - FLASK-DEBUG - create_features_string:
json_path=/var/lib/jenkins/ocpcf_tests/jenkinsData/Cust-Policy-Regression-
PCF.json
2026-02-16 06:00:56.044 - FLASK-DEBUG - create_sgfs_map_with_tags: called
2026-02-16 06:00:56.044 - FLASK-DEBUG - create_sgfs_map_with_tags:
building sgfs map
2026-02-16 06:00:56.045 - FLASK-DEBUG - create_sgfs_map_with_tags:
stages_count=7
2026-02-16 06:00:56.045 - FLASK-INFO - create_features_string:
filtered_features_count=10
2026-02-16 06:00:56.046 - FLASK-DEBUG - write_features_in_json (Filtered
features): called
2026-02-16 06:00:56.046 - FLASK-INFO - write_features_in_json:
json_file_to_read=/var/lib/jenkins/ocpcf_tests/jenkinsData/Cust-Policy-
Regression-PCF.json
2026-02-16 06:00:56.051 - FLASK-INFO - FormulatepageAndQuery: end
pageAndQuery=buildWithParameters?
Configuration_Type=Custom_Config&FilterWithTags=false,true,&Feature_Include
_Tags=false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,&Feature_Exclude_Tags=false,false,false,false,false,false,false,fa
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
lse,false,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,&Scenario_Include_Tags=false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,&Scenario_Exclude_Tags=false,false,false,false,false,fals
e,false,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fals
e, return_message_len=0
2026-02-16 06:00:56.051 - FLASK-DEBUG - create_session: called
user=guiadmin tls=True
2026-02-16 06:00:56.051 - FLASK-DEBUG - create_session:
session.verify=/tmp/caroot.cer
2026-02-16 06:00:56.051 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name]
has_data=False
2026-02-16 06:00:56.051 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name] timeout=(10, 27)
delete_session=False
2026-02-16 06:00:56.053 - FLASK-DEBUG - Starting new HTTPS connection (1):
127.0.0.1:8443
2026-02-16 06:00:56.105 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /api/
json?tree=jobs%5Bname%5D HTTP/1.1" 200 368
2026-02-16 06:00:56.106 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:00:56.106 - FLASK-DEBUG - valid_pipeline: called
pipeline_name=Policy-Regression
2026-02-16 06:00:56.106 - FLASK-DEBUG - valid_pipeline: checking existence
pipeline_name=Policy-Regression endpoint=https://127.0.0.1:8443/api/json?
tree=jobs[name]
2026-02-16 06:00:56.106 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name]
has_data=False
2026-02-16 06:00:56.107 - FLASK-DEBUG - jenkins_request: creating session
from request auth for endpoint=https://127.0.0.1:8443/api/json?
tree=jobs[name]
2026-02-16 06:00:56.107 - FLASK-DEBUG - create_session: called
user=guiadmin tls=True
2026-02-16 06:00:56.107 - FLASK-DEBUG - create_session:
session.verify=/tmp/caroot.cer
2026-02-16 06:00:56.107 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name] timeout=(10, 27)
delete_session=True
2026-02-16 06:00:56.108 - FLASK-DEBUG - Starting new HTTPS connection (1):
127.0.0.1:8443
2026-02-16 06:00:56.131 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /api/
json?tree=jobs%5Bname%5D HTTP/1.1" 200 368
2026-02-16 06:00:56.131 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:00:56.131 - FLASK-DEBUG - close_session: closing session
2026-02-16 06:00:56.131 - FLASK-INFO - valid_pipeline: SUCCESS
pipeline_name=Policy-Regression
2026-02-16 06:00:56.132 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/queue/api/json?
tree=items[id,task[name],actions[causes[userName]]] has_data=False
2026-02-16 06:00:56.132 - FLASK-DEBUG - jenkins_request: method=POST
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
endpoint=https://127.0.0.1:8443/queue/api/json?
tree=items[id,task[name],actions[causes[userName]]] timeout=(10, 27)
delete_session=False
2026-02-16 06:00:56.150 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /
queue/api/json?
tree=items%5Bid,task%5Bname%5D,actions%5Bcauses%5BuserName%5D%5D%5D HTTP/
1.1" 200 61
2026-02-16 06:00:56.152 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:00:56.152 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Abort-helper/lastBuild/api/
json?tree=building has_data=False
2026-02-16 06:00:56.152 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Abort-helper/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:163 - JENKINS-DEBUG - /job/Abort-helper/lastBuild/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.182 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Abort-helper/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.186 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.186 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/BSF_HealthCheck/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.186 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/BSF_HealthCheck/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:198 - JENKINS-DEBUG - /job/BSF_HealthCheck/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.214 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
BSF_HealthCheck/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.217 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.217 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/BSF_NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.217 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/BSF_NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:228 - JENKINS-DEBUG - /job/BSF_NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.241 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
BSF_NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.250 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.250 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/BSF_Performance/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.250 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/BSF_Performance/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:262 - JENKINS-DEBUG - /job/BSF_Performance/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.278 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
BSF_Performance/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.283 - FLASK-DEBUG - jenkins_request: response
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
status=404
2026-02-16 06:00:56.283 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/BSF_Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.283 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/BSF_Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:294 - JENKINS-DEBUG - /job/BSF_Regression/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.309 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
BSF_Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.314 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.314 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/CNCC-NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.314 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/CNCC-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:326 - JENKINS-DEBUG - /job/CNCC-NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.341 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
CNCC-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.346 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.346 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/CNCC-Performance/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.346 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/CNCC-Performance/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:357 - JENKINS-DEBUG - /job/CNCC-Performance/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.371 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
CNCC-Performance/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.375 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.375 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/CNCC-Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.375 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/CNCC-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:386 - JENKINS-DEBUG - /job/CNCC-Regression/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.399 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
CNCC-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.403 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.403 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/CnSIH-NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.403 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/CnSIH-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:414 - JENKINS-DEBUG - /job/CnSIH-NewFeatures/
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.427 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
CnSIH-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.431 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.431 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/CnSIH-Performance/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.432 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/CnSIH-Performance/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:443 - JENKINS-DEBUG - /job/CnSIH-Performance/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.456 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
CnSIH-Performance/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.461 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.461 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/CnSIH-Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.461 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/CnSIH-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:489 - JENKINS-DEBUG - /job/CnSIH-Regression/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.503 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
CnSIH-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.507 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.507 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/EIR-HealthCheck/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.507 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/EIR-HealthCheck/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:521 - JENKINS-DEBUG - /job/EIR-HealthCheck/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.534 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
EIR-HealthCheck/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.538 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.539 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/EIR-NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.539 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/EIR-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:550 - JENKINS-DEBUG - /job/EIR-NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.563 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
EIR-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.567 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.567 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/EIR-Regression/
lastBuild/api/json?tree=building has_data=False
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-16 06:00:56.567 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/EIR-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:578 - JENKINS-DEBUG - /job/EIR-Regression/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.591 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
EIR-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.595 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.595 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/NADD-NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.596 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/NADD-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:607 - JENKINS-DEBUG - /job/NADD-NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.620 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
NADD-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.624 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.624 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/NADD-Performance/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.624 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/NADD-Performance/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:635 - JENKINS-DEBUG - /job/NADD-Performance/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.648 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
NADD-Performance/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.652 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.653 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/NADD-Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.653 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/NADD-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:664 - JENKINS-DEBUG - /job/NADD-Regression/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.677 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
NADD-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.681 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.681 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/NEF-NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.681 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/NEF-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:692 - JENKINS-DEBUG - /job/NEF-NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.705 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
NEF-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.709 - FLASK-DEBUG - jenkins_request: response
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
status=404
2026-02-16 06:00:56.709 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/NEF-Performance/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.709 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/NEF-Performance/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:720 - JENKINS-DEBUG - /job/NEF-Performance/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.735 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
NEF-Performance/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.740 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.740 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/NEF-Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.741 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/NEF-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:752 - JENKINS-DEBUG - /job/NEF-Regression/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.765 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
NEF-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.769 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.769 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/NRF-NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.769 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/NRF-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:781 - JENKINS-DEBUG - /job/NRF-NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.793 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
NRF-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.797 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.797 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/NRF-Performance/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.797 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/NRF-Performance/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:809 - JENKINS-DEBUG - /job/NRF-Performance/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.839 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
NRF-Performance/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.842 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.842 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/NRF-Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.842 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/NRF-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:857 - JENKINS-DEBUG - /job/NRF-Regression/
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.873 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
NRF-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.875 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.875 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/NSSF-HealthCheck/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.876 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/NSSF-HealthCheck/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:888 - JENKINS-DEBUG - /job/NSSF-HealthCheck/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.901 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
NSSF-HealthCheck/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.903 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.903 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/NSSF-NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.903 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/NSSF-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:914 - JENKINS-DEBUG - /job/NSSF-NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.928 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
NSSF-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.931 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.931 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/NSSF-Performance/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.931 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/NSSF-Performance/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:943 - JENKINS-DEBUG - /job/NSSF-Performance/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.956 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
NSSF-Performance/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.958 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.958 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/NSSF-Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:56.958 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/NSSF-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:970 - JENKINS-DEBUG - /job/NSSF-Regression/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:56.984 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
NSSF-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:56.986 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:56.987 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/OCCM-NewFeatures/
lastBuild/api/json?tree=building has_data=False
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-16 06:00:56.987 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/OCCM-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:56:998 - JENKINS-DEBUG - /job/OCCM-NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:57.028 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
OCCM-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:57.031 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:57.031 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/OCCM-Performance/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:57.031 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/OCCM-Performance/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:57:045 - JENKINS-DEBUG - /job/OCCM-Performance/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:57.057 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
OCCM-Performance/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:57.059 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:57.060 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/OCCM-Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:57.060 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/OCCM-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:57:071 - JENKINS-DEBUG - /job/OCCM-Regression/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:57.096 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
OCCM-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:57.098 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:57.098 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/OCCNDRA-NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:57.098 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/OCCNDRA-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:57:113 - JENKINS-DEBUG - /job/OCCNDRA-NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:57.129 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
OCCNDRA-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:57.132 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:57.132 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/OCCNDRA-Performance/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:57.132 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/OCCNDRA-Performance/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:57:148 - JENKINS-DEBUG - /job/OCCNDRA-Performance/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:57.159 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
OCCNDRA-Performance/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:57.162 - FLASK-DEBUG - jenkins_request: response
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
status=404
2026-02-16 06:00:57.162 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/OCCNDRA-Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:57.162 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/OCCNDRA-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:57:177 - JENKINS-DEBUG - /job/OCCNDRA-Regression/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:57.206 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
OCCNDRA-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:57.219 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:57.219 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-HealthCheck/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:57.219 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-HealthCheck/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:57:232 - JENKINS-DEBUG - /job/Policy-HealthCheck/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:57.245 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-HealthCheck/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:57.248 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:57.248 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:57.248 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:57:259 - JENKINS-DEBUG - /job/Policy-NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:57.271 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:57.274 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:57.274 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Performance/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:57.274 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Performance/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:57:295 - JENKINS-DEBUG - /job/Policy-Performance/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:57.308 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Performance/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:57.310 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:57.310 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:00:57.310 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:00:57:322 - JENKINS-DEBUG - /job/Policy-Regression/
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:00:57.335 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:00:57.343 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:00:57.343 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/
buildWithParameters?
Configuration_Type=Custom_Config&FilterWithTags=false,true,&Feature_Include
_Tags=false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,&Feature_Exclude_Tags=false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,&Scenario_Include_Tags=false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,&Scenario_Exclude_Tags=false,false,false,false,false,fals
e,false,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fals
e, has_data=False
2026-02-16 06:00:57.343 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/buildWithParameters?
Configuration_Type=Custom_Config&FilterWithTags=false,true,&Feature_Include
_Tags=false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,&Feature_Exclude_Tags=false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,&Scenario_Include_Tags=false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,&Scenario_Exclude_Tags=false,false,false,false,false,fals
e,false,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fals
e, timeout=(10, 27) delete_session=False
2026-02-16 06:00:57:361 - JENKINS-DEBUG - /job/Policy-Regression/
buildWithParameters by guiadmin from 127.0.0.1
2026-02-16 06:00:58.019 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/buildWithParameters?
Configuration_Type=Custom_Config&FilterWithTags=false,true,&Feature_Include
_Tags=false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,&Feature_Exclude_Tags=false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,&Scenario_Include_Tags=false,false,false,
false,false,false,false,false,false,false,false,false,false,false,false,fal
se,false,false,false,false,false,false,false,false,false,false,false,false,
false,false,false,&Scenario_Exclude_Tags=false,false,false,false,false,fals
e,false,false,false,false,false,false,false,false,false,false,false,false,f
alse,false,false,false,false,false,false,false,false,false,false,false,fals
e, HTTP/1.1" 201 0
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-16 06:00:58.020 - FLASK-DEBUG - jenkins_request: response
status=201
2026-02-16 06:00:58.020 - FLASK-DEBUG - _get_build_number_from_queue:
called queue_item_id=1 timeout_s=60
2026-02-16 06:00:58.020 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] has_data=False
2026-02-16 06:00:58.020 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] timeout=(10, 10) delete_session=False
2026-02-16 06:00:58.040 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /queue/
item/1/api/json?tree=executable%5Bnumber%5D HTTP/1.1" 200 63
2026-02-16 06:00:58.040 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:00:58:611 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:00:59.041 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] has_data=False
2026-02-16 06:00:59.041 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] timeout=(10, 10) delete_session=False
2026-02-16 06:00:59.056 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /queue/
item/1/api/json?tree=executable%5Bnumber%5D HTTP/1.1" 200 63
2026-02-16 06:00:59.056 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:00.057 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] has_data=False
2026-02-16 06:01:00.057 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] timeout=(10, 10) delete_session=False
2026-02-16 06:01:00.071 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /queue/
item/1/api/json?tree=executable%5Bnumber%5D HTTP/1.1" 200 63
2026-02-16 06:01:00.071 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:01.072 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] has_data=False
2026-02-16 06:01:01.072 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] timeout=(10, 10) delete_session=False
2026-02-16 06:01:01.086 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /queue/
item/1/api/json?tree=executable%5Bnumber%5D HTTP/1.1" 200 63
2026-02-16 06:01:01.086 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:02.086 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] has_data=False
2026-02-16 06:01:02.087 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] timeout=(10, 10) delete_session=False
2026-02-16 06:01:02.101 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /queue/
item/1/api/json?tree=executable%5Bnumber%5D HTTP/1.1" 200 63
2026-02-16 06:01:02.102 - FLASK-DEBUG - jenkins_request: response
status=200
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-16 06:01:03.102 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] has_data=False
2026-02-16 06:01:03.102 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] timeout=(10, 10) delete_session=False
2026-02-16 06:01:03.116 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /queue/
item/1/api/json?tree=executable%5Bnumber%5D HTTP/1.1" 200 63
2026-02-16 06:01:03.116 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:03:572 - JENKINS-DEBUG - job/Policy-Regression/ #1
Started by user guiadmin, Parameters:[: {}, : {}, TestSuite: {}, SUT:
{PCF}, Configuration_Type: {Custom_Config}, Include_NewFeatures: {NO},
FilterWithTags: {false,true,}, shell_executor: {}, Feature_Include_Tags:
{false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,}, Feature_Exclude_Tags:
{false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,}, Scenario_Include_Tags:
{false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,},  Scenario_Exclude_Tags:
{false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,}, json_writer: {}, Submit_Tags: {},
Fetch_Log_Upon_Failure: {YES}, Log_Validation: {NO}, : {},
Select_Features_Option: {All}, Features: {}, : {},
Select_Testcases_Option: {}, TestCases: {}]
2026-02-16 06:01:04.117 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] has_data=False
2026-02-16 06:01:04.117 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/queue/item/1/api/json?
tree=executable[number] timeout=(10, 10) delete_session=False
2026-02-16 06:01:04.139 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /queue/
item/1/api/json?tree=executable%5Bnumber%5D HTTP/1.1" 200 126
2026-02-16 06:01:04.140 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:04.140 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/SCP-NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:01:04.140 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/SCP-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:152 - JENKINS-DEBUG - /job/SCP-NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.171 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
SCP-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.175 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:01:04.176 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/SCP-Performance/
lastBuild/api/json?tree=building has_data=False
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-16 06:01:04.176 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/SCP-Performance/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:189 - JENKINS-DEBUG - /job/SCP-Performance/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.201 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
SCP-Performance/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.236 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:01:04.236 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/SCP-Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:01:04.237 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/SCP-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:248 - JENKINS-DEBUG - /job/SCP-Regression/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.261 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
SCP-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.265 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:01:04.265 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/SEPP-HealthCheck/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:01:04.265 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/SEPP-HealthCheck/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:276 - JENKINS-DEBUG - /job/SEPP-HealthCheck/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.287 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
SEPP-HealthCheck/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.291 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:01:04.291 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/SEPP-NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:01:04.291 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/SEPP-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:303 - JENKINS-DEBUG - /job/SEPP-NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.314 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
SEPP-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.317 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:01:04.318 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/SEPP-Performance/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:01:04.318 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/SEPP-Performance/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:329 - JENKINS-DEBUG - /job/SEPP-Performance/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.339 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
SEPP-Performance/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.343 - FLASK-DEBUG - jenkins_request: response
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
status=404
2026-02-16 06:01:04.343 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/SEPP-Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:01:04.343 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/SEPP-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:354 - JENKINS-DEBUG - /job/SEPP-Regression/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.374 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
SEPP-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.379 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:01:04.379 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/SLF-HealthCheck/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:01:04.379 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/SLF-HealthCheck/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:391 - JENKINS-DEBUG - /job/SLF-HealthCheck/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.404 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
SLF-HealthCheck/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.410 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:01:04.410 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/SLF-NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:01:04.410 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/SLF-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:424 - JENKINS-DEBUG - /job/SLF-NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.438 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
SLF-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.442 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:01:04.442 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/SLF-Performance/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:01:04.442 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/SLF-Performance/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:458 - JENKINS-DEBUG - /job/SLF-Performance/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.471 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
SLF-Performance/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.475 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:01:04.475 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/SLF-Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:01:04.476 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/SLF-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:492 - JENKINS-DEBUG - /job/SLF-Regression/
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.504 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
SLF-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.516 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:01:04.516 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/UDR-HealthCheck/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:01:04.516 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/UDR-HealthCheck/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:538 - JENKINS-DEBUG - /job/UDR-HealthCheck/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.549 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
UDR-HealthCheck/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.554 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:01:04.554 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/UDR-NewFeatures/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:01:04.554 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/UDR-NewFeatures/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:567 - JENKINS-DEBUG - /job/UDR-NewFeatures/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.579 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
UDR-NewFeatures/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.583 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:01:04.583 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/UDR-Regression/
lastBuild/api/json?tree=building has_data=False
2026-02-16 06:01:04.583 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/UDR-Regression/lastBuild/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:04:589 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:01:04:594 - JENKINS-DEBUG - /job/UDR-Regression/
lastBuild/api/json by guiadmin from 127.0.0.1
2026-02-16 06:01:04.605 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
UDR-Regression/lastBuild/api/json?tree=building HTTP/1.1" 404 None
2026-02-16 06:01:04.609 - FLASK-DEBUG - jenkins_request: response
status=404
2026-02-16 06:01:04.610 - FLASK-INFO - triggerBuild: completed status=201
response_len=19
2026-02-16 06:01:04.610 - FLASK-DEBUG - response_generator: called
status=201 message_type=dict
2026-02-16 06:01:04.611 - FLASK-INFO - 127.0.0.1 - - [16/Feb/2026
06:01:04] "POST /build HTTP/1.1" 201 -
2026-02-16 06:01:10:589 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:01:16:588 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:01:16:872 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/19/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:01:22:597 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
runs by guiadmin from 10.233.101.0
2026-02-16 06:01:22:844 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/19/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:01:25.842 - FLASK-DEBUG - stop: called pipeline=Policy-
Regression job_id=1 method=POST
2026-02-16 06:01:25.842 - FLASK-INFO - route /job/Policy-Regression/1/
stop: called
2026-02-16 06:01:25.842 - FLASK-DEBUG - stop: immediate=True
2026-02-16 06:01:25.842 - FLASK-DEBUG - authenticate_request: called
2026-02-16 06:01:25.842 - FLASK-DEBUG - authenticate_request: received
authorization username=guiadmin
2026-02-16 06:01:25.842 - FLASK-DEBUG - create_session: called
user=guiadmin tls=True
2026-02-16 06:01:25.842 - FLASK-DEBUG - create_session:
session.verify=/tmp/caroot.cer
2026-02-16 06:01:25.842 - FLASK-DEBUG - authenticate_request: validating
creds via https to 127.0.0.1:8443
2026-02-16 06:01:25.842 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name] has_data=False
2026-02-16 06:01:25.842 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name] timeout=(10, 27)
delete_session=False
2026-02-16 06:01:25.844 - FLASK-DEBUG - Starting new HTTPS connection (1):
127.0.0.1:8443
2026-02-16 06:01:26.180 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /api/
json?tree=jobs%5Bname%5D HTTP/1.1" 200 368
2026-02-16 06:01:26.180 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:26.180 - FLASK-DEBUG - authenticate_request: auth
validation status_code=200
2026-02-16 06:01:26.181 - FLASK-DEBUG - valid_job_id: called job_id=1
2026-02-16 06:01:26.181 - FLASK-DEBUG - valid_pipeline: called
pipeline_name=Policy-Regression
2026-02-16 06:01:26.181 - FLASK-DEBUG - valid_pipeline: checking existence
pipeline_name=Policy-Regression endpoint=https://127.0.0.1:8443/api/json?
tree=jobs[name]
2026-02-16 06:01:26.181 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name]
has_data=False
2026-02-16 06:01:26.181 - FLASK-DEBUG - jenkins_request: creating session
from request auth for endpoint=https://127.0.0.1:8443/api/json?
tree=jobs[name]
2026-02-16 06:01:26.181 - FLASK-DEBUG - create_session: called
user=guiadmin tls=True
2026-02-16 06:01:26.181 - FLASK-DEBUG - create_session:
session.verify=/tmp/caroot.cer
2026-02-16 06:01:26.181 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/api/json?tree=jobs[name] timeout=(10, 27)
delete_session=True
2026-02-16 06:01:26.182 - FLASK-DEBUG - Starting new HTTPS connection (1):
127.0.0.1:8443
2026-02-16 06:01:26.473 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /api/
json?tree=jobs%5Bname%5D HTTP/1.1" 200 368
2026-02-16 06:01:26.473 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:26.473 - FLASK-DEBUG - close_session: closing session
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-16 06:01:26.474 - FLASK-INFO - valid_pipeline: SUCCESS
pipeline_name=Policy-Regression
2026-02-16 06:01:26.474 - FLASK-DEBUG - valid_job_id: called job_id=1
2026-02-16 06:01:26.474 - FLASK-DEBUG - create_session: called
user=guiadmin tls=True
2026-02-16 06:01:26.474 - FLASK-DEBUG - create_session:
session.verify=/tmp/caroot.cer
2026-02-16 06:01:26.474 - FLASK-DEBUG - is_build_running: called
pipeline=Policy-Regression job_id=1
2026-02-16 06:01:26.474 - FLASK-DEBUG - is_build_running: endpoint=https://
127.0.0.1:8443/job/Policy-Regression/1/api/json?tree=building
2026-02-16 06:01:26.474 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/
json?tree=building has_data=False
2026-02-16 06:01:26.474 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:26.476 - FLASK-DEBUG - Starting new HTTPS connection (1):
127.0.0.1:8443
2026-02-16 06:01:27.019 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/api/json?tree=building HTTP/1.1" 200 88
2026-02-16 06:01:27.019 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:27.020 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-16 06:01:27.020 - FLASK-DEBUG - is_build_running: called
pipeline=Policy-Regression job_id=1
2026-02-16 06:01:27.020 - FLASK-DEBUG - is_build_running: endpoint=https://
127.0.0.1:8443/job/Policy-Regression/1/api/json?tree=building
2026-02-16 06:01:27.020 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/
json?tree=building has_data=False
2026-02-16 06:01:27.020 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:27:469 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:01:27.471 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/api/json?tree=building HTTP/1.1" 200 88
2026-02-16 06:01:27.471 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:27.471 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-16 06:01:27.472 - FLASK-DEBUG - is_stage_execute_tests_running:
called pipeline=Policy-Regression job_id=1
2026-02-16 06:01:27.472 - FLASK-DEBUG - is_stage_execute_tests_running:
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]]
2026-02-16 06:01:27.472 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] has_data=False
2026-02-16 06:01:27.472 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] timeout=(10, 27)
delete_session=False
2026-02-16 06:01:28:149 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
json by guiadmin from 127.0.0.1
2026-02-16 06:01:28.161 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /job/
Policy-Regression/1/api/json?
depth=2&tree=actions%5Bnodes%5BdisplayName,running%5D%5D HTTP/1.1" 200 765
2026-02-16 06:01:28.161 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:28.161 - FLASK-DEBUG - is_stage_execute_tests_running:
execute-tests NOT running pipeline=Policy-Regression job_id=1
2026-02-16 06:01:28.161 - FLASK-DEBUG - send_abort_request: called
pipeline=Policy-Regression job_id=1 immediate=True
2026-02-16 06:01:28.161 - FLASK-INFO - send_abort_request: pipeline=Policy-
Regression job_id=1 immediate=True
2026-02-16 06:01:28.161 - FLASK-DEBUG - send_abort_request: stop
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop
2026-02-16 06:01:28.162 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop
has_data=False
2026-02-16 06:01:28.162 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop timeout=(10,
27) delete_session=False
2026-02-16 06:01:28:434 - JENKINS-DEBUG - /job/Policy-Regression/1/stop by
guiadmin from 127.0.0.1
2026-02-16 06:01:28.438 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/stop HTTP/1.1" 302 0
2026-02-16 06:01:28:596 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:01:28:752 - JENKINS-DEBUG - /job/Policy-Regression/1/ by
guiadmin from 127.0.0.1
2026-02-16 06:01:28:825 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/19/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:01:29.188 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /job/
Policy-Regression/1/ HTTP/1.1" 200 None
2026-02-16 06:01:29.340 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:29.340 - FLASK-INFO - send_abort_request: stop request
status=200
2026-02-16 06:01:34:614 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:01:34:869 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/95/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:01:39.341 - FLASK-DEBUG - is_build_running: called
pipeline=Policy-Regression job_id=1
2026-02-16 06:01:39.341 - FLASK-DEBUG - is_build_running: endpoint=https://
127.0.0.1:8443/job/Policy-Regression/1/api/json?tree=building
2026-02-16 06:01:39.341 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/
json?tree=building has_data=False
2026-02-16 06:01:39.341 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:39:387 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:01:39.390 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/api/json?tree=building HTTP/1.1" 200 88
2026-02-16 06:01:39.390 - FLASK-DEBUG - jenkins_request: response
status=200
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-16 06:01:39.391 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-16 06:01:39.391 - FLASK-DEBUG - is_stage_execute_tests_running:
called pipeline=Policy-Regression job_id=1
2026-02-16 06:01:39.391 - FLASK-DEBUG - is_stage_execute_tests_running:
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]]
2026-02-16 06:01:39.391 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] has_data=False
2026-02-16 06:01:39.391 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] timeout=(10, 27)
delete_session=False
2026-02-16 06:01:39:414 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:01:39.422 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /job/
Policy-Regression/1/api/json?
depth=2&tree=actions%5Bnodes%5BdisplayName,running%5D%5D HTTP/1.1" 200 1136
2026-02-16 06:01:39.422 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:39.423 - FLASK-DEBUG - is_stage_execute_tests_running:
execute-tests running pipeline=Policy-Regression job_id=1
2026-02-16 06:01:39.423 - FLASK-DEBUG - send_abort_request: called
pipeline=Policy-Regression job_id=1 immediate=True
2026-02-16 06:01:39.423 - FLASK-INFO - send_abort_request: pipeline=Policy-
Regression job_id=1 immediate=True
2026-02-16 06:01:39.423 - FLASK-DEBUG - send_abort_request: stop
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop
2026-02-16 06:01:39.423 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop
has_data=False
2026-02-16 06:01:39.423 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop timeout=(10,
27) delete_session=False
2026-02-16 06:01:39.549 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/stop HTTP/1.1" 302 0
2026-02-16 06:01:39:569 - JENKINS-DEBUG - /job/Policy-Regression/1/ by
guiadmin from 127.0.0.1
2026-02-16 06:01:39.598 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /job/
Policy-Regression/1/ HTTP/1.1" 200 None
2026-02-16 06:01:39.641 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:39.641 - FLASK-INFO - send_abort_request: stop request
status=200
2026-02-16 06:01:40:605 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:01:40:871 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/171/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:01:43.444+0000 [id=54]    WARNING
o.j.p.w.s.s.i.POSTHyperlinkNote#<init>: You need to define the root URL of
Jenkins
2026-02-16 06:01:46:595 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:01:46:965 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/247/wfapi/describe by guiadmin from 10.233.101.0
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-16 06:01:47:217 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/249/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:01:47:598 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/251/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:01:49.641 - FLASK-DEBUG - is_build_running: called
pipeline=Policy-Regression job_id=1
2026-02-16 06:01:49.642 - FLASK-DEBUG - is_build_running: endpoint=https://
127.0.0.1:8443/job/Policy-Regression/1/api/json?tree=building
2026-02-16 06:01:49.642 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/
json?tree=building has_data=False
2026-02-16 06:01:49.642 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:49:656 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:01:49.660 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/api/json?tree=building HTTP/1.1" 200 88
2026-02-16 06:01:49.660 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:49.660 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-16 06:01:49.660 - FLASK-DEBUG - is_stage_execute_tests_running:
called pipeline=Policy-Regression job_id=1
2026-02-16 06:01:49.660 - FLASK-DEBUG - is_stage_execute_tests_running:
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]]
2026-02-16 06:01:49.660 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] has_data=False
2026-02-16 06:01:49.660 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] timeout=(10, 27)
delete_session=False
2026-02-16 06:01:49:713 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:01:49.731 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /job/
Policy-Regression/1/api/json?
depth=2&tree=actions%5Bnodes%5BdisplayName,running%5D%5D HTTP/1.1" 200 None
2026-02-16 06:01:49.731 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:49.732 - FLASK-DEBUG - is_stage_execute_tests_running:
execute-tests running pipeline=Policy-Regression job_id=1
2026-02-16 06:01:49.732 - FLASK-DEBUG - send_abort_request: called
pipeline=Policy-Regression job_id=1 immediate=True
2026-02-16 06:01:49.732 - FLASK-INFO - send_abort_request: pipeline=Policy-
Regression job_id=1 immediate=True
2026-02-16 06:01:49.732 - FLASK-DEBUG - send_abort_request: stop
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop
2026-02-16 06:01:49.732 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop
has_data=False
2026-02-16 06:01:49.732 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop timeout=(10,
27) delete_session=False
2026-02-16 06:01:49.746 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
Policy-Regression/1/stop HTTP/1.1" 302 0
2026-02-16 06:01:49:762 - JENKINS-DEBUG - /job/Policy-Regression/1/ by
guiadmin from 127.0.0.1
2026-02-16 06:01:49.813 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /job/
Policy-Regression/1/ HTTP/1.1" 200 None
2026-02-16 06:01:49.841 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:49.841 - FLASK-INFO - send_abort_request: stop request
status=200
2026-02-16 06:01:52:591 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:01:52:852 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/329/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:01:54.549+0000 [id=53]    WARNING
o.j.p.w.s.s.i.POSTHyperlinkNote#<init>: You need to define the root URL of
Jenkins
2026-02-16 06:01:58:594 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:01:58:848 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/399/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:01:58:852 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/403/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:01:58:852 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/401/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:01:59.842 - FLASK-DEBUG - is_build_running: called
pipeline=Policy-Regression job_id=1
2026-02-16 06:01:59.842 - FLASK-DEBUG - is_build_running: endpoint=https://
127.0.0.1:8443/job/Policy-Regression/1/api/json?tree=building
2026-02-16 06:01:59.842 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/
json?tree=building has_data=False
2026-02-16 06:01:59.842 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:01:59:855 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:01:59.856 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/api/json?tree=building HTTP/1.1" 200 88
2026-02-16 06:01:59.857 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:59.857 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-16 06:01:59.857 - FLASK-DEBUG - is_stage_execute_tests_running:
called pipeline=Policy-Regression job_id=1
2026-02-16 06:01:59.857 - FLASK-DEBUG - is_stage_execute_tests_running:
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]]
2026-02-16 06:01:59.857 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] has_data=False
2026-02-16 06:01:59.857 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] timeout=(10, 27)
delete_session=False
2026-02-16 06:01:59:868 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
2026-02-16 06:01:59.875 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /job/
Policy-Regression/1/api/json?
depth=2&tree=actions%5Bnodes%5BdisplayName,running%5D%5D HTTP/1.1" 200 None
2026-02-16 06:01:59.876 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:59.877 - FLASK-DEBUG - is_stage_execute_tests_running:
execute-tests running pipeline=Policy-Regression job_id=1
2026-02-16 06:01:59.877 - FLASK-DEBUG - send_abort_request: called
pipeline=Policy-Regression job_id=1 immediate=True
2026-02-16 06:01:59.877 - FLASK-INFO - send_abort_request: pipeline=Policy-
Regression job_id=1 immediate=True
2026-02-16 06:01:59.877 - FLASK-DEBUG - send_abort_request: stop
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop
2026-02-16 06:01:59.877 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop
has_data=False
2026-02-16 06:01:59.877 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop timeout=(10,
27) delete_session=False
2026-02-16 06:01:59:887 - JENKINS-DEBUG - /job/Policy-Regression/1/stop by
guiadmin from 127.0.0.1
2026-02-16 06:01:59.889 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/stop HTTP/1.1" 302 0
2026-02-16 06:01:59:900 - JENKINS-DEBUG - /job/Policy-Regression/1/ by
guiadmin from 127.0.0.1
2026-02-16 06:01:59.923 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /job/
Policy-Regression/1/ HTTP/1.1" 200 None
2026-02-16 06:01:59.963 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:01:59.963 - FLASK-INFO - send_abort_request: stop request
status=200
2026-02-16 06:02:00:807 - JENKINS-DEBUG - /job/Policy-Regression/
buildHistory/ajax by guiadmin from 10.233.101.0
2026-02-16 06:02:02:152 - JENKINS-DEBUG - /job/Policy-Regression/1/
contextMenu by guiadmin from 10.233.101.0
2026-02-16 06:02:04:252 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:02:04:507 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/475/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:02:04.745+0000 [id=50]    WARNING
o.j.p.w.s.s.i.POSTHyperlinkNote#<init>: You need to define the root URL of
Jenkins
2026-02-16 06:02:05:798 - JENKINS-DEBUG - /job/Policy-Regression/
buildHistory/ajax by guiadmin from 10.233.101.0
2026-02-16 06:02:09:632 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:02:09:876 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/540/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:02:09.963 - FLASK-DEBUG - is_build_running: called
pipeline=Policy-Regression job_id=1
2026-02-16 06:02:09.963 - FLASK-DEBUG - is_build_running: endpoint=https://
127.0.0.1:8443/job/Policy-Regression/1/api/json?tree=building
2026-02-16 06:02:09.963 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/
json?tree=building has_data=False
2026-02-16 06:02:09.964 - FLASK-DEBUG - jenkins_request: method=POST
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:02:09:978 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:02:09.980 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/api/json?tree=building HTTP/1.1" 200 88
2026-02-16 06:02:09.980 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:02:09.980 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-16 06:02:09.981 - FLASK-DEBUG - is_stage_execute_tests_running:
called pipeline=Policy-Regression job_id=1
2026-02-16 06:02:09.981 - FLASK-DEBUG - is_stage_execute_tests_running:
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]]
2026-02-16 06:02:09.981 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] has_data=False
2026-02-16 06:02:09.981 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] timeout=(10, 27)
delete_session=False
2026-02-16 06:02:09:998 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:02:10.010 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /job/
Policy-Regression/1/api/json?
depth=2&tree=actions%5Bnodes%5BdisplayName,running%5D%5D HTTP/1.1" 200 None
2026-02-16 06:02:10.013 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:02:10.014 - FLASK-DEBUG - is_stage_execute_tests_running:
execute-tests running pipeline=Policy-Regression job_id=1
2026-02-16 06:02:10.014 - FLASK-DEBUG - send_abort_request: called
pipeline=Policy-Regression job_id=1 immediate=True
2026-02-16 06:02:10.014 - FLASK-INFO - send_abort_request: pipeline=Policy-
Regression job_id=1 immediate=True
2026-02-16 06:02:10.014 - FLASK-DEBUG - send_abort_request: stop
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop
2026-02-16 06:02:10.014 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop
has_data=False
2026-02-16 06:02:10.014 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/stop timeout=(10,
27) delete_session=False
2026-02-16 06:02:10:127 - JENKINS-DEBUG - /job/Policy-Regression/1/stop by
guiadmin from 127.0.0.1
2026-02-16 06:02:10.129 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/stop HTTP/1.1" 302 0
2026-02-16 06:02:10:142 - JENKINS-DEBUG - /job/Policy-Regression/1/ by
guiadmin from 127.0.0.1
2026-02-16 06:02:10.163 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /job/
Policy-Regression/1/ HTTP/1.1" 200 None
2026-02-16 06:02:10.187 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:02:10.187 - FLASK-INFO - send_abort_request: stop request
status=200
2026-02-16 06:02:10:800 - JENKINS-DEBUG - /job/Policy-Regression/
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
buildHistory/ajax by guiadmin from 10.233.101.0
2026-02-16 06:02:14.889+0000 [id=52]    WARNING
o.j.p.w.s.s.i.POSTHyperlinkNote#<init>: You need to define the root URL of
Jenkins
2026-02-16 06:02:15:000 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:02:15:250 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/595/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:02:15:803 - JENKINS-DEBUG - /job/Policy-Regression/
buildHistory/ajax by guiadmin from 10.233.101.0
2026-02-16 06:02:20.188 - FLASK-DEBUG - is_build_running: called
pipeline=Policy-Regression job_id=1
2026-02-16 06:02:20.188 - FLASK-DEBUG - is_build_running: endpoint=https://
127.0.0.1:8443/job/Policy-Regression/1/api/json?tree=building
2026-02-16 06:02:20.188 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/
json?tree=building has_data=False
2026-02-16 06:02:20.188 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:02:20:224 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:02:20.226 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/api/json?tree=building HTTP/1.1" 200 88
2026-02-16 06:02:20.227 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:02:20.227 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-16 06:02:20.227 - FLASK-DEBUG - is_stage_execute_tests_running:
called pipeline=Policy-Regression job_id=1
2026-02-16 06:02:20.227 - FLASK-DEBUG - is_stage_execute_tests_running:
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]]
2026-02-16 06:02:20.227 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] has_data=False
2026-02-16 06:02:20.227 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] timeout=(10, 27)
delete_session=False
2026-02-16 06:02:20:238 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:02:20.247 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /job/
Policy-Regression/1/api/json?
depth=2&tree=actions%5Bnodes%5BdisplayName,running%5D%5D HTTP/1.1" 200 None
2026-02-16 06:02:20.250 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:02:20.251 - FLASK-DEBUG - is_stage_execute_tests_running:
execute-tests NOT running pipeline=Policy-Regression job_id=1
2026-02-16 06:02:20:419 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:02:20:684 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/595/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:02:20:980 - JENKINS-DEBUG - /job/Policy-Regression/
buildHistory/ajax by guiadmin from 10.233.101.0
2026-02-16 06:02:25.130+0000 [id=52]    WARNING
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
o.j.p.w.s.s.i.POSTHyperlinkNote#<init>: You need to define the root URL of
Jenkins
2026-02-16 06:02:25.251 - FLASK-DEBUG - is_build_running: called
pipeline=Policy-Regression job_id=1
2026-02-16 06:02:25.251 - FLASK-DEBUG - is_build_running: endpoint=https://
127.0.0.1:8443/job/Policy-Regression/1/api/json?tree=building
2026-02-16 06:02:25.251 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/
json?tree=building has_data=False
2026-02-16 06:02:25.252 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:02:25:266 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:02:25.268 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/api/json?tree=building HTTP/1.1" 200 88
2026-02-16 06:02:25.268 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:02:25.269 - FLASK-INFO - is_build_running method: build is
running with values, pipeline=Policy-Regression job_id=1
2026-02-16 06:02:25.269 - FLASK-DEBUG - is_stage_execute_tests_running:
called pipeline=Policy-Regression job_id=1
2026-02-16 06:02:25.269 - FLASK-DEBUG - is_stage_execute_tests_running:
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]]
2026-02-16 06:02:25.269 - FLASK-DEBUG - jenkins_request: called method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] has_data=False
2026-02-16 06:02:25.269 - FLASK-DEBUG - jenkins_request: method=GET
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
depth=2&tree=actions[nodes[displayName,running]] timeout=(10, 27)
delete_session=False
2026-02-16 06:02:25:280 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:02:25.286 - FLASK-DEBUG - https://127.0.0.1:8443 "GET /job/
Policy-Regression/1/api/json?
depth=2&tree=actions%5Bnodes%5BdisplayName,running%5D%5D HTTP/1.1" 200 None
2026-02-16 06:02:25.288 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:02:25.289 - FLASK-DEBUG - is_stage_execute_tests_running:
execute-tests NOT running pipeline=Policy-Regression job_id=1
2026-02-16 06:02:25:851 - JENKINS-DEBUG - /job/Policy-Regression/wfapi/
runs by guiadmin from 10.233.101.0
2026-02-16 06:02:25:874 - JENKINS-DEBUG - /job/Policy-Regression/
buildHistory/ajax by guiadmin from 10.233.101.0
2026-02-16 06:02:26:095 - JENKINS-DEBUG - /job/Policy-Regression/1/
execution/node/595/wfapi/describe by guiadmin from 10.233.101.0
2026-02-16 06:02:28:622 - JENKINS-DEBUG - Policy-Regression #1 Started by
user guiadmin, Parameters:[: {}, : {}, TestSuite: {}, SUT: {PCF},
Configuration_Type: {Custom_Config}, Include_NewFeatures: {NO},
FilterWithTags: {false,true,}, shell_executor: {}, Feature_Include_Tags:
{false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,}, Feature_Exclude_Tags:
{false,false,false,false,false,false,false,false,false,false,false,false,fa
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
lse,false,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,}, Scenario_Include_Tags:
{false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,}, Scenario_Exclude_Tags:
{false,false,false,false,false,false,false,false,false,false,false,false,fa
lse,false,false,false,false,false,false,false,false,false,false,false,false
,false,false,false,false,false,false,}, json_writer: {}, Submit_Tags: {},
Fetch_Log_Upon_Failure: {YES}, Log_Validation: {NO}, : {},
Select_Features_Option: {All}, Features: {}, : {},
Select_Testcases_Option: {}, TestCases: {}] on Built-In Node started at
2026-02-16T06:01:03Z completed in 85018ms completed: ABORTED
2026-02-16 06:02:30.290 - FLASK-DEBUG - is_build_running: called
pipeline=Policy-Regression job_id=1
2026-02-16 06:02:30.290 - FLASK-DEBUG - is_build_running: endpoint=https://
127.0.0.1:8443/job/Policy-Regression/1/api/json?tree=building
2026-02-16 06:02:30.290 - FLASK-DEBUG - jenkins_request: called
method=POST endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/
json?tree=building has_data=False
2026-02-16 06:02:30.290 - FLASK-DEBUG - jenkins_request: method=POST
endpoint=https://127.0.0.1:8443/job/Policy-Regression/1/api/json?
tree=building timeout=(10, 27) delete_session=False
2026-02-16 06:02:30:307 - JENKINS-DEBUG - /job/Policy-Regression/1/api/
json by guiadmin from 127.0.0.1
2026-02-16 06:02:30.310 - FLASK-DEBUG - https://127.0.0.1:8443 "POST /job/
Policy-Regression/1/api/json?tree=building HTTP/1.1" 200 89
2026-02-16 06:02:30.311 - FLASK-DEBUG - jenkins_request: response
status=200
2026-02-16 06:02:30.311 - FLASK-INFO - is_build_running: build not running
with values, pipeline=Policy-Regression job_id=1
2026-02-16 06:02:30.311 - FLASK-DEBUG - close_session: closing session
2026-02-16 06:02:30.312 - FLASK-INFO - stop: completed pipeline=Policy-
Regression job_id=1 status=200 response_len=15
2026-02-16 06:02:30.312 - FLASK-DEBUG - response_generator: called
status=200 message_type=str
2026-02-16 06:02:30.313 - FLASK-INFO - 127.0.0.1 - - [16/Feb/2026
06:02:30] "POST /job/Policy-Regression/1/stop?immediate=true HTTP/1.1" 200
-
2026-02-16 06:02:30:884 - JENKINS-DEBUG - /job/Policy-Regression/
buildHistory/ajax by guiadmin from 10.233.101.0
```

2.6 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 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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>
  ```

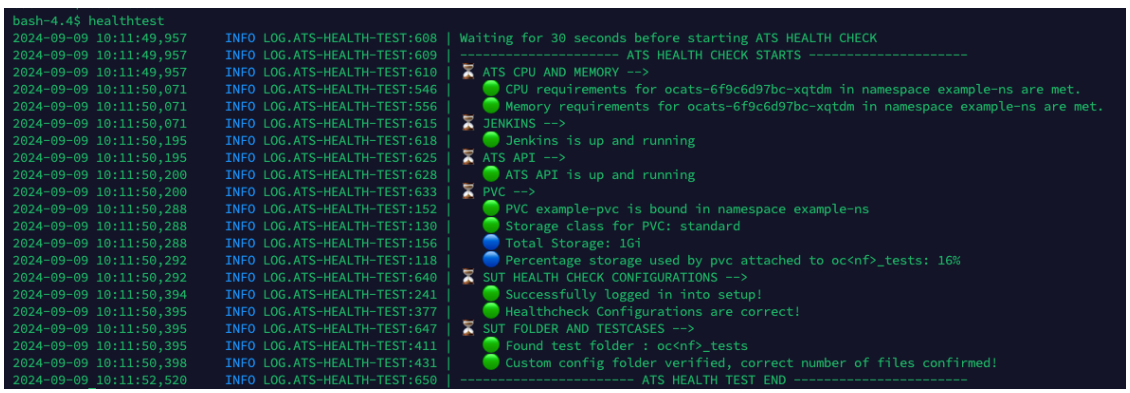
  .
- For subsequent checks, rerun the tool through ATS bash with the command:

  ```
  healthtest
  ```

  .

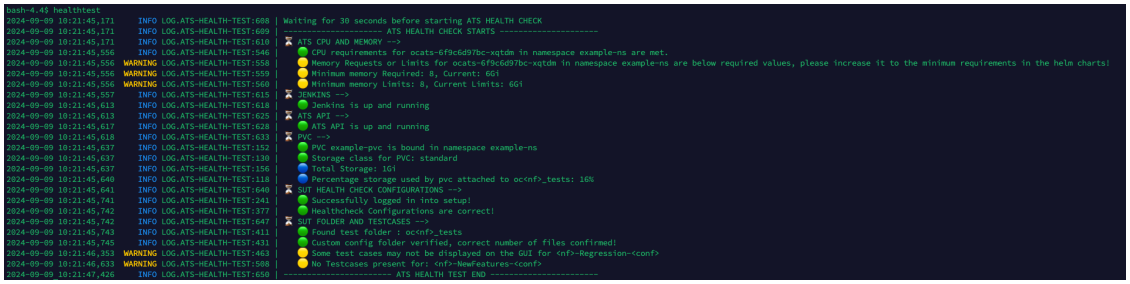
The following screenshots provide visual examples of different scenarios:

Figure 2-19 Success Health Check



Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-20 Warning and Errors



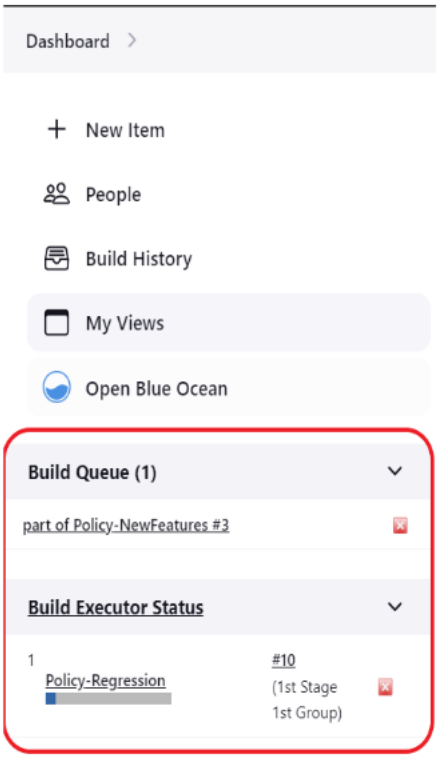
> **Note**
>
> The highlighted areas illustrate a health check with warnings and errors that require further investigation or action.

