Oracle® Communications 5G Automated Testing Suite Guide

Release 1.2.0
F37980-01
July 2020

ORACLE

Oracle Communications 5G Automated Testing Suite Guide , Release 1.2.0

F37980-01

Contents

List of Figures

List of Tables

What's New in This Guide

This section shares the list of new features introduced in every ATS release. For more release specific information, please refer to its release notes.

ATS Release 1.1.0
Following new features are introduced in ATS 1.1.0:

Table Features

Feature	NRF	PCF	SCP
One-click start of <NF> New-feature and Regression pipelines for **'All'** option after successfully logging into the Jenkins GUI.	Yes **Example:** • **New-Features One-Click** - To run **"All"** testcases in **NRF-NewFeatures** pipeline, use **http://<Jenkins_IP>:<Jenkins_Port>/view/NRF/job/NRF-NewFeatures/build** link to open Jenkins and click **"Build"**. • **Regression One-Click** - To run **"All"** testcases in **NRF-Regression** pipeline, use **http://<Jenkins_IP>:<Jenkins_Port>/view/NRF/job/NRF-Regression/build** link to open Jenkins and click **"Build"**.	Yes **Example:** • **New-Features One-Click** - To run **"All"** testcases in **PCF-NewFeatures** pipeline, use **http://<Jenkins_IP>:<Jenkins_Port>/view/PCF/job/PCF-NewFeatures/build** link to open Jenkins and click **"Build"**. • **Regression One-Click** - To run **"All"** testcases in **PCF-Regression** pipeline, use **http://<Jenkins_IP>:<Jenkins_Port>/view/PCF/job/PCF-Regression/build** link to open Jenkins and click **"Build"**.	Yes **Example:New-Features One-Click** - To run **"All"** testcases in **SCP-NewFeatures** pipeline, use **http://<Jenkins_IP>:<Jenkins_Port>/view/SCP/job/SCP-NewFeatures/build** link to open Jenkins and click **"Build"**.
Allows to deploy ATS using either **Helm2** or **Helm3** helm versions	Yes	Allows to deploy ATS using Helm 2 only.	Yes
Does not requires manual intervention to add environmental variable, like ATS-1.0.0	Yes	Yes	Yes
One-time configuration on Jenkins GUI to execute the test cases.	Yes	Yes	Yes

Table (Cont.) Features

Feature	NRF	PCF	SCP
No need to login to ATS pod through CLI. Users can perform all operations through Jenkins GUI.	Yes	Yes	Yes
Allows to execute Sanity cases to validate the NF and ATS deployment.	Yes	No	No
Provides an option to execute either **All**, **Sanity** and **Single/ Multiple** feature files in the New-Features pipeline.	Yes	Yes but it does not have Sanity option	Yes but it does not have Sanity option
Does not display **Skipped cases** in the console output when executing the cases using **Single/Multiple** feature files option.	Yes	No	No
The **Documentation** section shows all the testcases according to the service operation supported by NF.	Yes	Yes	Yes
Automatically executes the failed test cases. Default re-run count is 2. The overall pipeline status shows the result of re-run so, it is recommended to provide the value of re-run count variable.	Yes	Yes. Default re-run count is 1.	Yes
New TestCases added to Jenkins Pipeline	Provides a total of **251 scenarios** clubbed together in **100 feature files** of NRF ATS - 1.6.1 New Feature pipeline.	Provides 19 Feature files in PCF-NewFeatures pipeline.	Provides a total of **65 scenarios** clubbed together in **9 feature files** of SCP ATS - 1.6.0 New Feature pipeline.
Previous Release TestCases	Provides a total of **163 scenarios** clubbed together in **80 feature files** of NRF ATS - 1.6.1 Regression pipeline.	Provides 10 Feature files in PCF-Regression pipeline.	Not applicable

Table (Cont.) Features

Feature	NRF	PCF	SCP
Backward Compatibility	ATS is **NOT** backward compatible. It means NRF ATS - 1.6.1 will work only with NRF 1.6.1. For NRF 1.5.0, user still need to use ATS - 1.0.0 version.	ATS is **NOT** backward compatible. It means PCF ATS - 1.1.0 will work only with PCF 1.6.1. For PCF 1.5.0, user still need to use ATS - 1.0.0 version.	Not applicable
Supports NF with TLS Enabled (server side) and Disabled mode	Not applicable	Yes. PCF ATS supports PCF with TLS Enabled (server side) and Disabled mode	Not applicable
Test cases delivered in ATS Release 1.0.0 are added to its respective Regression Pipeline. User can run any pipeline but not parallel.	Yes	Yes	Not applicable