2.7 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.

Figure 2-21 Build Executor Status



Copyright © 2019, 2026, Oracle and/or its affiliates.

2.8 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.

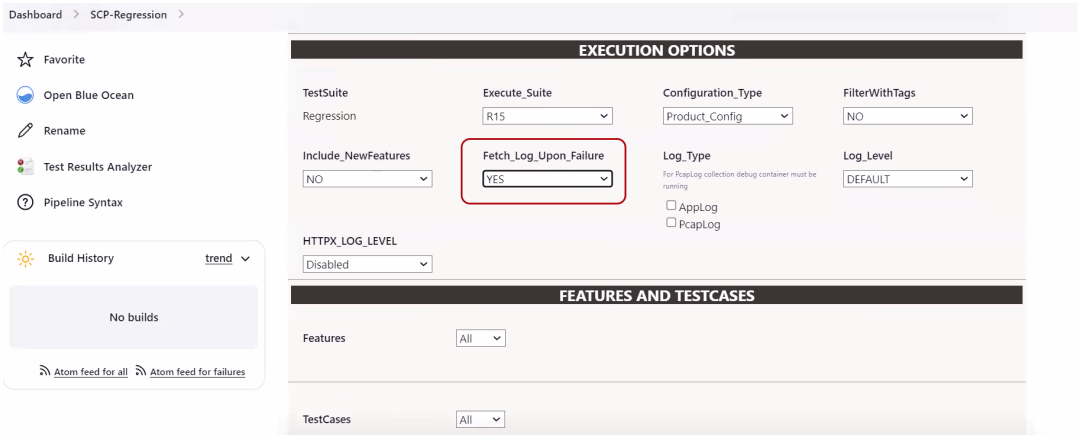
Application Log Collection can be implemented by using OpenSearch or Kubernetes Logs. In both these implementations, logs are collected per scenario for the failed scenarios.

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.

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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **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).
- 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

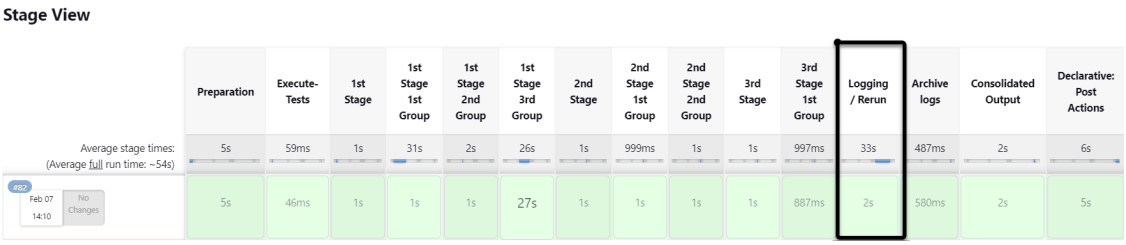
The following tasks are carried out in the background to collect logs:

- 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.

2.8.1 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 2-23 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.

2.9 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- 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
```

2.9.1 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

Copyright © 2019, 2026, Oracle and/or its affiliates.

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.

2.9.2 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 ####`.

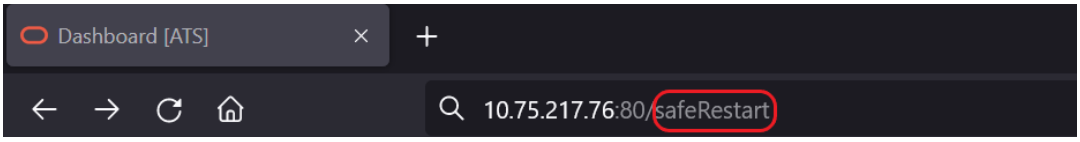
2.9.3 Restarting Jenkins without Restarting Pod

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`.

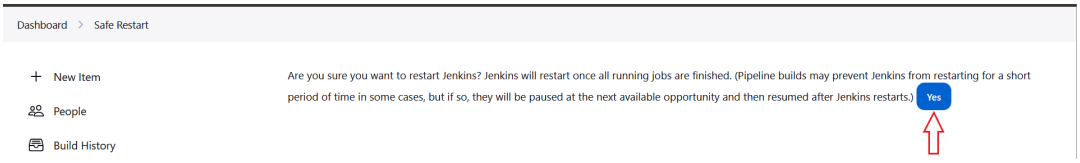
Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-24 Safe Restart



3. Click **Yes**.

Figure 2-25 Restart Jenkins



2.9.4 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **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.

2.9.5 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.

2.10 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

2.11 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:

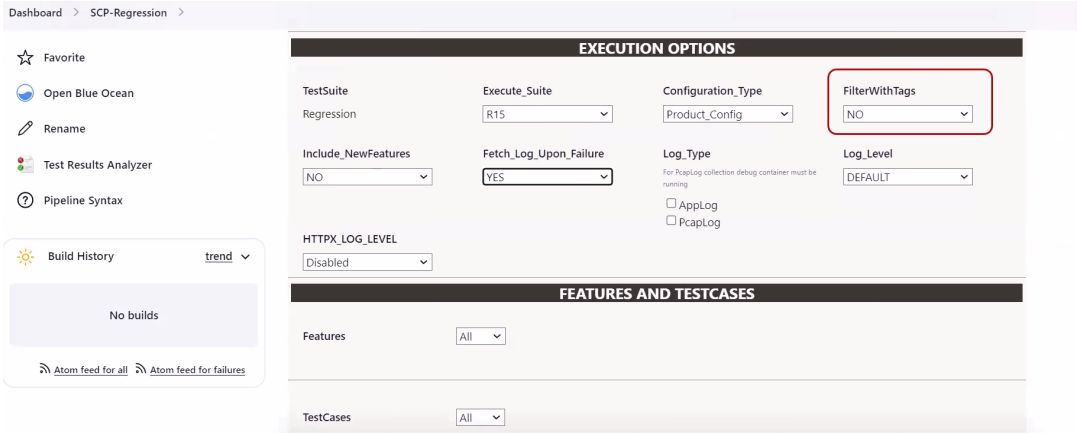
- Feature_Include_Tags: The features that contain either of the tags available in the **Feature_Include_Tags** field are considered for tagging.
  - For example, "cne-common", "config-server". All the features that have either "cne-common" or "config-server" tags are taken into consideration.
- Feature_Exclude_Tags: 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

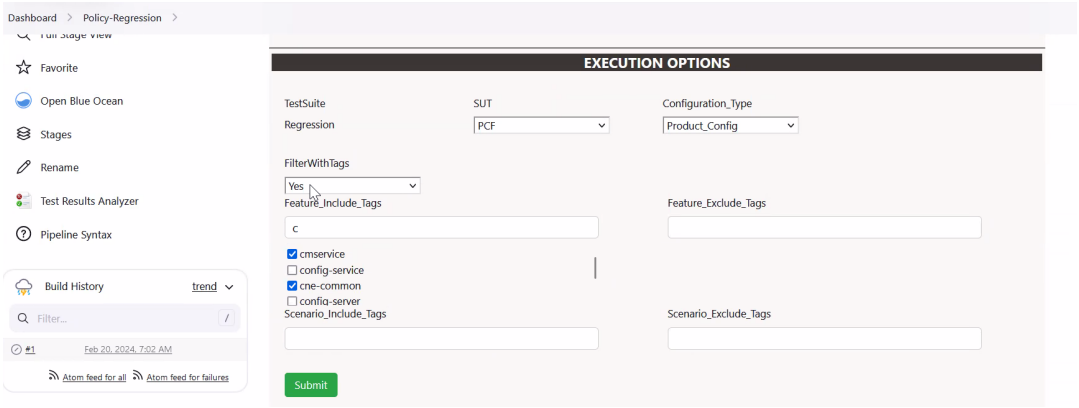


3. Select **Yes** from the **FilterWithTags** drop-down menu. The result shows four input fields. The default value of **FilterWithTags** field is "No".

Copyright © 2019, 2026, Oracle and/or its affiliates.

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.

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.

2.11.1 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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

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

Copyright © 2019, 2026, Oracle and/or its affiliates.

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

Feature_Include	Feature_Exclude	Scenario_Include	Scenario_Exclude	Results
-	"ghi"	-	"sanity","cne"	Features without the "ghi" tag and scenarios without the "sanity" and "cne" tags from filtered features are taken into consideration.
-	-	"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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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

Feature_Include	Feature_Exclude	Scenario_Include	Scenario_Exclude	Results
"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.

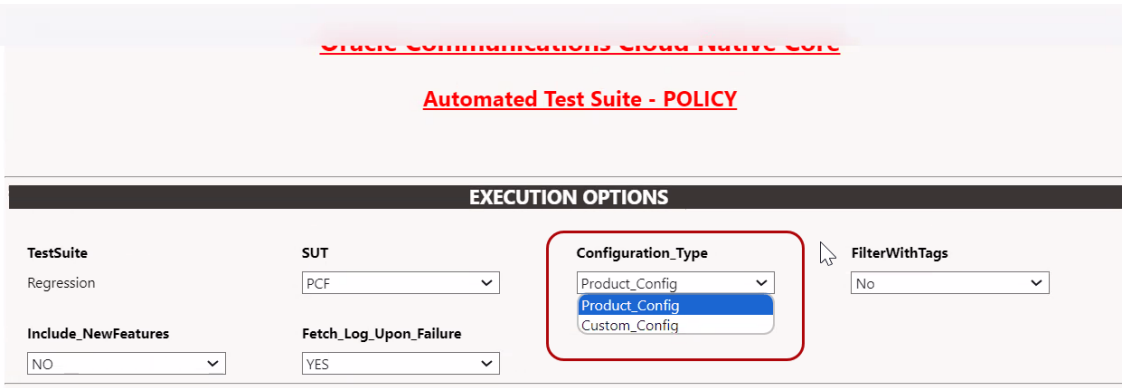
> **Note**
>
> - The tags mentioned in the table are just examples; they may or may not be actually used.
> - The **Replay** option in the Jenkins GUI is not supported for tag-related test case execution. Always trigger builds related to tagging from the **Build with Parameter** step, and do not replay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

2.12 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

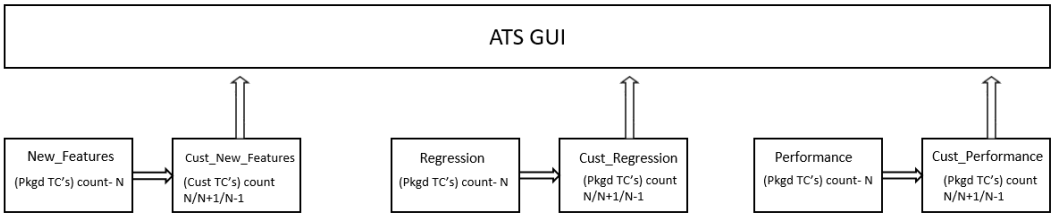


Copyright © 2019, 2026, Oracle and/or its affiliates.

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.

Figure 2-29 Summary of Custom Folder Implementation



2.13 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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
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
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

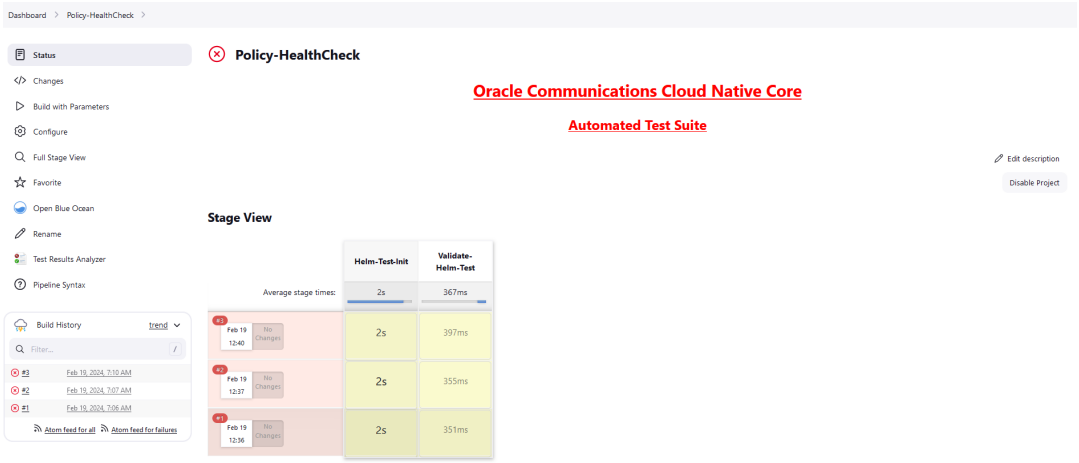
To run Health Check pipeline:

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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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]
```

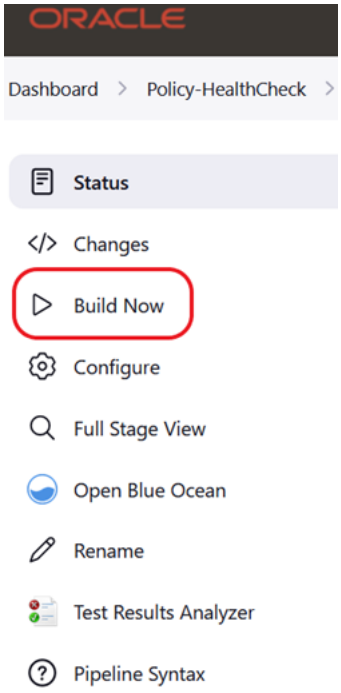

Copyright © 2019, 2026, Oracle and/or its affiliates.

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, 2026, 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, 2026, 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

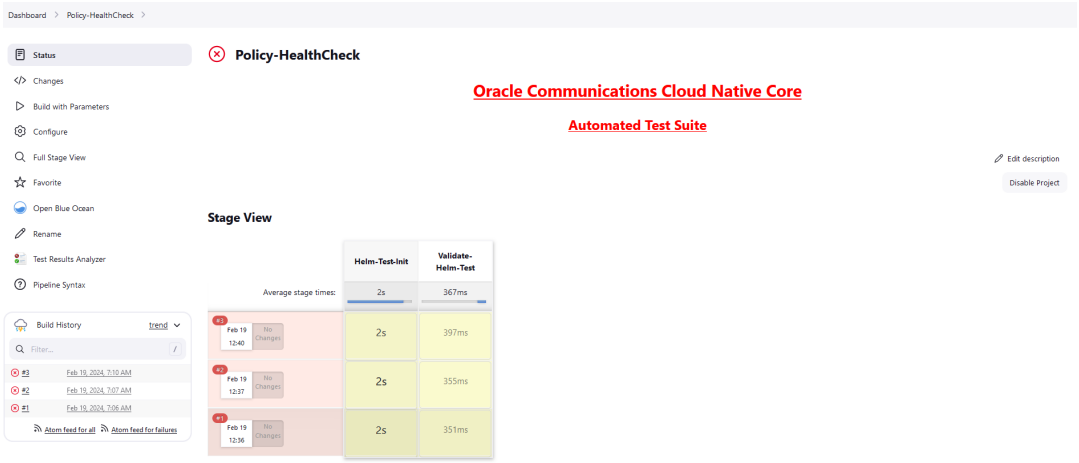
To run ATS Health Check pipeline:

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, 2026, 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]
```

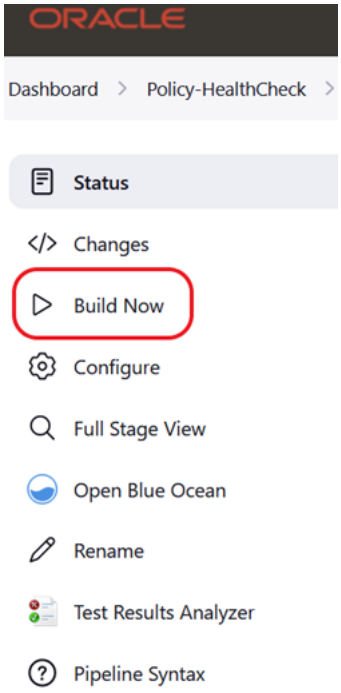

Copyright © 2019, 2026, Oracle and/or its affiliates.

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, 2026, 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, 2026, 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.

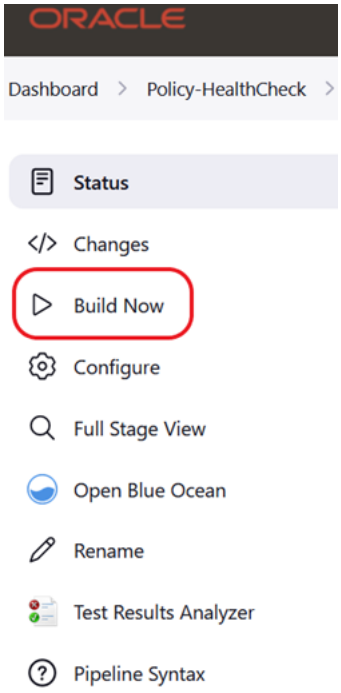
```
//a = helm releases [Provide Release Name with Comma Separated if more
than 1 ]
//b = Namespace, If not applicable to WEBSCALE environment then remove the
argument
//c = helm command name [helm or helm2 or helm3]
```

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, 2026, 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.

2.14 Individual Stage Group Selection

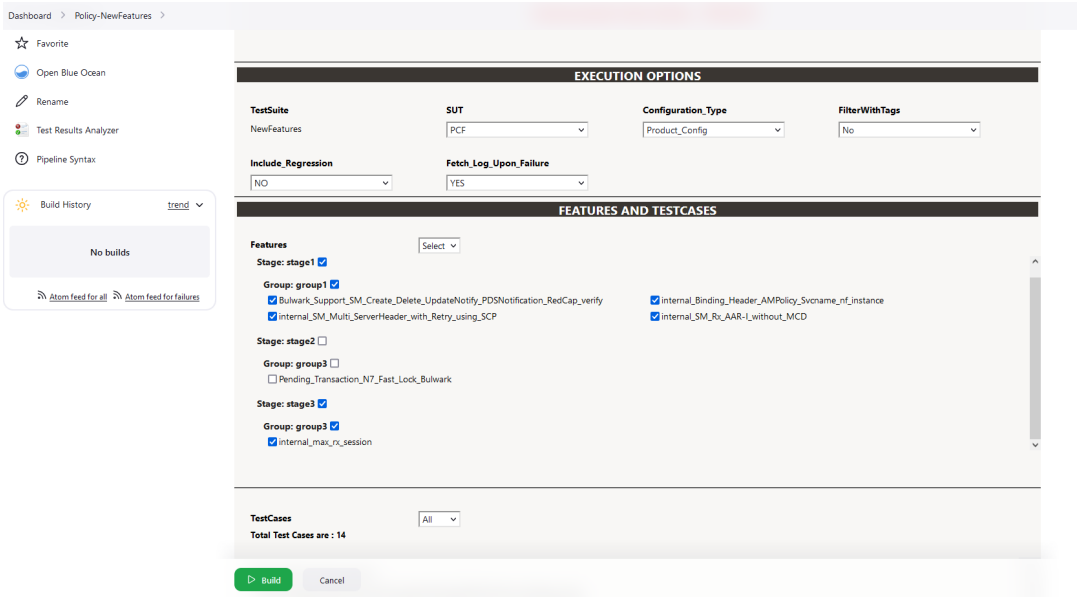
The Individual Stage Group Selection feature allows you to select and execute a single or multiple stages or groups by selecting a check box for the corresponding stage or group.

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, 2026, Oracle and/or its affiliates.

Figure 2-38 Stages or Groups Selection

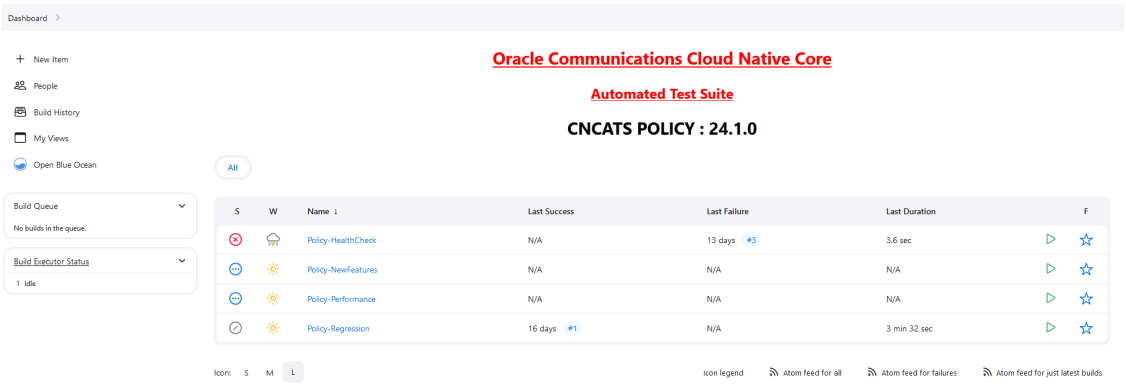


4. Scroll down to click **Build**.

2.15 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.

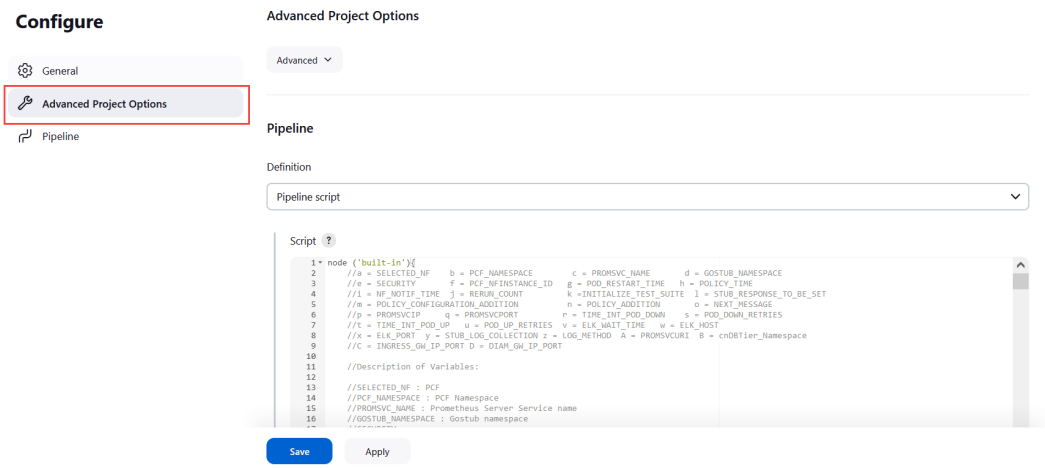
2.15.1 Configure <NF>-Performance Pipeline

Perform the following to configure the performance pipeline:

Copyright © 2019, 2026, 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.

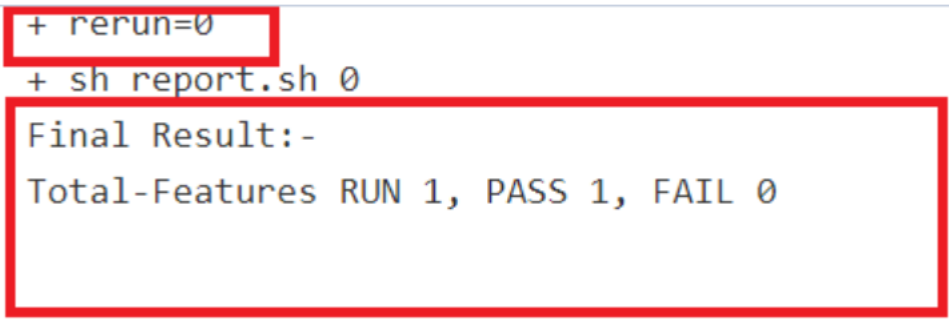
2.16 Managing Final Summary Report, Build Color, and Application Log

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, 2026, 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, 2026, 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
```

Display the Count and Names of Skipped Features Whenever Skipped During a Rerun
Starting from the 25.2.200 release, ATS includes additional details in the Overall Result Summary by showing the count and names of any features that are skipped during a rerun. The following screenshot depicts this scenario:

Figure 2-46 Overall Result Summary Whenever Features Skipped During a Rerun

```
+ set +x
**********************************Overall Result****************************************
Failing scenarios:
Stage:stage1 Group:group1
  YamlSchema_Import_Export/YamlSchema_Import_Export.feature:58  Re_Import_YamlSchema_Verify_SMPolicy
Stage:stage2 Group:group3
  N7_KPIs/N7_KPIs.feature:55  N7_KPIs_Create_Update_Delete
Stage:stage3 Group:group1
  Dynamic_Policy_Override/Dynamic_Policy_Override.feature:25  Dynamic_Override_Qos_Data

Took 0h12m45s

Initial Run:-
Features RUN 3, PASS 0, FAIL 3

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

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

Final Result:-
Total-Features RUN 3, PASS 0, FAIL 3


Total skipped features count: 1
stage: stage3  Group: group1
  Dynamic_Policy_Override/Dynamic_Policy_Override.feature:25
[Pipeline] }
[Pipeline] // script
[Pipeline] }
```

Implementing Build Colors

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

Copyright © 2019, 2026, Oracle and/or its affiliates.

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

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

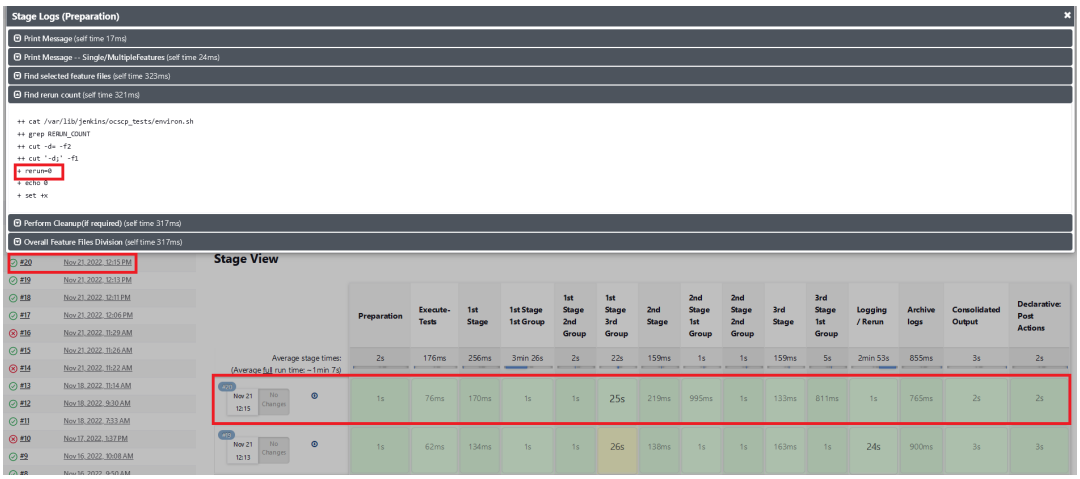
Rerun Values	Rerun set to zero			Rerun set to non-zero		
Passed/Failed	**All Passed in Initial Run**	**Some Failed in Initial Run and Failed in Rerun**	**Some Failed in Initial Run and Passed in Rerun**	**All Passed in Initial Run**	**Some Passed in Initial Run, Rest Passed in Rerun**	**Some Passed in 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:

Copyright © 2019, 2026, Oracle and/or its affiliates.

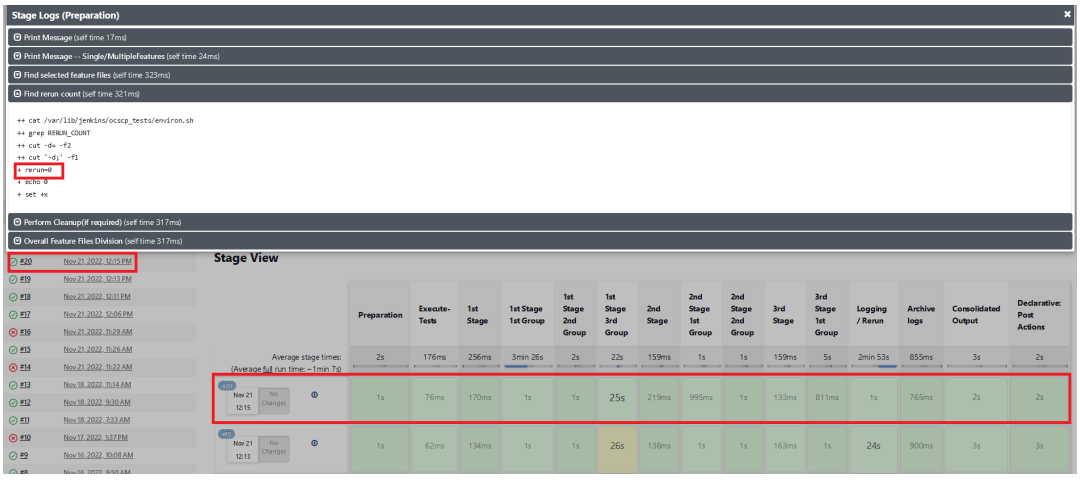
- When `Fetch_Log_Upon_Failure` is set to YES and `rerun_count` is set to 0, test cases pass in the initial run. The pipeline will be green, and its status will show as blue.

Figure 2-47 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.

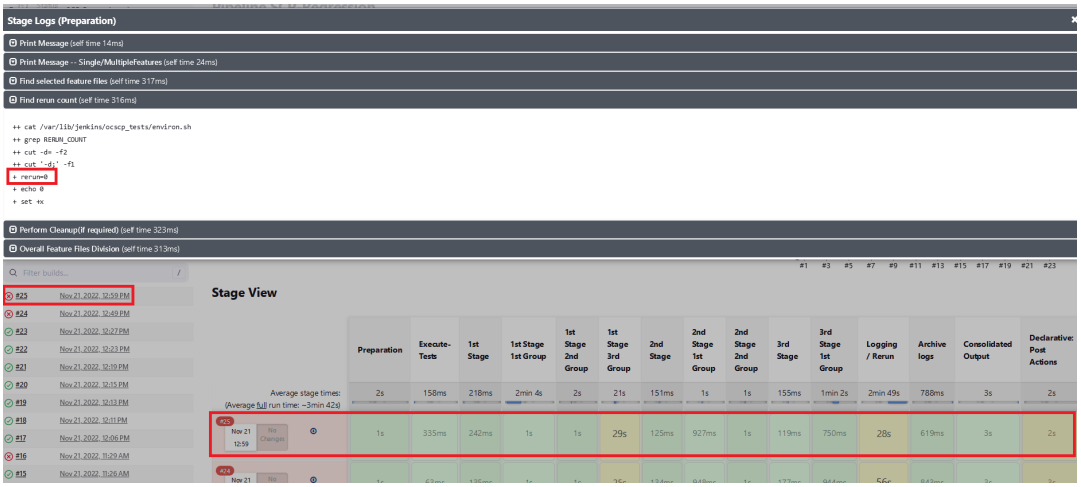
Figure 2-48 Test Cases Fail on the Initial Run but Pass in the Rerun



- 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.

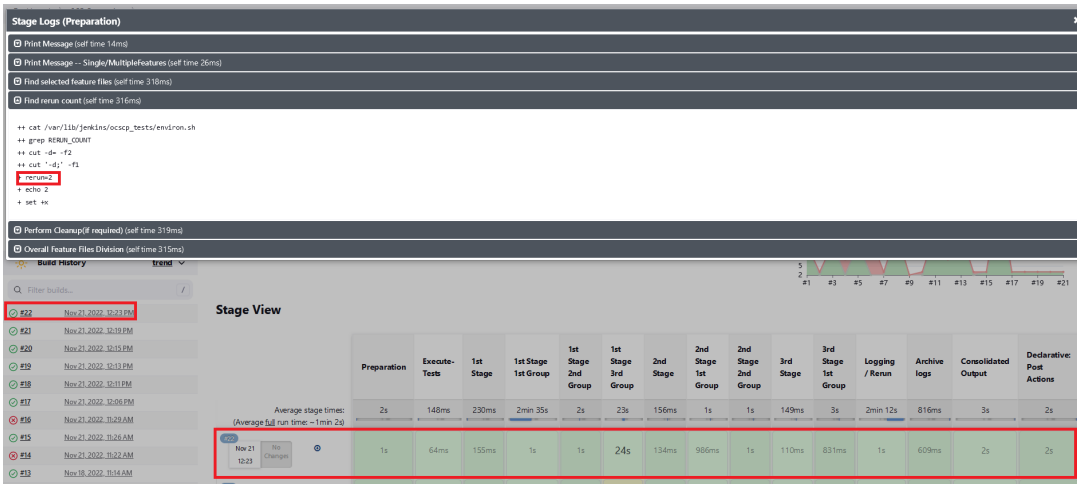
Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-49 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.

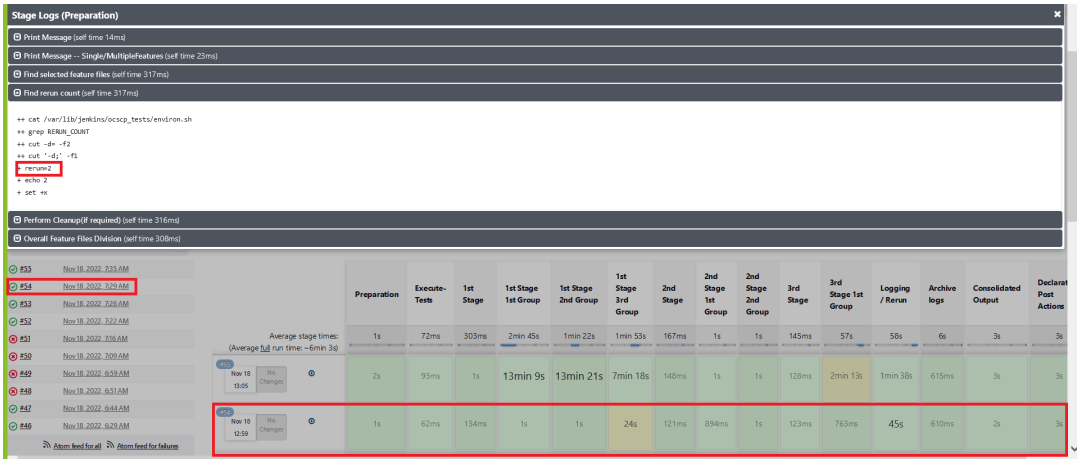
Figure 2-50 All of the Test cases Pass in the Initial Run



- 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.

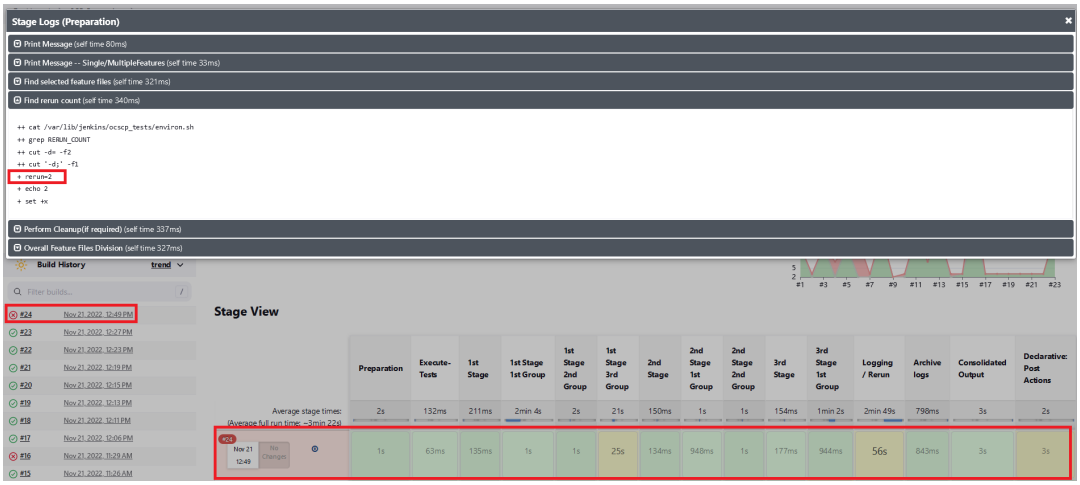
Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-51 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.

Figure 2-52 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-53 Stage View When Behave Process is Incomplete

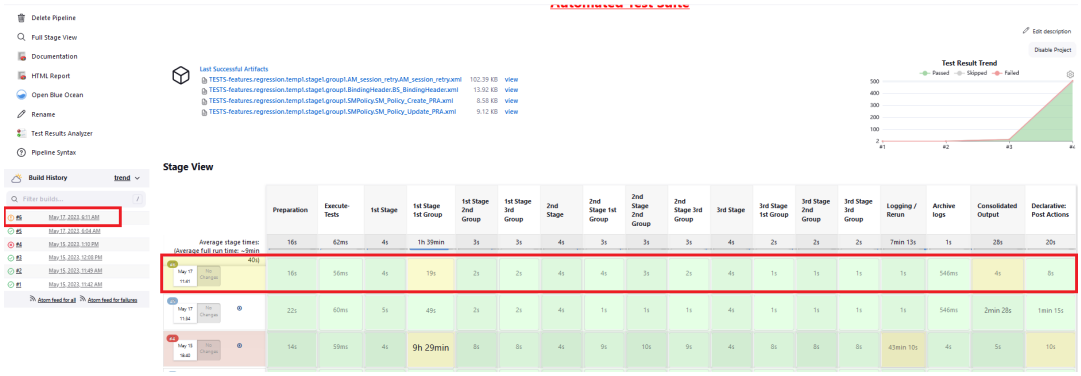
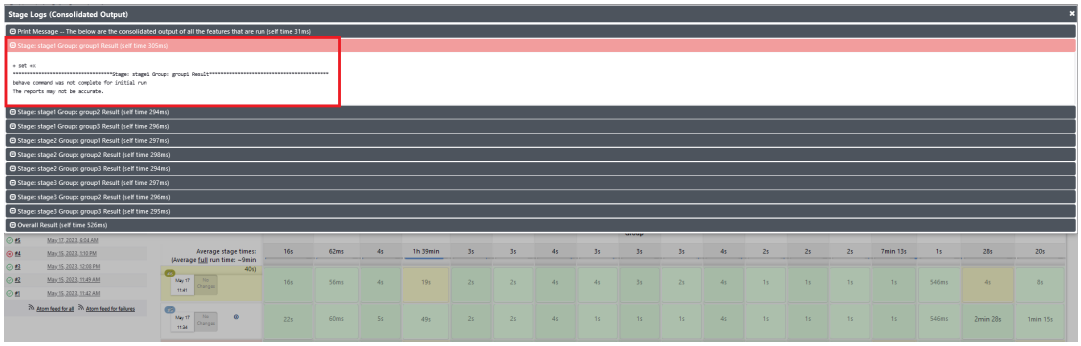


Figure 2-54 Consolidated Report for a Group When a Behave Process was Incomplete



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.
>
> .

Copyright © 2019, 2026, Oracle and/or its affiliates.

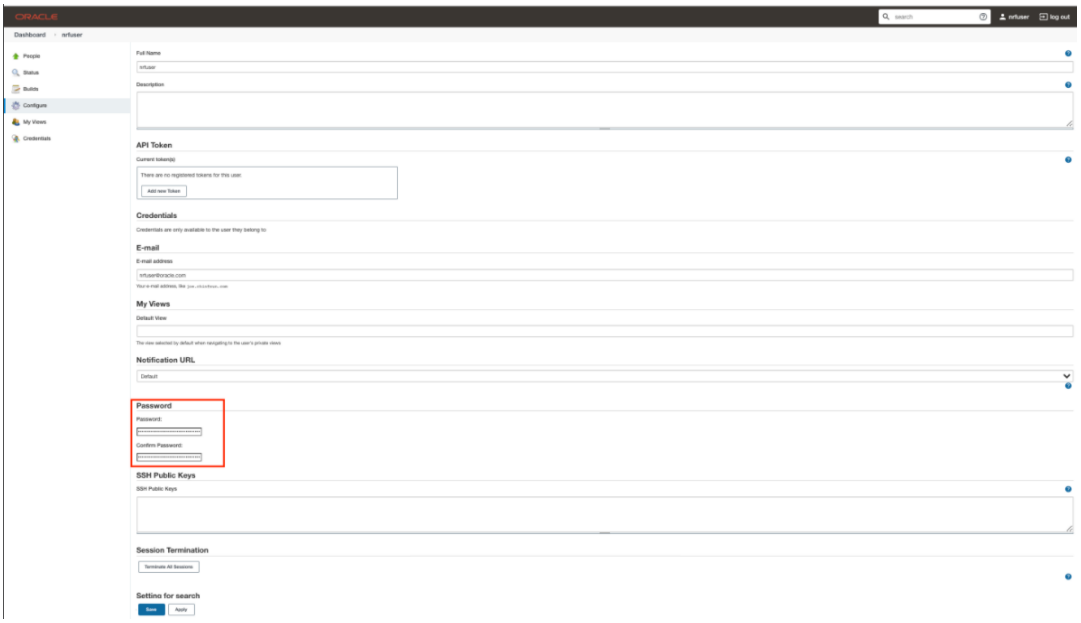
2.17 Modifying Login Password

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..

Figure 2-55 Logged-in User Details



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

2.18 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.

2.18.1 Feature Level Selection

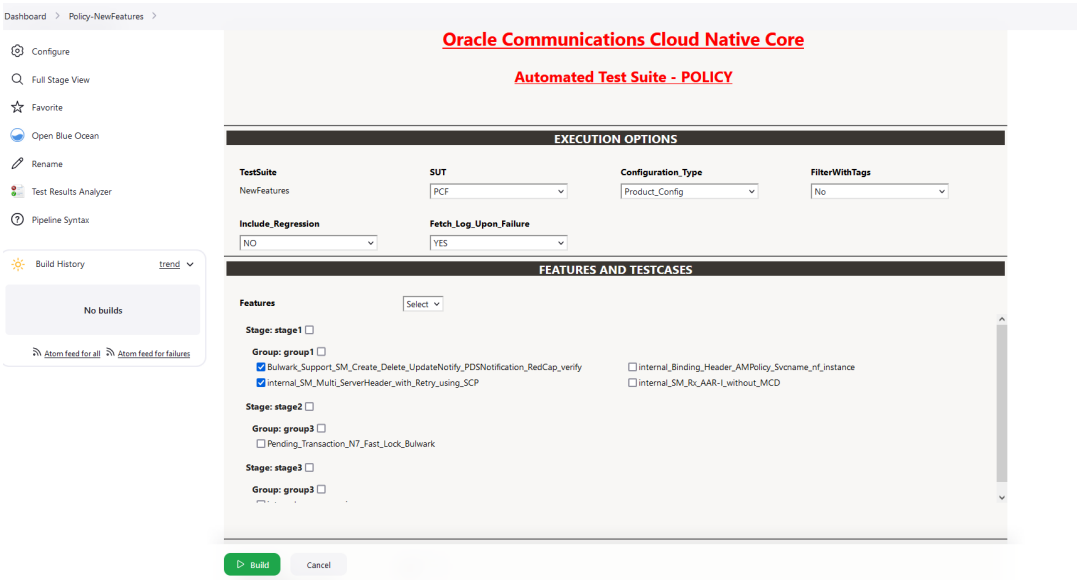
Perform the following procedure to run the single or multiple features:

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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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.

Figure 2-56 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**.

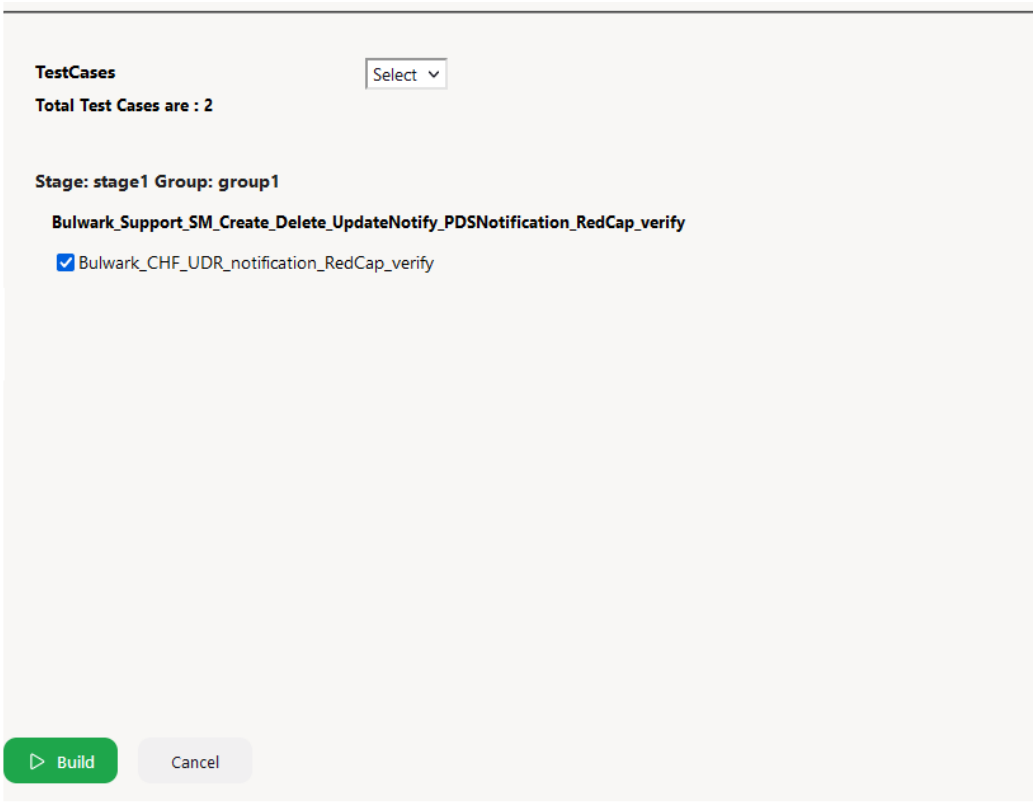
2.18.2 Scenario Selection

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, 2026, Oracle and/or its affiliates.

Figure 2-57 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**.

2.19 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.

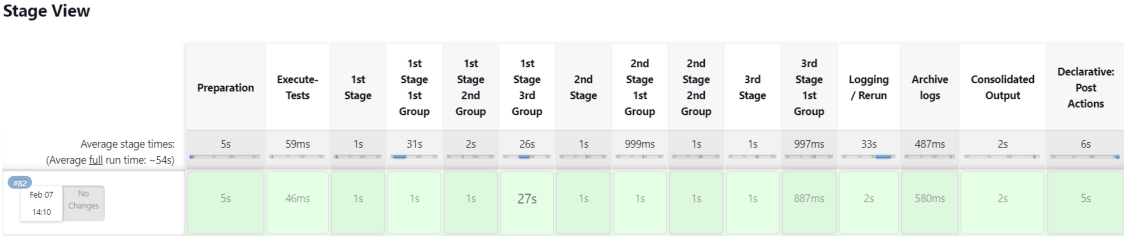
ATS currently runs all its tests in a sequential manner, which is time-consuming. With parallel test execution, tests can be run concurrently rather than sequentially or one at a time. Test cases or feature files are now separated into different folders, such as stages and groups, for concurrent test 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, 2026, Oracle and/or its affiliates.

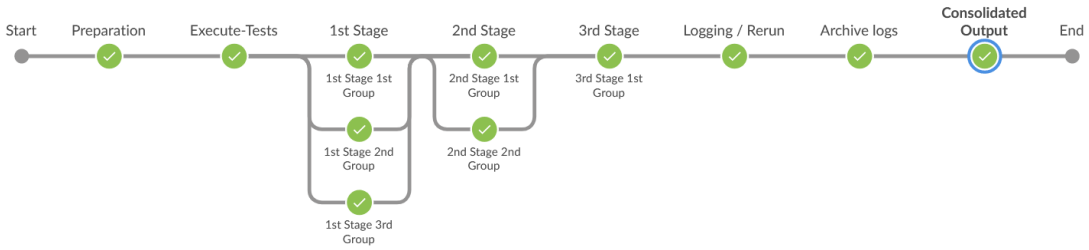
Figure 2-58 Pipeline Stage View



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 2-59 Pipeline Blue Ocean View



Merged Execution

The Merged Execution option allows you to run both new features and regression pipeline test cases together.