Following table shares information about pipeline statuses:

Table Pipeline Statuses

Condition	Result	Stage Status	Pipeline Status
If all the test cases pass without rerun	Re-run function executes. It does not initiate any test case as none of the test case has failed.	Green	All other successful stages appears as Green and Build Status appears as Blue.
If some test cases fail in actual execution and all of them pass in re-run stage	The Execution stage appears as **YELLOW**.	Yellow. It indicates that the stage was not successful in first attempt.	All other successful stages appears as Green and Build Status appears as Blue.
If some test cases fail in actual execution and some of them pass in re-run, and some test cases still have failed in re-run even after exhausting the re-run count	The Execution stage appears as YELLOW	Yellow. It indicates that the stage was not successful in first attempt.	All other successful stages appears as Green and Build Status appears as Red.

1

Understanding Automated Testing Suite (ATS)

In this chapter, you will get an overview about ATS, its need and its features.

Automated Testing Suite Overview

Automated Testing Suite (ATS) allows you to execute software test cases using an automated testing tool and then, compares the actual results with the expected or predicted results. In this process, there is no intervention from the user.

ATS for 5G Network Functions

For 5G Network Functions (NFs), ATS is built using **Oracle Linux 7-slim** as the base image. **Jenkins** is a part of the ATS image and it provides a GUI interface to the users to test either a single NF or multiple NFs independently in the same environment.

Along with the NF docker images, user are provided with the ATS image, simulator images, and test cases for the specific NF. All these are handed over to the customer as a fully automated suite so that they can directly perform Lab deployment and testing. You can combine it with any other **Continuous Integration (CI) pipeline** with minimal changes. Since, 5G ATS uses Jenkins as GUI.

Why Automated Testing Suite in 5G NFs?

Through Automated Testing Suite (ATS), Oracle Communications aims at providing an end-to-end solution to its customers for deploying and testing its 5G-NFs.

This guide covers implementation of ATS in 5G NFs like Network Repository Function (NRF), Policy Control Function (PCF) and Service Communication Proxy (SCP).

ATS Features

The ATS features are as follows:

- Provides an end-to-end solution to the customers for testing Oracle Communications 5G-NFs. The ATS package includes:
  - Test scripts and docker images of test container.
    - The docker images have complete framework and libraries installed, which is common for all NFs working with BDD framework.
  - Docker image of HTTP Server simulator
  - Helm chart to deploy the ATS (delivered as a tar file)
  - Readme text file (.txt file)
- Enables all the NF teams with the basic environment, framework and a GUI (Jenkins) to execute all the functional test cases.

2
ATS Deployment Models

In this chapter, you will learn about ATS Deployment Models. They are:

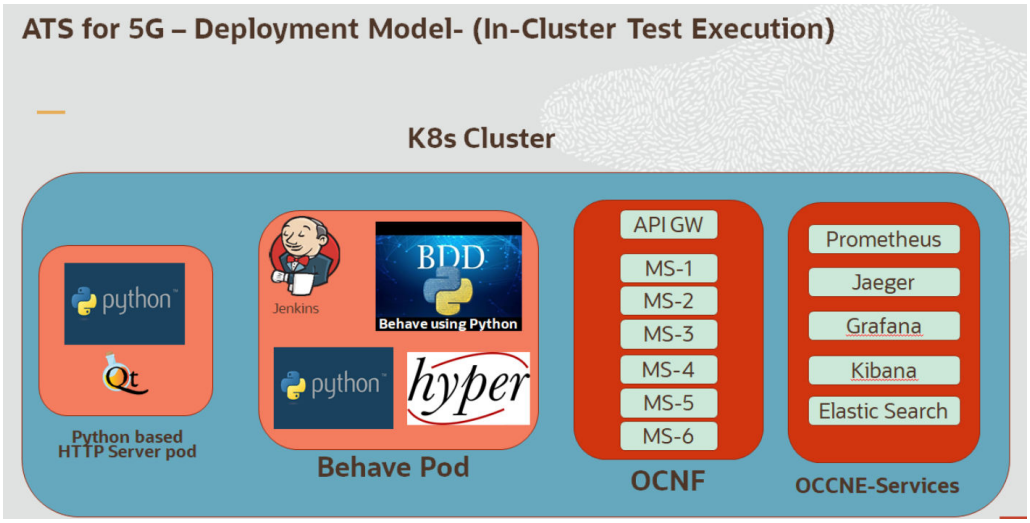
- In-Cluster Deployment
- Out-of Cluster Deployment

In-Cluster Deployment Model

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

> **Note:**
>
> The ATS 1.1.0 package supports only in-cluster deployment.

Figure 2-1 In-Cluster Deployment Model



> **Note:**
>
> GO Language is used to create stubs for PCF ATS and SCP ATS.

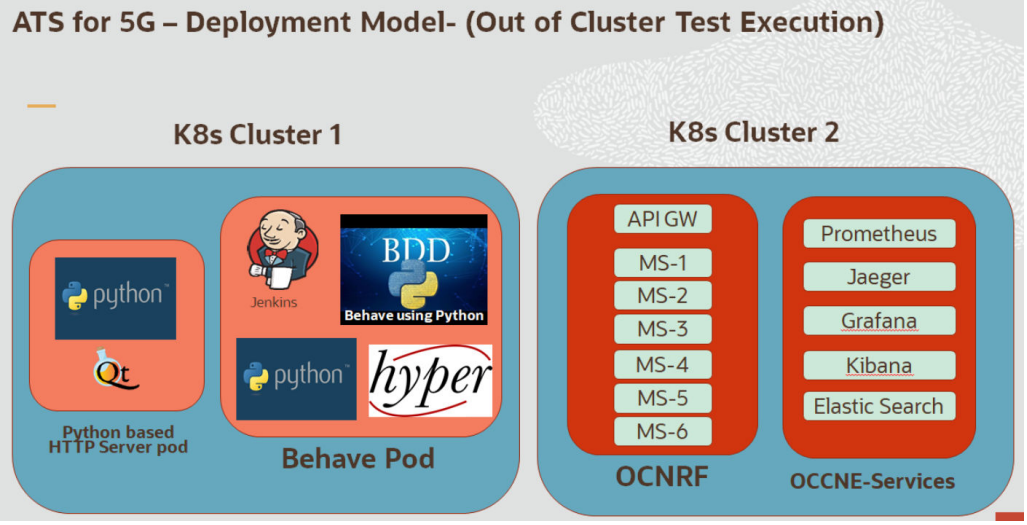
Out-of-Cluster Deployment Model

According to **Out-of-Cluster deployment model**, you can deploy ATS in a separate cluster other than the one where NFs are deployed.

This deployment model is useful to perform "Out-of-cluster" testing as it is:

- More aligned with the production use cases
- Rare for all NFs to co-exist in the same cluster

Figure 2-2 Out-of-Cluster Deployment Model



Note:

GO Language is used to create stubs for PCF ATS and SCP ATS.

3
NF ATS Installation Procedure

In this chapter, you will learn to install ATS for different network function platforms like,

- NRF
- PCF
- SCP

NRF ATS Installation Procedure

The NRF ATS installation procedure covers two steps:

1. Locating and downloading ATS and Simulator Images
2. Deploying ATS and Stub Pod in K8s Cluster as per NRF

Locating and Downloading ATS Images

The steps to locate and download ATS Images are as follows:

1. Login to My Oracle Support using the appropriate credentials.
2. Select **Patches & Updates** tab.
3. In **Patch Search** console, select **Product or Family (Advanced)** tab.
4. Enter *Oracle Communications Cloud Native Core - 5G* in **Product** field and select the product from the Product drop-down.
5. Select *Oracle Communications Cloud Native Core Network Repository Function <release_number>* in **Release** field.
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 on **Download**. File Download window appears.
9. Click on the **<p********_<release_number>_Tekelec>.zip** file.
10. Extract the ATS release package zip file to download the ATS images to the system where network function must be installed.
11. The `ocats-nrf` directory has following files:
    - `ocats-nrf-tools-pkg-1.6.1.0.0.tgz`
    - `ocats-nrf-tools-pkg-1.6.1.0.0-README.txt`
12. The `ocats-nrf-tools-pkg-1.6.1.0.0-README.txt` file contains all the information required for the package.
13. The `ocats-nrf-tools-pkg-1.6.1.0.0.tgz` file has following images and charts packaged as tar files:

    ```
    ocats-nrf-tools-pkg-1.6.1.0.0.tgz
    ```

```
        |
        |_ _ _ocats-nrf-pkg-1.6.1.0.0.tgz
        |        |_ _ _ _ _ _ ocats-nrf-1.6.1.tgz (Helm Charts)
        |        |_ _ _ _ _ _ ocats-nrf-image-1.6.1.tar (Docker
Images)
        |        |_ _ _ _ _ _ Readme.txt
        |
        |_ _ _ocstub-python-pkg-1.6.1.0.0.tgz
                 |_ _ _ _ _ _ ocstub-1.6.1.tgz (Helm Charts)
                 |_ _ _ _ _ _ ocstub-python-image-1.6.1.tar (Docker
Images)
                 |_ _ _ _ _ _ Readme.txt
```

14. The user can copy the tar file from here and copy in their OCCNE/ OCI/k8s cluster where they want to deploy ATS.

Deploying ATS in K8s Cluster

The steps to deploy ATS in K8s Cluster are as follows:

1. Execute the following command to extract tar file content.
   `tar -xvf ocats-nrf-tools-pkg-1.6.1.0.0.tgz`

   The output of this command is:

   ```
   ocats-nrf-pkg-1.6.1.0.0.tgz
   ocstub-python-pkg-1.6.1.0.0.tgz
   ```

2. Execute the given command to extract final helm charts and docker images of ATS.
   `tar -xvf ocats-nrf-pkg-1.6.1.0.0.tgz`

   The output of this command is:

   ```
   ocats-nrf-image-1.6.1.tar
   ocats-nrf-1.6.1.tgz

   Readme.txt
   ```

3. In your cluster, load the ATS image, '`ocats-nrf-image-1.6.1.tar`' and push to your registry.
   `docker load -i ocats-nrf-image-1.6.1.tar`
4. Untar the helm charts, `ocats-nrf-1.6.1.tgz`
   `tar -xvf ocats-nrf-1.6.1.tgz`
5. Update the image name and tag in the `ocats-nrf/values.yaml` file as required. For this, you need to open the `values.yaml` file and update the `image.repository` and `image.tag`
6. Deploy ATS using the updated helm charts after performing the previous **step 5**.

```
helm install ocats-nrf --name <release_name> --namespace
<namespace_name> -f ocats-nrf/values.yaml
```

For example:`helm install ocats-nrf --name ocats --namespace ocnrf -f ocats-nrf/values.yaml`

7. Verify ATS deployment by executing the following command.
   `helm status <release_name>`

A sample screen showing ATS HELM Release is given below:

Figure 3-1 Checking ATS Deployment

```
[root@oort5-master release_1.6.1]# helm status ocats
LAST DEPLOYED: Thu May 28 06:43:55 2020
NAMESPACE: ocnrf
STATUS: DEPLOYED

RESOURCES:
==> v1/ServiceAccount
NAME                                      SECRETS  AGE
ocnrf-ocats-ocats-ocats-nrf-serviceaccount  1        7m

==> v1/ClusterRole
NAME                                    AGE
ocnrf-ocats-ocats-ocats-nrf-clusterrole  7m

==> v1beta1/ClusterRoleBinding
NAME                                             AGE
ocnrf-ocats-ocats-ocats-nrf-clusterrolebinding  7m

==> v1/Service
NAME             TYPE          CLUSTER-IP     EXTERNAL-IP  PORT(S)          AGE
ocats-ocats-nrf  LoadBalancer  10.96.216.104  <pending>    8080:31611/TCP  7m

==> v1/Deployment
NAME             DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
ocats-ocats-nrf  1        1        1           1          7m

==> v1/Pod(related)
NAME                              READY  STATUS   RESTARTS  AGE
ocats-ocats-nrf-7795b9c77d-hfx2j  1/1    Running  0         7m


NOTES:
# Copyright 2018 (C), Oracle and/or its affiliates. All rights reserved.

Thank you for installing ocats-nrf.

Your release is named ocats , Release Revision: 1.
To learn more about the release, try:

  $ helm status ocats
  $ helm get ocats

[root@oort5-master release_1.6.1]#
```