Perform the following procedure to run the test cases of both pipelines:

1. Log in to ATS using the respective <NF> login credentials.
   Where, <NF> indicates any 5G CNC Network Function (NF).
2. On the <NF> home page, click any new feature or regression pipeline.
3. On the displayed <NF>-NewFeature or <NF>-Regression web page, click **Build with Parameters** from the left navigation pane.
4. In the EXECUTION OPTIONS section, configure **Include_NewFeatures** and **Include_Regression** to one of the values:
   - **NO**: This is the default value. Only the test cases of the current pipeline, either Regression or NewFeatures, are run.
   - **YES**: The test case names (feature file and scenario names) of both Regression and NewFeatures are displayed on the ATS GUI, and the test cases of both pipelines are run.
5. To select and run single or multiple features of both pipelines, in the FEATURES AND TESTCASES section, select an appropriate value from the **Features** drop-down list:
   - **All**: Selects all the features of the pipeline.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **Select**: Allows you to select the required features of the pipeline.

You can also select tags, stages, or groups as described in ATS Tagging Support and Individual Stage Group Selection.

6. Click **Build**.
Sample test case result summary report:

```
*****************************NewFeatures
Result*****************************

Final Result:-
Total-Features RUN 3, PASS 3, FAIL 0




*****************************Regression
Result*****************************
Failing scenarios:
Stage:stage1 Group:group1
  AM_Concurrency/AM_Concurrency.feature:27  AM_Concurrency_Unlocked
  AM_Concurrency_Extended_ID/AM_Concurrency_Extended_ID.feature:37
AM_Concurrency_Locked_Extended_Id
  Correlation_Header_Verification/
Correlation_Header_Verification_AM.feature:80
AM_Correlation_Generation_Verification
  Correlation_Header_Verification/
Correlation_Header_Verification_AM.feature:202
AM_Correlation_Forward_Verification
  Pending_Transaction_AM_Retry_NoRetry/
Pending_Transaction_AM_Retry_NoRetry.feature:56
Pending_Transaction_AM_Update_Notify_No_Retry
  Pending_Transaction_AM_Retry_NoRetry_Extended_ID_RedCap/
Pending_Transaction_AM_Retry_NoRetry_Extended_ID_RedCap.feature:57
Pending_Transaction_AM_Update_Notify_Retry_Extended_ID
  Revalidating_nUDR_async_nCHF_subscriptions_AM/
Revalidating_nUDR_async_nCHF_subscriptions_AM.feature:68
Revalidating_nUDR_async_nCHF_subscriptions_AM
  Subscriber_Activity_Logging_AM_ocLogId/
Subscriber_Activity_Logging_AM_ocLogId.feature:69
Subscriber_Activity_Logging_With_Subscribe_to_Notify_disabled

Initial Run:-
Features RUN 199, PASS 189, FAIL 10

1st Rerun:-
Features RUN 10, PASS 3, FAIL 7

2nd Rerun:-
Features RUN 7, PASS 0, FAIL 7

Final Result:-
Total-Features RUN 199, PASS 192, FAIL 7
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

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

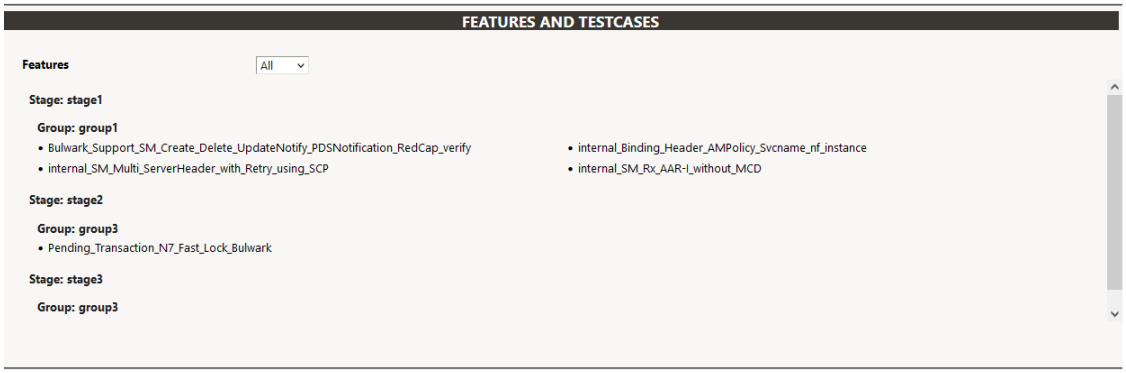
2.19.1 ATS GUI Page Changes

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.

Figure 2-60 Displays Stages and Groups



2.19.2 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.

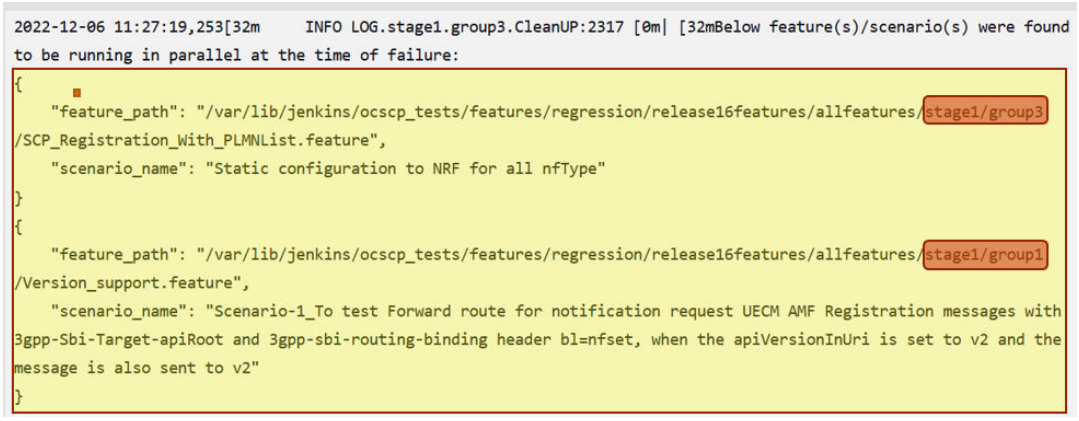
Figure 2-61 Logger Statement

```
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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-62 Absolute Path of Feature File



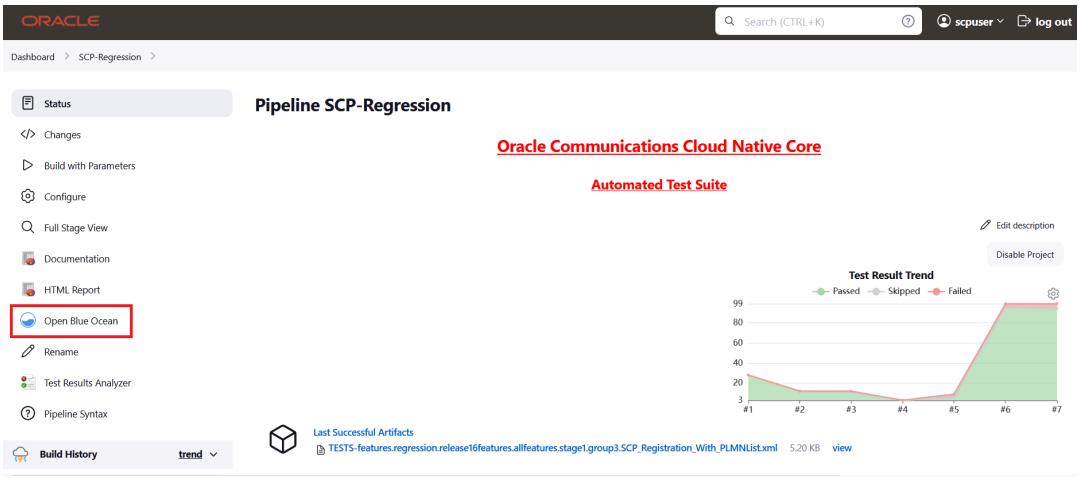
- The test result summary contains a summary for each group and an overall summary, along with the details of failing scenarios (stage-groupwise) and the total time taken by any pipeline execution. For further information, see the Managing Final Summary Report, Build Color, and Application Log.

2.19.3 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.

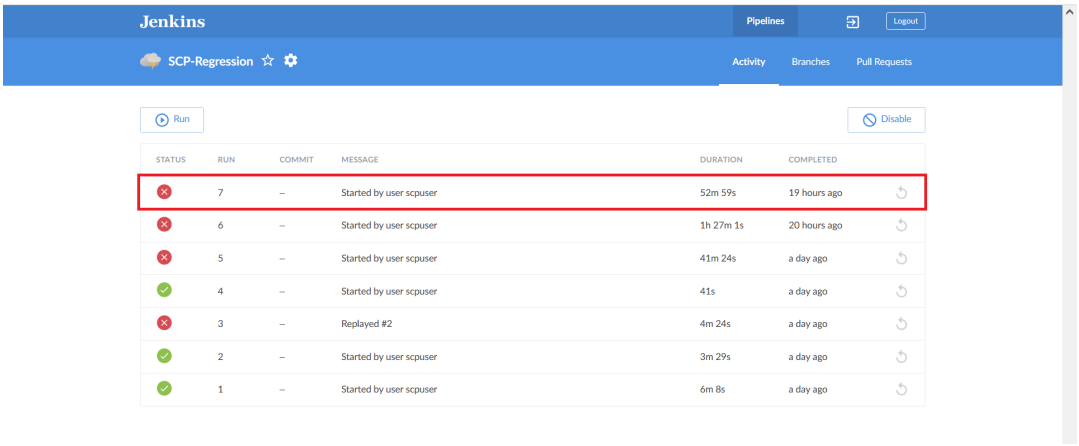
Figure 2-63 Jenkins Pipeline Page



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

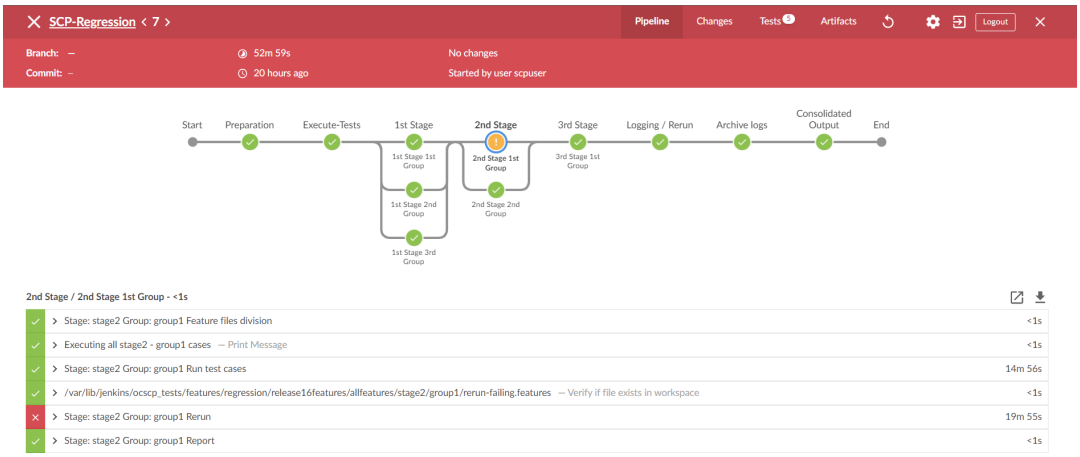
Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-64 Run the Build



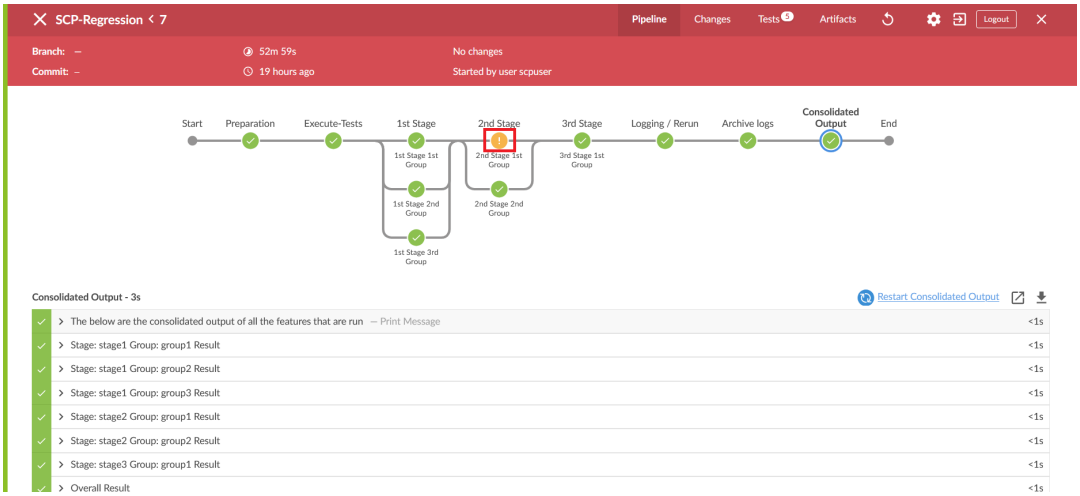
3. The selected build appears. The diagram displays the order in which the different stages, or groups, are executed.

Figure 2-65 Stage Execution



4. Click the desired group to download the logs.

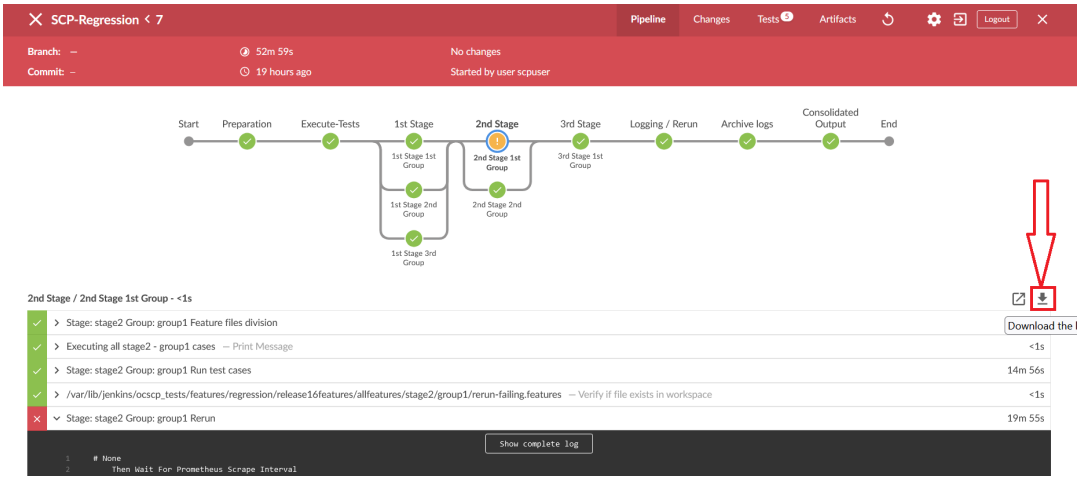
Figure 2-66 Executed Groups



Copyright © 2019, 2026, Oracle and/or its affiliates.

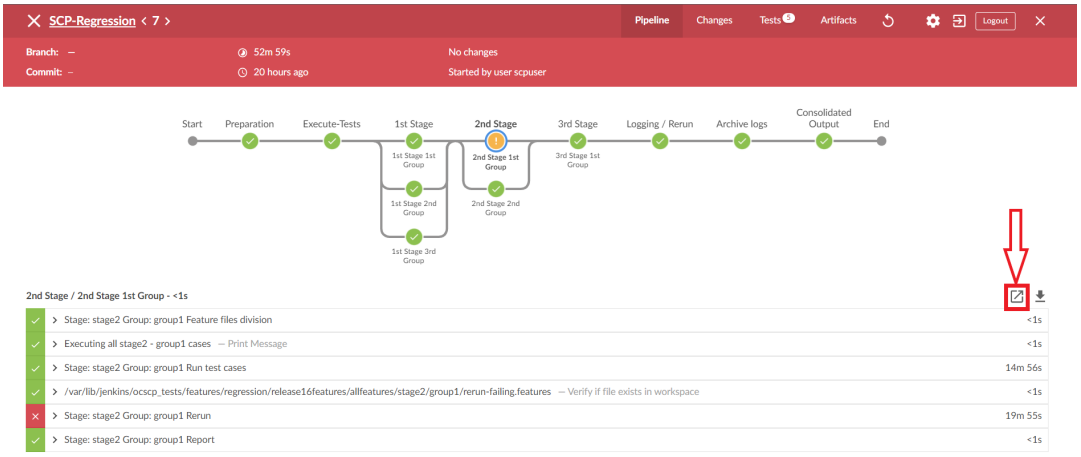
5. Click the **Download** icon on the bottom right of the pipeline. The log for the selected group is downloaded to the local system.

Figure 2-67 Download Logs



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

Figure 2-68 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

4. Click the row labeled **Stage: stage_name>." "group_name> Rerun** to display the rerun logs.

- Using Pipeline Steps Page

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.

2.20 Parameterization

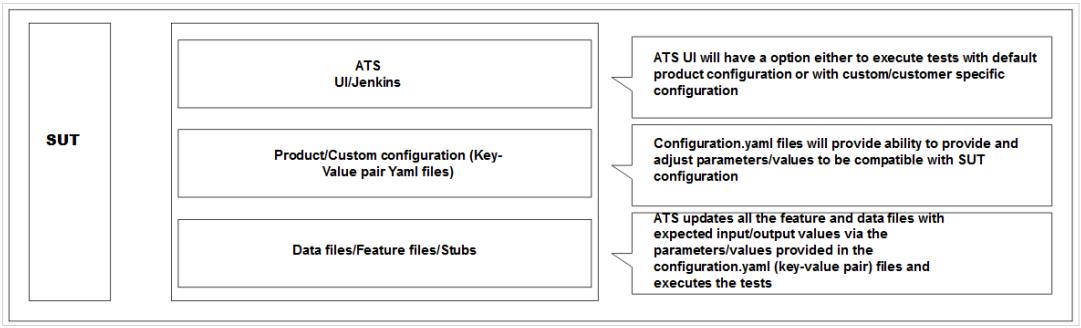
This feature allows you to provide or adjust values for the input and output parameters needed for the test cases to be compatible with the SUT configuration. You can update or adjust the key-value pair values in the `global.yaml` and `feature.yaml` files for each of the feature files so that they are 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

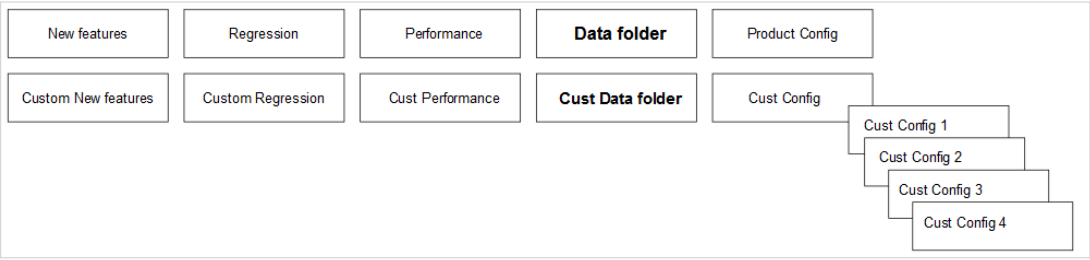
Figure 2-69 SUT Design Summary



In the folder structure:

- 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 2-70 Folder Structure



2.20.1 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.

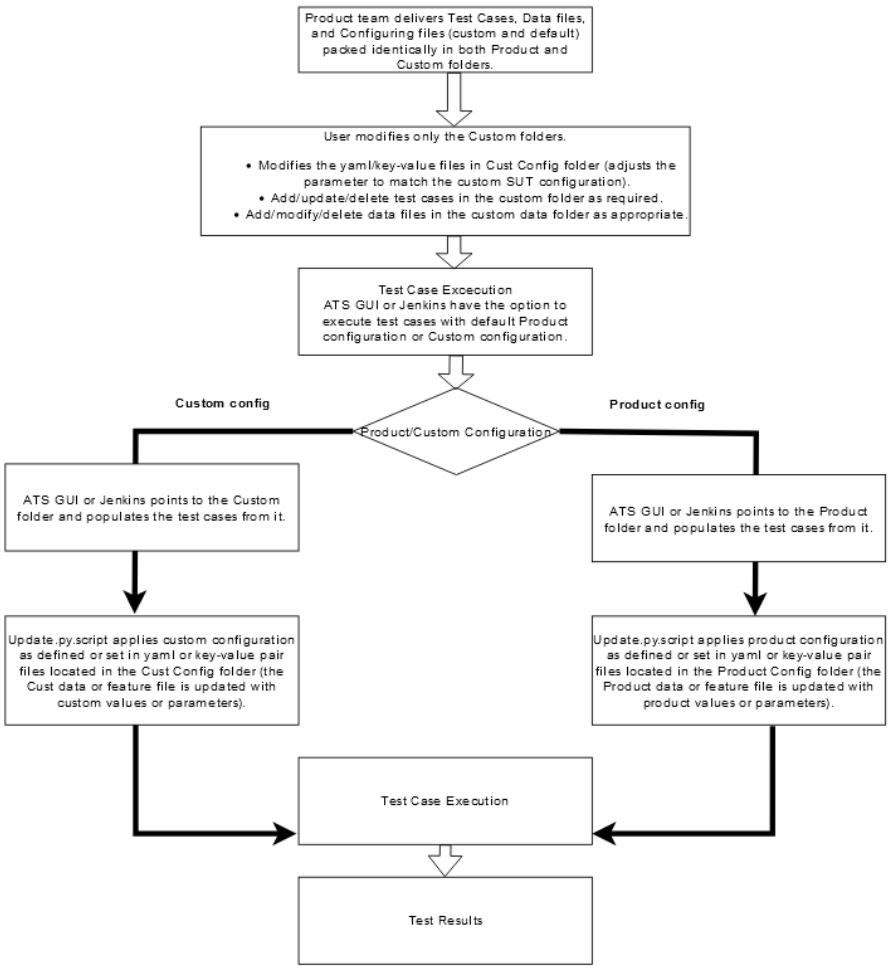
Copyright © 2019, 2026, Oracle and/or its affiliates.

To Run ATS Test Cases

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

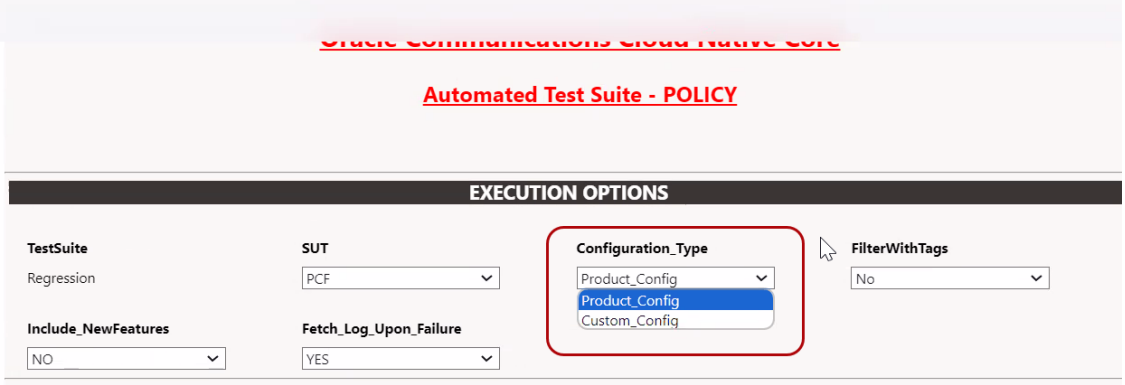
- If custom configuration is selected, then test cases from custom folders are populated on the ATS UI, and custom configuration is applied to them through the key-value pair per yaml file defined or present in the "Cust Config" folder.
- If product configuration is selected, then the test cases from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

Figure 2-71 ATS Execution Flow



Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-72 Sample: Configuration_Type



2.21 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/`

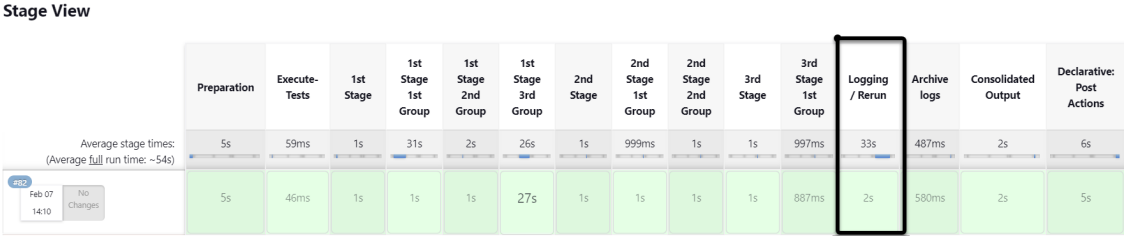
   Pcaplogs is present in zip form. Unzip it to get the log files.

2.21.1 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, 2026, Oracle and/or its affiliates.

Figure 2-73 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.

2.22 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.

2.22.1 Processing Flow

First Time Deployment

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

Subsequent Deployments

Copyright © 2019, 2026, Oracle and/or its affiliates.

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

2.22.2 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
   - 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:
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
>     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.

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 2-74 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

Copyright © 2019, 2026, Oracle and/or its affiliates.

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
```

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.

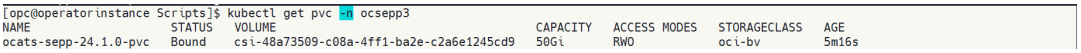
4. To verify whether PVC is bound to PV and is available for use, run the following command:

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


Copyright © 2019, 2026, Oracle and/or its affiliates.

The output of the above command is as follows:

Figure 2-75 Verifying PVC



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

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:

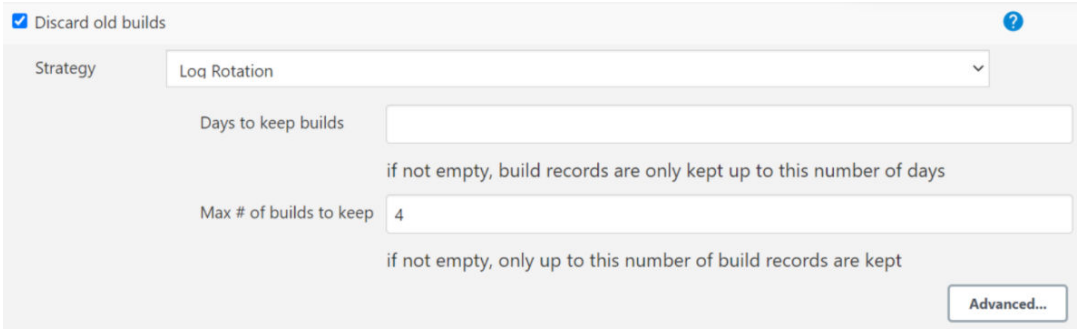
      ```
      kubectl delete po <pod-name> -n <namespace>
      ```

      > **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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-76 Discard Old Builds



> **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.

2.22.3 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**.

2.23 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.

2.23.1 Configuring Single Click Job

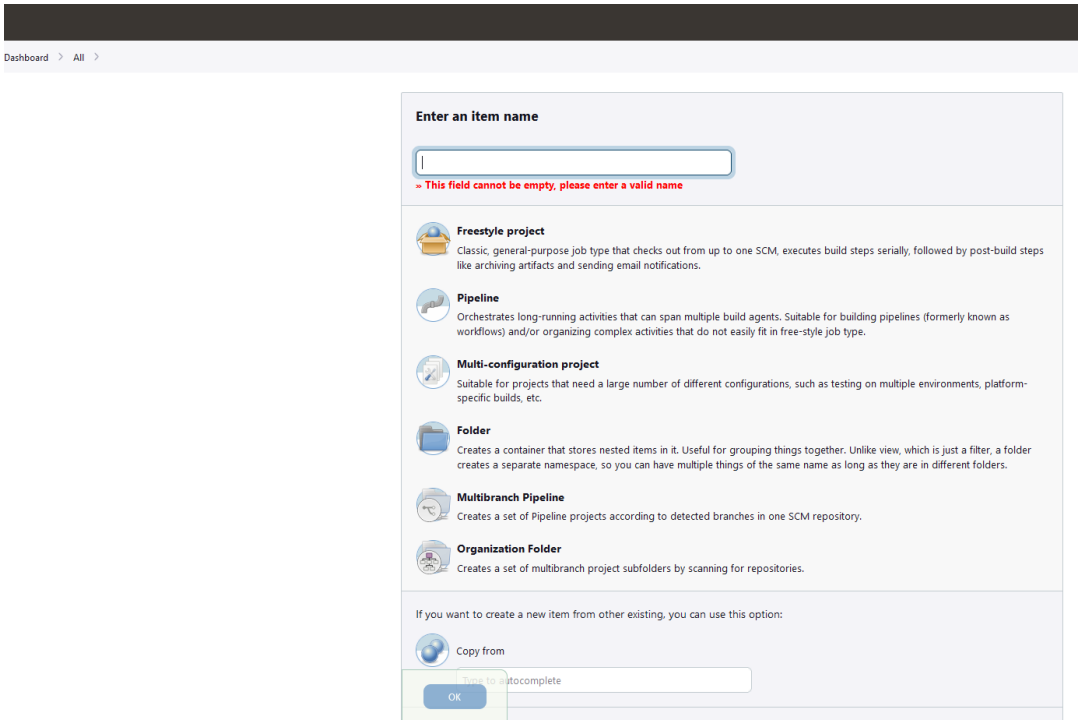
Prerequisite: The network function specific user should have 'Create Job' access.

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, 2026, Oracle and/or its affiliates.

Figure 2-77 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, 2026, 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, 2026, 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.

2.24 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.

2.24.1 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, 2026, 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

2.24.2 Adding Ingress and Egress Rules to Access the OCI Console

Perform the following procedure to add ingress and egress rules:

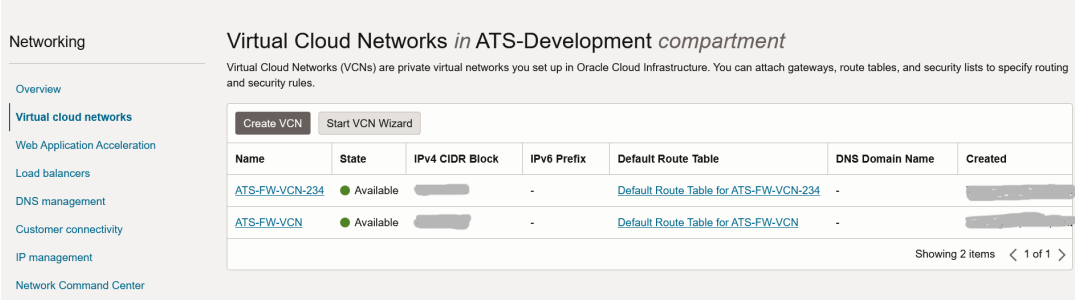
1. Log in to the OCI Console using your login credentials.
2. On the OCI Console Dashboard, click **Navigation menu**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-78 Navigation menu

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

Figure 2-79 Virtual cloud networks

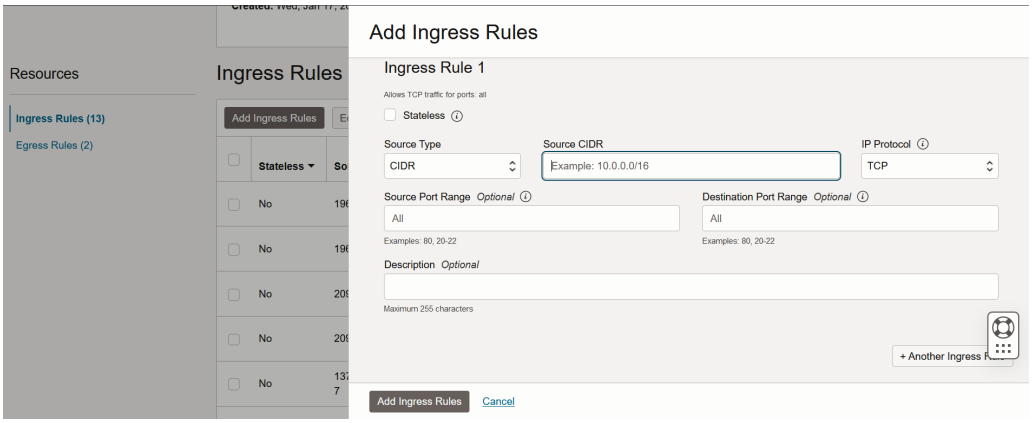


4. Click the required OKE cluster VCN.
   A list of all subnets in that VCN is displayed.
5. Save the value of the "IPv4 CIDR Block" field for **nf_lb_subnet**.
   This information can be used in the subsequent steps as value of `<CIDR for Load balancer subnet>`.
6. Save the value of the "IPv4 CIDR Block" 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 2-80 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, 2026, 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, 2026, 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 2-81 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, 2026, 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.

2.25 Support for Transport Layer Security

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

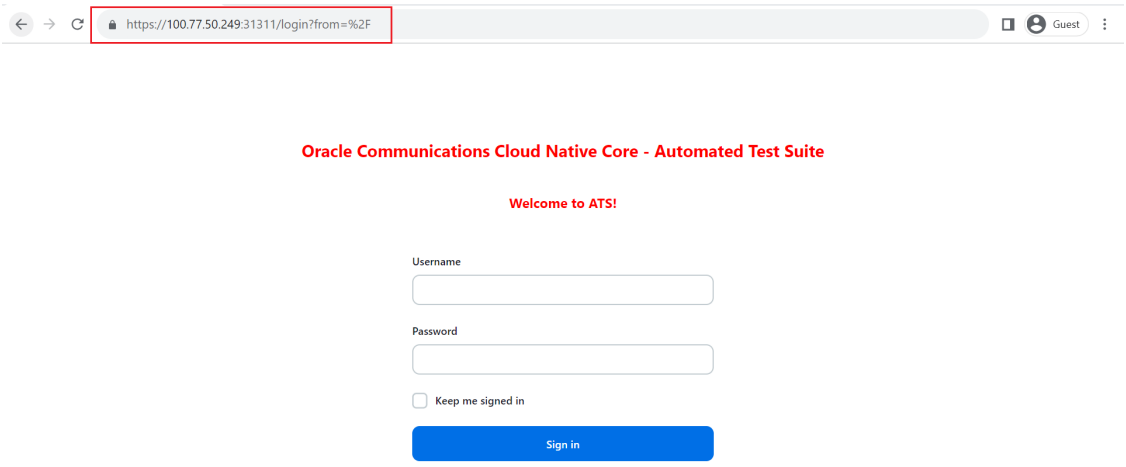
With the support of the TLS feature, Jenkins servers have been upgraded to support HTTPS, ensuring a secure and encrypted connection when accessing the ATS dashboard.

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, 2026, Oracle and/or its affiliates.

Figure 2-82 Access with TLS



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

2.25.1 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.

2.25.1.1 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, 2026, 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, 2026, 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, 2026, 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, 2026, 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, 2026, 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.

2.25.1.2 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, 2026, 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, 2026, 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.

2.26 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.

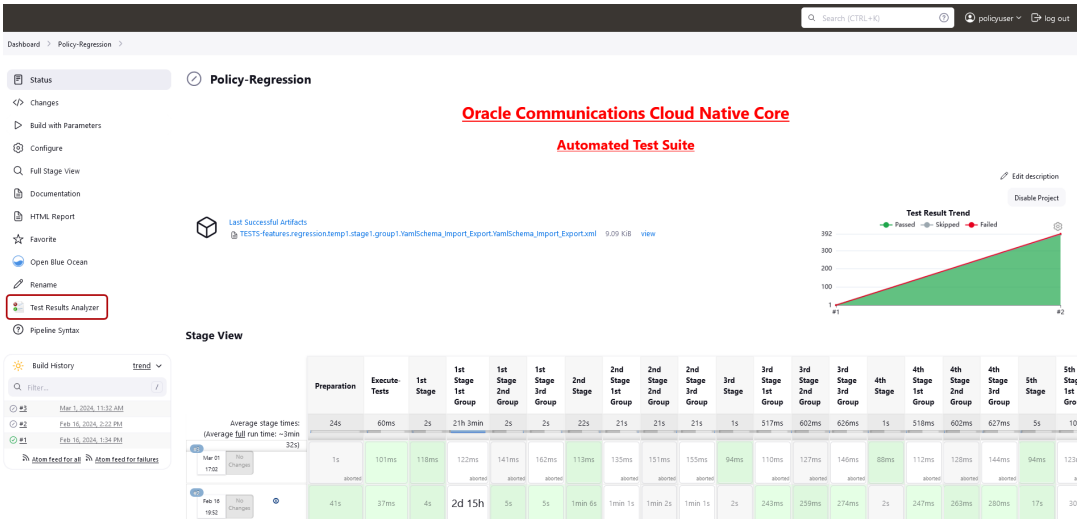
2.26.1 Accessing Test Results Analyzer Feature

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, 2026, Oracle and/or its affiliates.

Figure 2-83 Test Results Analyzer Option



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

Figure 2-84 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, 2026, Oracle and/or its affiliates.

Figure 2-85 Test Result

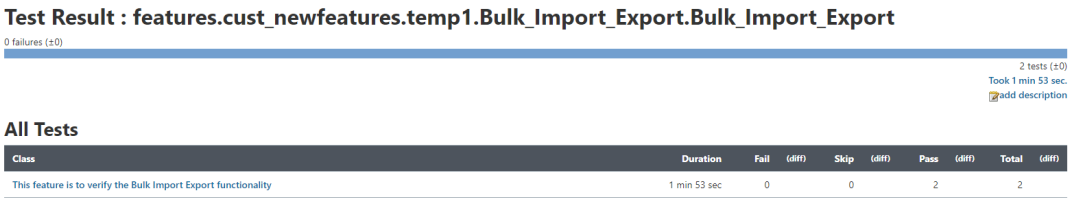
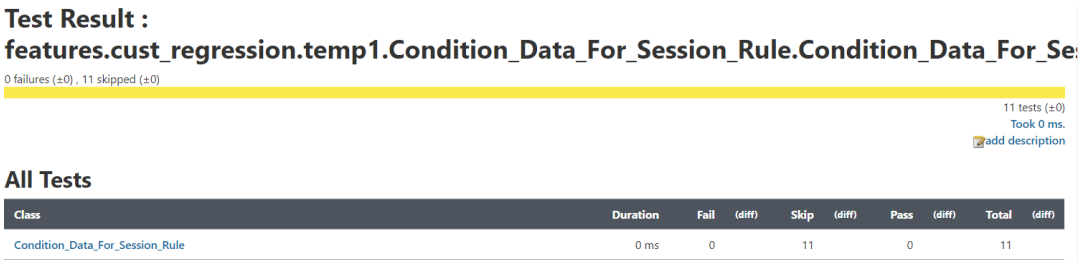


Figure 2-86 Test Result



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 2-87 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, 2026, Oracle and/or its affiliates.

Figure 2-88 Test Result Info

2.27 Support for Test Case Mapping and Count

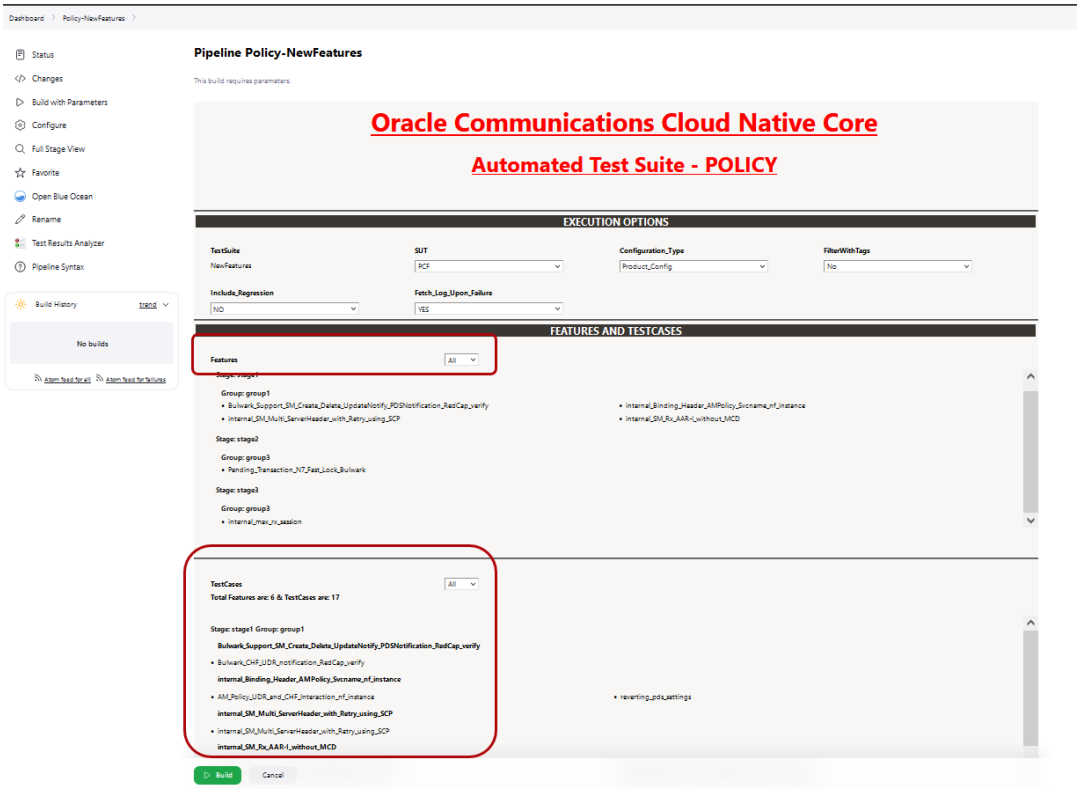
The Test Case Mapping and Count feature displays the total number of features, test cases, or scenarios and their mapping to each feature in the ATS GUI.

2.27.1 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.

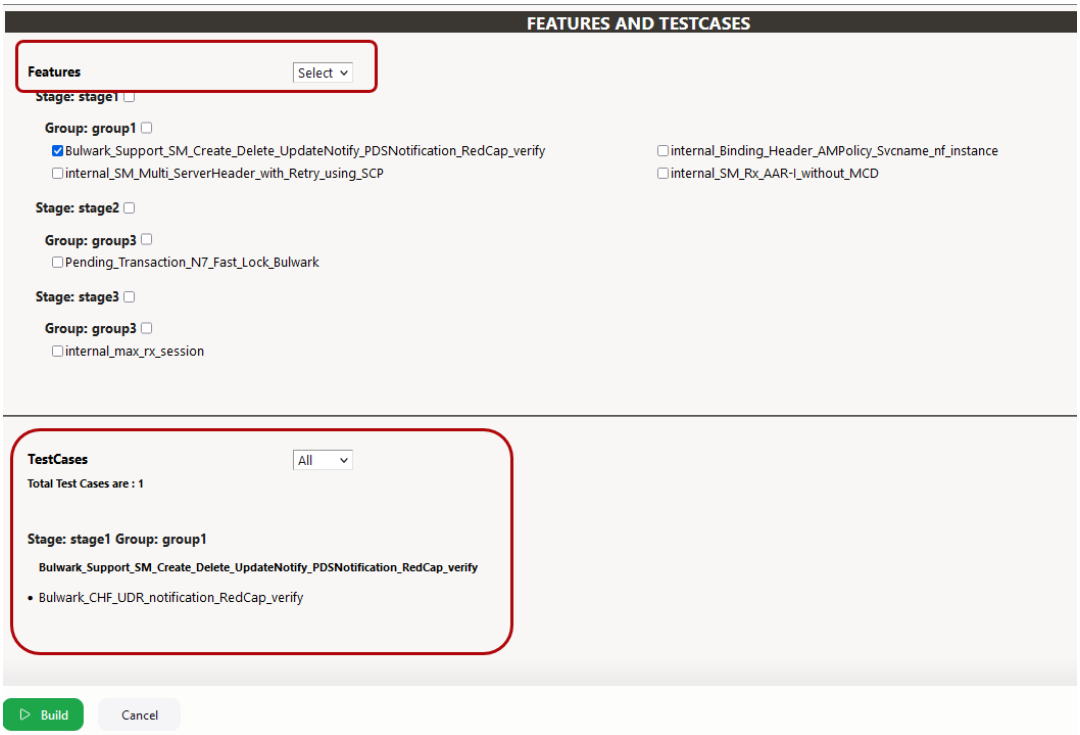
Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 2-89 Test Case Mapping



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

Figure 2-90 Test Cases Details When Select a Single/MultipleFeatures



Copyright © 2019, 2026, 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 NRF
- Installing ATS for NSSF
- Installing ATS for SCP
- Installing ATS for SEPP

3.1 Installing ATS for NRF

3.1.1 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 3-1 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	9	11	0
Grand Total NRF ATS	110.5	130.5	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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-2 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	4	6	1	1	1	4	6	0
ATS Stub (Python)	1	1	1	1	5	5	5	0
ATS Totals						9	11	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 3-3 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

3.1.2 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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 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_26_1_201_0_0.zip
ocats_ocnrf_csar_26_1_201_0_0.zip.sha256
ocats_ocnrf_csar_mcafee-26.1.201.0.0.log
```

11. **Note**

    The above zip file contains all the images and custom values required for 26.1.201 release of OCATS-NRF.

    The `ocats_ocnrf_csar_26_1_201_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-26.1.201.tgz
│    ├─── Licenses
│    ├─── ocats-nrf-26.1.201.tar
│    ├─── Oracle.cert
│    ├─── ocstub-py-26.1.201.tar
│    └─── Tests
├─── ocats_ocnrf.mf
├─── Scripts
│    ├─── ocats_ocnrf_custom_serviceaccount_26.1.201.yaml
│    ├─── ocats_ocnrf_custom_values_26.1.201.yaml
│    └─── ocats_ocnrf_tests_jenkinsjobs_26.1.201.tgz
└─── TOSCA-Metadata
     └─── TOSCA.meta
```

12. Copy the zip file to Kubernetes cluster where you want to deploy ATS.

3.1.3 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_26_1_201_0_0.zip
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

The following docker image tar files are located at **Files** folder:

- ocats-nrf-26.1.201.tar
- ocstub-py-26.1.201.tar

2. Run the following commands in your cluster to load the ATS docker image, '`ocats-nrf-26.1.201.tar`' and Stub docker image '`ocstub-py-26.1.201.tar`', and push it to your registry.

```
$ docker load -i ocats-nrf-26.1.201.tar
$ docker load -i ocstub-py-26.1.201.tar

$ docker tag ocats/ocats-nrf:26.1.201 <local_registry>/ocats/ocats-nrf:26.1.201

$ docker tag ocats/ocstub-py:26.1.201 <local_registry>/ocats/ocstub-py:26.1.201

$ docker push <local_registry>/ocats/ocats-nrf:26.1.201

$ docker push <local_registry>/ocats/ocstub-py:26.1.201
```

3. Create a copy of the custom values located at **Scripts/ocats_ocnrf_custom_values_26.1.201.yaml** and update it for image name, tag and other parameters as per the requirement.

3.1.4 Configuring ATS

3.1.4.1 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"
```

3.1.4.2 Enabling Aspen Service Mesh

To enable service mesh for ATS:

Copyright © 2019, 2026, 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, 2026, 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.

3.1.4.3 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 value of the `deploymentMode` parameter is defined in the global section of `ocats_ocnrf_custom_values_26.1.201.yaml` file.

  ```
  deploymentMode: &deploymentMode "clusterPreferred"
  ```

Figure 3-1 Global Dual Stack Configuration

```
global:
  # Dual-stack / ipFamilies control for the NRF ATS Service.
  # Supported: "IPv4", "IPv6", "IPv4_IPv6", "IPv6_IPv4","ClusterPreferred""
  deploymentMode: &deploymentMode "ClusterPreferred"
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

The default value of this parameter is `ClusterPreferred`. 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
```

The same variable is referred in the ocats-nrf and ocstub-py sections.