Deploying Stub Pod in K8s Cluster

The steps to deploy Stub Pod are as follows:

1. Execute the command to extract ocstub tar file content.
   `tar -xvf ocstub-python-pkg-1.6.1.0.0.tgz`

   The output of this command is:

   ```
   ocstub-python-image-1.6.1.tar
   ocstub-python-1.6.1.tgz
   ```

`Readme.txt`

2. In your cluster, load the STUB image, `ocstub-python-image-1.6.1.tar` and push to your registry.
   `docker load -i ocstub-python-image-1.6.1.tar`
3. Untar the helm charts, `ocstub-python-1.6.1.tgz`.
   `tar -xvf ocstub-python-1.6.1.tgz`
4. Update the image name and tag in `ocstub/values.yaml` file as required.
   Open the **values.yaml** file and update the `image.repository` and `image.tag`
5. Deploy Stub
   `helm install ocstub-python --set service.name=<stub-service-name> --name <release_name> --namespace <namespace_name> -f ocstub-python/values.yaml`

   For example:`helm install ocstub-python --set service.name=notify-stub-service --name ocstub --namespace ocnrf -f ocstub/values.yaml`

   > **Note:**
   >
   > In this version of ATS, you are provided with SLF and Forwarding functionality cases of NRF, which requires two STUBS to be deployed in the related scenarios. The service name can be updated using the above command. The service name for the STUBS must be "notify-stub-service" and "notify-stub-service02".

6. Check the Stub deployment using given command.
   `helm status <release_name>`

A sample screen showing how to check Stub HELM Release is given below:

Figure 3-2 Checking Stub Helm Release

```
[root@oort5-master release_1.6.1]# helm status ocstub
LAST DEPLOYED: Thu May 28 06:46:31 2020
NAMESPACE: ocnrf
STATUS: DEPLOYED

RESOURCES:
==> v1/Deployment
NAME                DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
ocstub-ocstub-python  1        1        1           1          7m

==> v1/Pod(related)
NAME                                READY  STATUS   RESTARTS  AGE
ocstub-ocstub-python-58d876b6b6-tchwc  1/1    Running  0         7m

==> v1/Service
NAME                TYPE          CLUSTER-IP    EXTERNAL-IP  PORT(S)                                        AGE
notify-stub-service  LoadBalancer  10.98.12.239  <pending>    8080:31315/TCP,8091:31573/TCP,8443:30718/TCP  7m


NOTES:
# Copyright 2018 (C), Oracle and/or its affiliates. All rights reserved.

Thank you for installing ocstub-python.

Your release is named ocstub , Release Revision: 1.
To learn more about the release, try:

  $ helm status ocstub
  $ helm get ocstub

[root@oort5-master release_1.6.1]#
```

PCF ATS Installation Procedure

The PCF ATS installation procedure covers two steps:

1. Locating and downloading the ATS images.
2. Deploying ATS images.

This includes installation of stubs (nf1stub, nf2stub and nf3stub) in any namespace and ATS in PCF's namespace. The release of ATS supports in-cluster deployment of PCF and ATS with both TLS (server side) enabled and disabled mode.

> **Note:**
>
> The Nrf-client pod of PCF has been restarted for UDR and CHF discovery as part of each test case.

Locating and Downloading ATS Images

The steps to locate and download ATS Images are as follows:

1. Login to My Oracle Support using the appropriate credentials.
2. Select **Patches & Updates** tab.
3. In **Patch Search** console, select **Product or Family (Advanced)** tab.
4. Enter *Oracle Communications Cloud Native Core - 5G* in **Product** field and select the product from the Product drop-down.
5. Select *Oracle Communications Cloud Native Core Policy <release_number>* in **Release** field.
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 on **Download**. File Download window appears.
9. Click on the **<p********_<release_number>_Tekelec>.zip** file.
10. Extract the ATS release package zip file to download the ATS images to the system where network function must be installed.
11. The `ocats-pcf`directory has following files:
    `ocats-pcf-tools-1.1.0.0.0.tgz`
12. The `ocats-pcf-tools-1.1.0.0.0.tgz` file has following images and charts packaged as tar files:

```
ocats-pcf-tools-1.1.0.0.0.tgz

      |

      |_ _ _ocats-pcf-pkg-1.1.0.0.0.tgz

      |        |_ _ _ _ _ _ ocats-pcf-1.1.0.tgz (Helm Charts)

      |        |_ _ _ _ _ _ ocats-pcf-images-1.1.0.tar (Docker
```

```
Images)

      |

      |_ _ _ocstub-pkg-1.1.0.0.0.tgz

               |_ _ _ _ _ _ ocstub-go-1.1.0.tgz(Helm Charts)

               |_ _ _ _ _ _ ocstub-go-image-1.0.0.tar (Docker
Images)
```

13. The user can copy the tar file from here to their K8s cluster where, they want to deploy ATS.

Deploying ATS in K8s Cluster

The steps to deploy ATS in K8s Cluster are as follows:

1. Execute the following command to extract the tar file content.
   `tar -xvf ocats-pcf-tools-1.1.0.0.0.tgz`

   The output of this command is:

   ```
   ocats-pcf-pkg-1.1.0.0.0.tgz
   ocstub-pkg-1.1.0.0.0.tgz
   ```

2. Go to the `ocats-pcf-tools-1.0.0.0.0` folder and execute the given command to extract final helm charts and docker images of ATS.
   `tar -xvf ocats-pcf-pkg-1.1.0.0.0.tgz`

   The output of this command is:

   ```
   ocats-pcf-1.1.0.tgz
   ocats-pcf-images-1.1.0.tar
   ```

3. In your cluster, execute the given command to load the ATS image.
   `docker load --input ocats-pcf-images-1.1.0.tar`
4. Execute the following commands to tag and push the ATS images

   ```
   docker tag ocatspcf:1.1.0 <registry>/ocatspcf:1.1.0
         docker push <registry>/ocatspcf:1.1.0
   ```

   Example:

   ```
   docker tag ocatspcf:1.1.0 localhost:5000/ocatspcf:1.1.0
         docker push localhost:5000/ocatspcf:1.1.0
   ```

5. Untar the helm charts, `ocats-pcf-1.1.0.tgz`
   `tar -xvf ocats-pcf-1.1.0.tgz`
6. Update the image name and tag in the `ocats-pcf/values.yaml` file as required. For this, you need to open the `values.yaml` file and update the `image.repository` and `image.tag`
7. Deploy ATS using the updated helm charts after performing the previous **step 5**.
   `helm install <chart_directory> --name <release_name> --namespace <namespace_name> -f <custom_values_filename>`

For example:`helm install ocats-pcf --name ocats --namespace ocpcf -f ocats-pcf/values.yaml`

8. Verify ATS deployment by executing the given command.
   `helm status <release_name>`

Figure 3-3 Helm Status

```
[cloud-user@keezyuvpcf-bastion-1 ats]$ helm status ocats
LAST DEPLOYED: Fri May 15 10:52:06 2020
NAMESPACE: ocpcf
STATUS: DEPLOYED

RESOURCES:
==> v1/ClusterRole
NAME                                      AGE
ocpcf-ocats-ocats-ocats-pcf-clusterrole   31s

==> v1/Pod(related)
NAME                                READY  STATUS   RESTARTS  AGE
ocats-ocats-pcf-545ccb9b69-524hp    1/1    Running  0         31s

==> v1/Service
NAME             TYPE          CLUSTER-IP    EXTERNAL-IP  PORT(S)         AGE
ocats-ocats-pcf  LoadBalancer  10.233.61.51  <pending>    8080:31358/TCP  31s

==> v1/ServiceAccount
NAME                                       SECRETS  AGE
ocpcf-ocats-ocats-ocats-pcf-serviceaccount 1        31s

==> v1beta1/ClusterRoleBinding
NAME                                              AGE
ocpcf-ocats-ocats-ocats-pcf-clusterrolebinding    31s


NOTES:
# Copyright 2018 (C), Oracle and/or its affiliates. All rights reserved.

Thank you for installing ocats-pcf.

Your release is named ocats , Release Revision: 1.
To learn more about the release, try:

  $ helm status ocats
  $ helm get ocats
```