Figure 3-2 ocats-nrf Dual Stack configuration

```
ocats-nrf:

  replicaCount: 1
  nrfReleaseName: "ocnrf"
  serviceMeshCheck: false
  #serviceAccountName: "ocats-custom-serviceaccount"

  image:
    repository: ocats/ocats-nrf
    tag: 25.2.200
    pullPolicy: Always

  service:
    customExtension:
      labels: {}
      annotations: {}
    # Dual-stack / ipFamilies control for the NRF ATS Service.
    # Supported: "IPv4", "IPv6", "IPv4_IPv6", "IPv6_IPv4","ClusterPreferred""
    deploymentMode: *deploymentMode
    type: LoadBalancer
```

Figure 3-3 nrf_ats Stub Dual Stack Configuration

```
 stub1:
   deploymentName: "stub1"
   deploymentMode: *deploymentMode
   service:
     name: "notify-stub-service"
     type: "LoadBalancer"
     ports:
       h2c: 8080
       http: 8081
       h2: 8443
```

3.1.4.4 Enabling Persistent Volume

ATS supports Persistent storage to retain ATS historical build execution data, test cases, and one-time environment variable configurations.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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-26.1.201-pvc"
```

For more details on Persistent Volume Storage, you can refer to Persistent Volume for 5G ATS.

3.1.4.5 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-26.1.201.yaml"** to create a service account. Add the following annotation in the **"ocats-nrf-custom-serviceaccount-26.1.201.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:

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
kubectl apply -f ocats-nrf-custom-serviceaccount-26.1.201.yaml
```

3. Update the service account name in the **ocats-ocnrf-custom-values-26.1.201.yaml** file as follows:

```
ocats-nrf:
  serviceAccountName: "ocats-custom-serviceaccount"
```

3.1.5 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.

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-26.1.201.tgz --namespace
<namespace_name> -f <values-yaml-file>
```

Example:

```
helm install ocats ocats-ocnrf-26.1.201.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.

3.1.6 Verifying ATS Deployment

Run the following command to verify ATS deployment.

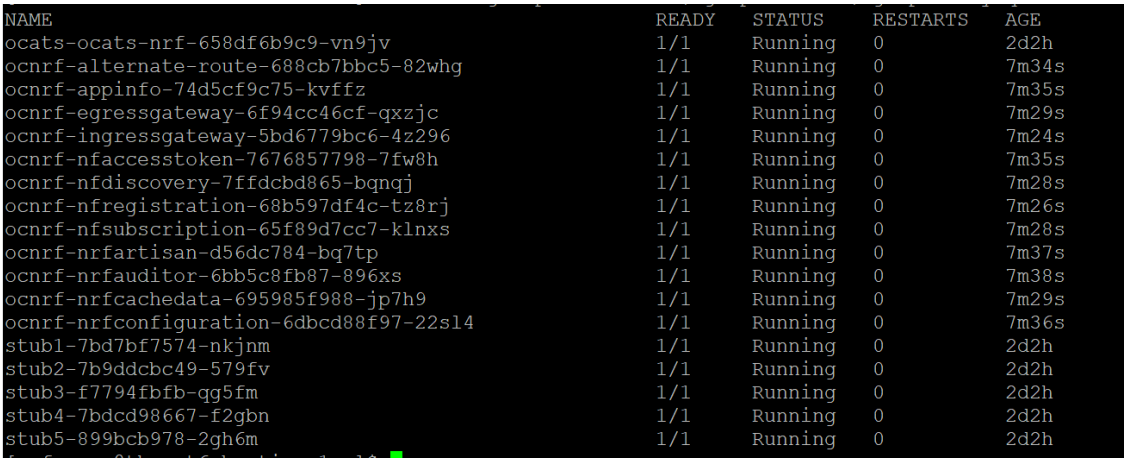
```
helm status <release_name>
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

Once ATS and Stub are deployed, run the following command to check the pod and service deployment.

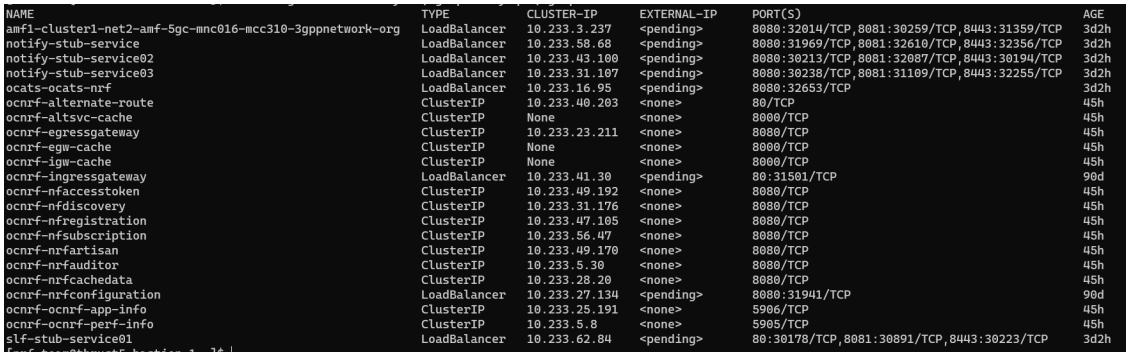
Checking Pod Deployment:
```
kubectl get pod -n ocnrf
```
Checking Service Deployment:
```
kubectl get service -n ocnrf
```

Figure 3-4 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 3-5 Checking Service Deployment without Service Mesh

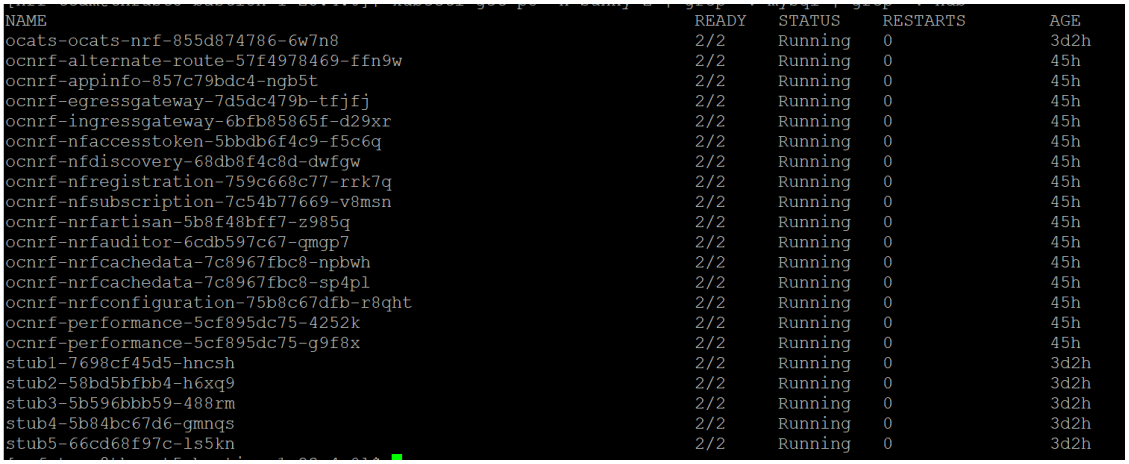


```
NAME                                                TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)                                        AGE
amf1-cluster1-net2-amf-5gc-mnc016-mcc310-3gppnetwork-org   LoadBalancer   10.233.3.237    <pending>     8080:32014/TCP,8081:30259/TCP,8443:31359/TCP   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:

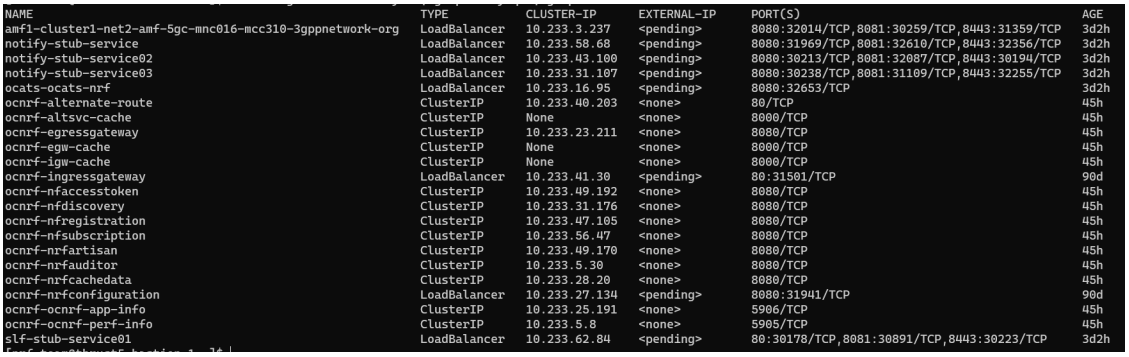
Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 3-6 ATS and Stub Deployed with Service Mesh



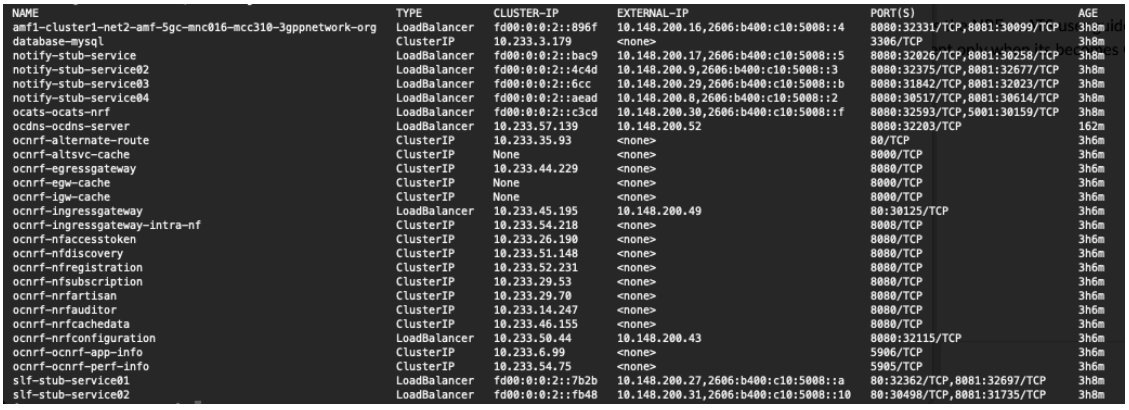
NAME	READY	STATUS	RESTARTS	AGE
ocats-ocats-nrf-855d874786-6w7n8	2/2	Running	0	3d2h
ocnrf-alternate-route-57f4978469-ffn9w	2/2	Running	0	45h
ocnrf-appinfo-857c79bdc4-ngb5t	2/2	Running	0	45h
ocnrf-egressgateway-7d5dc479b-tfjfj	2/2	Running	0	45h
ocnrf-ingressgateway-6bfb85865f-d29xr	2/2	Running	0	45h
ocnrf-nfaccesstoken-5bbdb6f4c9-f5c6q	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

Figure 3-7 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

Figure 3-8 ATS and Stub Deployed with Dual Stack



NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
amf1-cluster1-net2-amf-5gc-mnc016-mcc310-3gppnetwork-org	LoadBalancer	fd00:0:0:2::896f	10.148.200.16,2606:b400:c10:5008::4	8080:32331/TCP,8081:30099/TCP	3h8m
database-mysql	ClusterIP	10.233.3.179	<none>	3306/TCP	3h8m
notify-stub-service	LoadBalancer	fd00:0:0:2::bac9	10.148.200.17,2606:b400:c10:5008::5	8080:32026/TCP,8081:30258/TCP	3h8m
notify-stub-service02	LoadBalancer	fd00:0:0:2::4c4d	10.148.200.9,2606:b400:c10:5008::3	8080:32375/TCP,8081:32677/TCP	3h8m
notify-stub-service03	LoadBalancer	fd00:0:0:2::6cc	10.148.200.29,2606:b400:c10:5008::b	8080:31842/TCP,8081:32023/TCP	3h8m
notify-stub-service04	LoadBalancer	fd00:0:0:2::aead	10.148.200.8,2606:b400:c10:5008::2	8080:30517/TCP,8081:30614/TCP	3h8m
ocats-ocats-nrf	LoadBalancer	fd00:0:0:2::c3cd	10.148.200.30,2606:b400:c10:5008::f	8080:32593/TCP,5001:30159/TCP	3h8m
ocdns-ocdns-server	LoadBalancer	10.233.57.139	10.148.200.52	8080:32203/TCP	162m
ocnrf-alternate-route	ClusterIP	10.233.35.93	<none>	80/TCP	3h6m
ocnrf-altsvc-cache	ClusterIP	None	<none>	8000/TCP	3h6m
ocnrf-egressgateway	ClusterIP	10.233.44.229	<none>	8080/TCP	3h6m
ocnrf-egw-cache	ClusterIP	None	<none>	8000/TCP	3h6m
ocnrf-igw-cache	ClusterIP	None	<none>	8000/TCP	3h6m
ocnrf-ingressgateway	LoadBalancer	10.233.45.195	10.148.200.49	80:30125/TCP	3h6m
ocnrf-ingressgateway-intra-nf	ClusterIP	10.233.54.218	<none>	8008/TCP	3h6m
ocnrf-nfaccesstoken	ClusterIP	10.233.26.190	<none>	8080/TCP	3h6m
ocnrf-nfdiscovery	ClusterIP	10.233.51.148	<none>	8080/TCP	3h6m
ocnrf-nfregistration	ClusterIP	10.233.52.231	<none>	8080/TCP	3h6m
ocnrf-nfsubscription	ClusterIP	10.233.29.53	<none>	8080/TCP	3h6m
ocnrf-nrfartisan	ClusterIP	10.233.29.70	<none>	8080/TCP	3h6m
ocnrf-nrfauditor	ClusterIP	10.233.14.247	<none>	8080/TCP	3h6m
ocnrf-nrfcachedata	ClusterIP	10.233.46.155	<none>	8080/TCP	3h6m
ocnrf-nrfconfiguration	LoadBalancer	10.233.50.44	10.148.200.43	8080:32115/TCP	3h6m
ocnrf-ocnrf-app-info	ClusterIP	10.233.6.99	<none>	5906/TCP	3h6m
ocnrf-ocnrf-perf-info	ClusterIP	10.233.54.75	<none>	5905/TCP	3h6m
slf-stub-service01	LoadBalancer	fd00:0:0:2::7b2b	10.148.200.27,2606:b400:c10:5008::a	80:32362/TCP,8081:32697/TCP	3h8m
slf-stub-service02	LoadBalancer	fd00:0:0:2::fb48	10.148.200.31,2606:b400:c10:5008::10	80:30498/TCP,8081:31735/TCP	3h8m

3.1.7 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

3.2 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.

3.2.1 Resource Requirements

Total Number of Resources

The total number of resource requirements are as follows:

Table 3-4 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:

Table 3-5 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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-5 (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)
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
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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-5 (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-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**

3.2.2 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.
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*.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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=https
```

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=https
```

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
```

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-26.1.201-README.txt   - file
contains all the information required for the package.
├── ocats-nssf-custom-configtemplates-26.1.201.zip   - contains
serviceaccount,PVC File and Custom values file
├── ocats-nssf-tools-pkg-26.1.201-README.txt   - file contains all
the information required for the package.
└── ocats-nssf-tools-pkg-26.1.201.tgz   - file has the following
images and charts packaged as tar files
```

11. Untar the `ocats-nssf-tools-pkg-26.1.201.tgz` tar file

Copyright © 2019, 2026, Oracle and/or its affiliates.

The structure of the file looks as given below:

```
ocats-nssf-tools-pkg-26.1.201
├─── amfstub-26.1.201.tar      - AMF Stub Server Docker image
├─── amfstub-26.1.201.tar.sha256
├─── ats_data-26.1.201.tar     - ATS data, After untar "ocnssf_tests"
folder will gets created in which ATS feature files present
├─── ats_data-26.1.201.tar.sha256
├─── ocats-nssf-26.1.201.tar   - NSSF ATS Docker Image
├─── ocats-nssf-26.1.201.tar.sha256
├─── ocats-nssf-26.1.201.tgz   - ATS Helm Charts, after untar "ocats-
nssf" ats charts folder gets created.
├─── ocats-nssf-26.1.201.tgz.sha256
├─── ocdns-bind-26.1.201.tar.  - NSSF DNS Stub Server Docker Image
├─── ocdns-bind-26.1.201.tar.sha256
└─── README.md
```

12. Copy the `ocats-nssf-tools-pkg-26.1.201.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-26.1.201.zip` at the same location.
    The readme file `ocats-nssf-custom-configtemplates-26.1.201-README.txt` contains information about the content of this zip file.

    Content of `ocats-nssf-custom-configtemplates-26.1.201.zip` is as below:

```
 Archive:  ocats-nssf-custom-configtemplates-26.1.201.zip
  inflating: nssf_ats_pvc_26.1.201.yaml
  inflating: ocats_nssf_custom_values_26.1.201.yaml
  inflating: ocats_ocnssf_custom_serviceaccount_26.1.201.yaml
```

    Copy these files to CNE or Kubernetes cluster where you want to deploy ATS.

```
ocats-nssf-tools-pkg-26.1.201
├─── amfstub-26.1.201.tar       - AMF Stub Server Docker image
├─── ats_data-26.1.201.tar      - ATS data, After untar "ocnssf_tests"
folder will gets created in which ATS feature files present
├─── ocats-nssf-26.1.201.tar    - NSSF ATS Docker Image
├─── ocats-nssf-26.1.201.tgz    - ATS Helm Charts, after untar "ocats-
nssf" ats charts folder gets created.
├─── ocdns-bind-26.1.201.tar.   - NSSF DNS Stub Server Docker Image
```

3.2.3 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-26.1.201.tgz`

Copyright © 2019, 2026, Oracle and/or its affiliates.

The output of this command will return the following files:

```
ocats-nssf-tools-pkg-26.1.201
├─── amfstub-26.1.201.tar      - AMF Stub Server Docker image
├─── amfstub-26.1.201.tar.sha256
├─── ats_data-26.1.201.tar     - ATS data, After untar "ocnssf_tests"
folder will gets created in which ATS feature files present
├─── ats_data-26.1.201.tar.sha256
├─── ocats-nssf-26.1.201.tar   - NSSF ATS Docker Image
├─── ocats-nssf-26.1.201.tar.sha256
├─── ocats-nssf-26.1.201.tgz   - ATS Helm Charts, after untar "ocats-
nssf" ats charts folder gets created.
├─── ocats-nssf-26.1.201.tgz.sha256
├─── ocdns-bind-26.1.201.tar.  - NSSF DNS Stub Server Docker Image
├─── ocdns-bind-26.1.201.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
   ```

   Examples:

   ```
   docker load -i ocats-nssf-26.1.201.tar
   ```

   ```
   docker load -i amfstub-26.1.201.tar
   ```

   ```
   docker load -i ocdns-bind-26.1.201.tar
   ```

   Podman Commands:

   ```
   podman load -i ocats-nssf-<version>.tar
   ```

   ```
   podman load -i amfstub-<version>.tar
   ```

   ```
   podman load -i ocdns-bind-<version>.tar
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

Examples:

```
podman load -i ocats-nssf-26.1.201.tar


podman load -i amfstub-26.1.201.tar


podman load -i ocdns-bind-26.1.201.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>>
      ```

      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>>
      ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

5. **ATS Helm Charts** : Run the following command to get "ATS" Helm charts as shown below:

   ```
   tar -xvzf ocats-nssf-26.1.201.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-26.1.201.tar
   ```

   The above command creates "ocnssf_tests" and "jobs" folders, 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/
      ```

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_26.1.201.yaml` file as required.

   a. For this, open the `ocats_nssf_custom_values_26.1.201.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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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. **NSSF ATS PVC Creation — Ordered List**
    You can create the NSSF ATS PVC using either of the two available methods:
    a. Apply the NSSF ATS PVC YAML file manually after updating the namespace.
    b. Enable PVC creation directly from the NSSF ATS CV file.
    a. ***To create the PVC manually:***
        i. Update the namespace in the `nssf_ats_pvc_26.1.201.yaml` file using:

        ```
        sed -i "s/ocnssf/${namespace}/g" nssf_ats_pvc_26.1.201.yaml
        ```

        ii. Apply the updated PVC YAML using:

        ```
        kubectl apply -f <nssfatspvc.yaml file> -n <namespace_name>
        ```

        Example:

        ```
        kubectl apply -f nssf_ats_pvc_26.1.201.yaml
        ```

    b. ***To enable PVC creation in the NSSF ATS CV file***:
        i. Update the following parameters in the CV file:

        ```
          #The below section is related to the deployment of ATS with
        persistant volume
          #To enable PV set PVEnabled to true
          #To use an already existing pvc in the namespace, set the flag
        useExistingPVC to true, the name of it will be PVClaimName value.
          #retainPVC is to not delete the pvc even after un-installation of
        ATS. Make this true for this behavior, or leave it false for
        default behavior
          PVEnabled: false
          PVClaimName: "ocats-nssf-pvc"
          PVStorageClassName: "standard"
          PVStorage: "1Gi"
          RetainPVC: false
        ```

        When `PVEnabled` is set to `true`, a PVC is automatically created in the namespace using the name specified in `PVClaimName` (for example, `ocats-nssf-pvc`).

Copyright © 2019, 2026, Oracle and/or its affiliates.

- PVC behavior after ATS uninstallation depends on the value of RetainPVC:
  - If `RetainPVC` is `false`, the PVC will be deleted when ATS is uninstalled.
  - If `RetainPVC` is `true`, the PVC will be retained even after ATS is uninstalled.

For example: `PVEnabled: true` and `RetainPVC: false`

- **Example 1**:

```
PVEnabled: true
PVClaimName: "<pvc_name>"
PVStorageClassName: "<storage_class>"
PVStorage: "<storage_size>"
RetainPVC: false
```

Before installation, no PVC exists:

```
kubectl get pvc -n <namespace>
NAME   STATUS   VOLUME   CAPACITY   ACCESS MODES
STORAGECLASS   AGE
```

Install ATS:

```
helm install <release_name> <chart_name> -n <namespace> -f
<ats_cv_file>
```

Check PVC after installation:

```
kubectl get pvc -n <namespace>
NAME             STATUS
VOLUME                                     CAPACITY   ACCESS
MODES   STORAGECLASS   AGE
<pvc_name>       Bound
<pvc_volume_id>                            1Gi
RWO            <storage_class>   <age>
```

Since `RetainPVC` is set to `false`, the PVC will be deleted after the uninstallation of ATS.

```
helm uninstall <release_name> -n <namespace>
release "<release_name>" uninstalled
```

After uninstallation, check the PVC status (PVC is deleted because `RetainPVC: false`):

```
kubectl get pvc -n <namespace>
NAME              STATUS        VOLUME              CAPACITY
ACCESS MODES   STORAGECLASS   AGE
<pvc_name>        Terminating   <pvc_volume_id>     1Gi
RWO            <storage_class>   <age>
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

- **Example 2**: `PVEnabled: true` and `RetainPVC: true`

  ```
  PVEnabled: true
  PVClaimName: "ocats-nssf-pvc"
  PVStorageClassName: "standard"
  PVStorage: "1Gi"
  RetainPVC: true
  ```

  Before installation, no PVC exists:

  ```
  kubectl get pvc -n <namespace>
  NAME   STATUS   VOLUME   CAPACITY   ACCESS MODES
  STORAGECLASS   AGE
  ```

  Install ATS:

  ```
  helm install <release_name> <chart_name> -n <namespace> -f
  <ats_cv_file>
  ```

  After installation, the PVC is created (because `PVEnabled: true`):

  ```
  kubectl get pvc -n <namespace>
  NAME                STATUS   VOLUME              CAPACITY
  ACCESS MODES   STORAGECLASS   AGE
  <pvc_name>          Bound    <pvc_volume_id>     1Gi
  RWO            <storage_class>   <age>
  ```

  Since `RetainPVC` is set to `true`, the PVC will be retained after the uninstallation of ATS.

  ```
  helm uninstall <release_name> -n <namespace>
  ```

  If `RetainPVC: true`, the PVC is not deleted and will be retained. The CLI output will indicate that the PVC is kept due to the resource policy. Check the PVC status after uninstallation:

  ```
  kubectl get pvc -n <namespace>
  NAME                STATUS   VOLUME              CAPACITY
  ACCESS MODES   STORAGECLASS   AGE
  <pvc_name>          Bound    <pvc_volume_id>     1Gi
  RWO            <storage_class>   <age>
  ```

13. **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_26.1.201.yaml
    ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

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
# 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
```

14. 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_26.1.201.yaml --version 26.1.201
    ```

    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
    ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
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
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
```

15. 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: <namespace>
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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    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"

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-
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
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
      enableF5=true
      renewalTimeBeforeExpiry=3600
      validityTime=30
      enableSubscriptionAutoRenewal=true
      nfHeartbeatRate=80
      acceptAdditionalAttributes=false
      retryForCongestion=5
      enableVirtualNrfResolution=true
      virtualNrfFqdn=nrfstub.changeme-ocats.svc
      virtualNrfScheme=https
```

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>
      ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

For example:

```
$ kubectl get pods -n ocnssf
```

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>
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

For example:

```
$ kubectl edit deployment ocnssf-alternate-route -n ocnssf


terminationMessagePath: /dev/termination-log
  terminationMessagePolicy: File
dnsPolicy: ClusterFirst
restartPolicy: Always
```

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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

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"
```

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]$
```

Prometheus TLS Warning in ATS

To avoid the following warning in ATS logs:

```
Request returned with status code 503

upstream connect error or disconnect/reset before headers.
reset reason: connection failure,
transport failure reason: TLS error:
268435703:SSL routines:OPENSSL_internal:WRONG_VERSION_NUMBER
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

Configure the Istio `DestinationRule` for the Prometheus service to disable TLS communication:

```
apiVersion: networking.istio.io/v1beta1
kind: DestinationRule
metadata:
  name: occne-prometheus-server-dr
  namespace: <nssf-ats-namespace>
spec:
  host: occne-prometheus-server.occne-infra.svc.cluster.local
  trafficPolicy:
    tls:
      mode: DISABLE
```

3.3 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.

3.3.1 Prerequisites

To run SCP test cases, the following prerequisites are required.

3.3.1.1 Software Requirements

This section lists the software that must be installed before installing ATS.

Table 3-6 Preinstalled Software

Software	Version
Kubernetes	1.35.2, 1.34.1
Helm	3.20.0
Podman	5.6.0

To check the versions of the preinstalled software in the cloud native environment, run the following commands:

```
kubectl version
```

```
helm version
```

```
podman version
```

3.3.1.2 Environment Setup Requirements

This section describes the requirements for the client machine, that is, the machine used by the user to run deployment commands.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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.
- 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.

3.3.1.3 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 3-7 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	91	97	10
Grand Total SCP ATS	192	203	30

SCP Pods Resource Requirements

This section describes the resources required to deploy SCP ATS.

Table 3-8 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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-8 (Cont.) 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-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

Table 3-9 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-10 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	-	1	6	12	10 (if debug applog collection is enabled, PVC needs to be 50GB)
ATS pystub	1	1	-	82	82	82	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					**91**	**97**	**4**

Table 3-11 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

3.3.1.4 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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_26_1_201_0_0.pkg` file to download the CNC SCP ATS package file.
   The `ocats_ocscp_csar_26_1_201_0_0.pkg` package contains the following files:

   ```
   ocats_ocscp_csar_26_1_201_0_0.zip
   mcafee-gen-ats-csar-26.1.201.log
   ```

   > **Note**
   >
   > The above zip file contains all the images and custom values required for 26.1.201 release of OCATS-OCSCP.

   Unzip the `ocats_ocscp_csar_26_1_201_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-26.1.201.tgz
   |  |-- Licenses
   |  |-- ocats-ddclientstub-26.1.201.tar
   |  |-- ocats-dnsstub-26.1.201.tar (Docker Image)
   |  |-- ocats-pystub-26.1.201.tar (Docker Image)
   |  |-- ocats-scp-26.1.201.tar (Docker Image)
   |  |-- ocats-scpglbratelimitstub-26.1.201.tar (Docker Image)
   |  |-- Oracle.cert
   |  `-- Tests
   |-- ocats_ocscp_csar_26_1_201_0_0.zip
   |-- ocats_ocscp.mf
   |-- Scripts
   |  |-- ocats_ocscp_custom_serviceaccount_26.1.201.yaml (Template to create
   custom service account)
   |  |-- ocats_ocscp_tests_jenkinsjobs_26.1.201.tgz. ( ocscp_tests and
   jenkins jobs folder to be copied if persistent volume is deployed)
   |  `-- ocats_ocscp_values_26.1.201.yaml (Custom values file for
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
installation)
`-- TOSCA-Metadata
  `-- TOSCA.meta
```

10. Copy the umbrella Helm chart `ocats-ocscp-26.1.201.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_26.1.201.yaml` file:

Table 3-12 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.)
`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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-12 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
`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","8443","5001"	-	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`	3	Value in milliseconds	-
`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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-12 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description

`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`	ocats-ocscp-pvc	Name	Persistent volume claim name.
`PVStorageClassName`	standard	-	Persistent volume storage class name.
`PVStorage`	4Gi	Allocated storage value	Persistent volume storage.
`retainPVC`	false	true, false	Enables or disables the PVC deletion after the uninstallation of ATS.
`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.
`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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-12 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
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.
SELECTED_NF	SCP	NF name	ATS parameters are set with default values in `ocats_ocscp_values_26.1.201.yaml` file. You must update the ATS parameters with the actual value based on the deployment environment, and then deploy the OCATS_OCSCP chart using this `ocats_ocscp_values_26.1.201.yaml` file. The updated ATS parameters are automatically applied during the deployment process, and ATS is installed 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 the SCP Namespace	
PrometheusAuthEnabled	false	false, true	Set this parameter to true when Prometheus requests require a token in the header for authentication. For more information about token-based authentication for Prometheus, see Table 2-5.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-12 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
`PrometheusTLSEnabled`	false	false, true	Set this parameter to true when Prometheus has HTTPS deployment. **Note:** When the connection is established using HTTPS, it remains insecure because no certificates are maintained with ATS for this connection.
`PrometheusScrapeInterval`	60 seconds (commented out)	According to setup Prometheus scrape interval	Uncomment this parameter only if this value is other value than 60 seconds. ATS assumes 60s as the default value if no response comes from the Prometheus config.
`prometheusClusterRole.rules`	`[]`	Default rules when rules are empty in the values.yaml files. - apiGroups: [""] resources: ["namespaces"] verbs: ["get"] - apiGroups: ["monitoring.coreos.com"] resources: ["prometheuses/api"] resourceNames: ["k8s"] verbs: ["get", "create", "update"] - nonResourceURLs: - /* verbs: - get	If PrometheusAuthEnabled is true, override the ClusterRole rules.
`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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-12 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
`prometheusClusterRole.rules`	`[]`	Default rules when rules are empty in cv files. • apiGroups: [""] resources: ["namespaces"] verbs: ["get"] • apiGroups: ["monitoring.coreos.com"] resources: ["prometheuses/api"] resourceNames: ["k8s"] verbs: ["get", "create", "update"] • nonResourceURLs: – /* verbs: – get	Optionally override ClusterRole rules used when PrometheusAuthEnabled is set to true.
`ocats-dnsstub.image.repository`	<docker-registry IP:docker-registry port>/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 ocats-dnsstub service. This value is referred from whatever is defined at serviceIpFamilyPolicy.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, 2026, Oracle and/or its affiliates.

Table 3-12 (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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-12 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
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.
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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-12 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
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	-
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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-12 (Cont.) ATS Parameters of the YAML File

Parameter	Default Value	Possible Values	Description
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.

3.3.1.5 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_26_1_201_0_0.pkg
   ```

   The package contains the following files:

   ```
   ocats_ocscp_csar_26_1_201_0_0.zip
   mcafee-gen-ats-csar-26.1.201.log
   ```

   Unzip the `ocats_ocscp_csar_26_1_201_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-26.1.201.tgz
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
|  |-- Licenses
|  |-- ocats-ddclientstub-26.1.201.tar
|  |-- ocats-dnsstub-26.1.201.tar (Docker Image)
|  |-- ocats-pystub-26.1.201.tar (Docker Image)
|  |-- ocats-scp-26.1.201.tar (Docker Image)
|  |-- ocats-scpglbratelimitstub-26.1.201.tar (Docker Image)
|  |-- Oracle.cert
|  `-- Tests
|-- ocats_ocscp_csar_26_1_201_0_0.zip
|-- ocats_ocscp.mf
|-- Scripts
|  |-- ocats_ocscp_custom_serviceaccount_26.1.201.yaml (Template to create
custom service account)
|  |-- ocats_ocscp_tests_jenkinsjobs_26.1.201.tgz. ( ocscp_tests and
jenkins jobs folder to be copied if persistent volume is deployed)
|  `-- ocats_ocscp_values_26.1.201.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-26.1.201.tar`, and push it to your registry:

   ```
   docker load --input ocats-scp-26.1.201.tar
   docker load --input ocats-dnsstub-26.1.201.tar
   docker load --input ocats-pystub-26.1.201.tar
   docker load --input ocats-scpglbratelimitstub.tar
   docker load --input ocats-ddclientstub-26.1.201.tar
   ```

3. Run the following command in your cluster to load the ATS image:

   ```
   docker load --input ocats-scp-images-26.1.201.tar
   ```

4. Run the following commands to push the ATS image to the registry:

   ```
   docker tag ocats/ocats-scp:26.1.201 <local_registry>/ocats/
   ocatsscp:26.1.201
   docker push <local_registry>/ocats/ocats-scp:26.1.201
   ```

   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:26.1.201 <local_registry>/ocats/ocats-
   pystub:26.1.201
   docker push <local_registry>/ocats/ocats-pystub:26.1.201
   ```

6. Run the following command to push the DNS Stub Image to the registry:

   ```
   docker tag ocats/ocats-dnsstub:26.1.201 <local_registry>/ocats/ocats-
   dnsstub:26.1.201
   docker push <local_registry>/ocats/ocats-dnsstub:26.1.201
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

7. Run the following command to push the Global Rate Limiting Stub Image to the registry:

```
docker tag ocats/ocats-scpglbratelimitstub:26.1.201 <local_registry>/ocats/
ocats-scpglbratelimitstub:26.1.201
docker push <local_registry>/ocats/ocats-scpglbratelimitstub:26.1.201
```

8. Run the following command to push the Data Director Stub Image to the registry:

```
docker tag ocats/ocats-ddclientstub:26.1.201 <local_registry>/ocats/ocats-
ddclientstub:26.1.201
docker push <local_registry>/ocats/ocats-ddclientstub:26.1.201
```

9. In the **Scripts** folder, extract the following content:

```
ocats-ocscp-values-26.1.201.yaml
ocats-ocscp-custom-serviceaccount-26.1.201.yaml
ocats-ocscp-tests-jenkinsjobs-26.1.201.tgz
```

10. Update the image name and tag in the `ocats-ocscp-values-26.1.201.yaml` file as required.

3.3.1.6 Preinstall Preparation of SCP for SCP-ATS

Complete the following steps before performing an installation:

- 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_26.1.201.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:
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    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
    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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
      #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
      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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
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 run SCP ATS under the `scp values.yaml`, ensure that `dnsSrvSchemeConfig.defaultScheme` is set to 'https'.
- To run ATS, the SCP deployment file should have the `sNssais` parameter under `scpProfileInfo`.
- 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.
  - Ensure the `preventiveAuditOnLastNFInstanceDeletion` parameter is set to `false` in the `ocscp_values.yaml` file.
  - Deploy SCP with max replica count as 1 unless specified.
  - 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- 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.
- 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

3.3.1.7 Preinstallation Preparation for SCP-ATS

Complete the following steps before performing an installation of SCP-ATS.

3.3.1.7.1 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
   ```

   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_26.1.201.yaml` file:

   ```
   traffic.sidecar.istio.io/excludeOutboundPorts: "8091"
   traffic.sidecar.istio.io/excludeInboundPorts: "8080,8443,5001"
   ```

   Sample file with annotations:

   ```
   lbDeployments:
   labels: {}
   annotations:
   traffic.sidecar.istio.io/excludeOutboundPorts: "8091"
   traffic.sidecar.istio.io/excludeInboundPorts: "8080,8443,5001"
   ```

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
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    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:
  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_manager.v2.HttpConnectionManage
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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    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
```

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
```

3.3.1.7.2 Enabling Persistent Volume

ATS supports Persistent storage to retain ATS historical build performance data, test cases and one-time environment variable configurations. With this enhancement, you can decide whether to use persistent volume based on their resource requirements. By default, the persistent volume feature is disabled.

> **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 procedure:

1. PVC creation is part of Helm charts. To create the PVC, configure the following set of parameters in `ocats_ocscp_values_26.1.201.yaml` file.
   `PVEnabled`: Indicates whether to enable PVC or not.

Copyright © 2019, 2026, Oracle and/or its affiliates.

`PVClaimName`: Indicates the name of persistent volume.

`PVStorageClassName`: Indicates the storage class name.

`PVStorage`: Indicates the size of the persistent volume.

`retainPVC`: Iindicates whether to delete or retain created PVC during uninstallation of ATS.

A sample edited `26.1.201.yaml` file for enabling PVC:

```
PVEnabled: true
PVClaimName: "ocats-scp-26.1.201-pvc"

PVStorageClassName: "standard"
PVStorage:"4Gi"
retainPVC: false
```

> **Note**
>
> - The value of `PVClaimName` 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. Run the following command to verify if the created PVC is bound to the Persistent Volume and available for use:

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

```
kubectl get pvc -n ocats-scp
```

Sample Output:

```
kubectl get pvc -n ocats-scp
NAME                  STATUS   VOLUME
CAPACITY       ACCESS MODES   STORAGECLASS   AGE
ocats-scp-26.1.201-pvc   Bound    pvc-65484045-3805-4064-9fc3-
f9eeeaccc8b8   1Gi        RWO            standard       11s
```

> **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.
> - Ensure 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

a. Extract the tar file.

```
tar -xvf ocats-scp-26.1.201.tgz
```

b. Run the following commands to copy the desired folder.

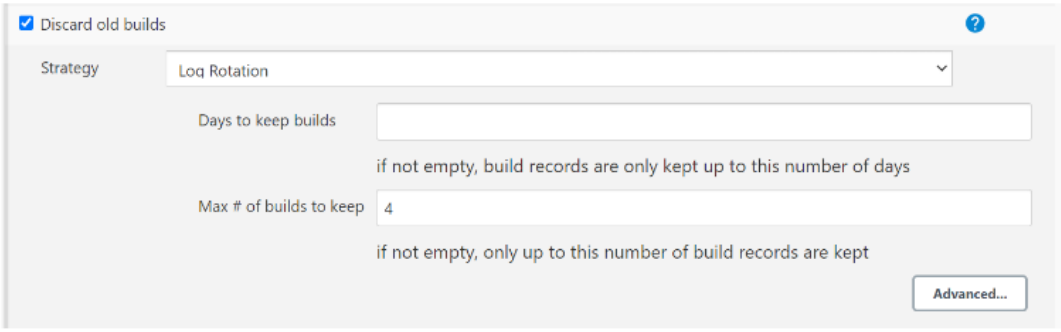
```
kubectl cp ocats-scp-26.1.201/ocscp_tests <namespace>/<pod-
name>:/var/lib/jenkins/


 kubectl cp ocats-scp-26.1.201/jobs <namespace>/<pod-name>:/var/lib/
jenkins/.jenkins/
```

c. Restart the pod.

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

4. When 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 information about Persistent Volume Storage, see Persistent Volume for 5G ATS.

3.3.2 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **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*.

3.3.2.1 Updating ATS Configuration

The following section covers updating the ATS deployment configuration and ATS input parameter configuration.

3.3.2.1.1 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_26.1.201.yaml` file, you can update the values for ATS parameters and then deploy the OCATS_OCSCP charts using this modified `ocats_ocscp_values_26.1.201.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_26.1.201.yaml` file.

The `ocats_ocscp_values_26.1.201.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_26.1.201.yaml` file:

Figure 3-10 ATS Deployment Configuration

```
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"
  LOG_SOURCE: kubernetes
  SCPSVCHTTPSPORT: 9443
  SCPNRFPROXYSVCNAME: ocscp-scp-nrfproxy
  KUBERNETES_HOST: kubernetes.default
  KAFKANAMESPACE: scpsvc
  SCPDDCLIENTSVCNAME: ocats-scpddclientstub
  SCPGLBRATELIMITSTUBSVCNAME: ocats-scpglbratelimitstub
  SCPCONFIGSVCPORT: 8081
  SCP_METRICS_VERSION: v1
  #Possible values of SCPSVCIP are IPv4 address or IPv6 address or scpSignallingFqdn
  SCPSVCIP: scpSignallingFqdn
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> Initially, at the time of deployment, you can configure or modify the parameter in the `ocats_ocscp_values_26.1.201.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.

3.3.2.1.2 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
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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
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_26.1.201.yaml` file. To update the ATS parameters in the `ocats_ocscp_values_26.1.201.yaml` file, see the Updating ATS Deployment Configuration section.

Modify the `scpMetricVersion` parameter to "v2" in the `ocats_ocscp_values_26.1.201.yaml` file. For more information, see *Oracle Communications Cloud Native Core, Service Communication Proxy Installation, Upgrade, and Fault Recovery Guide*.

3.3.2.1.3 Updating ATS Input Parameter Configuration

This section provides information about how to modify different services in SCP and configure SCP test cases in ATS.

3.3.2.1.3.1 Enabling or Disabling Stubs

By default, all the stubs are enabled.

> **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_26.1.201.yaml` file:

```
global:
  # ********  Sub-Section Start: Custom Extension Global Parameters ********
  #************************************************************************
  ausf1Stubs: true
  ausf2Stubs: true
  ausf3Stubs: true
  ausf4Stubs: true
  ausf5Stubs: true
  ausf6Stubs: true
  ausf11Stubs: true
  ausf12Stubs: true
  ausf13Stubs: true
  ausf14Stubs: true
  ausf15Stubs: true
  ausf16Stubs: true
  ausf21Stubs: true
  ausf22Stubs: true
  ausf23Stubs: true
  chf1Stubs: true
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
chf2Stubs: true
nrf1Stubs: true
nrf2Stubs: true
nrfr2l1Stubs: true
nrfr2l2Stubs: true
nrfr3l1Stubs: true
nrfr3l2Stubs: true
pcf1Stubs: true
pcf2Stubs: true
pcf3Stubs: true
pcf4Stubs: true
pcf5Stubs: true
pcf6Stubs: true
pcf7Stubs: true
pcf8Stubs: true
pcf11Stubs: true
pcf12Stubs: true
pcf13Stubs: true
pcf14Stubs: true
pcf15Stubs: true
pcf16Stubs: true
pcf21Stubs: true
pcf22Stubs: true
pcf23Stubs: true
pcf24Stubs: true
pcf25Stubs: true
pcf26Stubs: true
pcf27Stubs: 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
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
sepp1Stubs: true
sepp2Stubs: true
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
sepp3Stubs: true
nef1Stubs: true
nef2Stubs: true
nef3Stubs: true
nef4Stubs: true
nef5Stubs: true
gen1Stubs: true
gen2Stubs: true
custom5gnif1Stubs: true
custom5gnif2Stubs: true
```

> **Note**
>
> Replica count of 'scp51svcxxxx' 'scp52svcxxxx' 'scp61svcxxxx' stubs must be set to Zero.

3.3.2.1.3.1.1 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.

3.3.2.1.3.2 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_26.1.201.yaml` file: **ocats-scp**

```
ocats-scp:
  image:
    repository: cgbu-ocscp-dev-docker.dockerhub-phx.oci.oraclecorp.com/
ocats/ocats-scp
    tag: 26.1.201
    pullPolicy: Always
  replicaCount: 1
  resources:
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    limits:
      cpu: 3
      memory:23Gi
      #ephemeral-storage: 4Gi
    requests:
      cpu: 3
      memory: 3Gi
```

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:
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
      name: ausf1svc
      type: ClusterIP
      ports:
        port: 8080
    deploymentName: ausf1
  ausf2:
    service:
      name: ausf2svc
      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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
      #ephemeral-storage: 55Mi
  log:
    level: INFO
  extraContainers: USE_GLOBAL_VALUE

  kafka_broker: "kafka-broker1-0.kafka-
broker1.ddkafkanamespace.svc.cluster.local:9092"
  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
```

3.3.2.1.3.3 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)

Copyright © 2019, 2026, Oracle and/or its affiliates.

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:

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.

3.3.2.1.3.4 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}
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
DNS.2 = localhost
EOF
```

b. Make changes to the SERVER_EXT. Set the IP.2 value as the local setup IP as mentioned below:

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_26.1.201.yaml`file, replace the value of `atsCommunicationTLSEnabled` with `true`.

3.3.2.1.3.5 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.

3.3.2.1.3.6 Configuring ATS for CCA Test Cases

Run CCA Validation feature tests with ATS.

- Enable HTTPs to run these feature tests.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- 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:
  - **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]
    ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
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
```

- **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}
```


Copyright © 2019, 2026, 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://[2001:db8:85a3:0:0:8a2e:370:7334]: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 = 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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
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}
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
```

3.3.2.1.3.7 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`].

3.3.2.1.3.8 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, 2026, Oracle and/or its affiliates.

3.3.2.1.3.9 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.

3.3.2.1.3.10 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.

3.3.2.1.3.11 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, 2026, 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.

3.3.2.1.3.12 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.

3.3.2.1.3.13 Configuring SCP to Run Roaming Proxy Test Cases in ATS

To run ATS for the Roaming Proxy feature, perform the following configuration:

```
sNssais:
- sd: "012AFC"
  sst: 27
```

3.3.3 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_26.1.201.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-26.1.201.tgz --name <release_name> --namespace
   <namespace_name> -f ocats-ocscp-values-26.1.201.yaml
   ```

   Example:

   ```
   helm install ocats-ocscp-26.1.201.tgz --name ocats-ocscp --namespace scpsvc-f
   ocats-ocscp-values-26.1.201.yaml
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> Update image name, tag, service name, and deployment name in ocats-pystub of the `ocats_pystub_values_26.1.201.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_26.1.201.yaml`.

3.3.4 Post Installation and Deployment Steps

This section describes the post-installation steps for SCP.

3.3.4.1 Verifying ATS Deployment

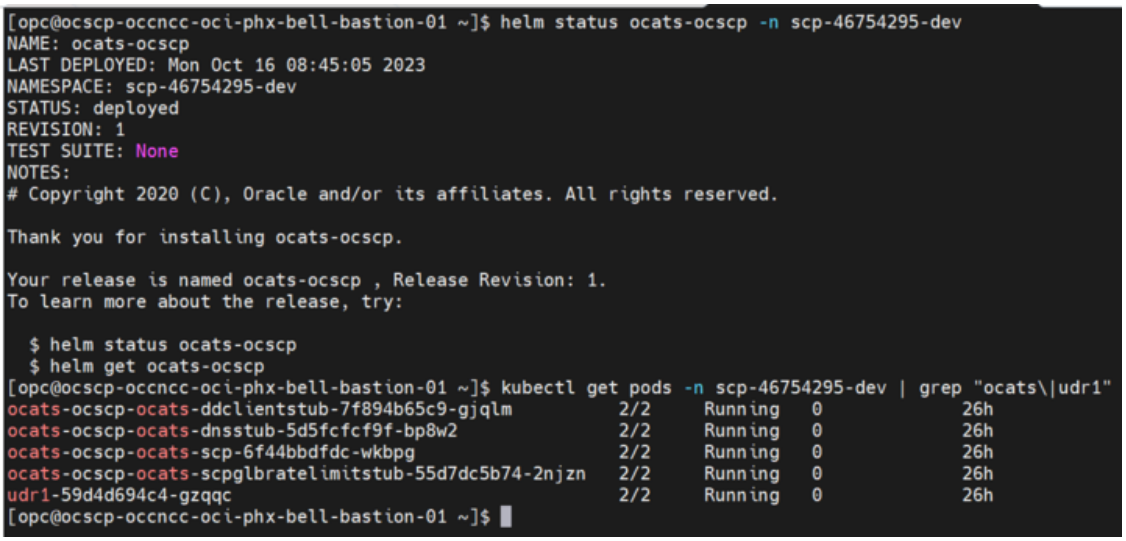
Run the following command to verify the ATS deployment status:

```
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 3-11 Checking ATS Helm Release and ATS related Pod Status



3.3.4.2 Modifying the scpc-alternate-resolution Microservice

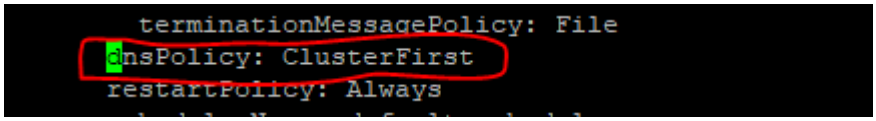
Perform the following procedure to modify the scpc-alternate-resolution microservice to point to the DNS Stub for the Alternate Routing based on the Domain Name System (DNS) Service Record (SRV) Records feature.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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.

   Figure 3-12 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.

3.3.4.3 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

Copyright © 2019, 2026, Oracle and/or its affiliates.

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.

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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    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
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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    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
    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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
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

    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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
proportionate as the traffice sent
          sc_http_requests_total_udm2: 100     #To be configured in the same
proportionate as the traffice sent

  "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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
messages with 3gpp-Sbi-Target-apiRoot missing and 3gpp-sbi-routing-binding
Header with bl=nfset is present":
      Input:
        File_Parameters:
          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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
  "Scenario_Scenario-14- Forward route initial 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

  "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:
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
        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
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
```

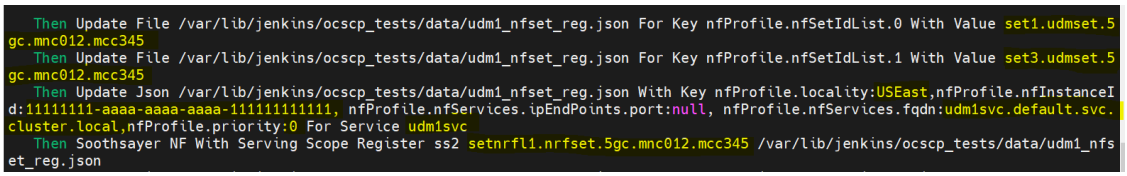

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
          sc_http_requests_total_pcf2: 100     #To be configured in the same
proportionate as the traffice sent
```

Updates in Feature Files

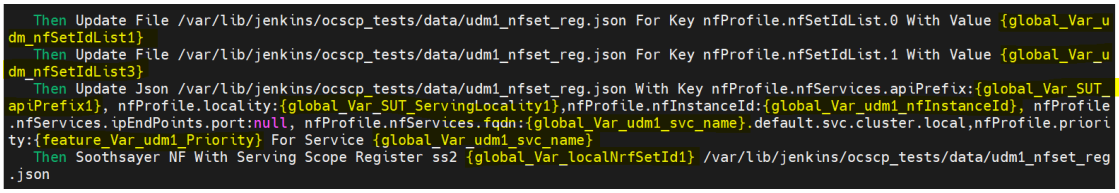
The variables are defined in either global.yaml or <feature_name>.yaml is used in the feature files by enclosing the variable name in curly brackets. The steps in a feature file without parameterization are as follows:

Figure 3-13 Feature File without Parameterization



The steps in a feature file when it is parameterized:

Figure 3-14 Feature File when it is Parameterized



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.

3.3.4.4 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:

- @asmonly: Any feature file with this tag should run only on an ASM setup.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- @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.
- @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

3.3.4.5 Accessing Collected Logs

Perform the following procedure to access the collected logs:

1. Access the ATS pod.
2. Navigate to the Jenkins job directory that applies to the following feature type:
   - For regression features, go to `/var/lib/jenkins/.jenkins/jobs/SCP-Regression/builds`.
   - For new features, go to `/var/lib/jenkins/.jenkins/jobs/SCP-NewFeatures/builds`.
3. Select the required build number, then open its directory.
4. Locate the log archive. For "AppLog", the archive is named `applogs.zip`.
5. Extract the archive and open the extracted directory.
6. Review the logs for failed scenarios. Log files use names such as `Initial_Run_<scenario_name>.log`, `1stRerun_<scenario_name>.log`, and `2ndRerun_<scenario_name>.log`.
   The logs are organized as follows:
   - "Initial_Run" logs are the default logs for scenarios that failed during the initial run.
   - When the rerun count is greater than zero, numbered rerun logs, such as "1stRerun" and "2ndRerun", are also available. These logs correspond to the selected "Log_Level".

> **Note**
>
> When a scenario passes during a rerun, logs are available only for the failed runs before the passing rerun. For example, if a scenario fails during the initial run and the first rerun but passes during the second rerun, only the "Initial_Run" and "1stRerun" logs are available. No "2ndRerun" or subsequent rerun logs are generated, even if the configured rerun count is three.

Copyright © 2019, 2026, Oracle and/or its affiliates.

3.3.4.6 Enabling PCAP Logs

Perform the following procedure to enable Packet Capture (PCAP) logs for ATS functional test execution:

1. In `ocscp_values.yaml`, enable extra containers:

   ```
   extraContainers: ENABLED
   ```

2. Add the required debug-tools image/tag and update `runAsUser`, for example value, 2).
3. Deploy the updated configuration.
4. In the ATS screen:
   a. Set **Fetch Logs Upon Failure** to Yes.
   b. Enable PCAP Logs.
5. Run the Functional Test (FT).
6. Navigate to the Jenkins job directory:
   - For regression features: `/var/lib/jenkins/.jenkins/jobs/SCP-Regression/builds`.
   - For new features: `/var/lib/jenkins/.jenkins/jobs/SCP-NewFeatures/builds`.

3.3.4.7 Data Collection after Feature File Failure

After a feature file fails, dumps are collected at `/tmp/scale_logs`. For 50 scenarios, the observed folder size is approximately 57 MB. By default, `/tmp/scale_logs` is automatically cleaned up after 5 days (default LOG_RETENTION_DAYS). This retention period can be modified by updating the LOG_RETENTION_DAYS variable in the CNCATS `custom_values.yaml` file.

The following dump types are captured in `/tmp/scale_logs`:

- DB dump
- Metric dump
- Worker routing rules

3.3.5 Appendix

This section provides supplementary information that may be helpful for a more comprehensive understanding of installing and running SCP test cases in ATS.

3.3.5.1 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_26.1.201.yaml` file.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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"]
```

3.3.5.2 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
```

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_26.1.201.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 3-15 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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

3.4 Installing ATS for SEPP

This section describes Automated Testing Suite (ATS) installation procedures for Security Edge Protection Proxy (SEPP) in a cloud native environment.

3.4.1 Resource Requirements

Total Number of Resources

The resources required to install SEPP-ATS are as follows:

Table 3-13 Total Number of Resources

Resource	CPUs	Memory(GB)	Storage(GB)
SEPP SUT Total	15.1	29.128	0
cnDBTier Total	40	40	20
ATS Total	5	5	1
Grand Total SEPP ATS	60.1	74.128	21

Resource Details

The details of resources required to install SEPP-ATS are as follows:

Table 3-14 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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-14 (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)
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
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 Stub-4 (Python)	.5	.5	0	1	1	.5	.5	0
ATS Totals						**5**	**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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-14 (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-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
DB Tier Totals						40	40	20

3.4.1.1 SEPP ATS Compatibility Matrix

The following table lists the versions SEPP ATS and the comapability with the SEPP and ATS framework:

Table 3-15 SEPP ATS Compatibility Matrix

SEPP ATS Release	SEPP Release	ATS Framework version
26.1.200	26.1.200	26.1.201
25.2.200	25.2.200	25.2.204
25.2.100	25.2.100	25.2.100

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-15 (Cont.) SEPP ATS Compatibility Matrix

SEPP ATS Release	SEPP Release	ATS Framework version
25.1.201	25.1.200	25.1.202
25.1.200	25.1.200	25.1.202

3.4.2 Downloading the ATS Package

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.
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_26.1.201_0_0.zip
ocats_ocsepp_csar_26.1.201_0_0.zip.sha256
```

> **Note**
>
> The above zip file contains all the images and custom values required for 26.1.201 release of OCATS-OCSEPP.

11. The `ocats_ocsepp_csar_26.1.201_0_0.zip` file has following definitions:

```
├──── Definitions
│     ├──── ocats_ocsepp_cne_compatibility.yaml
│     └──── ocats_ocsepp.yaml
├──── Files
│     ├──── ChangeLog.txt
│     ├──── Helm
│     │     └──── ocats-sepp-26.1.201.tgz (Helm Charts)
│     ├──── Licenses
│     ├──── ocats-sepp-26.1.201.tar (BDD client image)
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
│    ├──── Oracle.cert
│    ├──── seppstub-26.1.201.tar (Stub server image)
│    └──── Tests
├──── Scripts/
│    ├──── ocats_ocsepp_tests_jenkinsjobs_26.1.201.tgz
│    ├────       ├────jobs (For Persistent Volume)
│    └────       ├────ocsepp_tests (For Persistent Volume)
│    └────  ocats_ocsepp_values_26.1.201.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

3.4.3 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_26_1_201_0_0.zip
   ```

   The following docker image tar files are located at **Files** folder:

   - ocats-sepp-26.1.201.tar
   - seppstub-26.1.201.tar

2. Run the following commands in your cluster to load the ATS docker image, '`ocats-sepp-26.1.201.tar`'' and Stub docker image `seppstub-26.1.201.tar`, and push it to your registry.

   ```
   $ docker load -i ocats-sepp-26.1.201.tar
   $ docker load -i seppstub-26.1.201.tar

   $ docker tag ocats/ocats-sepp:26.1.201 <local_registry>/ocats/ocats-
   sepp:26.1.201

   $ docker tag ocats/seppstub:26.1.201 <local_registry>/ocats/
   seppstub:26.1.201

   $ docker push <local_registry>/ocats/ocats-sepp:26.1.201

   $ docker push <local_registry>/ocats/seppstub:26.1.201
   ```

3. Run the following command to get the helm charts are located in Helm directory of files folder:

   ```
   tar -xvf ocats-sepp-26.1.201.tgz
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

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_26.1.201.yaml** and update it for image name, tag and other parameters as per the requirement.

3.4.4 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.

3.4.4.1 Configuring Root Certificate

The following are the steps to configure the root certificate (caroot.cer):

> **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 and also note down the {PEM_PHRASE} which was used to create cakey.pem file for further ats certificate creation steps.
> - 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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
commonName_max = 64
commonName_default = sepp1.inter.oracle.com
[ req_ext ]
subjectAltName = @alt_names

[ v3_ca ]
basicConstraints = critical,CA:TRUE,pathlen:0
keyUsage = critical,keyCertSign,cRLSign
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer

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

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

[ mtls_ext ]
basicConstraints = critical,CA:FALSE
keyUsage = critical,digitalSignature,keyEncipherment
extendedKeyUsage = serverAuth,clientAuth
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
export SEPP_FQDN="${DEPLOYMENT_NAMESPACE}.inter.oracle.com"
```

3. Run the following command to create the required files:

```
# Generate the Root CA:


openssl genrsa -out cakey.pem 4096
openssl req -new -key cakey.pem -out careq.pem \
  -passout pass:${PEM_PHRASE} \
  -subj "/C=IN/ST=Karnataka/L=Bangalore/O=Oracle/CN=sepp1.inter.oracle.com"


openssl x509 -signkey cakey.pem \
  -req \
  -days 3650 \
  -in careq.pem \
  -out caroot.cer \
  -sha256 \
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
  -extensions v3_ca \
  -extfile ssl.conf

# Generate RSA certificates:

openssl genrsa -out rsa_private_key 2048

openssl rsa -in rsa_private_key \
  -outform PEM \
  -out rsa_private_key_pkcs1.pem

openssl req -new \
  -key rsa_private_key \
  -out rsa_certificate.csr \
  -config ssl.conf \
  -subj "/C=IN/ST=Karnataka/L=Bangalore/O=Oracle/CN=${SEPP_FQDN}"

openssl x509 -req \
  -in rsa_certificate.csr \
  -CA caroot.cer \
  -CAkey cakey.pem \
  -CAcreateserial \
  -out rsa_certificate.crt \
  -days 365 \
  -sha256 \
  -extfile ssl.conf \
  -extensions mtls_ext

openssl req -new \
  -key rsa_private_key \
  -out ocsepp.csr \
  -config ssl.conf \
  -subj "/C=IN/ST=Karnataka/L=Bangalore/O=Oracle/CN=${SEPP_FQDN}"

openssl x509 -req \
  -in ocsepp.csr \
  -CA caroot.cer \
  -CAkey cakey.pem \
  -CAserial caroot.srl \
  -out ocsepp.cer \
  -days 365 \
  -sha256 \
  -extfile ssl.conf \
  -extensions mtls_ext

# Generate ECDSA certificates:
openssl ecparam -genkey -name prime256v1 -out host.key

openssl req -new \
  -sha256 \
  -key host.key \
  -nodes \
  -out ecdsa_certificate.csr \
  -config ssl.conf \
  -subj "/C=IN/ST=Karnataka/L=Bangalore/O=Oracle/CN=${SEPP_FQDN}"
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
openssl x509 -req \
  -in ecdsa_certificate.csr \
  -CA caroot.cer \
  -CAkey cakey.pem \
  -CAserial caroot.srl \
  -extfile ssl.conf \
  -extensions mtls_ext \
  -out ssl_ecdsa_certificate.crt \
  -days 730 \
  -sha256

openssl pkcs8 -topk8 \
  -in host.key \
  -out ssl_ecdsa_private_key.pem \
  -nocrypt


echo "${PEM_PHRASE}" > trust.txt
echo "${PEM_PHRASE}" > key.txt
echo 1234 > serial.txt
```

> **Note**
>
> Both ATS and SEPP must have same root certificate.

3.4.4.2 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.

   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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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 = ${ATS_HELM_RELEASE_NAME}-stubserver-4.${DEPLOYMENT_NAMESPACE}

DNS.6 = localhost
```

2. Run the following command to create a `certificate signing request` or csr:

```
[cloud-user@star23-bastion-1 ocats]$ openssl req -config ssl.conf -newkey
rsa:2048 -days 1000 -nodes -keyout rsa_private_key_pkcs1.key >
ssl_rsa_certificate.csr
```

Output:

```
Ignoring -days; not generating a certificate
Generating a RSA private key
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
...+++++
.........+++++
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]$ openssl rsa -in
rsa_private_key_pkcs1.key -outform PEM -out rsa_private_key_pkcs1.pem

[cloud-user@star23-bastion-1 ocats]$ openssl req -new -key rsa_private_key
-out ocats.csr -config ssl.conf -subj '/C=IN/ST=Karnataka/L=Bangalore/
O=Oracle/CN=sepp1.inter.oracle.com/emailAddress=xyz@oracle.com'

[cloud-user@star23-bastion-1 ocats]$ openssl x509 -CA caroot.cer -CAkey
cakey.pem -CAserial serial.txt -req -in ocats.csr -out ocats.cer -days 365
-extfile ssl.conf -extensions req_ext -passin pass:${PEM_PHRASE}



[cloud-user@star23-bastion-1 ocats]$ openssl ecparam -genkey -name
prime256v1 -out host.key

[cloud-user@star23-bastion-1 ocats]$  openssl req -new -sha256 -key
host.key -nodes -out ecdsa_certificate.csr -config ssl.conf -subj "/C=IN/
ST=Karnataka/L=Bangalore/O=Oracle/CN='${HELM_ATS_RELEASE_NAME}'-ocats.'$
{DEPLOY_NAMESPACE}'.'${DOMAIN}'"

[cloud-user@star23-bastion-1 ocats]$ openssl x509 -CA caroot.cer -req -
sha256 -days 730 -in ecdsa_certificate.csr -CAkey cakey.pem -
CAcreateserial -extfile ssl.conf -extensions req_ext -out
ssl_ecdsa_certificate.crt -passin pass:${PEM_PHRASE}

[cloud-user@star23-bastion-1 ocats]$ openssl pkcs8 -topk8 -in host.key -
out ssl_ecdsa_private_key.pem -nocrypt

[cloud-user@star23-bastion-1 ocats]$ echo ${PEM_PHRASE} > trust.txt

[cloud-user@star23-bastion-1 ocats]$ echo ${PEM_PHRASE} > key.txt
```

3. Run the following command to verify whether all configurations are done:

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


Copyright © 2019, 2026, Oracle and/or its affiliates.

4. Run the following command to sign the certificate signing request or csr file with root certificate:

```
[cloud-user@star23-bastion-1 ocats]$ 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:

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

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:

```
[cloud-user@star23-bastion-1 ocats]$ 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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.

```
[cloud-user@star23-bastion-1 ocats]$ 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.

3.4.4.3 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

2. Run the following command to verify the secret:

```
[cloud-user@star23-bastion-1 ~]$ 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:

- 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"

   ca_cert: "caroot.cer"

   client_cert: "ssl_rsa_certificate.crt"

   private_key: "rsa_private_key_pkcs1.key"
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
   # 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.
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.

3.4.5 Generating Hosted SEPP multi-fqdn Certificates and Secrets

> **Note**
>
> The following steps are steps are only required for Hosted SEPP deployment.

This section provides the steps for generating the Hosted SEPP multi-fqdn certificates and secrets.

Copyright © 2019, 2026, Oracle and/or its affiliates.

1. Both ATS and SEPP certificates must be created before generating Hosted SEPP certificate.
2. Create a separate directory to keep the Hosted SEPP multi-FQDN certificates.

   ```
   $ mkdir hs-multifqdn-certs
   ```

3. Copy the following default SEPP certificate files generated in the SEPP certificate generation section into this directory:

   ```
   caroot.cer
   rsa_private_key
   rsa_private_key_pkcs1.pem
   rsa_certificate.crt
   ocsepp.cer
   host.key
   ssl_ecdsa_certificate.crt
   ssl_ecdsa_private_key.pem
   trust.txt
   key.txt
   ssl.conf
   ```

4. Run the following command to copy these into the new directory:

   ```
   cp <default-sepp-cert-file> hs-multifqdn-certs/<default-sepp-cert-file>
   ```

5. Copy the ATS certificate files generated in the ATS certificate generation section into the Hosted SEPP multi-FQDN certificate directory with ats_certs_ prefix:

   ```
   cp caroot.cer hs-multifqdn-certs/ats_certs_caroot.cer
   cp ssl_rsa_certificate.crt hs-multifqdn-certs/
   ats_certs_ssl_rsa_certificate.crt
   cp ocats.cer hs-multifqdn-certs/ats_certs_ocats.cer
   cp ssl_ecdsa_certificate.crt hs-multifqdn-certs/
   ats_certs_ssl_ecdsa_certificate.crt
   cp rsa_private_key_pkcs1.key hs-multifqdn-certs/
   ats_certs_rsa_private_key_pkcs1.key
   cp rsa_private_key_pkcs1.pem hs-multifqdn-certs/
   ats_certs_rsa_private_key_pkcs1.pem
   cp host.key hs-multifqdn-certs/ats_certs_host.key
   cp ssl_ecdsa_private_key.pem hs-multifqdn-certs/
   ats_certs_ssl_ecdsa_private_key.pem
   cp ssl.conf hs-multifqdn-certs/ats_certs_ssl.conf
   cp certificate.p12 hs-multifqdn-certs/ats_certs_certificate.p12
   cp jenkinsserver.jks hs-multifqdn-certs/ats_certs_jenkinsserver.jks
   ```

6. Change directory to hs-multifqdn-certs:

   ```
   cd hs-multifqdn-certs
   ```

7. Hosted SEPP Multi-FQDN support requires separate certificate material per Hosted SEPP FQDN. The deployment does not rely on a single shared server certificate covering all Hosted SEPP FQDNs. Instead, each FQDN is configured with its own certificate chain, and

Copyright © 2019, 2026, Oracle and/or its affiliates.

all corresponding CA certificates are added to the ATS trust bundle. The Hosted SEPP FQDNs are:

```
ocats.hp1.inter.oracle.com

ocats.hp2.inter.oracle.com

ocats.rp1.inter.oracle.com

ocats.rp2.inter.oracle.com
```

8. Repeat the following steps a to f for certificate-generation and configuration steps for each of the above FQDNs:
   a. Start with one FQDN, for example "ocats.hp1.inter.oracle.com". Then repeat steps for remaining fqdns.
   b. Set the following environment variables:

      ```
      export PEM_PHRASE=NextGen1
      export DEPLOYMENT_NAMESPACE=<DEPLOYMENT_NAMESPACE>
      ```

   c. Create the SSL config file (Replace <HS_FQDN> with one of the FQDNs mentioned above for example "ocats.hp1.inter.oracle.com"):

      ```
      cat > "${HS_FQDN}_ssl.conf" << EOF
      ```

      Output:

      ```
      #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 = ${HS_FQDN}
      commonName_max = 64
      commonName_default = ${HS_FQDN}

      [ req_ext ]
      subjectAltName = @alt_names

      [alt_names]
      ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
IP.1 = 127.0.0.1
DNS.1 = ${HS_FQDN}
EOF
```

d. Create the CA config file:

```
cat > "${HS_FQDN}_ca_ssl.conf" << EOF



#ca_ssl.conf
[v3_ca]
basicConstraints = critical,CA:TRUE
keyUsage = critical,keyCertSign,cRLSign
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
EOF
```

e. Run the following commands after exporting the prefix:

```
export prefix=ocats.hp1.inter.oracle.com
```

Output:

```
openssl req -new -newkey rsa:4096 -nodes -keyout "${prefix}_cakey.pem" -
out "${prefix}_careq.pem" -subj "/C=IN/ST=Karnataka/L=Bangalore/
O=Oracle/CN=${prefix}"

openssl x509 -signkey "${prefix}_cakey.pem" -req -days 3650 -in "$
{prefix}_careq.pem" -out "${prefix}_caroot.cer" -extfile "$
{prefix}_ca_ssl.conf" -extensions v3_ca

openssl req -x509 -nodes -sha256 -days 365 -newkey rsa:2048 -keyout "$
{prefix}_rsa_private_key" -out "${prefix}_rsa_certificate.crt" -subj "/
C=IN/ST=Karnataka/L=Bangalore/O=Oracle/CN=${prefix}"

openssl rsa -in "${prefix}_rsa_private_key" -outform PEM -out "$
{prefix}_rsa_private_key_pkcs1.pem"

openssl req -new -key "${prefix}_rsa_private_key" -out "$
{prefix}_ocsepp.csr" -config "${prefix}_ssl.conf" -subj "/C=IN/
ST=Karnataka/L=Bangalore/O=Oracle/CN=${prefix}"

echo 1234 > "${prefix}_serial.txt"

openssl x509 -CA "${prefix}_caroot.cer" -CAkey "${prefix}_cakey.pem" -
CAserial "${prefix}_serial.txt" -req -in "${prefix}_ocsepp.csr" -out "$
{prefix}_ocsepp.cer" -days 365 -extfile "${prefix}_ssl.conf" -
extensions req_ext -passin pass:${PEM_PHRASE}

openssl ecparam -genkey -name prime256v1 -out "${prefix}_host.key"

openssl req -new -sha256 -key "${prefix}_host.key" -nodes -out "$
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
{prefix}_ecdsa_certificate.csr" -config "${prefix}_ssl.conf" -subj "/
C=IN/ST=Karnataka/L=Bangalore/O=Oracle/CN=${prefix}"

openssl x509 -CA "${prefix}_caroot.cer" -req -sha256 -days 730 -in "$
{prefix}_ecdsa_certificate.csr" -CAkey "${prefix}_cakey.pem" -
CAcreateserial -extfile "${prefix}_ssl.conf" -extensions req_ext -out "$
{prefix}_ssl_ecdsa_certificate.crt" -passin pass:${PEM_PHRASE}

openssl pkcs8 -topk8 -in "${prefix}_host.key" -out "$
{prefix}_ssl_ecdsa_private_key.pem" -nocrypt

echo "${PEM_PHRASE}" > "${prefix}_trust.txt"
echo "${PEM_PHRASE}" > "${prefix}_key.txt"
```

f. Run the following commands after exporting the prefix:

```
export prefix=ocats.hp1.inter.oracle.com
```

Output:

```
openssl req -new -newkey rsa:4096 -nodes -keyout "${prefix}_cakey.pem" -
out "${prefix}_careq.pem" -subj "/C=IN/ST=Karnataka/L=Bangalore/
O=Oracle/CN=${prefix}"

openssl x509 -signkey "${prefix}_cakey.pem" -req -days 3650 -in "$
{prefix}_careq.pem" -out "${prefix}_caroot.cer" -extfile "$
{prefix}_ca_ssl.conf" -extensions v3_ca

openssl req -x509 -nodes -sha256 -days 365 -newkey rsa:2048 -keyout "$
{prefix}_rsa_private_key" -out "${prefix}_rsa_certificate.crt" -subj "/
C=IN/ST=Karnataka/L=Bangalore/O=Oracle/CN=${prefix}"

openssl rsa -in "${prefix}_rsa_private_key" -outform PEM -out "$
{prefix}_rsa_private_key_pkcs1.pem"

openssl req -new -key "${prefix}_rsa_private_key" -out "$
{prefix}_ocsepp.csr" -config "${prefix}_ssl.conf" -subj "/C=IN/
ST=Karnataka/L=Bangalore/O=Oracle/CN=${prefix}"

echo 1234 > "${prefix}_serial.txt"

openssl x509 -CA "${prefix}_caroot.cer" -CAkey "${prefix}_cakey.pem" -
CAserial "${prefix}_serial.txt" -req -in "${prefix}_ocsepp.csr" -out "$
{prefix}_ocsepp.cer" -days 365 -extfile "${prefix}_ssl.conf" -
extensions req_ext -passin pass:${PEM_PHRASE}

openssl ecparam -genkey -name prime256v1 -out "${prefix}_host.key"

openssl req -new -sha256 -key "${prefix}_host.key" -nodes -out "$
{prefix}_ecdsa_certificate.csr" -config "${prefix}_ssl.conf" -subj "/
C=IN/ST=Karnataka/L=Bangalore/O=Oracle/CN=${prefix}"

openssl x509 -CA "${prefix}_caroot.cer" -req -sha256 -days 730 -in "$
{prefix}_ecdsa_certificate.csr" -CAkey "${prefix}_cakey.pem" -
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
CAcreateserial -extfile "${prefix}_ssl.conf" -extensions req_ext -out "$
{prefix}_ssl_ecdsa_certificate.crt" -passin pass:${PEM_PHRASE}

openssl pkcs8 -topk8 -in "${prefix}_host.key" -out "$
{prefix}_ssl_ecdsa_private_key.pem" -nocrypt

echo "${PEM_PHRASE}" > "${prefix}_trust.txt"
echo "${PEM_PHRASE}" > "${prefix}_key.txt"
```

g. Verify whether the certificate is properly signed by root certificate:

```
openssl verify -CAfile "${HS_FQDN}_caroot.cer" "${HS_FQDN}_ocsepp.cer"
"${HS_FQDN}_ssl_ecdsa_certificate.crt"
```

h. Now, create the ATS CA bundle in hs-multifqdn-certs directory:

```
: > ats_bundle.cer

sed -n '/-----BEGIN CERTIFICATE-----/,/-----END CERTIFICATE-----/p'
ocats.hp1.inter.oracle.com_caroot.cer >> ats_bundle.cer
printf '%s\n' '--------' >> ats_bundle.cer

sed -n '/-----BEGIN CERTIFICATE-----/,/-----END CERTIFICATE-----/p'
ocats.hp2.inter.oracle.com_caroot.cer >> ats_bundle.cer
printf '%s\n' '--------' >> ats_bundle.cer

sed -n '/-----BEGIN CERTIFICATE-----/,/-----END CERTIFICATE-----/p'
ocats.rp1.inter.oracle.com_caroot.cer >> ats_bundle.cer
printf '%s\n' '--------' >> ats_bundle.cer

sed -n '/-----BEGIN CERTIFICATE-----/,/-----END CERTIFICATE-----/p'
ocats.rp2.inter.oracle.com_caroot.cer >> ats_bundle.cer
printf '%s\n' '--------' >> ats_bundle.cer

sed -n '/-----BEGIN CERTIFICATE-----/,/-----END CERTIFICATE-----/p'
ats_certs_caroot.cer >> ats_bundle.cer
```

i. After doing the abovesteps in hs-multifqdn-certs directory, following files will be present:

Figure 3-16 Hosted SEPP Multi FQDN files

```
[cloud-user@occne-26-1-200-nss-bastion-1 hs-certificates-sepp.hosted.26226260-backup]$ ls
ats_bundle.cer                                ocats.hp1.inter.oracle.com_rsa_private_key_pkcs1.pem  ocats.rp1.inter.oracle.com_ssl.conf
ats_certs_caroot.cer                          ocats.hp1.inter.oracle.com_ssl.conf                   ocats.rp1.inter.oracle.com_ssl_ecdsa_certificate.crt
ats_certs_certificate.p12                     ocats.hp1.inter.oracle.com_ssl_ecdsa_certificate.crt  ocats.rp1.inter.oracle.com_ssl_ecdsa_private_key.pem
ats_certs_host.key                            ocats.hp1.inter.oracle.com_ssl_ecdsa_private_key.pem  ocats.rp1.inter.oracle.com_trust.txt
ats_certs_jenkinsserver.jks                   ocats.hp1.inter.oracle.com_trust.txt                  ocats.rp2.inter.oracle.com_caroot.cer
ats_certs_ocats.cer                           ocats.hp2.inter.oracle.com_caroot.cer                 ocats.rp2.inter.oracle.com_ca_ssl.conf
ats_certs_ocsepp.cer                          ocats.hp2.inter.oracle.com_ca_ssl.conf                ocats.rp2.inter.oracle.com_host.key
ats_certs_rsa_certificate.crt                 ocats.hp2.inter.oracle.com_host.key                   ocats.rp2.inter.oracle.com_key.txt
ats_certs_rsa_private_key                     ocats.hp2.inter.oracle.com_key.txt                    ocats.rp2.inter.oracle.com_ocsepp.cer
ats_certs_rsa_private_key_pkcs1.key           ocats.hp2.inter.oracle.com_ocsepp.cer                 ocats.rp2.inter.oracle.com_rsa_certificate.crt
ats_certs_rsa_private_key_pkcs1.pem           ocats.hp2.inter.oracle.com_rsa_certificate.crt        ocats.rp2.inter.oracle.com_rsa_private_key
ats_certs_ssl.conf                            ocats.hp2.inter.oracle.com_rsa_private_key            ocats.rp2.inter.oracle.com_rsa_private_key_pkcs1.pem
ats_certs_ssl_ecdsa_certificate.crt           ocats.hp2.inter.oracle.com_rsa_private_key_pkcs1.pem  ocats.rp2.inter.oracle.com_ssl.conf
ats_certs_ssl_ecdsa_private_key.pem           ocats.hp2.inter.oracle.com_ssl.conf                   ocats.rp2.inter.oracle.com_ssl_ecdsa_certificate.crt
ats_certs_ssl_rsa_certificate.crt             ocats.hp2.inter.oracle.com_ssl_ecdsa_certificate.crt  ocats.rp2.inter.oracle.com_ssl_ecdsa_private_key.pem
caroot.cer                                    ocats.hp2.inter.oracle.com_ssl_ecdsa_private_key.pem  ocats.rp2.inter.oracle.com_trust.txt
host.key                                      ocats.hp2.inter.oracle.com_trust.txt                  ocsepp.cer
key.txt                                       ocats.rp1.inter.oracle.com_caroot.cer                 rsa_certificate.crt
ocats.hp1.inter.oracle.com_caroot.cer         ocats.rp1.inter.oracle.com_ca_ssl.conf                rsa_private_key
ocats.hp1.inter.oracle.com_ca_ssl.conf        ocats.rp1.inter.oracle.com_host.key                   rsa_private_key_pkcs1.pem
ocats.hp1.inter.oracle.com_host.key           ocats.rp1.inter.oracle.com_key.txt                    ssl.conf
ocats.hp1.inter.oracle.com_key.txt            ocats.rp1.inter.oracle.com_ocsepp.cer                 ssl_ecdsa_certificate.crt
ocats.hp1.inter.oracle.com_ocsepp.cer         ocats.rp1.inter.oracle.com_rsa_certificate.crt        ssl_ecdsa_private_key.pem
ocats.hp1.inter.oracle.com_rsa_certificate.crt ocats.rp1.inter.oracle.com_rsa_private_key           tag.txt
ocats.hp1.inter.oracle.com_rsa_private_key    ocats.rp1.inter.oracle.com_rsa_private_key_pkcs1.pem  trust.txt
```

j. Create or update the Hosted SEPP secret (ocats-hs-secret) and the n32-ingress-gateway-secret and n32-egress-gateway-secret:

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> If n32-ingress-gateway-secret and n32-ingress-gateway-secret have already been created while Hosted SEPP installation, delete them before creating the following secrets

```
kubectl create secret generic <n32-ingress-gateway-secret> \
  --from-file=hs-multifqdn-certs \
  -n "${DEPLOYMENT_NAMESPACE}" \
  --dry-run=client -o yaml | kubectl apply -n "${DEPLOYMENT_NAMESPACE}"
-f -

kubectl create secret generic <n32-egress-gateway-secret> \
  --from-file=hs-multifqdn-certs \
  -n "${DEPLOYMENT_NAMESPACE}" \
  --dry-run=client -o yaml | kubectl apply -n "${DEPLOYMENT_NAMESPACE}"
-f -

kubectl create secret generic ocats-hs-secret \
  --from-file=hs-multifqdn-certs \
  -n "${DEPLOYMENT_NAMESPACE}" \
  --dry-run=client -o yaml | kubectl apply -n "${DEPLOYMENT_NAMESPACE}"
-f -
```

3.4.6 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.

```
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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

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:

```
ocats:
  atsFeatures:
    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: ""
```

3.4.7 Configuring ATS

This section describes how to configure ATS for SEPP.

3.4.7.1 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
   ```

   Example:

   ```
   kubectl label --overwrite namespace seppsvc istio-injection=enabled
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

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
  plmnIgwPort: 80 # plmn ingress gateway service port
  n32IgwPort: 443 # 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 "null" in ASM mode to deploy stub server in HTTP mode and HTTPS mode to support for bypassing ASM sidecars architecture.

```
stubserver:
  service:
    expose_tls_service: null
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

7. Update the following parameters in the `ocats_ocsepp_values_<values>.yaml` file file with the appropriate certificate values, as secret creation is now required for ASM mode due to bypassing ASM sidecars architecture.
   In ocats section:

```
ocats:
    certificates:
        cert_secret_name: "ocats-sepp-secret"
        ca_cert: "caroot.cer"
        client_cert: "ssl_rsa_certificate.crt"
        private_key: "rsa_private_key_pkcs1.key"
        jks_file: "jenkinsserver.jks" # This parameter is needed when
atsGuiTLSEnabled is set to true. This file is necessary for ATS GUI to be
opened with secured TLS protocol.
```

In stubserver section:

```
tubserver:
  env:
    cert_secret_name: "ocats-sepp-secret"
    ca_cert: "caroot.cer"
    client_cert: "ssl_rsa_certificate.crt"
    private_key: "rsa_private_key_pkcs1.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.

3.4.7.2 Configuring NIF

This section provides the details of configuring Network Intelligence Fabric (NIF).

1. Update the the following configuration in the `ocsepp-custom-values.yaml` file to run the ATS NIF test cases:

```
nrfclient:
    nrf-client:
        configmapApplicationConfig:
            profile:
                primaryNrfApiRoot: <ats-release-name>-
stubserver.<namespace>:8080
                nrfScheme=http
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> Replace <ats-release-name> with the actual ATS Helm release name.

2. Update the the following configuration in the ocats_ocsepp_values_<version>.yaml file to run the enable the stubserver-4:

```
stubserver:
  stubserver2:
    enabled: true
  stubserver3:
    enabled: true
  stubserver4:
    enabled: true
```

Configuring NIF Discovery Initial and Scheduled Delays

- nifDiscoveryInitialDelay parameter configures the initial delay period after the starting the pod and before NIF discovery is initiated from the config manager to the NRF.

```
nifDiscoveryInitialDelay:2000
```

- nifDiscoveryScheduledDelay parameter configures the scheduled delay before the config manager initiates NIF discovery queries to the NRF.

```
nifDiscoveryScheduledDelay:2000
```

- The following annotation is required to support ATS run as per bypassing ASM sidecars architecture, do the following changes in the ocats_ocsepp_values_<values>.yaml file:

```
deployment:
      customExtension:
        annotations:
          traffic.sidecar.istio.io/excludeInboundPorts: "8443"
```

3.4.7.3 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.

3.4.7.4 Enabling Static Port

To enable static port:

> **Note**
>
> ATS supports static port. By default, this feature is disabled.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- 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: ""
        http:
          port: "8080"
          staticNodePortEnabled: false
          staticNodePort: ""
```

3.4.7.5 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
```

3.4.7.6 Configuring Hosted SEPP Mode

SEPP ATS now supports three deployment modes: SEPP, Roaming Hub, and Hosted SEPP. A flag `HSenabled` has been introduced to select the Hosted SEPP deployment mode for running ATS test cases.

Enabling Roaming Hub mode

1. Update `ocats_custom_values.yaml` to enable Roaming Hub Mode:
   To enable Roaming Hub, do the following in the `ocats_custom_values.yaml` file:

```
#Flag to enable Roaming Hub mode
 RHenabled: false
```

Enabling Hosted SEPP mode

1. Update `ocats_custom_values.yaml` to enable Hosted SEPP Mode:

Copyright © 2019, 2026, Oracle and/or its affiliates.

To enable Hosted SEPP, do the following in the `ocats_custom_values.yaml` file:

```
ocats:
  #Flag to enable hosted Sepp mode (values can be : true /false)
  HSenabled: false

  #HSSecret specifies the Kubernetes secret that contains the Hosted SEPP
Multi-FQDN certificate bundle used by ATS.
  HSSecret: "ocats-hs-secret"
```

Enable the Hosted SEPP parameter:

```
global:
   #Flag to enable hosted Sepp mode (values can be : true /false)
  hostedSEPP: &hostedSEPPEnabled true
```

> **Note**
>
> The secret "ocats-hs-secret" should be present prior to ATS installation. See, Generating Hosted SEPP multi-fqdn Certificates and Secrets section to support TLSv1.2 and TLSv1.3 in SEPP Enable HS Mode.

2. Enable the `multiFqdn` parameter and add the required `fqdnDetails` under both n32-ingress-gateway and n32-egress-gateway sections. Refer to the certificate files created earlier in the "ATS Framework to Support TLSv1.2 and TLSv1.3 in SEPP" section for updates.
   Example for the n32-ingress-gateway and n32-egress-gateway:

```
n32-ingress-gateway:
  multiFqdn:
    enabled: true
    fqdnDetails:
      - fqdn: ocats.hp1.inter.oracle.com
        caroot_file: "ats_bundle.cer"
        rsa_certificate_file: "ocats.hp1.inter.oracle.com_ocsepp.cer"
        rsa_private_key_file:
"ocats.hp1.inter.oracle.com_rsa_private_key_pkcs1.pem"
        key_store_passwd_file: "ocats.hp1.inter.oracle.com_key.txt"
        trust_store_passwd_file: "ocats.hp1.inter.oracle.com_trust.txt"
        ssl_ecdsa_certificate_file:
"ocats.hp1.inter.oracle.com_ssl_ecdsa_certificate.crt"
        ssl_ecdsa_private_key_file:
"ocats.hp1.inter.oracle.com_ssl_ecdsa_private_key.pem"

n32-egress-gateway:
  multiFqdn:
    enabled: true
    fqdnDetails:
      - fqdn: ocats.hp1.inter.oracle.com
        caroot_file: "ats_bundle.cer"
        rsa_certificate_file: "ocats.hp1.inter.oracle.com_ocsepp.cer"
        rsa_private_key_file:
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
"ocats.hp1.inter.oracle.com_rsa_private_key_pkcs1.pem"
       key_store_passwd_file: "ocats.hp1.inter.oracle.com_key.txt"
       trust_store_passwd_file: "ocats.hp1.inter.oracle.com_trust.txt"
       ssl_ecdsa_certificate_file:
"ocats.hp1.inter.oracle.com_ssl_ecdsa_certificate.crt"
       ssl_ecdsa_private_key_file:
"ocats.hp1.inter.oracle.com_ssl_ecdsa_private_key.pem"
```

> **Note**
>
> After configuring all the ats certificates files, create Kubernetes secrets for the n32-ingress-gateway and n32-egress-gateway using the certificate files listed above.

3. Update the retryInterval parameter to 5000 in ocsepp-custom-values.yaml. This change reduces the remote PLMN retry interval from the default 30 minutes (300000 ms) to 5 seconds. It enables Config Manager to retry pending N32 handshake attempts quickly, significantly reducing ATS testcase run time and avoiding a 30-minute wait for a single handshake retry.

```
global:
  localProfile:
    retryInterval: 5000
```

4. Configure dedicated N32 gateway TLS Secrets: Update ocsepp-custom-values.yaml so that the N32 ingress and egress gateways use their dedicated TLS Secrets.

```
Update the k8SecretName fields under service.ssl for both gateways.n32-
ingress-gateway:
n32-ingress-gateway:
  volumeMountSecrets:
    featureSecrets:
      - ocsepp-n32-igw-secret

  service:
    ssl:
      privateKey:
        k8SecretName: ocsepp-n32-igw-secret
      certificate:
        k8SecretName: ocsepp-n32-igw-secret
      caBundle:
        k8SecretName: ocsepp-n32-igw-secret
      keyStorePassword:
        k8SecretName: ocsepp-n32-igw-secret
      trustStorePassword:
        k8SecretName: ocsepp-n32-igw-secret

n32-egress-gateway:
  volumeMountSecrets:
    featureSecrets:
      - ocsepp-n32-egw-secret
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
  service:
    ssl:
      privateKey:
        k8SecretName: ocsepp-n32-egw-secret
      certificate:
        k8SecretName: ocsepp-n32-egw-secret
      caBundle:
        k8SecretName: ocsepp-n32-egw-secret
      keyStorePassword:
        k8SecretName: ocsepp-n32-egw-secret
      trustStorePassword:
        k8SecretName: ocsepp-n32-egw-secret
```

5. Disable `n32cHandshakePlmnIdListValidationEnabled` parameter in global section of ocsepp-custom-values.yaml file:

```
global:
   #Disable for Hosted Sepp (value can be: true/false)
   n32cHandshakePlmnIdListValidationEnabled: false
```

6. For Hosted SEPP deployments, update the SAN extraction regular expression present in the ocsepp-custom-values.yaml to support the format used by the new certificates. It should be updated at two places:

```
pn32f-svc:
  configs:
    extractSANRegex: "SAN=([^ ]+)\\s"

pn32c-svc:
  configs:
    extractSANRegex: "SAN=([^ ]+)\\s"
```

7. Update the first occurence of "target" value in ATS Load Sharing to <ats-release-name>-stubserver-4:

```
alternate-route:
  staticVirtualFqdns:
  #Below Config will be used while running ATS Load SHaring CN32F test
cases
    - 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
        priority: 1
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
                weight: 90
```

Configuring Customized Hosted SEPP Error Code

To handle failure scenarios, a customized error code variable has been introduced for Hosted SEPP. Ensure the value matches your configuration:

```
#Customized error code variable for Hosted SEPP
HSErrCode:  "400"
```

3.4.7.7 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*.

3.4.7.8 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*.

3.4.7.9 Configuring Single Service Account

> **Note**
>
> If the user doesn't want to configure a single service account, skip the following steps.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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: ""
```

3.4.7.10 Configuring the Cache Refresh Timeout Value

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).

3.4.7.11 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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
```

3.4.7.12 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.

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)
```

3.4.7.13 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"
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    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"},
{"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"},
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

{"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"},
{"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"},

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
{"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"},
{"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"}]
```


3.4.7.14 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- 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 3-16 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

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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

3.4.7.15 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".

3.4.7.16 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 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
```

3.4.7.17 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)
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> If ATS is not configured `evictSanHeaderCacheDelay` must be set to 50000 in the pn32f-svc section of `ocsepp_custom_values_<values>.yaml` file.

3.4.7.18 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}}]
```

3.4.7.19 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:

```
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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
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
> ```

> **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"
>     ]
> ```

3.4.7.20 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:

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
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
         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
```

3.4.7.21 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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
```

3.4.7.22 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 3-17 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.
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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-17 (Cont.) Feature Level Tags in SEPP Mode

Sl.No.	Tag	Selected Feature Files
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 3-18 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.
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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 3-18 (Cont.) Feature Level Tags in Roaming Hub Mode

Sl.No.	Tag	Selected Feature Files
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 3-19 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.

3.4.7.23 ATS Helm Enhancement

ATS Helm enhancement enables configuring the ATS pipeline configuration variables directly through the ATS Helm charts.

In the ATS custom values file, the following variables are available. Update the variables in the same way to configure them through the ATS Jenkins UI:

Regression:

```
SELECTED_NF_a: "SEPP"

 SEPPCONFIGSVCNAME_b: "ocsepp-release-config-mgr-svc.csepp"

 CSEPPIGWNAME_c: "ocsepp-release-plmn-ingress-gateway.csepp"

 PSEPPIGWNAME_d: "ocsepp-release-n32-ingress-gateway.csepp"

 SEPPSTUBNAME_e: "sepp-ats-rel-stubserver.csepp"

 NFINSTANCEID_f: "9faf1bbc-6e4a-4454-a507-aef01a101a06"

 PROMSVCIP_g: "prometheus.cne-infra"

 PROMSVCPORT_h: "9090"

 PROMSVCURI_i: "/api/"

 RERUN_COUNT_j: "1"

 ELK_WAIT_TIME_k: "10"

 ELK_HOSTNAME_l: "occne-elastic-elasticsearch-master.occne-infra"
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    ELK_PORT_m: "9200"

    STUB_LOGS_n: "yes"

    LOG_TYPE_o: "kubernetes"

    GLOBAL_WAIT_TIME_p: "12
```

NewFeatures:

```
    SELECTED_NF_a: "SEPP"

    SEPPCONFIGSVCNAME_b: "ocsepp-release-config-mgr-svc.csepp"

    CSEPPIGWNAME_c: "ocsepp-release-plmn-ingress-gateway.csepp"

    PSEPPIGWNAME_d: "ocsepp-release-n32-ingress-gateway.csepp"

    SEPPSTUBNAME_e: "sepp-ats-rel-stubserver.csepp"

    NFINSTANCEID_f: "9faf1bbc-6e4a-4454-a507-aef01a101a06"

    PROMSVCIP_g: "prometheus.cne-infra"

    PROMSVCPORT_h: "9090"

    PROMSVCURI_i: "/api/"

    RERUN_COUNT_j: "1"

    ELK_WAIT_TIME_k: "10"

    ELK_HOSTNAME_l: "occne-elastic-elasticsearch-master.occne-infra"

    ELK_PORT_m: "9200"

    STUB_LOGS_n: "yes"

    LOG_TYPE_o: "kubernetes"

    GLOBAL_WAIT_TIME_p: "120"
```

Advantage of this configuration:

- No need to configure the pipeline through the ATS UI.
- Simply select the desired options on the Build with Parameters page and trigger the build to start ATS execution.
- Pipeline configuration remains intact even after ATS pod restarts.

3.4.7.24 Configuring LCI and OCI Header Support

Update the following configuration to run LCI and OCI header test cases:

Copyright © 2019, 2026, Oracle and/or its affiliates.

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
    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
```

3.4.7.25 NRF Selection Mechanisms Using nrf client

This feature allows the SEPP to dynamically select NRF instances based on real-time availability and site redundancy through DNS SRV configurations. In addition to the static configurations by operators, the SEPP can now resolve NRFs using DNS SRV-based Fully Qualified Domain Names (FQDNs). The SEPP is configured with a primary NRF and multiple fallback NRFs, which take over if the primary NRF becomes unreachable

In `ocsepp_custom_values_<version>.yaml` update the following:

Copyright © 2019, 2026, Oracle and/or its affiliates.

In ASM Mode:

```
nrfclient:
    nrf-client:
        configmapApplicationConfig:
            profile:
                primaryNrfApiRoot: <ats-release-name>-
stubserver.<namespace>:8080
                nrfScheme=http
```

In non-ASM Mode:

```
nrfclient:
    nrf-client:
        configmapApplicationConfig:
            profile:
                primaryNrfApiRoot: <ats-release-name>-
stubserver.<namespace>:8443
                nrfScheme=https
```

3.4.7.26 Enabling Persistant Volume Storage

ATS supports Persistent storage to retain ATS historical build execution data, test cases, and one-time environment variable configurations.

> **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

Copyright © 2019, 2026, Oracle and/or its affiliates.

3.4.8 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
```

3.4.9 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 3-17 Checking Pod and Service Deployment

```
sepp-ats-rel-ocats                LoadBalancer   10.96.30.119    <pending>     8080:30049/TCP,5001:32644/TCP
sepp-ats-rel-stubserver           LoadBalancer   10.96.50.127    <pending>     8080:30732/TCP,8081:30267/TCP
sepp-ats-rel-stubserver-2         LoadBalancer   10.96.43.73     <pending>     8080:32685/TCP,8081:30934/TCP
sepp-ats-rel-stubserver-3         LoadBalancer   10.96.127.151   <pending>     8080:32762/TCP,8081:30018/TCP
```

> **Note**
>
> bddclient name is now updated to ocats.

Copyright © 2019, 2026, Oracle and/or its affiliates.

3.4.10 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.

Copyright © 2019, 2026, 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 NRF Test Cases using ATS
- Running NSSF Test Cases using ATS
- Running SCP Test Cases using ATS
- Running SEPP Test Cases using ATS

4.1 Running NRF Test Cases using ATS

4.1.1 Prerequisites

To run NRF test cases using NRF ATS 26.1.201, you need to ensure that following prerequisites are met.

- For three-site georedundancy test cases, deploy NRF 26.1.201 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 26.1.201 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 26.1.201 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- Deploy NRF 26.1.201 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.
- For NRF 26.1.201, 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- 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.
- 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:
  ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    - protocol: TCP
      port: 8080


kubectl apply -f allowEgressPrometheus.yaml -n ocnrf
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-egress-prometheus
spec:
  policyTypes:
    - Egress
  podSelector:
    matchLabels:
      app: ocats-nrf
  egress:
    - ports:
        - port: 9090
          protocol: TCP
      to:
        - namespaceSelector:
            matchLabels:
              kubernetes.io/metadata.name: occne-infra
        - podSelector:
            matchLabels:
              app.kubernetes.io/name: prometheus
```

- 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.

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": {
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
                        "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"
                                        }
                                },
                                {
                                        "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"
                                        }
                                }
                        ]
                }
        }
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

> **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" : {
        "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"
    }
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
 }
}
```

```
$ 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"
    },
    "certificate" : {
        "k8sSecretName" : "ocnrfaccesstoken-secret",
        "fileName" : "rsa_certificate.crt"
    }
 }
}
```

> **Note**
>
> Here, user should update k8SecretName fileName (name of private keys and the corresponding certificates.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **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
       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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **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_26.1.201.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.
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
   ```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    //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, notify-stub-service03.occne-cndbtierone,
notify-stub-service04.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 0.25 \
        -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"
    }
    else{
        load "/var/lib/jenkins/ocnrf_tests/jenkinsData/Jenkinsfile-
Regression"
    }

    }
}
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

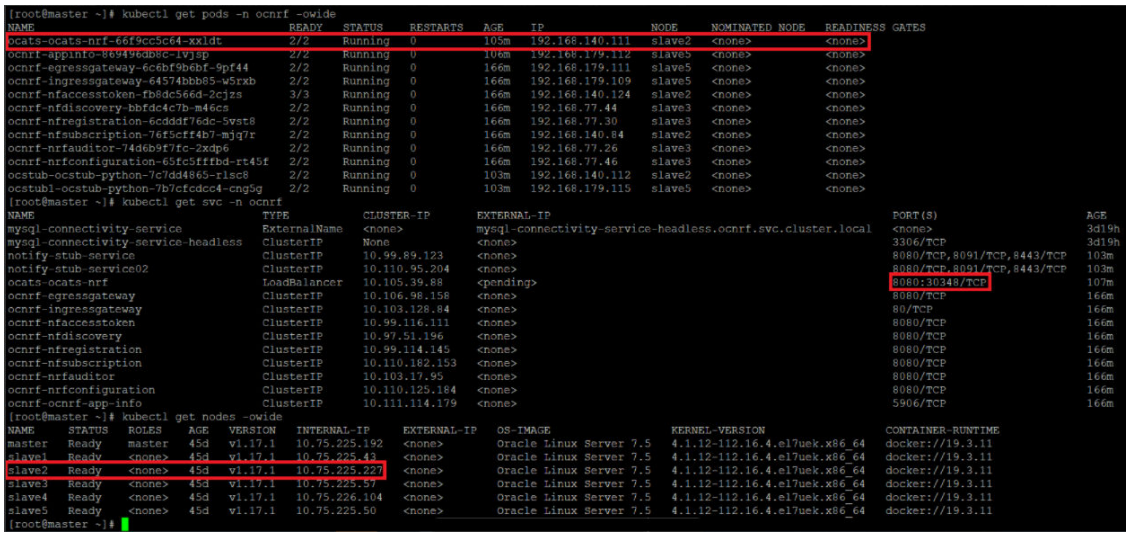
4.1.2 Logging into ATS

Running 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 4-1 Verifying ATS Pod



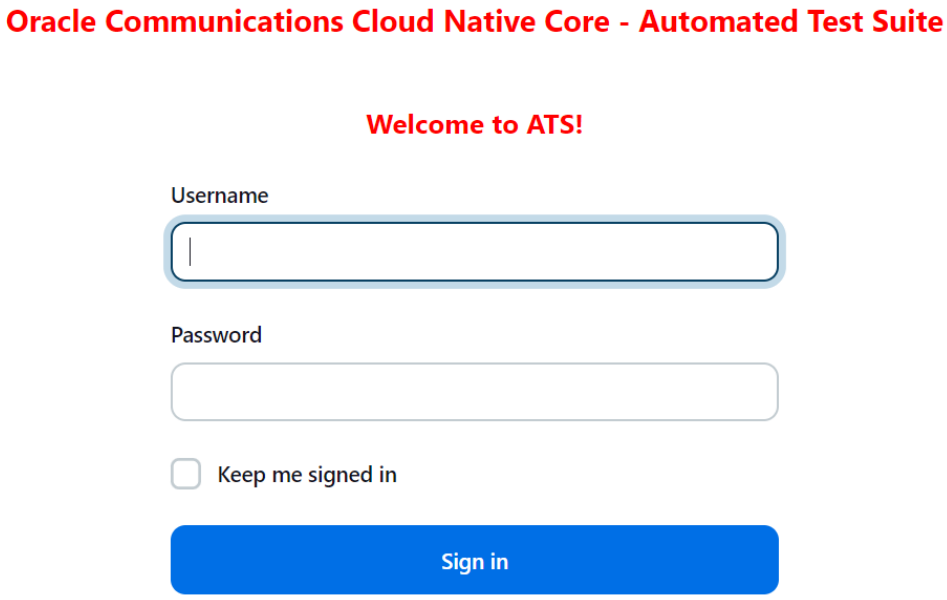
For more information on verifying ATS deployment, see Verifying ATS Deployment.

> **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, 2026, Oracle and/or its affiliates.

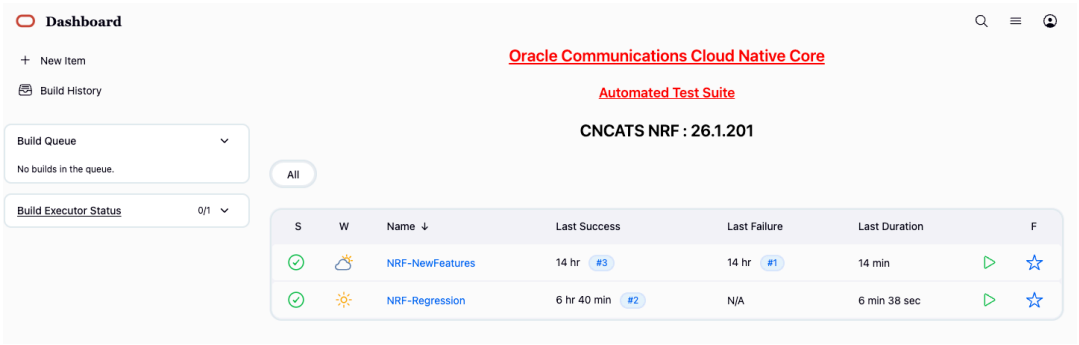
Figure 4-2 ATS Login



To run ATS:

1. Enter the login credentials. Click **Sign in**. The following screen appears.

Figure 4-3 NRF Preconfigured Pipelines



NRF ATS has two pre-configured pipelines.

- **NRF-NewFeatures:** This pipeline has all the test cases delivered as part of NRF ATS - 26.1.201.
- **NRF-Regression:** This pipeline has all the test cases delivered so far in the previous releases.

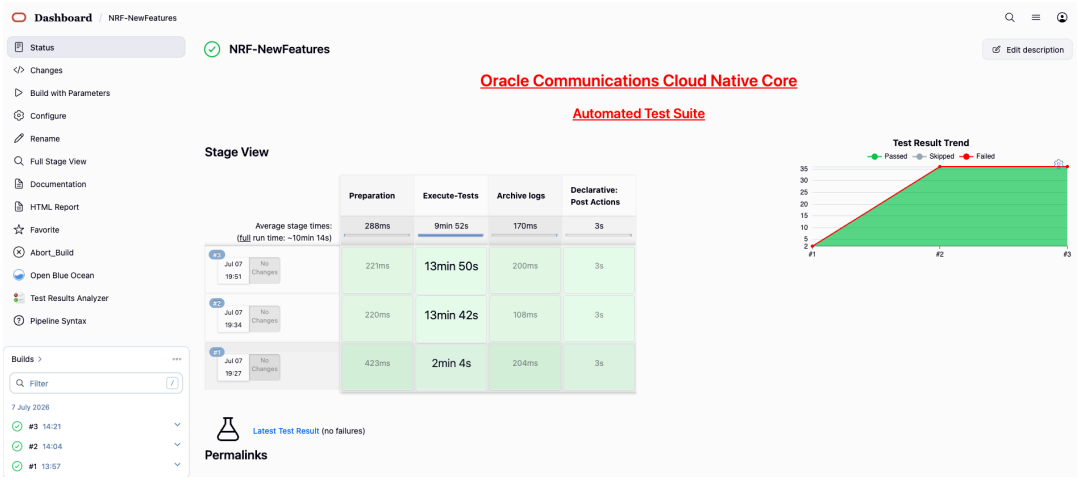
4.1.3 NRF-NewFeatures Pipeline

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:

Copyright © 2019, 2026, Oracle and/or its affiliates.

1. Click **NRF-NewFeatures** in the **Name** column. The following screen appears:Configuring NRF-NewFeatures

Figure 4-4 Configuring NRF-NewFeatures



In the above screen:

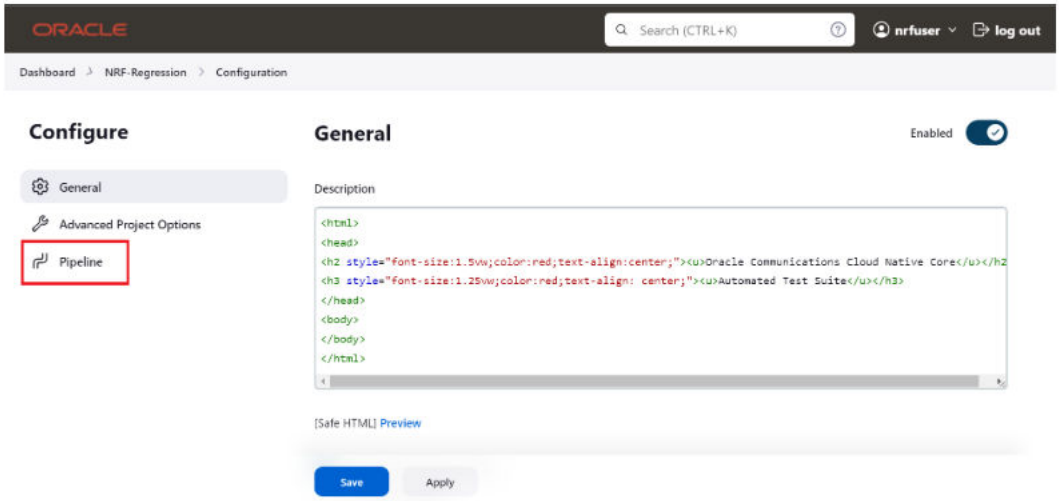
- Click **Configure** to configure NRF-New Features.
- Click **Documentation** to view the documented test cases, which are part of this NRF release.
- Click the blue dots inside **Build History** box to view the success console logs of the "Sanity", "All-NewFeatures" respectively.
- The **Stage View** 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, 2026, Oracle and/or its affiliates.

Figure 4-5 Pipeline Tab



The **Pipeline** section of the configuration page appears as follows:

Figure 4-6 Preconfigured pipeline

```
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 nrf-ci-ats-regression-26176196-26176203 \
        -c nrf-26176203-ingressgateway:80 \
        -d nrf-26176203-ingressgateway \
        -e 80 \
        -f nrf-26176203-nrfconfiguration \
        -g 8080 \
        -h notify-stub-service.nrf-ci-ats-regression-26176196-26176203 \
        -i 8080 \
        -j 6faf1bbc-6e4a-4454-a507-a14ef8e1bc5c \
        -k oso-26176203-prom-svr.nrf-ci-ats-regression-26176196-26176203 \
        -l 80 \
        -m 1 \
        -n /nrf-ci-ats-regression-26176196-26176203/prometheus/api/ \
        -o nrf-26176203 \
        -p database-mysql.nrf-ci-ats-regression-26176196-26176203 \
        -q privilegeduser-secret \
        -r 0.5 \
        -s http://ocnrf-ocnrf-app-info:5906/status/category/replicationstatus \
        -t svc.cluster.local \
        -u 0\
        -v 0.25 \
        -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, 2026, Oracle and/or its affiliates.

Figure 4-7 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 nrf-ci-ats-regression-26176196-26176203 \
        -c nrf-26176203-ingressgateway:80 \
        -d nrf-26176203-ingressgateway \
        -e 80 \
        -f nrf-26176203-nrfconfiguration \
        -g 8080 \
        -h notify-stub-service.nrf-ci-ats-regression-26176196-26176203 \
        -i 8080 \
        -j 6faf1bbc-6e4a-4454-a507-a14ef8e1bc5c \
        -k oso-26176203-prom-svr.nrf-ci-ats-regression-26176196-26176203 \
        -l 80 \
        -m 1 \
        -n /nrf-ci-ats-regression-26176196-26176203/prometheus/api/ \
        -o nrf-26176203 \
        -p database-mysql.nrf-ci-ats-regression-26176196-26176203 \
        -q privilegeduser-secret \
        -r 0.5 \
        -s http://ocnrf-ocnrf-app-info:5906/status/category/replicationstatus \
        -t svc.cluster.local \
        -u 0\
        -v 0.25 \
        -w 0\
        -x 0
    '''
```

> **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).

Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> Ensure that all the stubs are added to the pretest config after bringing up the deployment.

- **i:** Port of stub service (8080).
- **j:** NRF_Instance ID (6faf1bbc-6e4a-4454-a507-a14ef8e1bc5c).
- **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, 2026, 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 0.25 \        -w 0 \
        -x 0
    '''
    if(env.Include_NewFeatures && "${Include_NewFeatures}" ==
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
"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 \
        -j 6faf1bbc-6e4a-4454-a507-
a14ef8e1bc5c,6faf1bbc-6e4a-4454-a507-
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
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 0.25 \
        -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.
   - **Sanity:** It is recommended to run sanity before running any test case. It ensures all the deployments are done properly.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **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.

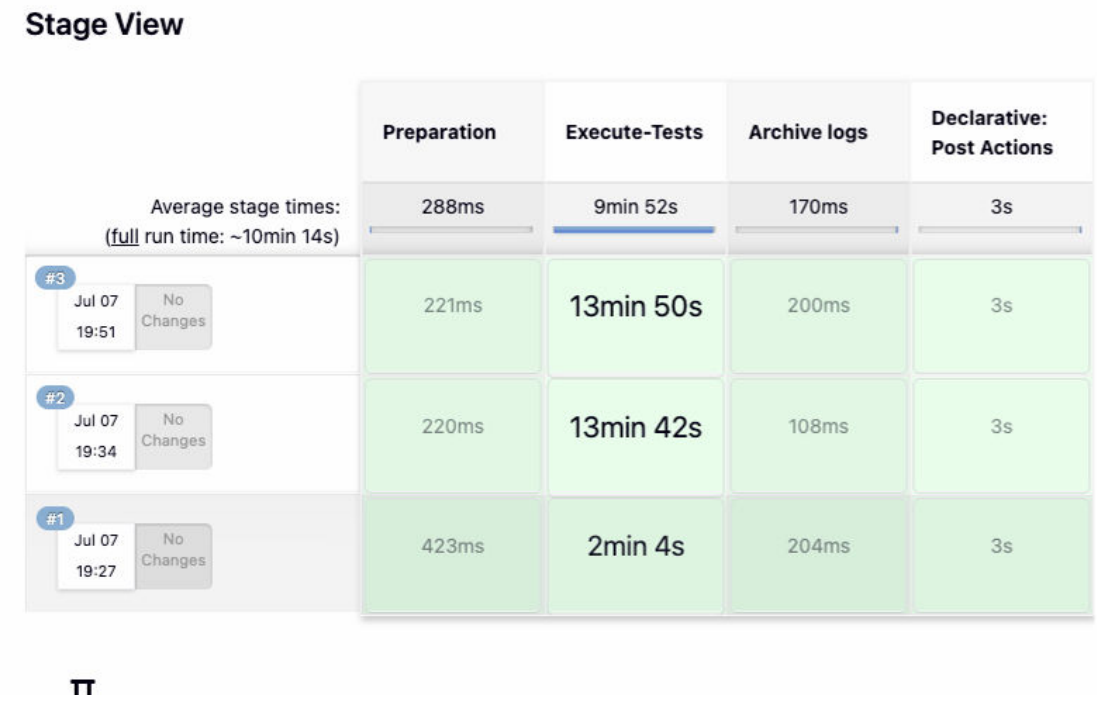
4. Select one of the following configuration types:
   - **Product_Config**: On selecting this option, test cases from product folders are populated on ATS UI and product configuration is applied to them via the key-value pair and yaml files defined or present in the "Product Config" folder.
   - **Custom_Config**: 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, 2026, Oracle and/or its affiliates.

Figure 4-8 Sample Screen: NRF ATS Full Execution



A sample consolidated test report is shown below, when rerun is set to 2:

Figure 4-9 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.

Test Cases Result - All-NewFeatures

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 4-10 Test Cases Result-All NewFeatures

```
2026-07-07 14:18:21,042 [32m     INFO LOG.AllHooks:306  [0m|  [32mInside after_all [0m
2026-07-07 14:18:21,042 [32m     INFO LOG.AllHooks:420  [0m|  [32mExiting after_all [0m
11 features passed, 0 failed, 0 skipped
36 scenarios passed, 0 failed, 0 skipped
648 steps passed, 0 failed, 0 skipped
Took 7min 34.728s
[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
```

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, 2026, Oracle and/or its affiliates.

> **Note**
>
> For NRF-ATS release 26.1.201, parameterization is supported for NRF NewFeatures pipeline only.

4.1.4 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 -26.1.201 along with sanity cases.

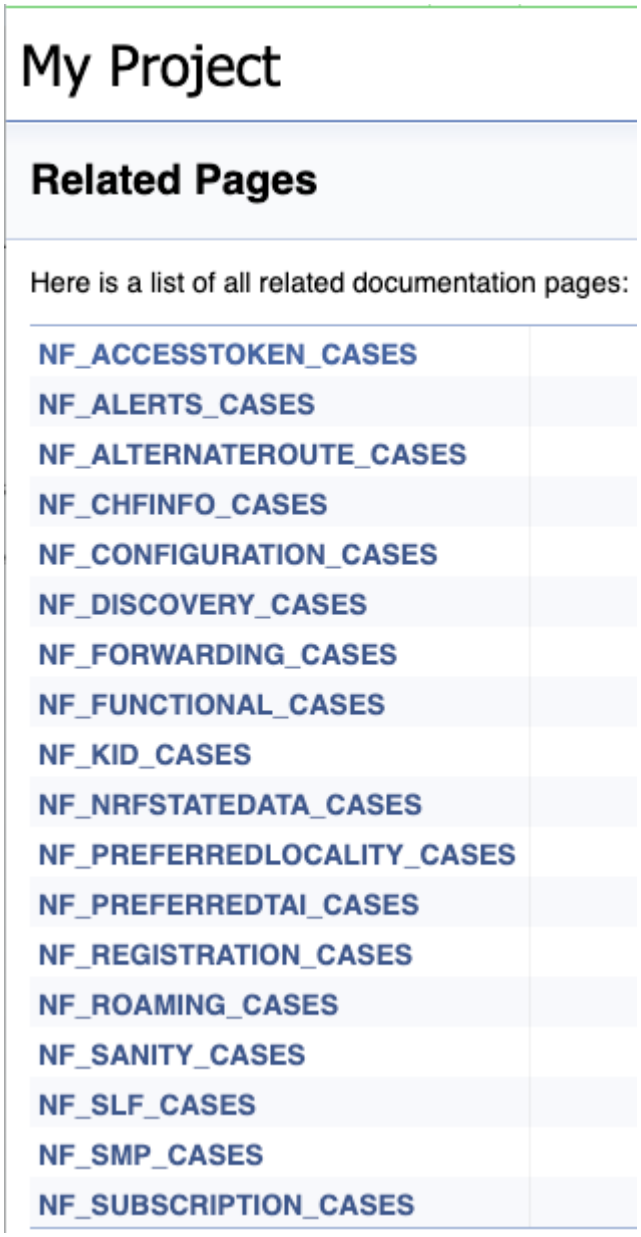
The NRF test cases are divided into multiple groups based on the functionality.

- **NF_ACCESSTOKEN_CASES** - All the test cases for validating accesstoken operation will be listed once this option is selected.
- **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_CHFINFO_CASES** - All the ChfInfo enhancement cases will be listed once this option is selected.
- **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_KID_CASES** - All the K-ID feature 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 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_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.
- **NF_SMP_CASES**- All the setting message priority cases will be listed once this option is selected.
- **NF_SUBSCRIPTION_CASES** - All subscription related cases will be listed once this option is selected.

The following screenshot shows all the NRF features:

Copyright © 2019, 2026, Oracle and/or its affiliates.

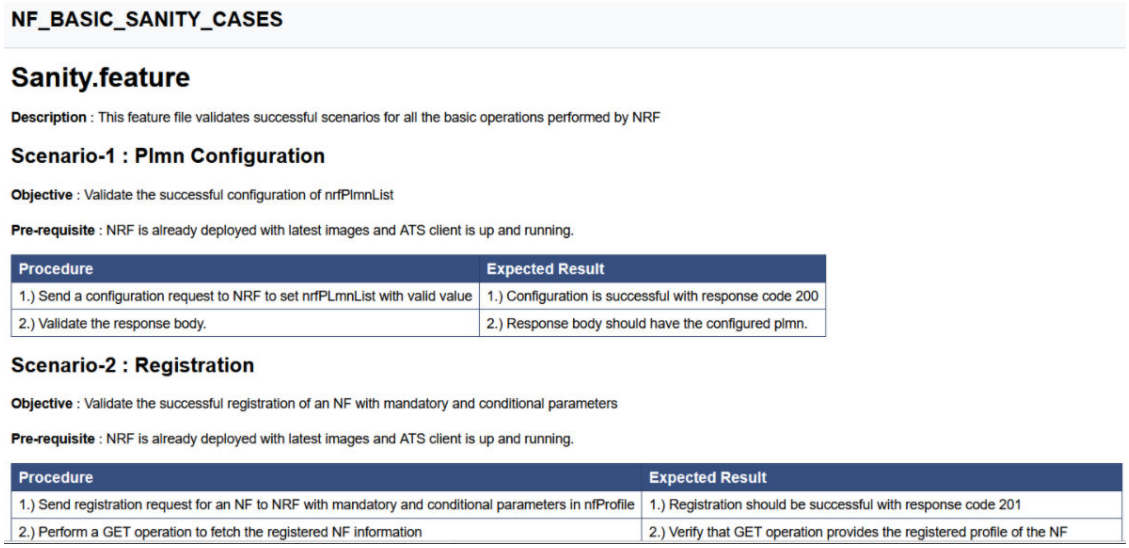
Figure 4-11 NRF-NewFeatures Documentation



Click any functionality to view its test cases and scenarios of each test case as shown below:

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 4-12 Sample Feature: NF_BASIC_SANITY_CASES



NF_BASIC_SANITY_CASES

Sanity.feature

Description : This feature file validates successful scenarios for all the basic operations performed by NRF

Scenario-1 : Plmn Configuration

Objective : Validate the successful configuration of nrfPlmnList

Pre-requisite : NRF is already deployed with latest images and ATS client is up and running.

Procedure	Expected Result
1.) Send a configuration request to NRF to set nrfPLmnList with valid value	1.) Configuration is successful with response code 200
2.) Validate the response body.	2.) Response body should have the configured plmn.

Scenario-2 : Registration

Objective : Validate the successful registration of an NF with mandatory and conditional parameters

Pre-requisite : NRF is already deployed with latest images and ATS client is up and running.

Procedure	Expected Result
1.) Send registration request for an NF to NRF with mandatory and conditional parameters in nfProfile	1.) Registration should be successful with response code 201
2.) Perform a GET operation to fetch the registered NF information	2.) Verify that GET operation provides the registered profile of the NF

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.

4.1.5 NRF-Regression Pipeline

This pre-configured pipeline contains all the test cases delivered till NRF ATS 26.1.201. 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:

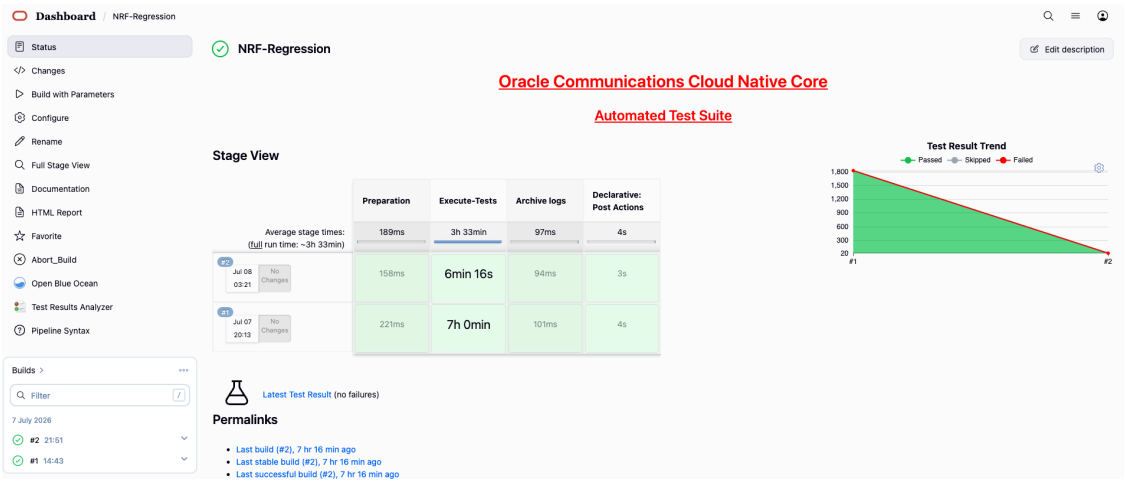
- **2-Site-Geo** - These feature files are listed with a prefix as **"Geo_Redundancy"**.
- **3-Site-Geo**- These feature files contain all the cases which are executed on 3 Site GEO NRF.
- **CCA-Header**- These feature files contain all cases related to CCA Header feature.
- **NF-FQDN-Authentication** - 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"**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **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.

Figure 4-13 NRF-Regression



The following screenshot shows successful execution as part of ATS image.

Figure 4-14 NRF-Regression - All-default-Regression

```
2026-07-08 13:52:11,062 [32m     INFO LOG.AllHooks:306  [0m|  [32mInside after_all [0m
2026-07-08 13:52:11,062 [32m     INFO LOG.AllHooks:420  [0m|  [32mExiting after_all [0m
655 features passed, 0 failed, 0 skipped
1825 scenarios passed, 0 failed, 0 skipped
29770 steps passed, 0 failed, 0 skipped
Took 240min 12.426s
[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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

4.1.6 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 26.1.201.

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.
- **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_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.

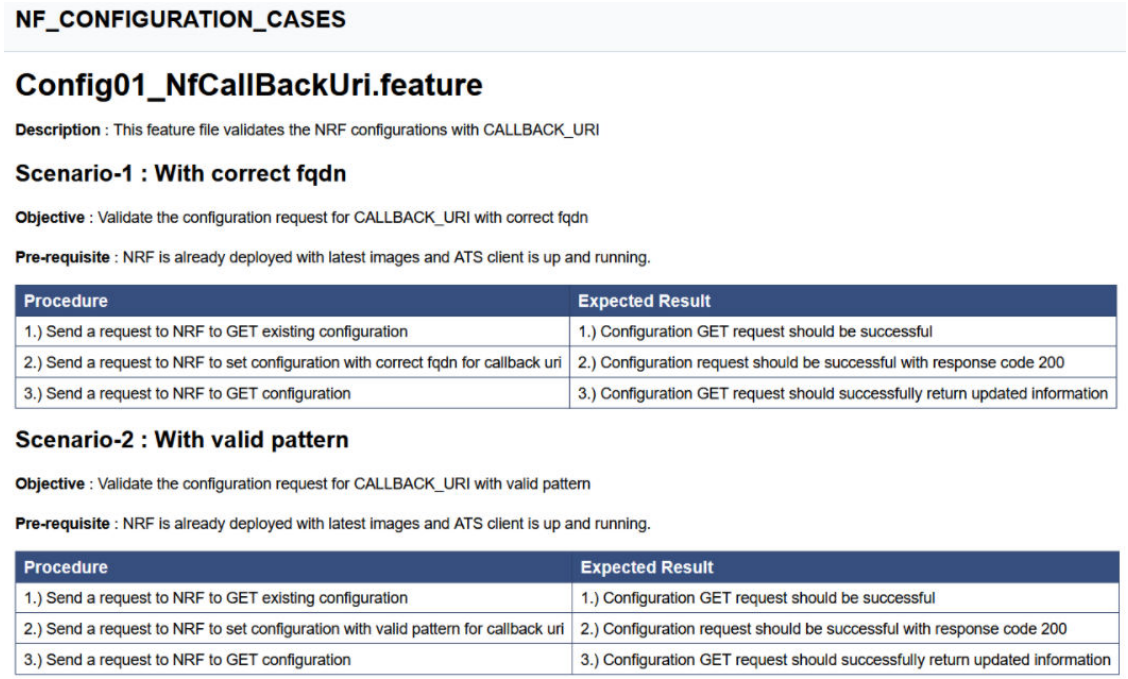
Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 4-15 NRF-Regression Documentation

The following sample screen shows documentation of Regression pipeline for NRF ATS.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 4-16 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

4.2 Running NSSF Test Cases using ATS

This section describes how to run NSSF test cases using ATS.

4.2.1 Prerequisites

The prerequisites to run NSSF Test Cases using NSSF ATS 26.1.201 are:

- Deploy NSSF 26.1.201 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 26.1.201, 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 26.1.201, 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 26.1.201, 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 26.1.201, 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 26.1.201, deploy one DNS stub server and the service name should be "ocdns-bind". It is required to run NRF-selection based on DNS SRV.

4.2.2 Logging into ATS

Before logging into ATS, deploy ATS using HELM charts as shown below:

Copyright © 2019, 2026, 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
```

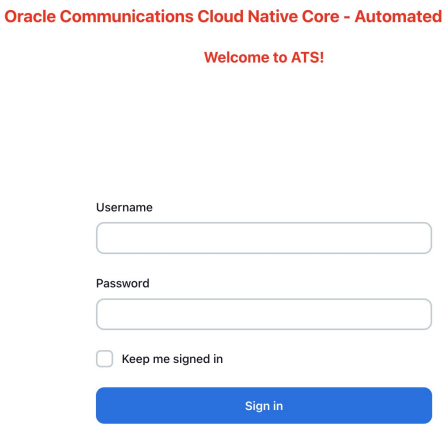
There are two ways to log in to ATS GUI.

- 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.



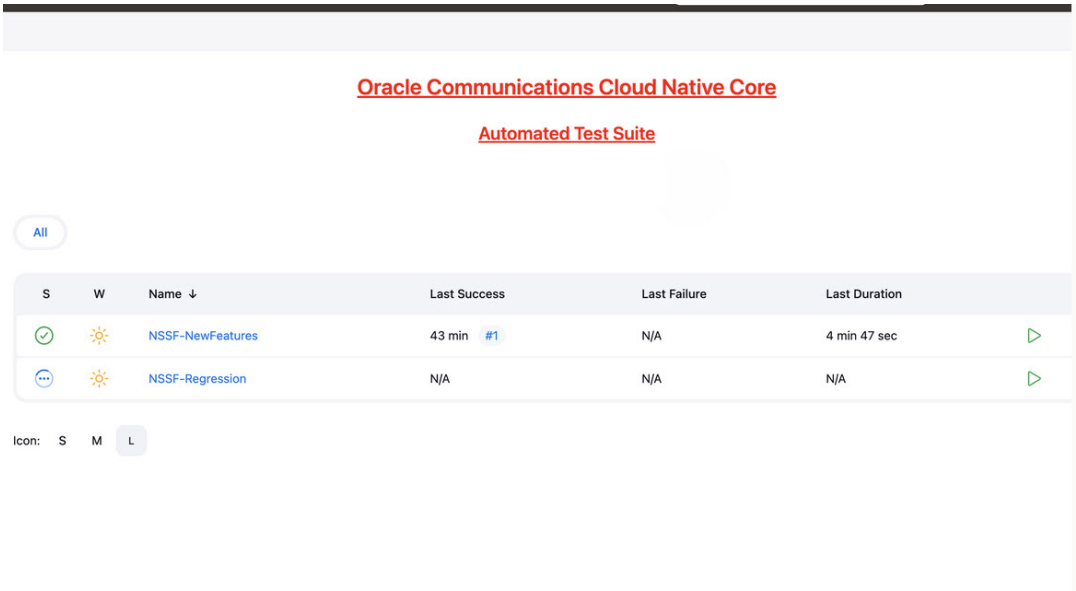
Copyright © 2019, 2026, Oracle and/or its affiliates.

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.



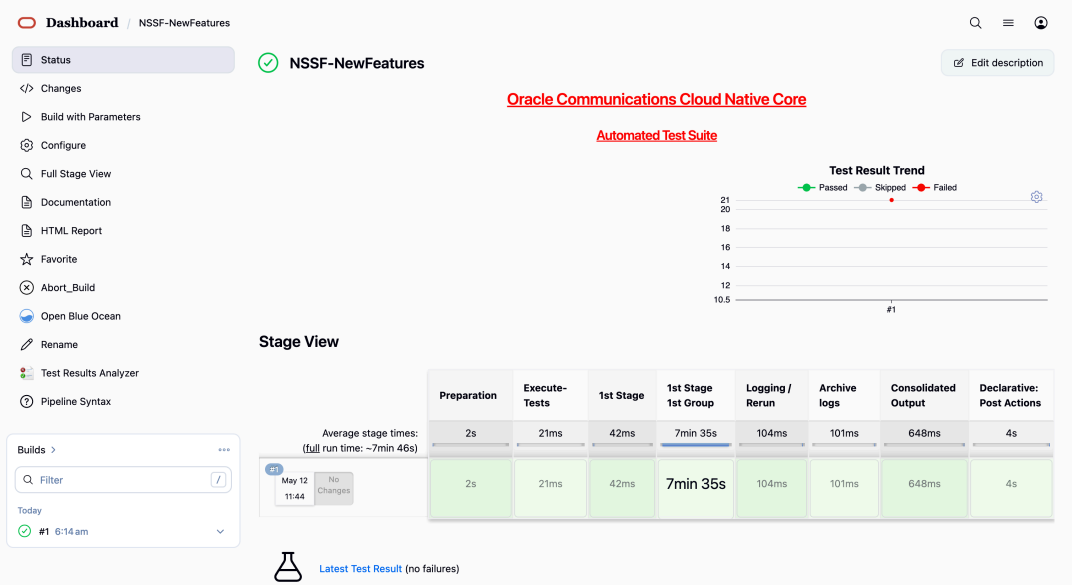
- **NSSF-New-Features:** This pipeline has all the test cases delivered as part of NSSF ATS - 26.1.201.
- **NSSF-Regression:** This pipeline has the test cases of all the previous releases.

4.2.3 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, 2026, 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.

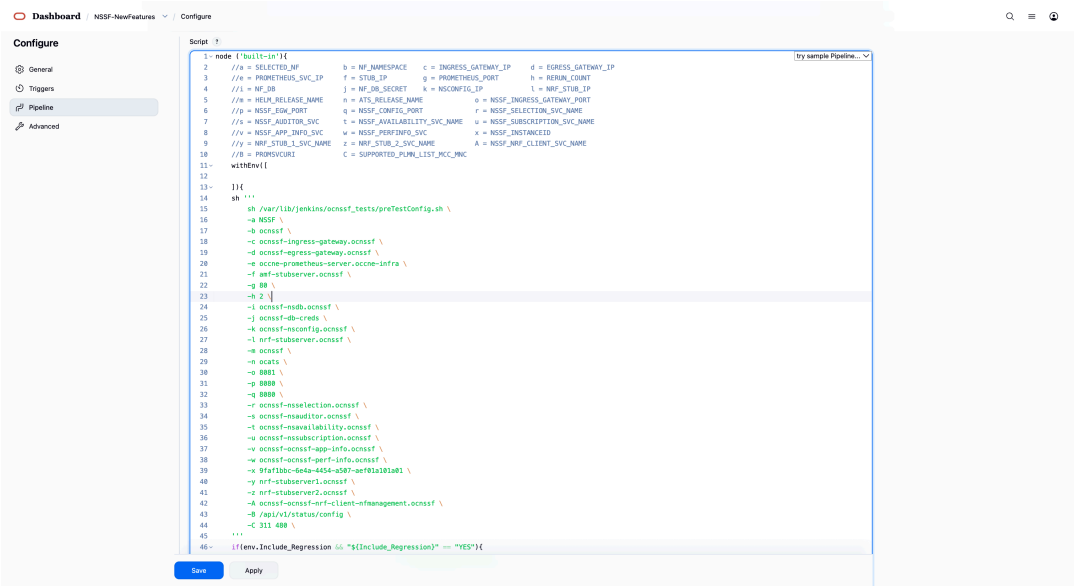
c. Click blue dots inside the **Build History** box to view the success console logs of the "All" and "Sanity" respectively.

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, 2026, 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') {

    // Parameter mappings:
    // 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 = PROMSVCURI
    // C = SUPPORTED_PLMN_LIST_MCC_MNC
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
    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 \
                -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 /api/v1/status/config \
                -C 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"

        }
    }
}
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

> **Note**
>
> The User MUST NOT change any other value apart from these parameters.

The descriptions of these parameters are as follows:

- **a:** Name of the NF to be tested in capital letters (`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 the Prometheus service (default is `80`).
- **h:** Number of times rerun of a failed case is allowed (default is `2`).
- **i:** Database name (default is `ocnssf-nsdb.ocnssf`).
- **j:** Database secrets (default is `ocnssf-db-creds`).
- **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 Supported PLMN.
- **A:** NRF Client Management Service Name (`ocnssf-ocnssf-nrf-client-nfmanagement.ocnssf`).
- **B:** PROMSVCURI (`/api/v1/status/config`).
- **C:** NSSF PLMN List (`311 480`).

Copyright © 2019, 2026, Oracle and/or its affiliates.

Note

Note

- The variable **B** is `promsvcurl`, and its current value is `/api/v1/status/config`.
- In other environments, if `promsvcurl` is set to `/prometheus/api/v1/status/config`, it should be updated to `/prometheus/api/`, as the remaining path (`v1/status/config`) will be appended automatically.
- Do not change any value if the OCCNE cluster is used and NSSF, ATS, and STUB are deployed in the `ocnssf` namespace.
- In the image above, the NSSF Helm release name is `ocnssf`, the ATS Helm release name is `ocats`, and the namespace is `ocnssf`. If there are any changes to the Helm release name of NSSF, ATS, or the namespace, update the configuration accordingly.

For example, if the NSSF Helm release name is `ocnssfats`, the ATS Helm release name is `ocatsnssf`, and the namespace is `ocnssf2510`, then the Pipeline Configuration should be edited as shown below.

- The NSSF Helm release name should be updated in `c`, `d`, `i`, `k`, `m`, `r`, `s`, `t`, `u`, `v`, and `w` if the NSSF Helm release name is anything other than `ocnssf`.
- The NSSF Instance ID must 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 change must be reflected in the Jenkins pipeline parameters.

By default, the NSSF Instance ID in the NSSF custom values file is as follows:

```
#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 changed from `9faf1bbc-6e4a-4454-a507-aef01a101a01` to `9faf1bbc-6e4a-4454-a507-aef01a101a20`, the same change must be updated in the `x` Jenkins pipeline parameter 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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
        // o = NSSF_INGRESS_GATEWAY_PORT
        // p = NSSF_EGW_PORT
        // q = NSSF_CONFIG_PORT
        // r = NSSF_SELECTION_SVC_NAME
        // s = NSSF_AUDITOR_SVC_NAME
        // 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 = PROMSVCURI
        // C = 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 \
                -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 /prometheus \
                -C 311 480
            '''

            if (env.Include_Regression && "${Include_Regression}" ==
"YES") {

                sh '''
                sh /var/lib/jenkins/common_scripts/merge_jenkinsfile.sh
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

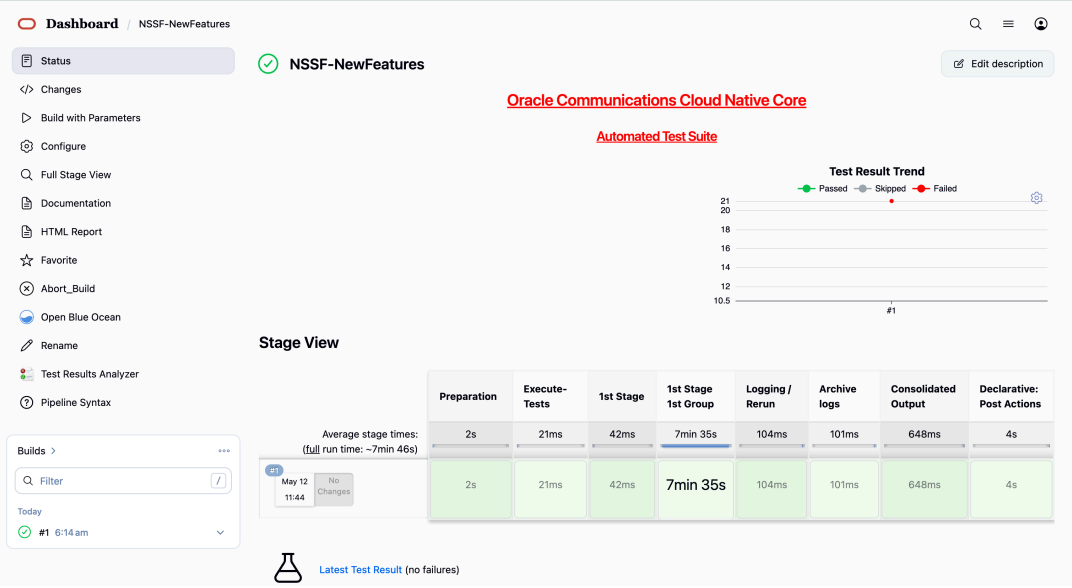
```
                '''

                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

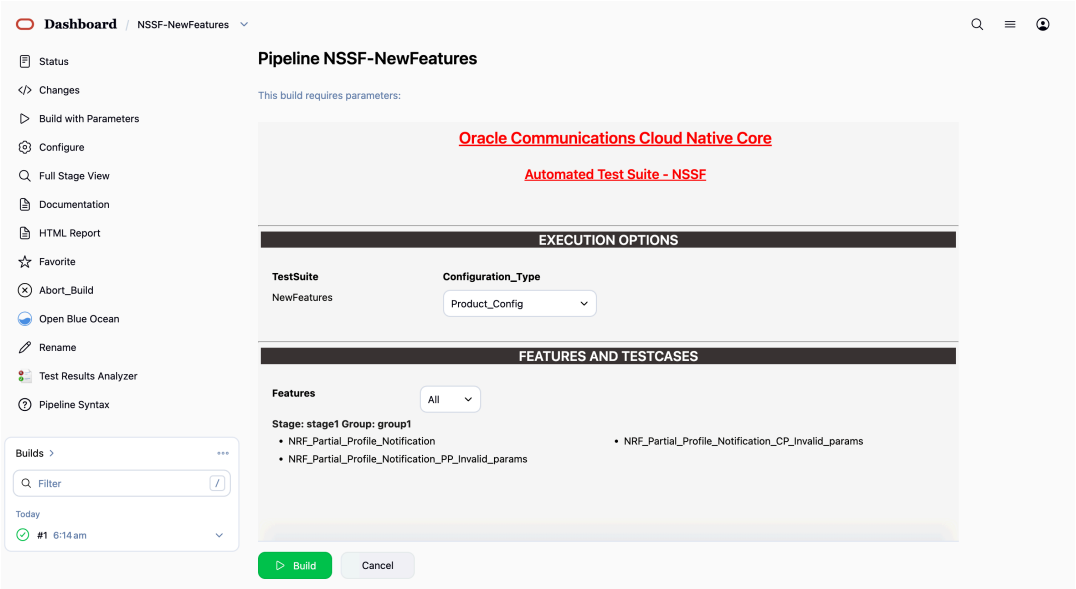
To run NSSF New Features test cases:

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:



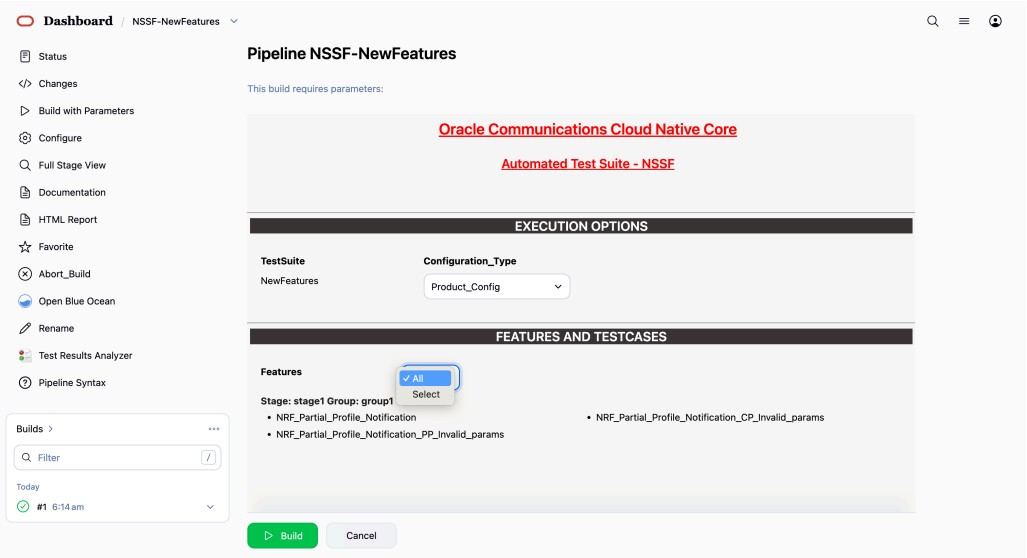
2. The following screen appears:

Copyright © 2019, 2026, Oracle and/or its affiliates.

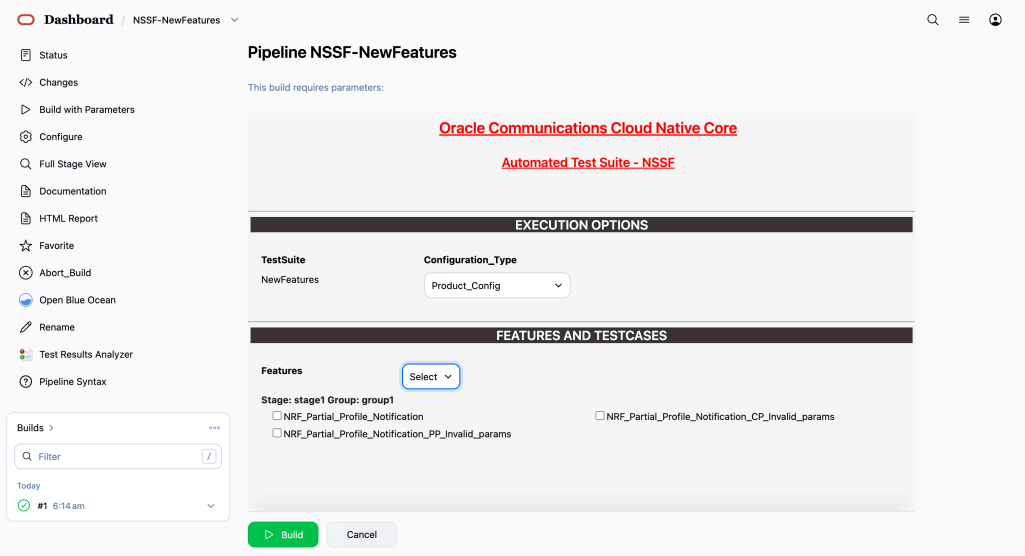


In the above screen, there are three **Select_Option(s)**, which are:

- 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, 2026, Oracle and/or its affiliates.



3. Select one of the following configuration types:

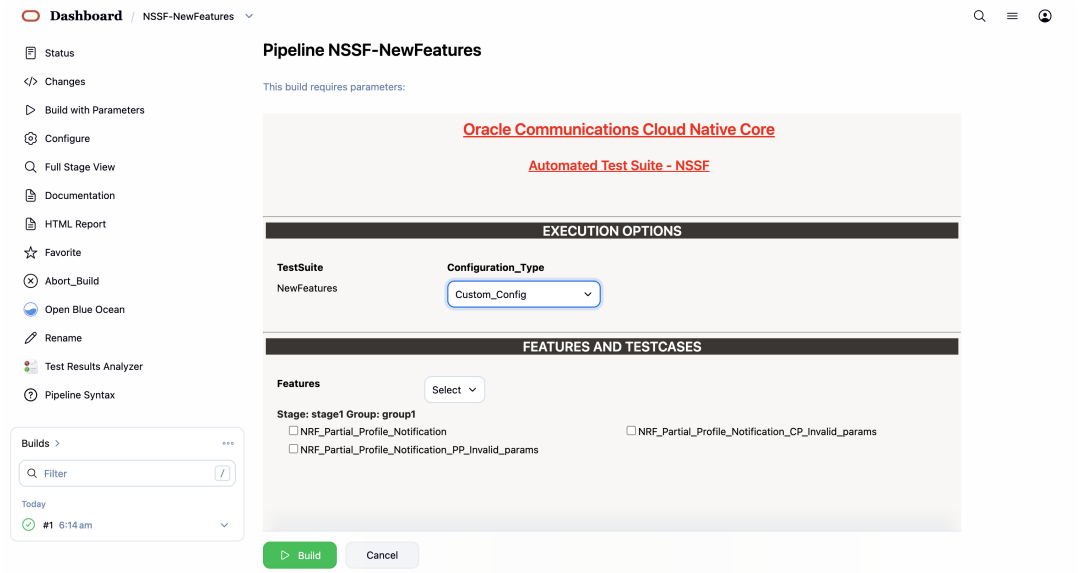
   - **Product_Config**: On selecting this option, test cases from product folders are populated on ATS UI and product configuration is applied to them via the keyvalue pair and yaml files defined or present in the "Product Config" folder.
   - **Custom_Config**: On selecting this option, test cases from 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:

   - `NRF_Partial_Profile_Notification` — This feature file contains test cases related to NRF partial profile notification.
   - `NRF_Partial_Profile_Notification_PP_Invalid_params` — This feature file contains test cases related to NRF partial profile invalid parameters.
   - `NRF_Partial_Profile_Notification_CP_Invalid_params` — This feature file contains test cases related to NRF complete profile invalid parameters.

   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.

Copyright © 2019, 2026, Oracle and/or its affiliates.




```
3 features passed, 0 failed, 0 skipped
21 scenarios passed, 0 failed, 0 skipped
484 steps passed, 0 failed, 0 skipped
Took 7min 5.486s
```

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.

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.

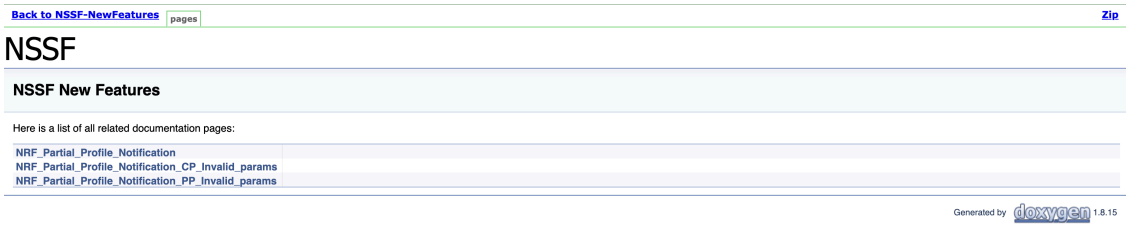
Copyright © 2019, 2026, Oracle and/or its affiliates.

- Values for parameters tagged with #START_GLOBAL and #END_GLOBAL tags take values from global.yaml.

> **Note**
>
> For NSSF-ATS release 26.1.201, parameterization is supported for NSSF NewFeatures pipeline only.

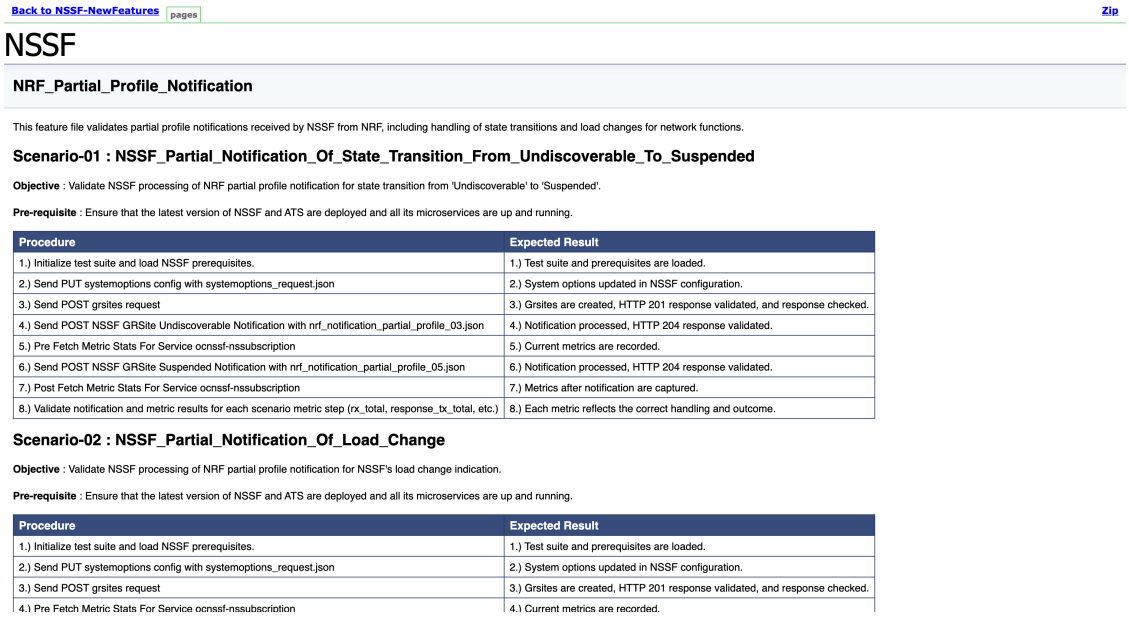
4.2.4 NSSF-NewFeatures Documentation

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, 2026, Oracle and/or its affiliates.



Back to NSSF-NewFeatures pages Zip

NSSF

NRF_Partial_Profile_Notification

This feature file validates partial profile notifications received by NSSF from NRF, including handling of state transitions and load changes for network functions.

Scenario-01 : NSSF_Partial_Notification_Of_State_Transition_From_Undiscoverable_To_Suspended

Objective : Validate NSSF processing of NRF partial profile notification for state transition from 'Undiscoverable' to 'Suspended'.

Pre-requisite : Ensure that the latest version of NSSF and ATS are deployed and all its microservices are up and running.

Procedure	Expected Result
1.) Initialize test suite and load NSSF prerequisites.	1.) Test suite and prerequisites are loaded.
2.) Send PUT systemoptions config with systemoptions_request.json	2.) System options updated in NSSF configuration.
3.) Send POST grsites request	3.) Grsites are created, HTTP 201 response validated, and response checked.
4.) Send POST NSSF GRSite Undiscoverable Notification with nrf_notification_partial_profile_03.json	4.) Notification processed, HTTP 204 response validated.
5.) Pre Fetch Metric Stats For Service ocnssf-nssubscription	5.) Current metrics are recorded.
6.) Send POST NSSF GRSite Suspended Notification with nrf_notification_partial_profile_05.json	6.) Notification processed, HTTP 204 response validated.
7.) Post Fetch Metric Stats For Service ocnssf-nssubscription	7.) Metrics after notification are captured.
8.) Validate notification and metric results for each scenario metric step (rx_total, response_tx_total, etc.)	8.) Each metric reflects the correct handling and outcome.

Scenario-02 : NSSF_Partial_Notification_Of_Load_Change

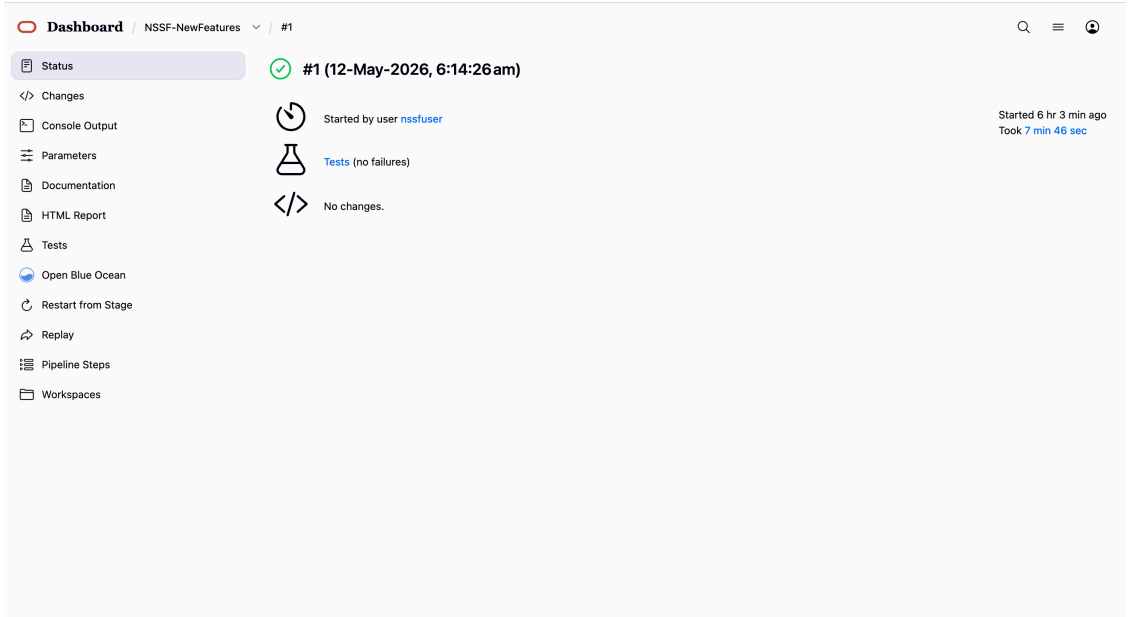
Objective : Validate NSSF processing of NRF partial profile notification for NSSF's load change indication.

Pre-requisite : Ensure that the latest version of NSSF and ATS are deployed and all its microservices are up and running.

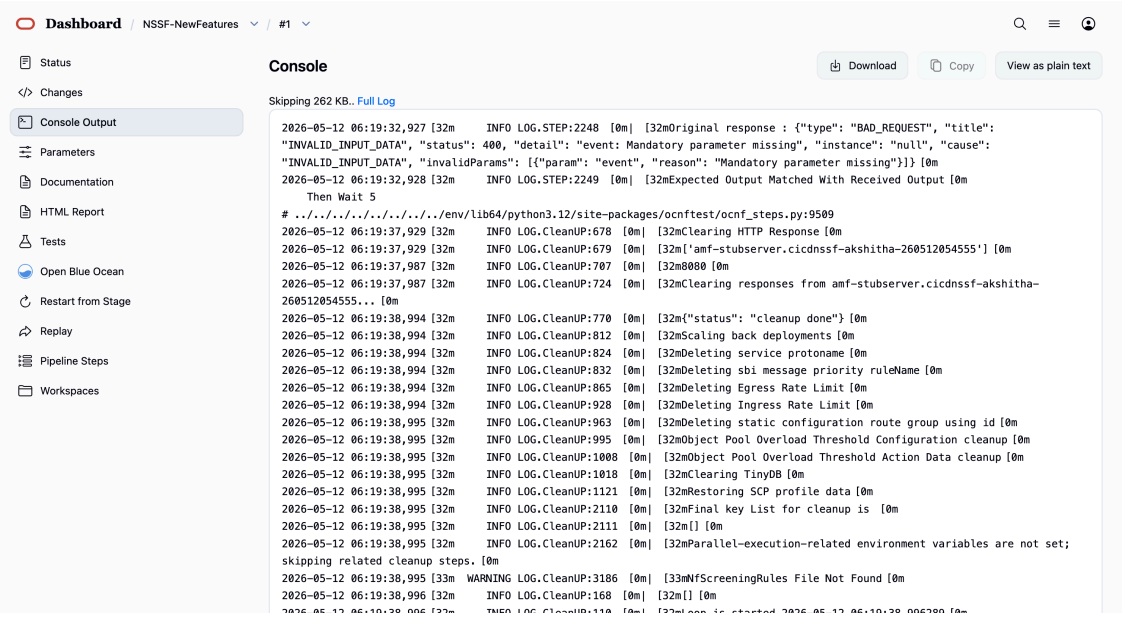
Procedure	Expected Result
1.) Initialize test suite and load NSSF prerequisites.	1.) Test suite and prerequisites are loaded.
2.) Send PUT systemoptions config with systemoptions_request.json	2.) System options updated in NSSF configuration.
3.) Send POST grsites request	3.) Grsites are created, HTTP 201 response validated, and response checked.
4.) Pre Fetch Metric Stats For Service ocnssf-nssubscription	4.) Current metrics are recorded.

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, 2026, Oracle and/or its affiliates.



4.2.5 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:

- **NSSF_OAuth_Response_Header**: Validates OAuth 2.0-related HTTP error codes and response headers for invalid, expired, or insufficient-scope OAuth tokens.
- **NSSF_SCP_Monitoring_OPTIONS**: Validates SCP peer monitoring and routing behavior, including configuration, `healthPath` enforcement, priorities, and error handling for SBI and indirect-routing scenarios.
- **NSSF_Sending_Notification_via_ARS_SCP**: Validates NSSF's ability to route and send notifications to SCPs through ARS, including scenarios involving single and multiple SCPs, priorities, virtual hosts, alternate routes, and associated error handling.
- **NSSF_Server_Header**: Validates the Server Header feature.
- **NSSF_User_Agent_Header**: Validates User-Agent header behavior when the feature is enabled and disabled.
- **oauth_signValidationServiceMeshEnabled_false**: Validates CRUD operations on OAuth configuration in the NSSF Ingress Gateway.
- **Virtual_Host_NRF_Resolution_By_NSSF_Using_DNSSRV**: Validates virtual-host NRF resolution by NSSF using DNS SRV records.
- **NSSF_Sanity**: Validates NSSF sanity test cases.
- **NSSF_NSAvailability_OPTIONS**: Validates NSSF responses to OPTIONS requests for supported communication methods.
- **Nnssf_NSAvailability_Invalid_Accept_ContentType_Header**: Validates rejection of NSAvailability requests with invalid or missing Accept and Content-Type headers.

Copyright © 2019, 2026, Oracle and/or its affiliates.

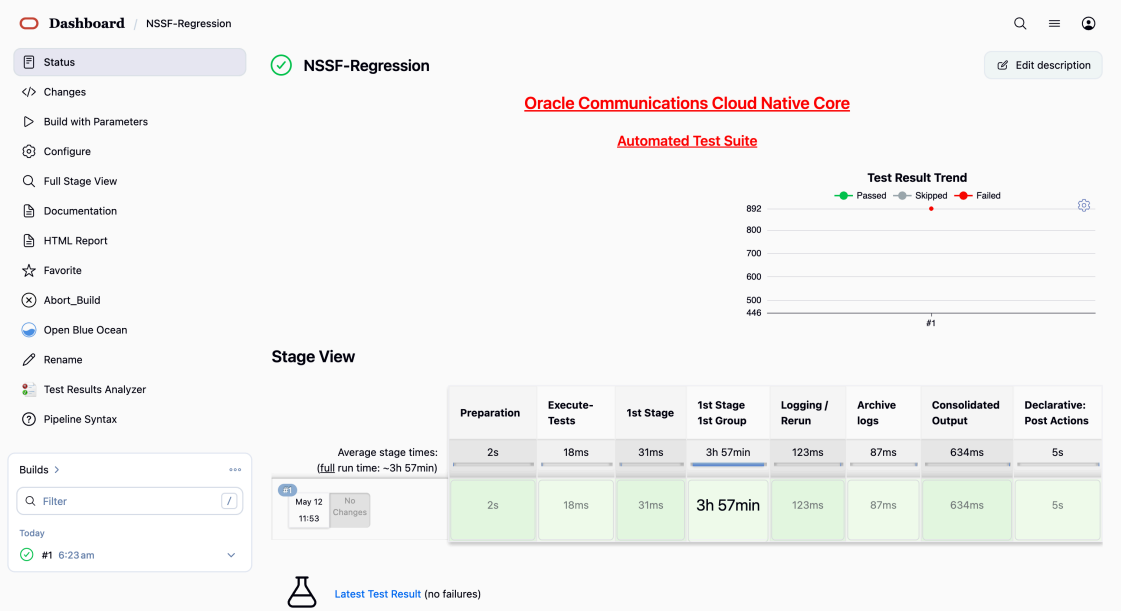
- **Nnssf_Nssai_Availability_patch_remove_204_NoContent**: Validates PATCH operations with and without auto-configuration, including enhanced PATCH behavior and expected `204 No Content` responses.
- **Nnssf_Nssai_Availability_patch_with_autoconfiguration**: Validates PATCH operations on NS availability resources with auto-configuration enabled.
- **Nnssf_Nssai_Availability_patch_without_autoconfiguration**: Validates PATCH operations on NS availability resources with auto-configuration disabled.
- **Nnssf_Nssai_Availability_put_delete_with_autoconfiguration**: Validates PUT and DELETE operations on NS availability resources with auto-configuration enabled.
- **Nnssf_Nssai_Availability_put_delete_without_autoconfiguration**: Validates PUT and DELETE operations on NS availability resources with auto-configuration disabled.
- **Nnssf_Nssai_Availability_put_patch_delete_error_scenarios**: Validates error scenarios for PUT, PATCH, and DELETE operations on NS availability resources with auto-configuration enabled and disabled.
- **Nnssf_Nssai_Availability_put_patch_with_supported_features**: Validates PUT and PATCH operations on NS availability resources with supported features enabled.
- **Nnssf_NsConfig_Invalid_Accept_ContentType_Header**: Validates rejection of NSConfig requests with invalid or missing Accept and Content-Type headers.
- **barred_slices_mapping**: Validates barred-slices mapping operations in NSConfig.
- **grsites**: Validates GRSites operations in NSConfig.
- **nsi_profiles**: Validates NSI profile operations in NSConfig.
- **plmn_config**: Validates PLMN configuration operations in NSConfig.
- **supported_slices_mapping**: Validates supported-slices mapping operations in NSConfig.
- **system_options**: Validates system-options operations in NSConfig.
- **MultiplePLMN**: Validates multiple-PLMN NSSelection scenarios.
- **NSSF_Unknown_URI_Unsupported_Method**: Validates behavior for unknown URIs and unsupported HTTP methods.
- **Nnssf_NSSelection_Initial_Reg_AutoConfiguration_OFF_EnhancedNssai_OFF**: Validates initial-registration NSSelection behavior with auto-configuration disabled and enhanced NSSAI disabled.
- **Nnssf_NSSelection_Initial_Reg_AutoConfiguration_OFF_EnhancedNssai_ON**: Validates initial-registration NSSelection behavior with auto-configuration disabled and enhanced NSSAI enabled.
- **Nnssf_NSSelection_Initial_Reg_AutoConfiguration_ON_EnhancedNssai_OFF**: Validates initial-registration NSSelection behavior with auto-configuration enabled and enhanced NSSAI disabled.
- **Nnssf_NSSelection_Initial_Reg_AutoConfiguration_ON_EnhancedNssai_ON**: Validates initial-registration NSSelection behavior when both `autoAuthorizeNssaiAvailabilityDataEnable` and `enhancedAllowedNssaiEnable` are enabled.
- **Nnssf_NSSelection_Initial_Reg_Invalid_Query_Parameters**: Validates handling of invalid query parameters during initial-registration NSSelection.
- **Nnssf_NSSelection_Invalid_Accept_ContentType_Header**: Validates rejection of NSSelection GET requests with invalid or missing Accept and Content-Type headers.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **Nnssf_NSSelection_PDU_AutoConfiguration_OFF**: Validates PDU session NSSelection behavior with auto-configuration disabled.
- **Nnssf_NSSelection_PDU_AutoConfiguration_ON**: Validates PDU session NSSelection behavior with auto-configuration enabled.
- **Nnssf_NSSelection_PDU_Miscellaneous_AutoConfiguration_OFF**: Validates miscellaneous PDU session NSSelection scenarios with auto-configuration disabled, including session establishment and error handling.
- **Nnssf_NSSelection_PDU_Miscellaneous_AutoConfiguration_ON**: Validates miscellaneous PDU session NSSelection scenarios with auto-configuration enabled, including configuration changes and error handling.
- **Nnssf_NSSelection_PDU_Session_Invalid_Query_Parameters**: Validates handling of invalid query parameters in PDU session NSSelection requests.
- **Nnssf_NSSelection_UE_CU_AutoConfiguration_OFF_EnhancedNssai_OFF**: Validates UE/CU NSSelection behavior when both `autoAuthorizeNssaiAvailabilityDataEnable` and `enhancedAllowedNssaiEnable` are disabled.
- **Nnssf_NSSelection_UE_CU_AutoConfiguration_OFF_EnhancedNssai_ON**: Validates UE/CU NSSelection behavior with auto-configuration disabled and enhanced NSSAI enabled.
- **Nnssf_NSSelection_UE_CU_AutoConfiguration_ON_EnhancedNssai_OFF**: Validates UE/CU NSSelection behavior with auto-configuration enabled and enhanced NSSAI disabled.
- **Nnssf_NSSelection_UE_CU_AutoConfiguration_ON_EnhancedNssai_ON**: Validates UE/CU NSSelection behavior when both auto-configuration and enhanced NSSAI are enabled.
- **Nnssf_NSSelection_UE_CU_Invalid_Query_Parameters**: Validates handling of invalid query parameters in UE/CU NSSelection requests.
- **NSSF_Subscription_Delete**: Validates deletion operations and error handling for NSSubscription resources.
- **Nnssf_NSSubscription_Indirect_Communication**: Validates indirect communication behavior for the NSSubscription service.
- **Nnssf_NSSubscription_Indirect_Communication_Negative**: Validates negative indirect-communication scenarios involving incorrect `3gpp-Sbi-Binding` headers.
- **Nnssf_NSSubscription_Invalid_Accept_ContentType_Header**: Validates subscription request handling with invalid or missing Accept and Content-Type headers.
- **Nnssf_NSSubscription_Notification_Auto_Config_OFF**: Validates NSSubscription notification behavior when auto-configuration is disabled.
- **Nnssf_NSSubscription_Notification_Auto_Config_ON**: Validates NSSubscription notification behavior when auto-configuration is enabled.
- **Nnssf_NSSubscription_Notification_Response_Handling**: Validates NSSubscription notification handling based on various AMF stub responses.
- **Nnssf_NSSubscription_Post**: Validates CRUD operations and request handling for NSSubscription POST requests.
- **Nnssf_NSSubscription_Subscription_Invalid_Params**: Validates CRUD operations and parameter validation for NSSubscription requests.
- **Nnssf_NSSubscription_Subscription_Update**: Validates CRUD operations and input validation for NSSubscription update requests.

Copyright © 2019, 2026, Oracle and/or its affiliates.

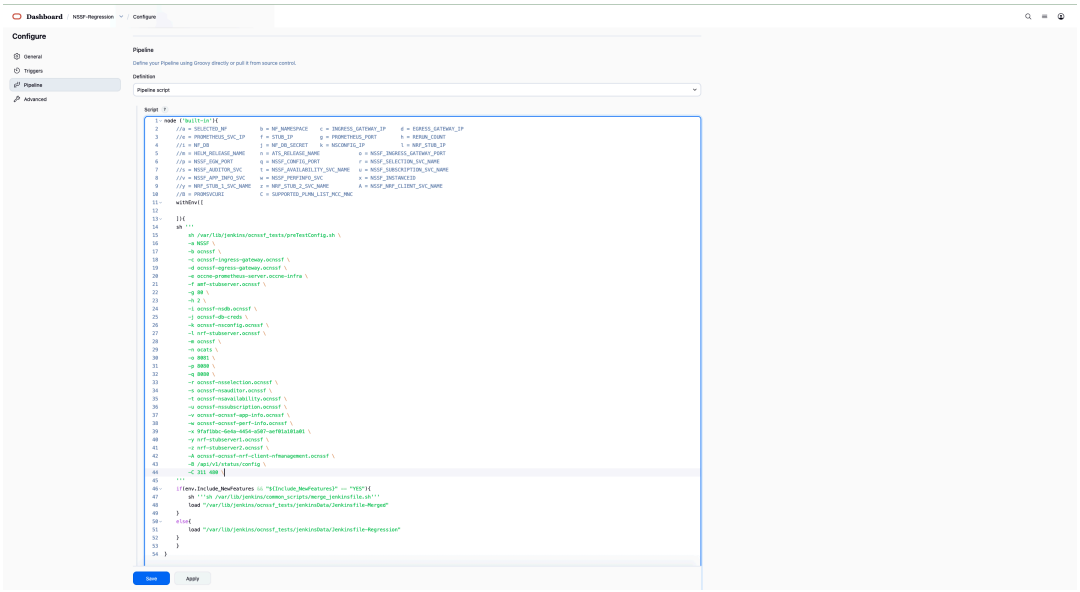
- **Nnssf_NSSubscription_post_with_supported_features**: Validates NSSubscription POST request handling with `supportedFeatures` values.



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, 2026, 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 = PROMSVCURI
    // C = SUPPORTED_PLMN_LIST_MCC_MNC

    withEnv([]) {
        sh '''
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
                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 \
                    -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 /api/v1/status/config \
                    -C 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)

Copyright © 2019, 2026, Oracle and/or its affiliates.

- 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)
- 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: PROMSVCURI (/api/v1/status/config)
- C: NSSF PLMN List (311 480)

Copyright © 2019, 2026, 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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

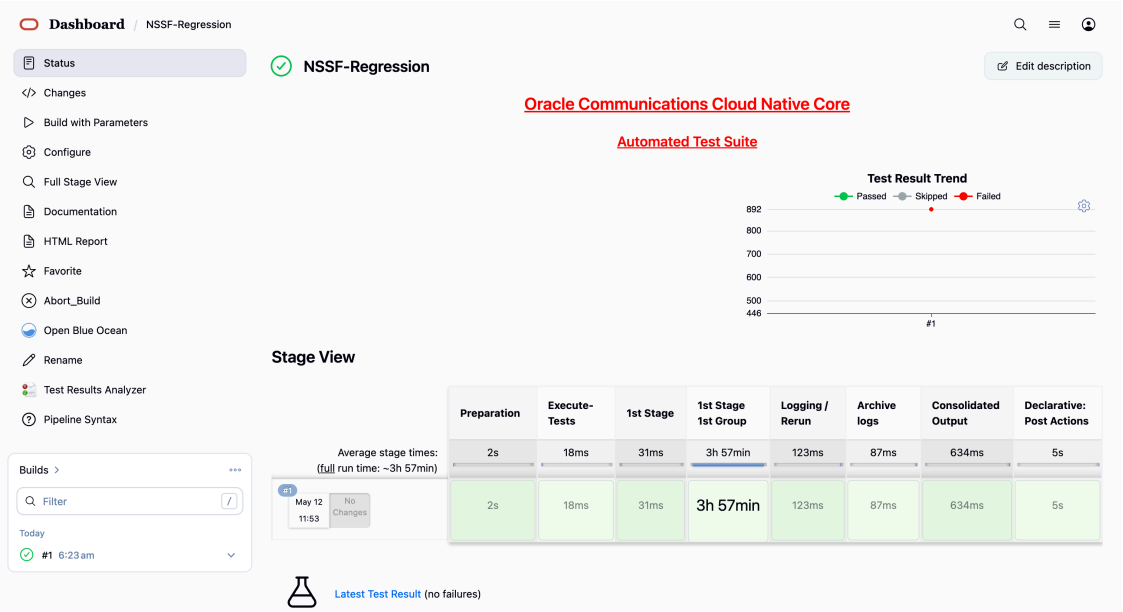
```
    // x = NSSF_INSTANCEID
    // y = NRF_STUB_1_SVC_NAME
    // z = NRF_STUB_2_SVC_NAME
    // A = NSSF_NRF_CLIENT_SVC_NAME
    // B = SUPPORTED_PLMN_LIST_MCC_MNC

    withEnv([]) {
        sh '''
            sh /var/lib/jenkins/ocnssf_tests/preTestConfig.sh \
                -a NSSF \
                -b 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 /prometheus \
                -C 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"
        }
```

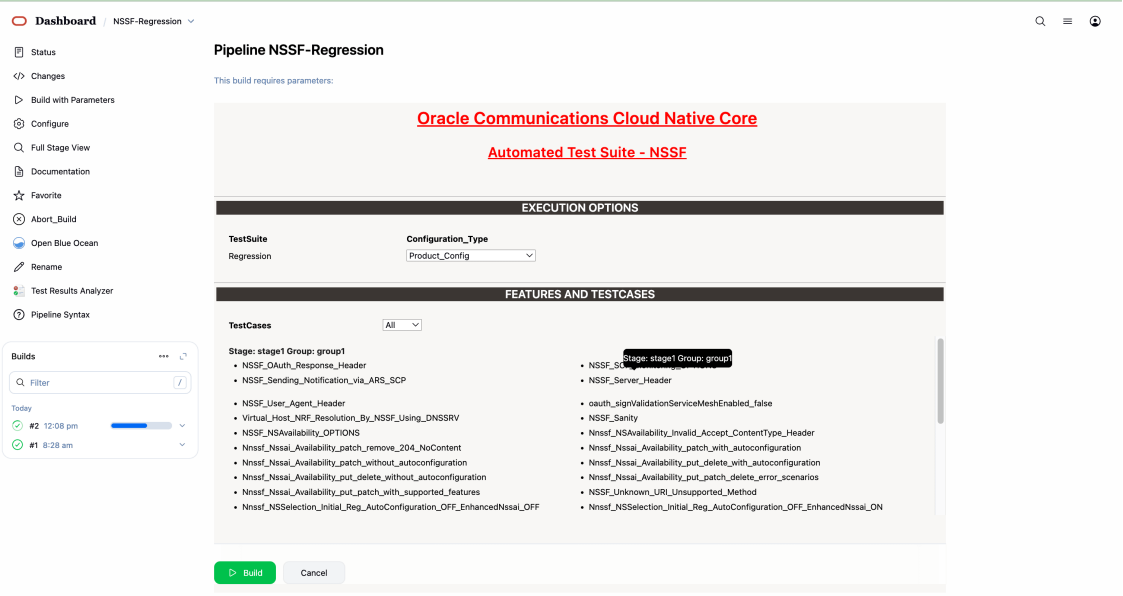

Copyright © 2019, 2026, Oracle and/or its affiliates.

```
        }
    }
```

4. Click **Save** after making necessary changes. The **Pipeline NSSF-Regression** screen appears.

NSSF-Regression - Build with Parameters

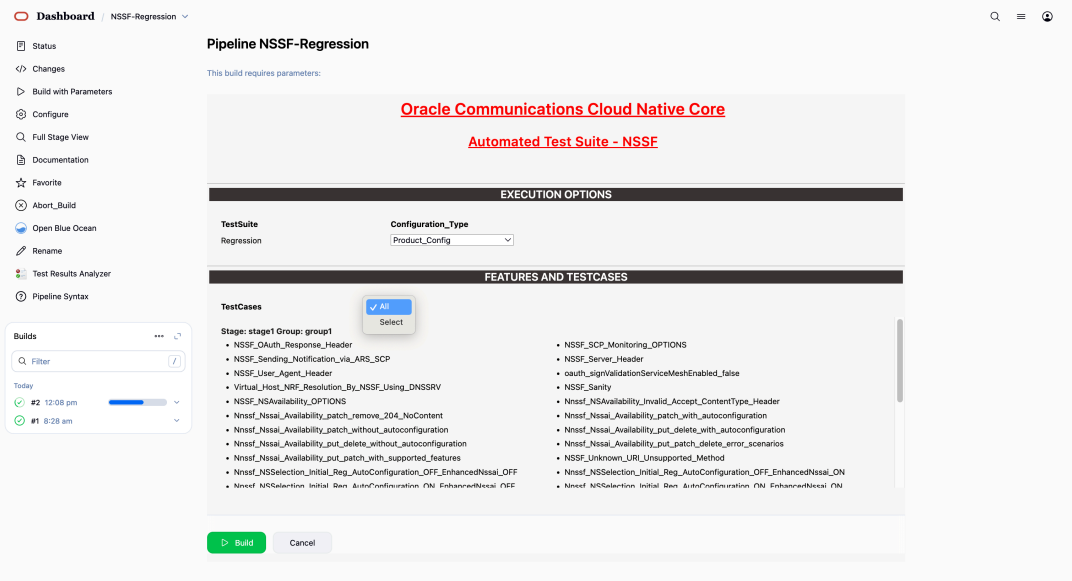


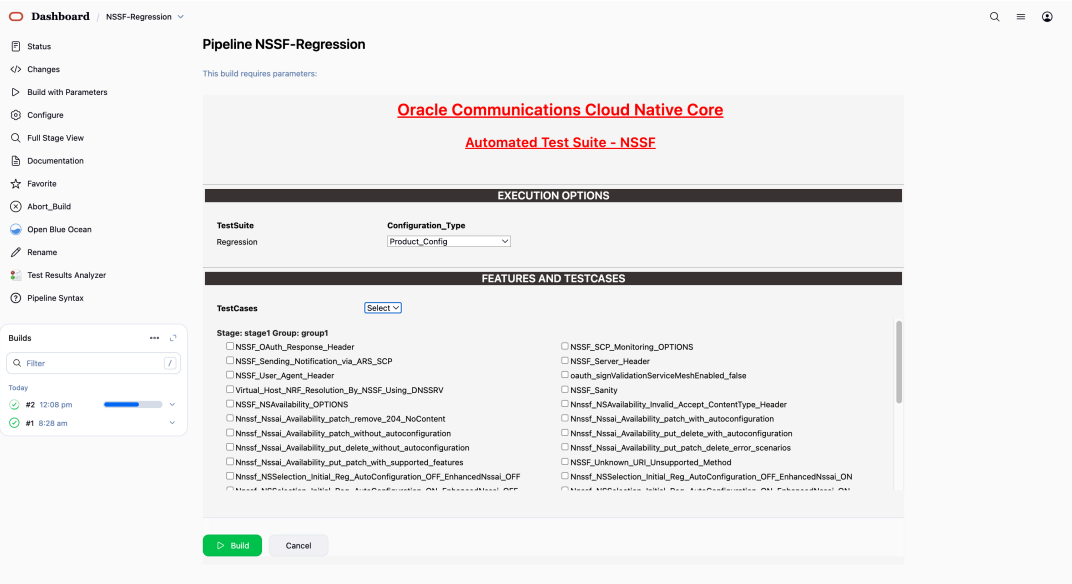


- By default, features "ALL," configuration type "Product_config," and include new features "NO" will be selected. All test cases will be executed once you click on BUILD, as shown above.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- 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.



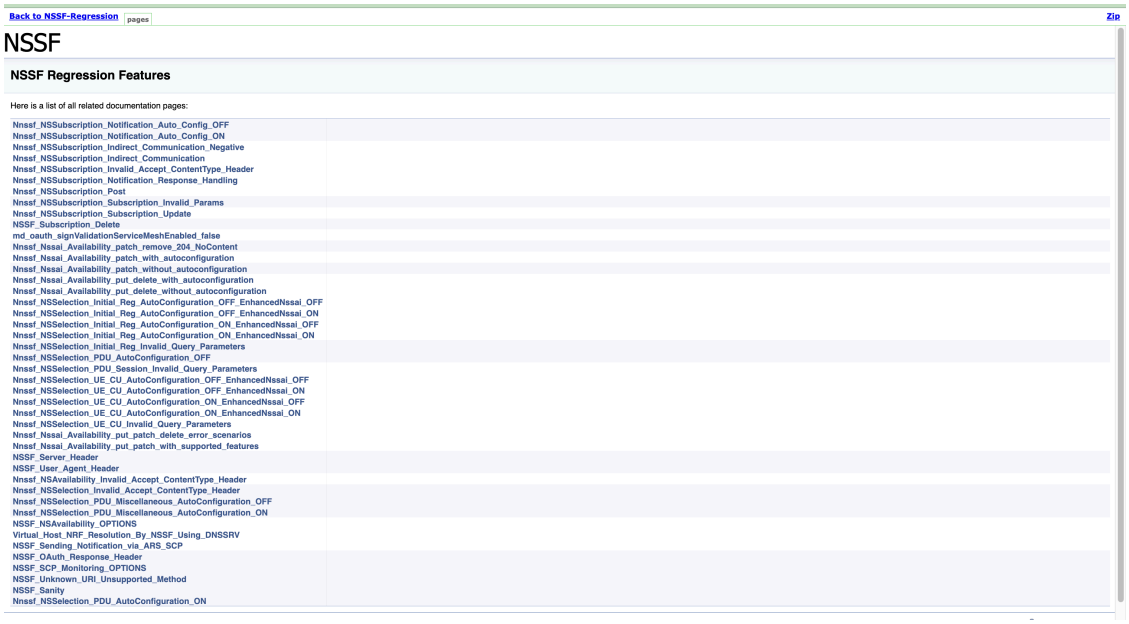



```
53 features passed, 0 failed, 0 skipped
892 scenarios passed, 0 failed, 0 skipped
17792 steps passed, 0 failed, 0 skipped
Took 229min 51.609s
```

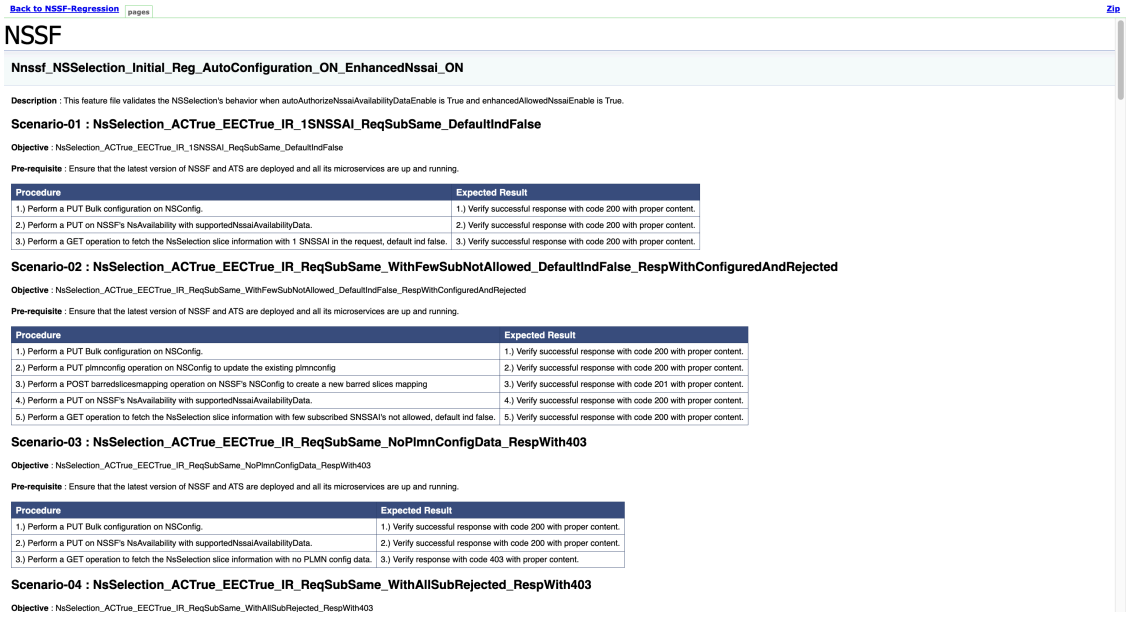

Copyright © 2019, 2026, Oracle and/or its affiliates.

4.2.6 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:



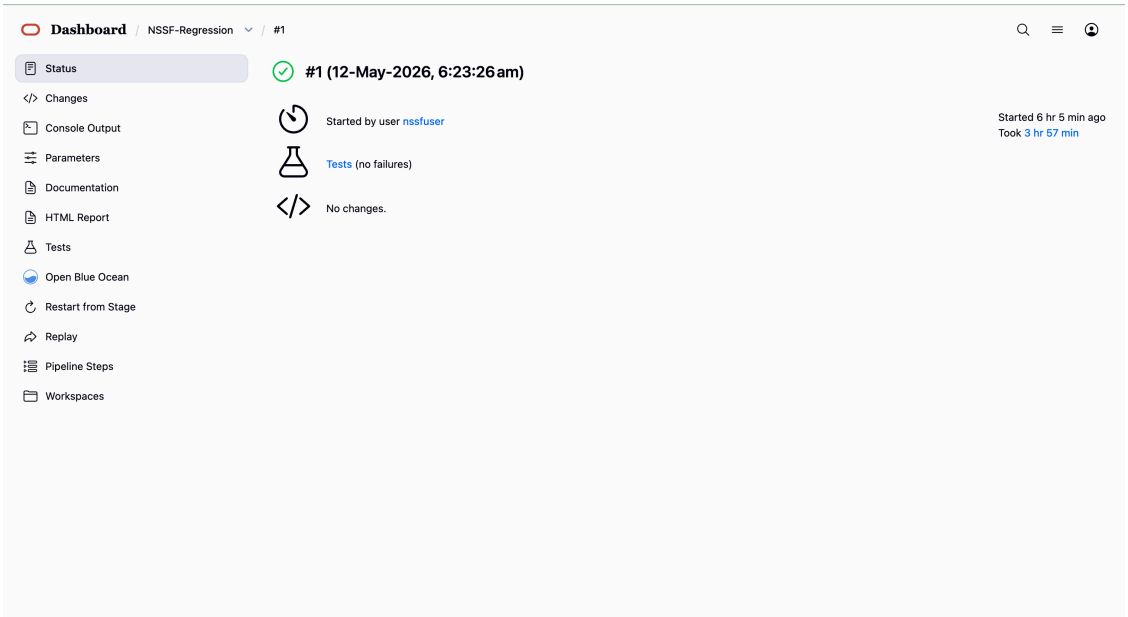
A sample of a few documentation features are as follows:

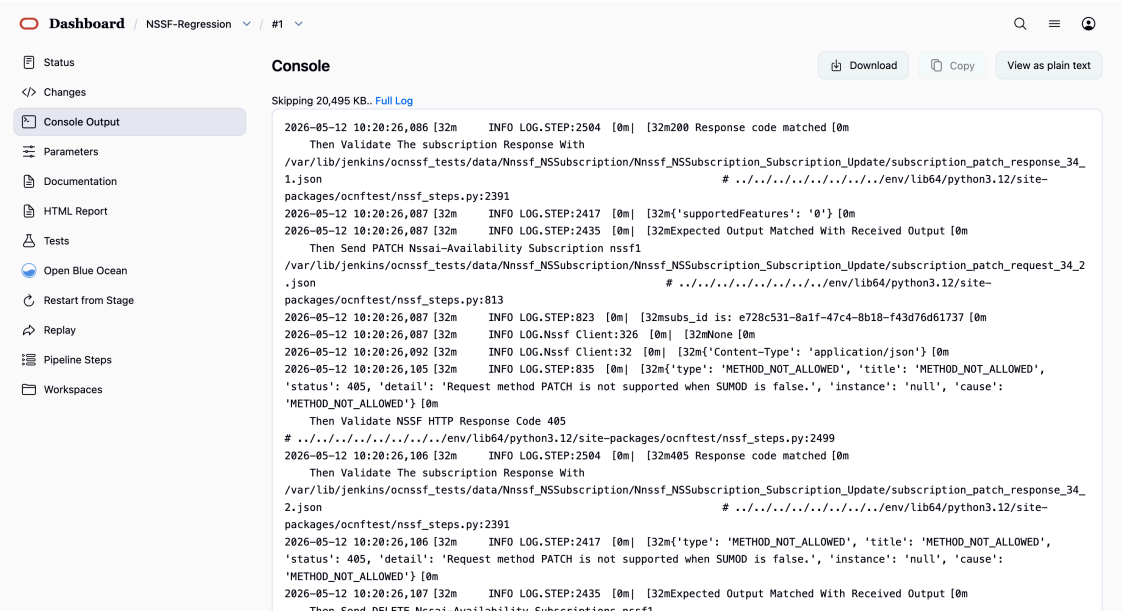


Once the run of regression features is done successfully, click on the build number whose logs are to be viewed and then click on **Console Output** on left side navigation.

Wait till the page is loaded completely, then click **Download** to download the regression features logs.

Copyright © 2019, 2026, Oracle and/or its affiliates.





4.3 Running SCP Test Cases using ATS

This section describes how to run SCP test cases using ATS.

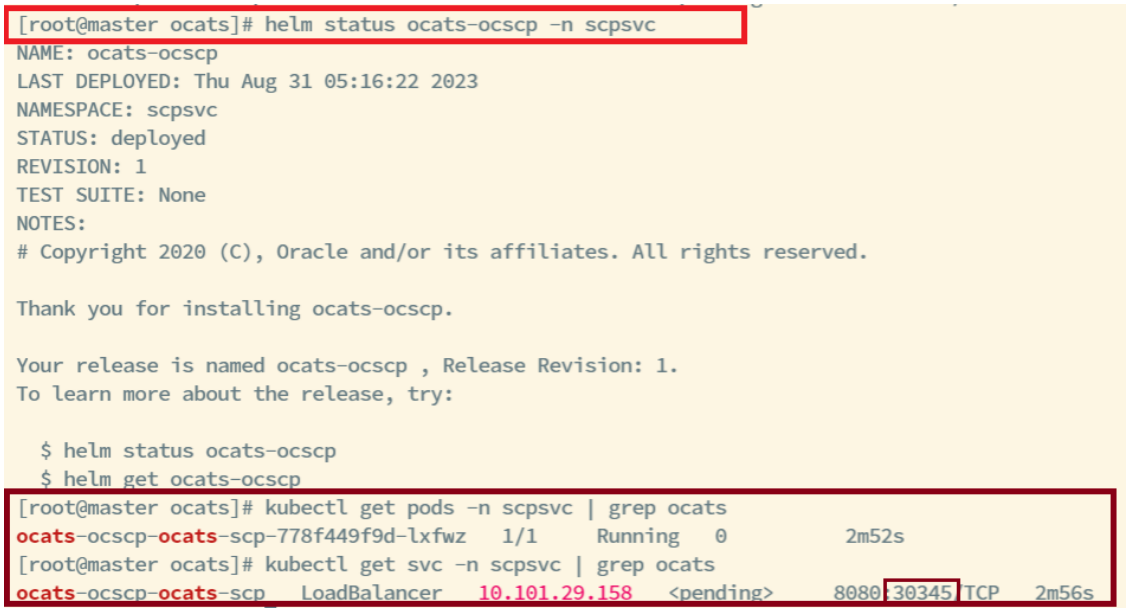
4.3.1 Logging in to ATS

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:

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 4-17 Verifying ATS Pod



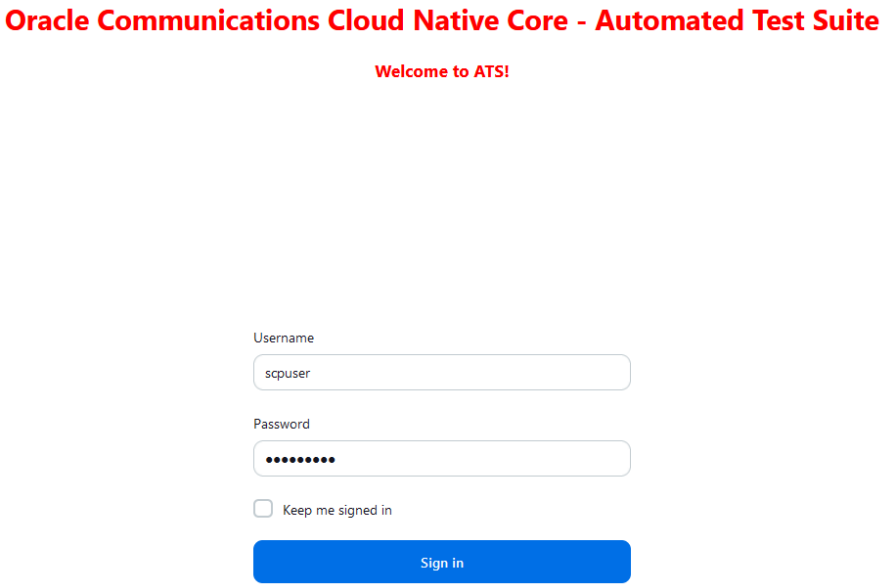
> **Note**
>
> In the Verifying ATS Pod image, the ATS nodeport is highlighted in red as **30345**. For more information about ATS deployment, see Deploying ATS and Stub in the Kubernetes Cluster. Similarly, verify Stub deployments.

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, 2026, Oracle and/or its affiliates.

Figure 4-18 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:

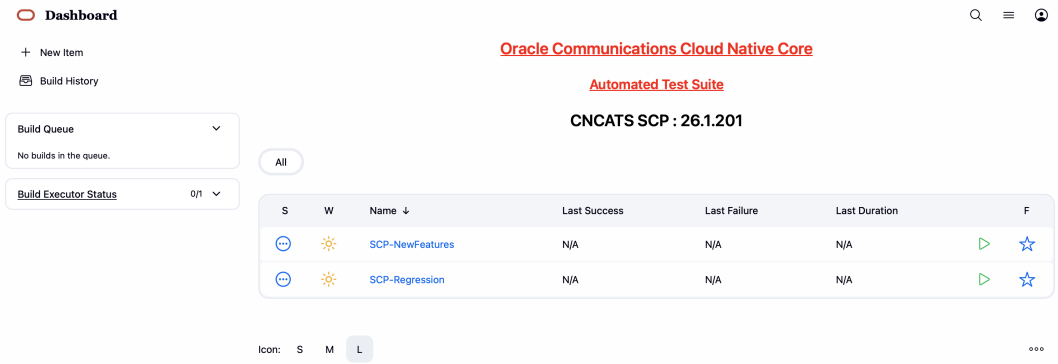
1. Enter the login credentials and click **Sign in**.

   > **Note**
   >
   > To modify default login password, see Modifying Login Password.

   The following screens appear to display preconfigured pipelines for SCP individually.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 4-19 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.

4.3.2 SCP-NewFeatures Pipeline

This section describes how to configure new feature pipeline parameters, run SCP test cases, and view functionalities.

4.3.2.1 Configuring New Feature Pipelines

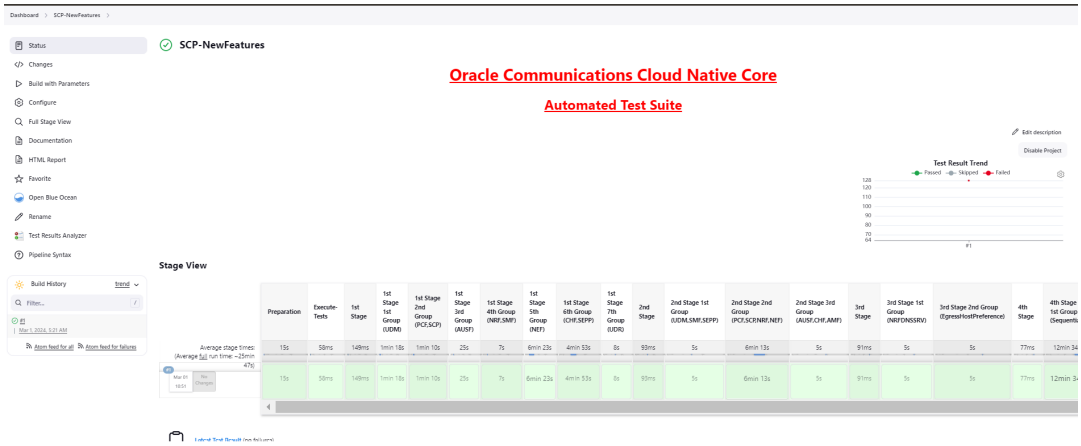
This is a pre-configured pipeline where all the SCP test cases are run.

To configure its parameters, do the following:

1. Click **SCP-NewFeatures** in the **Name** column.
   The SCP-NewFeatures screen appears.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 4-20 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.

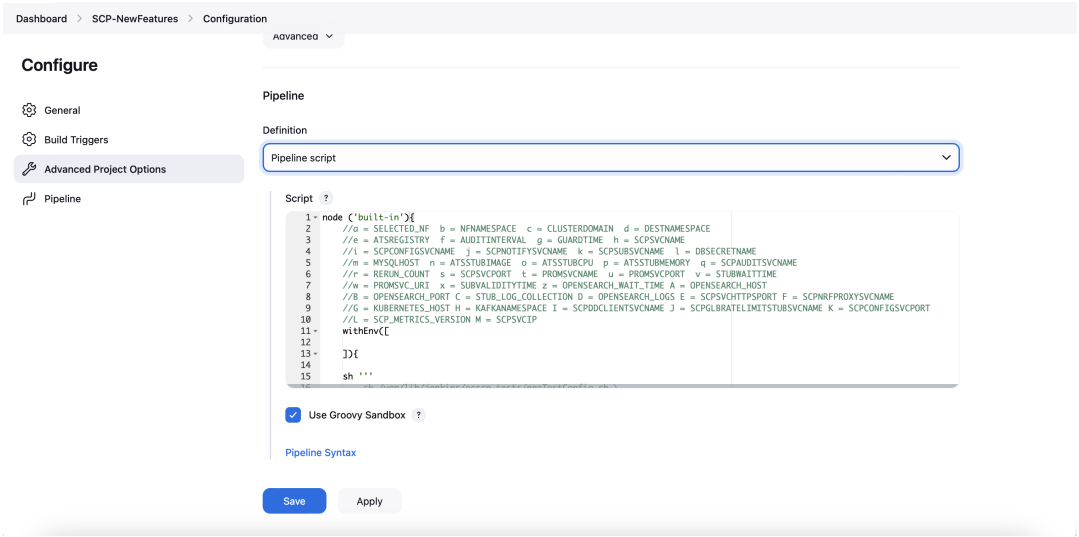
2. In the left navigation pane, click **Configure** to provide input parameters and scroll down to the pipeline script as shown in the following image:
The General tab appears.

> **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, 2026, Oracle and/or its affiliates.

Figure 4-21 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 N = KAFKANAMESPACE1
    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, 2026, Oracle and/or its affiliates.

```
            -k cicdocscp-scpc-subscription \
            -l appuser-secret \
            -m mysql.scp-73999881-dev.svc.cluster.local \
            -n ocats-pystub \
            -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 \
            -N scpsvc1 \

        '''
        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

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **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
- **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, 2026, Oracle and/or its affiliates.

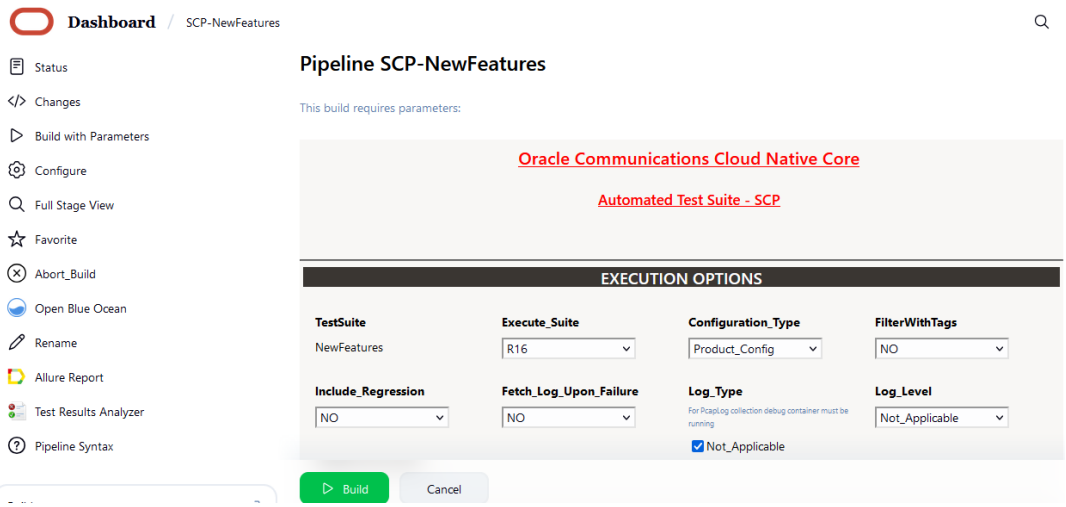
- **M**: scp worker FQDN or cluster IP: The default value is scpSignallingFqdn, which forwards requests to the worker using the worker's FQDN. Alternatively, an IPv4 or IPv6 address (enclosed in square brackets) can also be provided.
- **N**: scpsvc1 \

4.3.2.2 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 4-22 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, 2026, 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 4-23 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.

4.3.3 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.

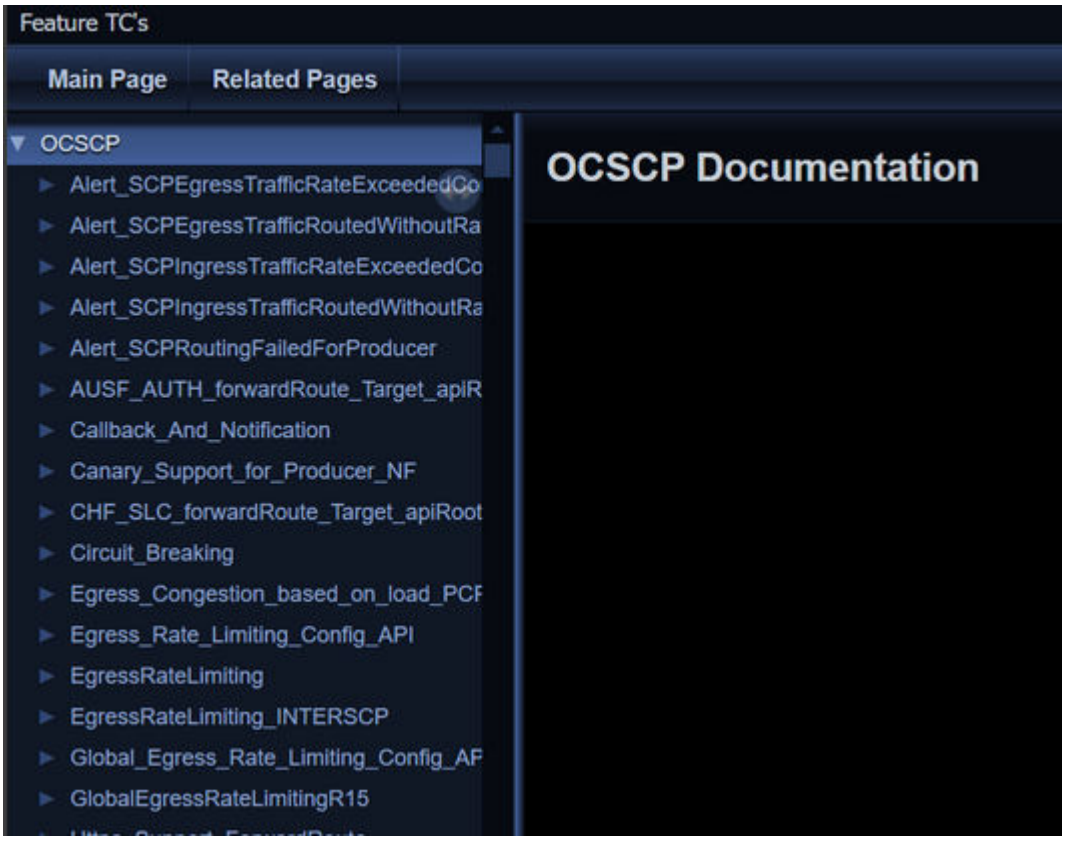
Copyright © 2019, 2026, Oracle and/or its affiliates.

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 -26.1.201.

> **Note**
>
> The **Documentation** option appears only if the **SCP-NewFeatures** pipeline test cases are run at least once.

The following screen appears:

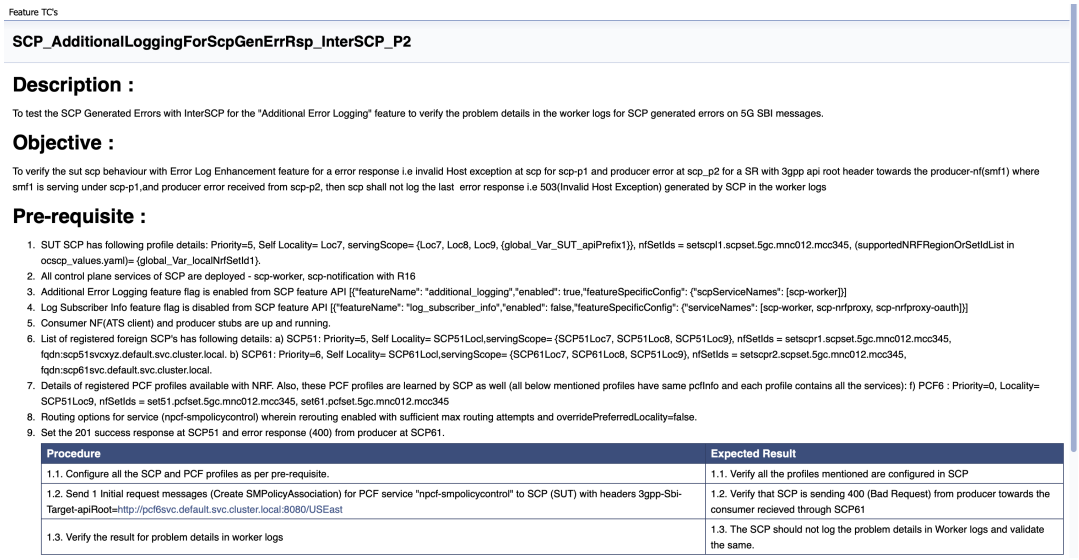
Figure 4-24 SCP-NewFeatures Documentation



2. Click any functionality to view its test cases and scenarios for each test case.

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 4-25 Sample Scenario



Feature TC's

SCP_AdditionalLoggingForScpGenErrRsp_InterSCP_P2

Description :

To test the SCP Generated Errors with InterSCP for the "Additional Error Logging" feature to verify the problem details in the worker logs for SCP generated errors on 5G SBI messages.

Objective :

To verify the sut scp behaviour with Error Log Enhancement feature for a error response i.e invalid Host exception at scp for scp-p1 and producer error at scp_p2 for a SR with 3gpp api root header towards the producer-nf(smf1) where smf1 is serving under scp-p1,and producer error received from scp-p2, then scp shall not log the last error response i.e 503(Invalid Host Exception) generated by SCP in the worker logs

Pre-requisite :

1. SUT SCP has following profile details: Priority=5, Self Locality= Loc7, servingScope= {Loc7, Loc8, Loc9, {global_Var_SUT_apiPrefix1}}, nfSetIds = setscpl1.scpset.5gc.mnc012.mcc345, (supportedNRFRegionOrSetIdList in ocscp_values.yaml)= {global_Var_localNrfSetId1}.
2. All control plane services of SCP are deployed - scp-worker, scp-notification with R16
3. Additional Error Logging feature flag is enabled from SCP feature API [{"featureName": "additional_logging","enabled": true,"featureSpecificConfig": {"scpServiceNames": [scp-worker]}}]
4. Log Subscriber Info feature flag is disabled from SCP feature API [{"featureName": "log_subscriber_info","enabled": false,"featureSpecificConfig": {"serviceNames": [scp-worker, scp-nrfproxy, scp-nrfproxy-oauth]}}]
5. Consumer NF(ATS client) and producer stubs are up and running.
6. List of registered foreign SCP's has following details: a) SCP51: Priority=5, Self Locality= SCP51Locl,servingScope= {SCP51Loc7, SCP51Loc8, SCP51Loc9}, nfSetIds = setscpr1.scpset.5gc.mnc012.mcc345, fqdn:scp51svcxyz.default.svc.cluster.local. b) SCP61: Priority=6, Self Locality= SCP61Locl,servingScope= {SCP61Loc7, SCP61Loc8, SCP51Loc9}, nfSetIds = setscpr2.scpset.5gc.mnc012.mcc345, fqdn:scp61svc.default.svc.cluster.local.
7. Details of registered PCF profiles available with NRF. Also, these PCF profiles are learned by SCP as well (all below mentioned profiles have same pcfInfo and each profile contains all the services): f) PCF6 : Priority=0, Locality= SCP51Loc9, nfSetIds = set51.pcfset.5gc.mnc012.mcc345, set61.pcfset.5gc.mnc012.mcc345
8. Routing options for service (npcf-smpolicycontrol) wherein rerouting enabled with sufficient max routing attempts and overridePreferredLocality=false.
9. Set the 201 success response at SCP51 and error response (400) from producer at SCP61.

Procedure	Expected Result
1.1. Configure all the SCP and PCF profiles as per pre-requisite.	1.1. Verify all the profiles mentioned are configured in SCP
1.2. Send 1 Initial request messages (Create SMPolicyAssociation) for PCF service "npcf-smpolicycontrol" to SCP (SUT) with headers 3gpp-Sbi-Target-apiRoot=http://pcf6svc.default.svc.cluster.local:8080/USEast	1.2. Verify that SCP is sending 400 (Bad Request) from producer towards the consumer recieved through SCP61
1.3. Verify the result for problem details in worker logs	1.3. The SCP should not log the problem details in Worker logs and validate the same.

Based on the functionalities covered under Documentation, the screen displays test cases.

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.

4.3.4 SCP-Regression Pipeline

This section describes how to configure regression pipeline parameters, run SCP test cases, and view functionalities.

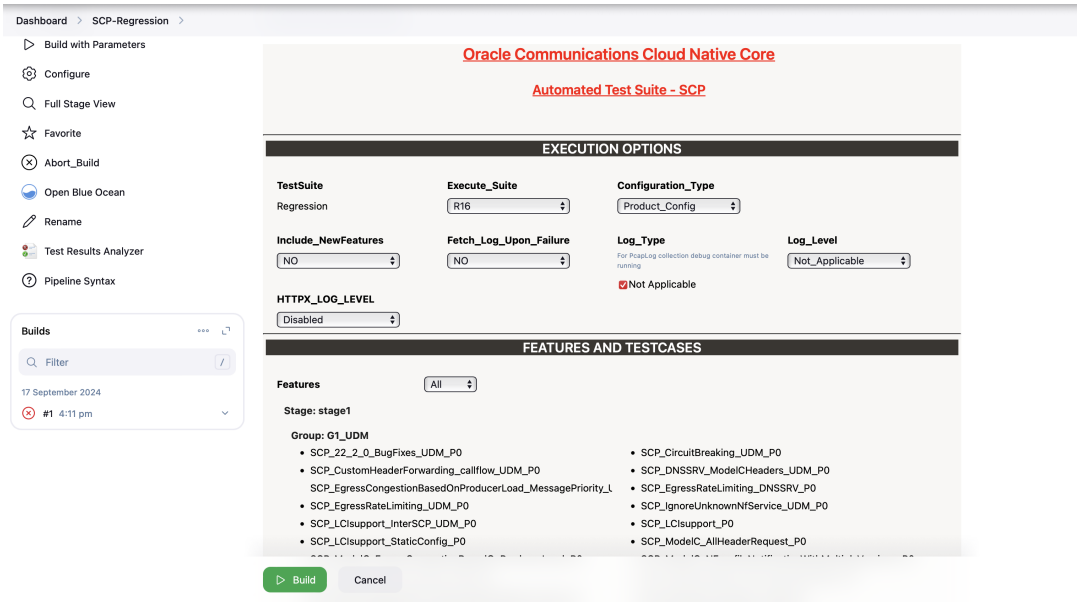
4.3.4.1 Configuring Regression Pipelines

This pre-configured pipeline has all the test cases from previous releases.

1. Click **SCP-Regression Pipeline**:

Copyright © 2019, 2026, Oracle and/or its affiliates.

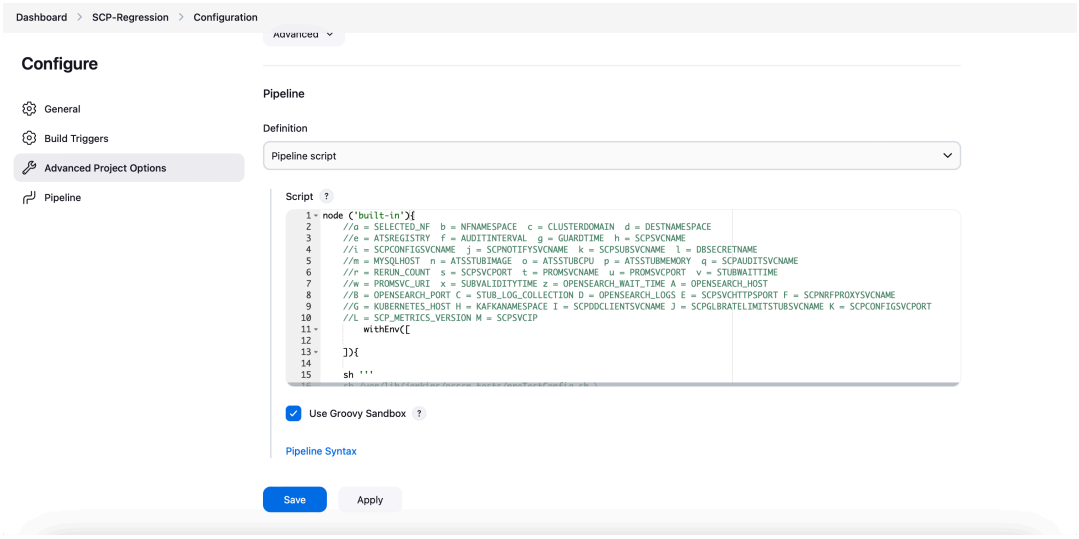
Figure 4-26 SCP-Regression Pipeline



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:

Figure 4-27 Regression - Pipeline Script



3. Modify the values of the Pipeline script as required, and then click **Save**.

Copyright © 2019, 2026, Oracle and/or its affiliates.

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 = LOG_SOURCE E =
SCPSVCHTTPSPORT F = SCPNRFPROXYSVCNAME
    //G = KUBERNETES_HOST H = KAFKANAMESPACE I = SCPDDCLIENTSVCNAME J =
SCPGLBRATELIMITSTUBSVCNAME K = SCPCONFIGSVCPORT
    //L = SCP_METRICS_VERSION M = SCPSVCIP N = KAFKANAMESPACE1
        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 \
        -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 kubernetes \
        -E 9443 \
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

```
            -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 \
            -N scpsvc1 \

        '''
        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
- **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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **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:** User can provide Kubernetes/opensearch for ATS to use for app log collection (Fetch_Log_Upon_Failure needs to 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
- **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.
- **N**: scpsvc1 \

4.3.4.2 Running Regression Pipelines

This section describes how to run test cases for SCP Regression pipeline.

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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- **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.

4.3.5 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.

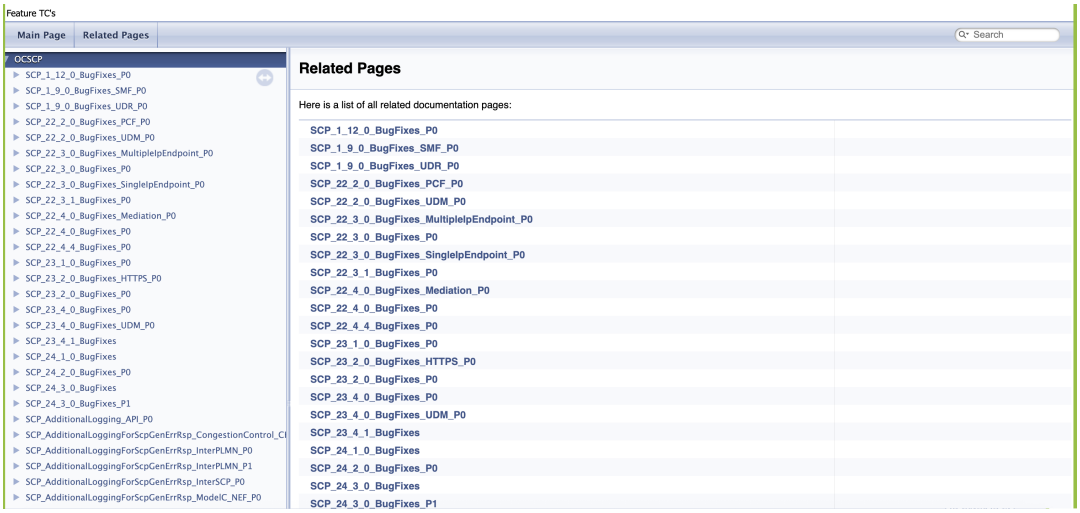
1. To view SCP functionalities, go to the **SCP-Regression** pipeline and click the **Documentation** option in the left navigation pane. It lists all the test cases provided as part of SCP ATS -26.1.201.

> **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:

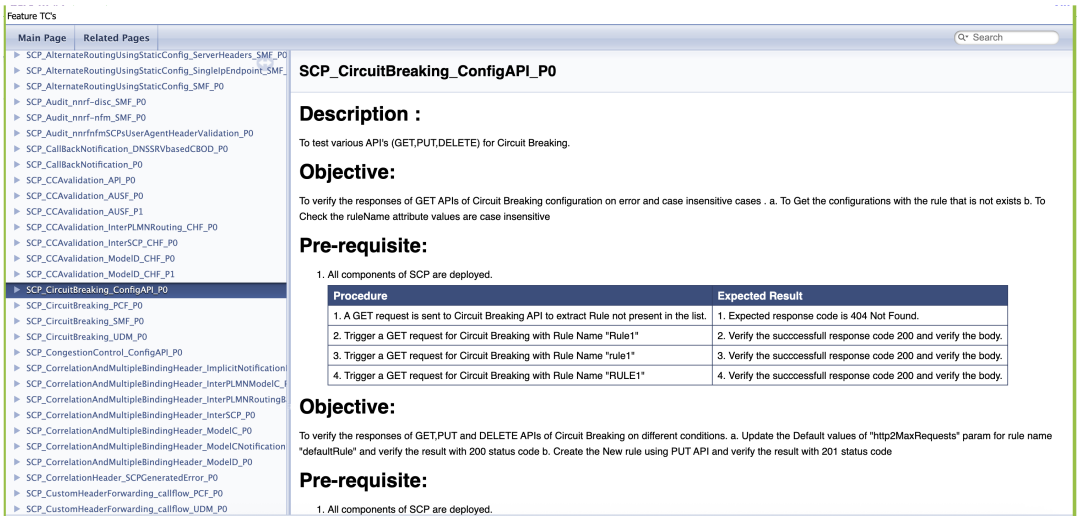
Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 4-28 SCP-Regression Documentation



2. Click any functionality to view its test cases and scenarios for each test case.

Figure 4-29 Sample Scenario



Based on the functionalities covered under Documentation, the screen displays test cases.

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.

4.3.5.1 Renaming of Regression Feature Files

The following tables detail the feature file that has been renamed in alignment with the naming convention:

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 4-1 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
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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 4-1 (Cont.) Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
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
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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 4-1 (Cont.) Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
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
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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 4-1 (Cont.) Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
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
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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 4-1 (Cont.) Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
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
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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 4-1 (Cont.) Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
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
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

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 4-1 (Cont.) Regression Feature File

Feature Names prior to SCP 24.1.0	New Feature Names for SCP 24.1.0
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
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:

Copyright © 2019, 2026, Oracle and/or its affiliates.

Table 4-2 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

4.4 Running SEPP Test Cases using ATS

This section describes how to run SEPP test cases using ATS.

4.4.1 Prerequisites

To run SEPP Test Cases using SEPP ATS 26.1.201, you need to ensure that following prerequisites are fulfilled:

- Debug container should be DISABLED for all microservices during ATS runs.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- 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 26.1.201 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_26.1.201.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.
- 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.

Copyright © 2019, 2026, Oracle and/or its affiliates.

- 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_<version>.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_<version>.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_<version>.yaml file`.
- For Hosted SEPP, set the `HSenabled` parameter to "true" in the ocats section of the `ocats_ocsepp_values_<version>.yaml` file.
- For Hosted SEPP, set the `hostedSEPP` parameter to "true" in the global section of the `ocsepp_custom_values.yaml` file.
- For Hosted SEPP, set the `multiFqdn` parameter to "true" in both the n32-ingress-gateway and n32-egress-gateway sections of the `ocsepp-custom-values.yaml` file.
- For Hosted SEPP, set the `n32cHandshakePlmnIdListValidationEnabled` parameter to "false" in the global section and the Prefix/Suffix Global Parameters subsection of the `ocsepp-custom-values.yaml` file.
- For Hosted SEPP, set the `retryInterval` parameter to "30000" in the global section of the `ocsepp-custom-values.yaml` file.
- For Hosted SEPP, update the certificates for hostedpartner1, hostedpartner2, roamingpartner1, roamingpartner2, stubserver2, and stubserver3 in both the n32-ingress-gateway and n32-egress-gateway sections of the `ocsepp-custom-values.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.

4.4.2 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 4-30 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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.

There are following ways to login to ATS GUI.

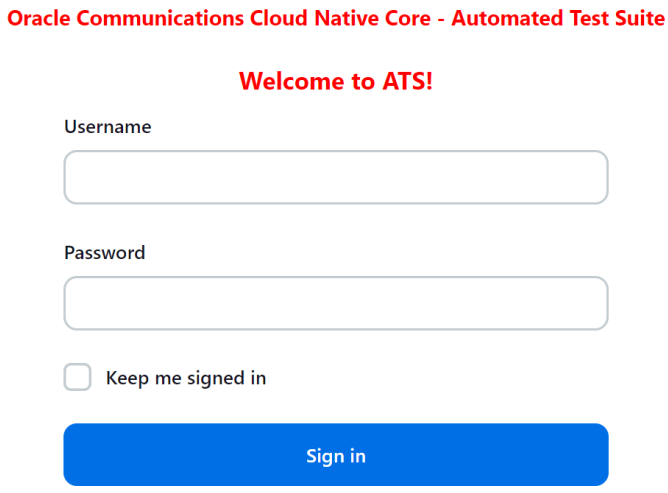
- When an external load balancer (metalLB in case of OCCNE) is available and an external IP Address is provided to the ATS service, login to ATS GUI using <External-IP>:8080.
- When an external IP Address is not provided to the ATS service, open the browser and provide the external IP Address of the 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 4-31 ATS Login



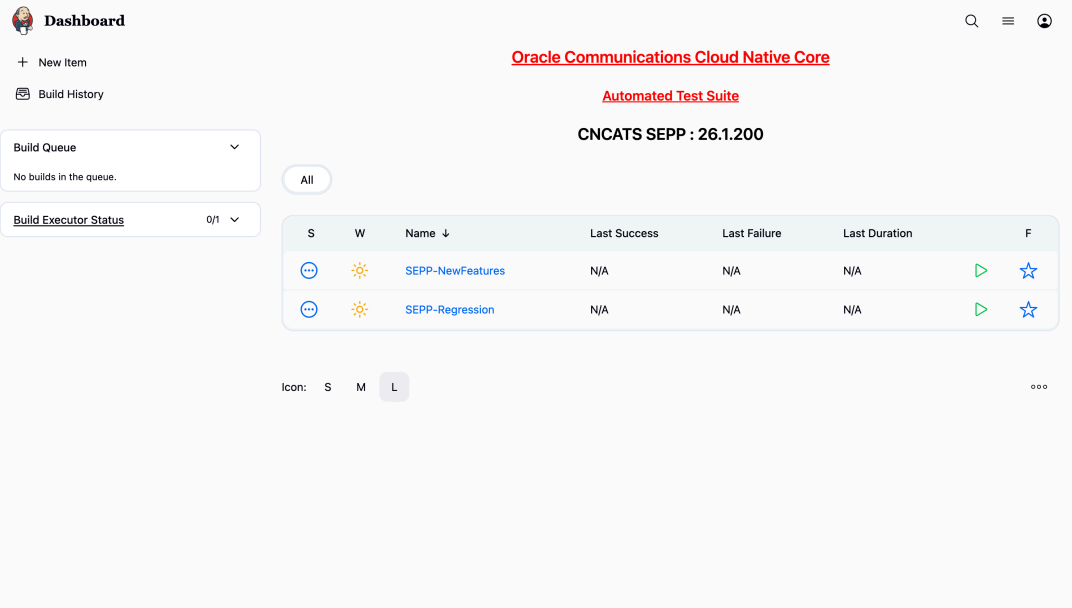
Copyright © 2019, 2026, Oracle and/or its affiliates.

To run ATS:

- 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 4-32 Pre-Configured Pipelines

- **Figure 4-33 SEPP ATS 26.1.200**



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.

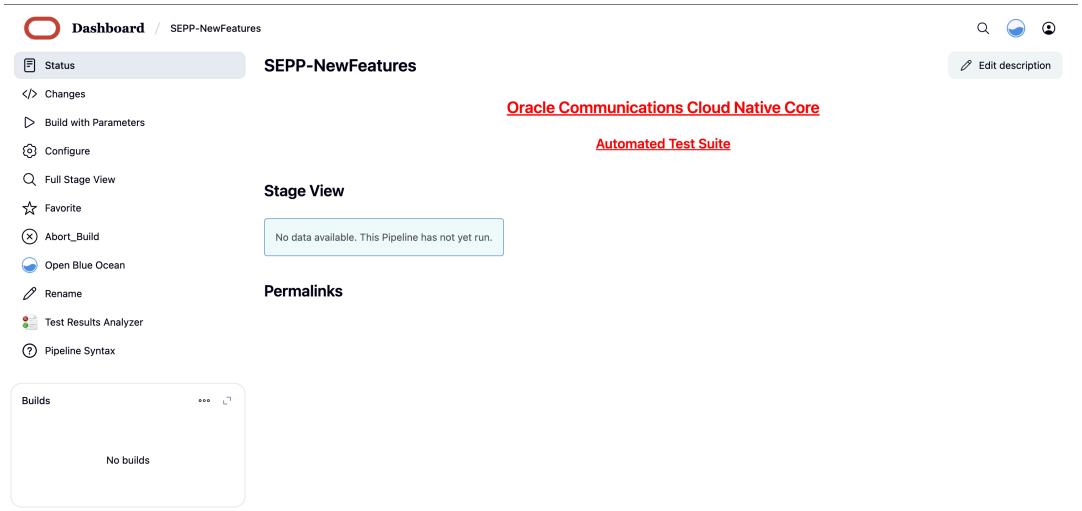
4.4.3 SEPP NewFeatures Pipeline

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, 2026, Oracle and/or its affiliates.

Figure 4-34 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, 2026, Oracle and/or its affiliates.

Figure 4-35 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, 2026, 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.

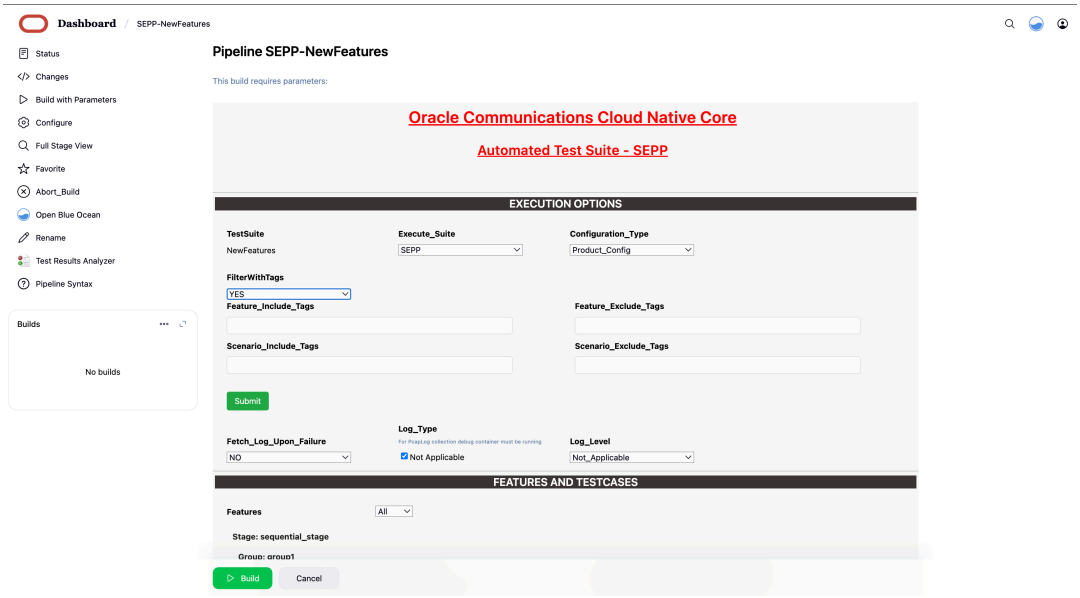
1. Click **Save** after making necessary changes. The **Pipeline SEPP-NewFeatures** screen appears.

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 4-36 SEPP-NewFeatures



2. In the above screen, there is an option **Execute_Suite** to select SEPP, Roaming_Hub, or Hosted SEPP. 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, 2026, 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.

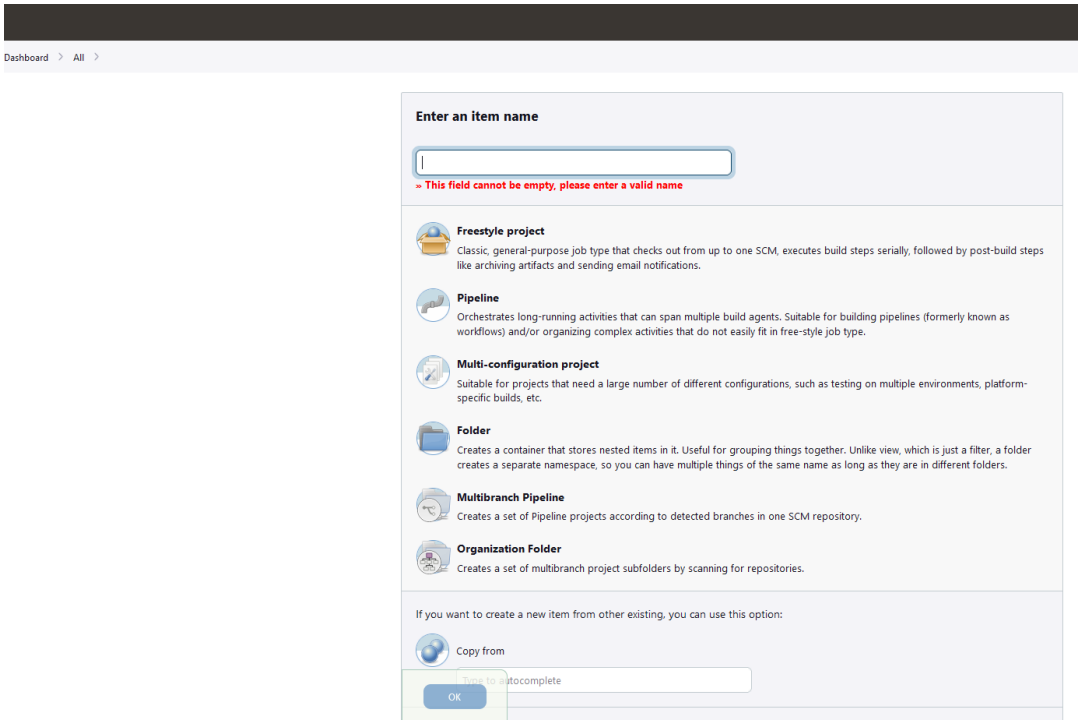
4.4.3.1 Single Click Job Creation for SEPP New Features

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, 2026, Oracle and/or its affiliates.

Figure 4-37 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, 2026, 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.

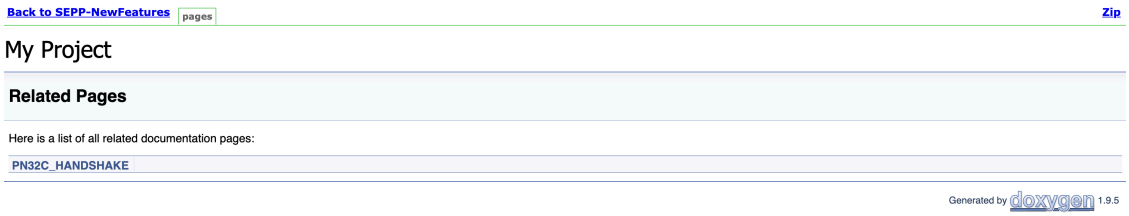
> **Note**
>
> Value of `j (RERUN_COUNT) = 3` is recommended. But it can be changed as per operator environment.

4.4.4 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, 2026, Oracle and/or its affiliates.

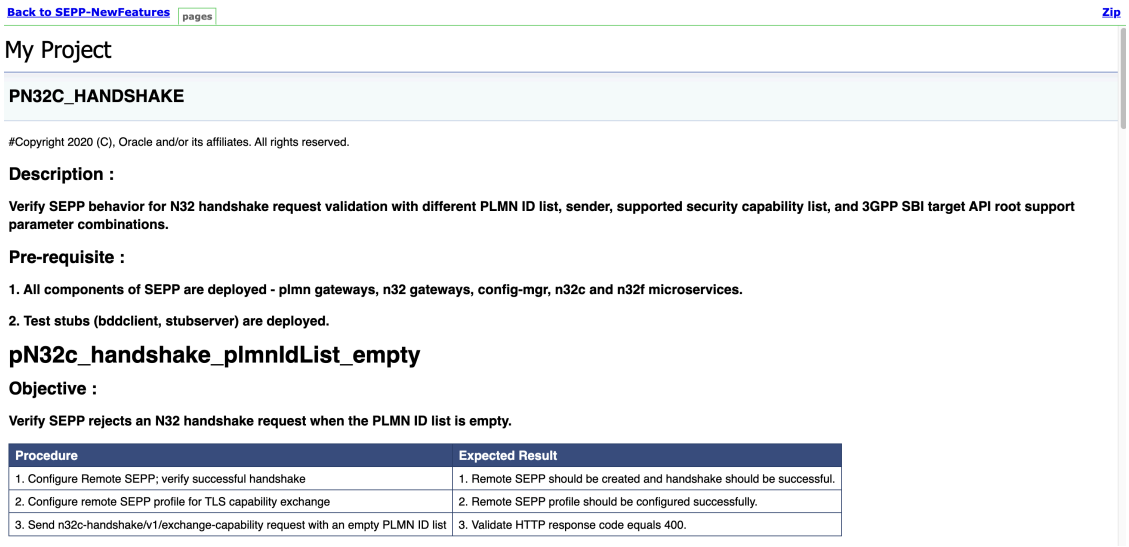
Figure 4-38 Documentation



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 4-39 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.

4.4.5 SEPP Regression Pipeline

This section describes how to run test cases for SEPP Regression pipeline.

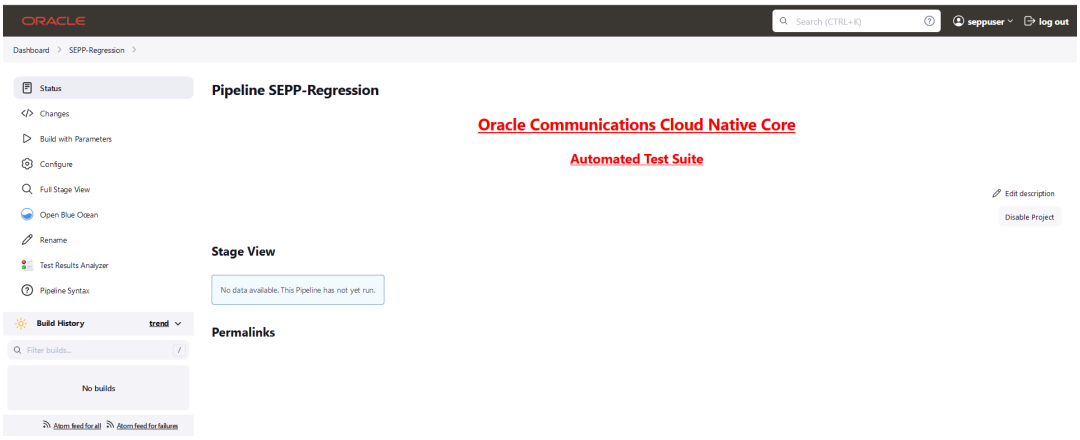
The **SEPP_Regression** pipeline is a pre-configured pipeline where all the test cases of previous releases are available. For example, for SEPP 25.1.2xx, this pipeline has all the test cases released till SEPP 25.1.1xx.

To view **SEPP_Regression** pipeline details, perform the following steps:

1. Click **SEPP_Regression** in the **Name** column. The following screen appears:

Copyright © 2019, 2026, Oracle and/or its affiliates.

Figure 4-40 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.

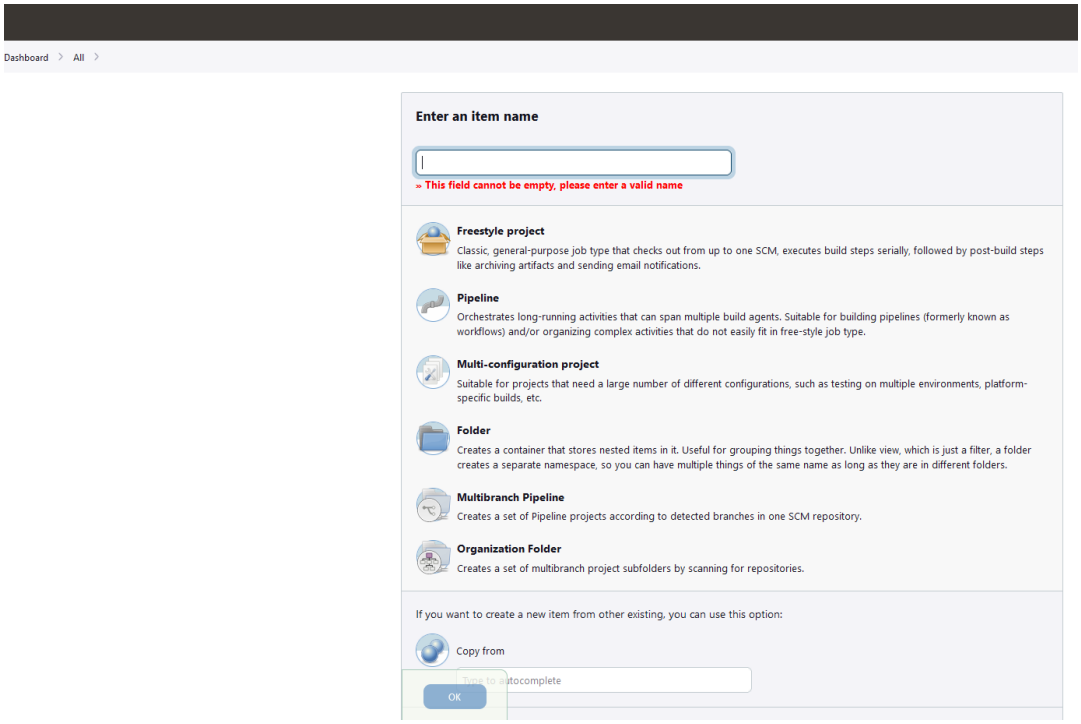
4.4.5.1 Single Click Job Creation for SEPP Regression Features

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, 2026, Oracle and/or its affiliates.

Figure 4-41 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, 2026, 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.

4.4.6 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, 2026, Oracle and/or its affiliates.

The following screen shows all the documentation features:

Figure 4-42 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 4-43 CN32C_SUCCESS_SECURITY_CAPABILITY_TLS

4.4.7 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, 2026, 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 4-44 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, 2026, 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
```


Copyright © 2019, 2026, Oracle and/or its affiliates.