Deploying Stub Pod in K8s Cluster

The steps to deploy Stub Pod in K8s cluster are as follows:

1. Go to the `ocats-pcf-tools-1.1.0.0.0` folder and execute the command to extract the ocstub tar file content.
   `tar -xvf ocstub-pkg-1.1.0.0.0.tgz`

   The output of this command is:

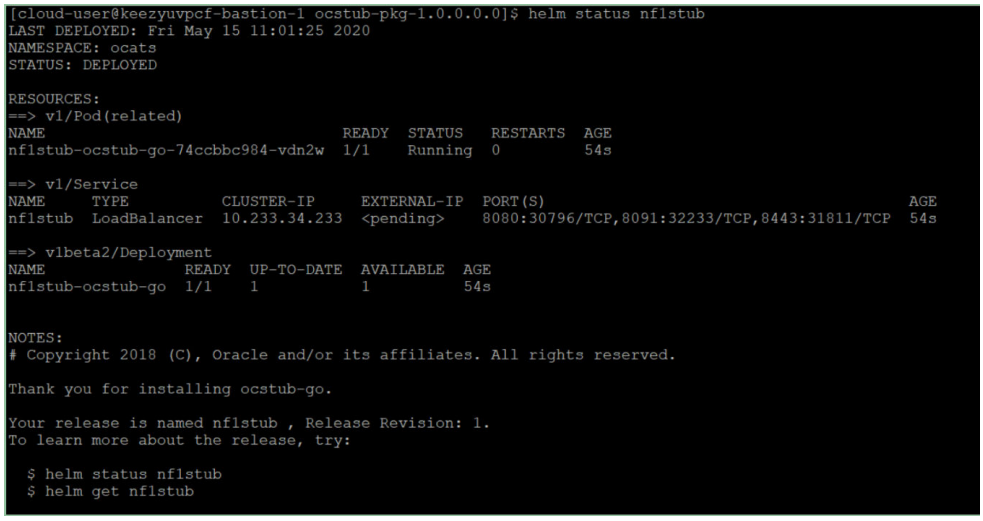
   ```
   ocstub-1.1.0.tgz
   ocstub-go-image-1.0.0.tar
   ```

2. In your cluster, execute the following command to load the STUB image and then, push it to your registry.
   `docker load --input ocstub-go-image-1.0.0.tar`
3. Untar the helm charts, `ocstub-go-1.1.0.tgz`.
   `tar -xvf ocstub-go-1.1.0.tgz`
4. Update the registry name, image name and tag (if required) in the `ocstub/values.yaml` file as required.

Open the **values.yaml** file and update the `image.repository` and `image.tag`

5. Deploy Stub.
   ```
   helm install <chart_directory> --set service.name=<service_name>
   --name <release_name> --namespace <namespace_name> -f
   <custom_values_filename>
   ```
   For example: `helm install ocstub-go --set service.name=nf1stub --name nf1stub --namespace ocats -f ocstub-go/values.yaml`
6. Similarly, execute the following commands to install all the stubs.
   ```
   helm install ocstub-go --set service.name=nf2stub --name nf2stub --
   namespace ocats -f ocstub-go/values.yaml
   ```
   ```
   helm install ocstub-go --set service.name=nf3stub --name nf3stub --
   namespace ocats -f ocstub-go/values.yaml
   ```

Figure 3-4 Stub - Check Helm Status



7. Execute the following command to check the Stub deployment.
   ```
   helm status <release_name>
   ```

A sample screen showing stubs deployment is given below:

Figure 3-5 Stubs After Installation

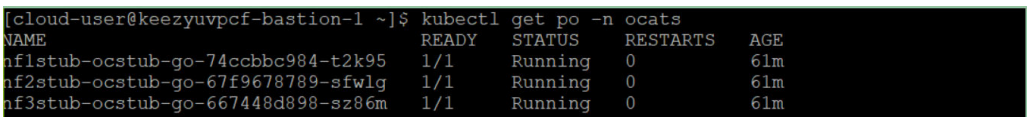
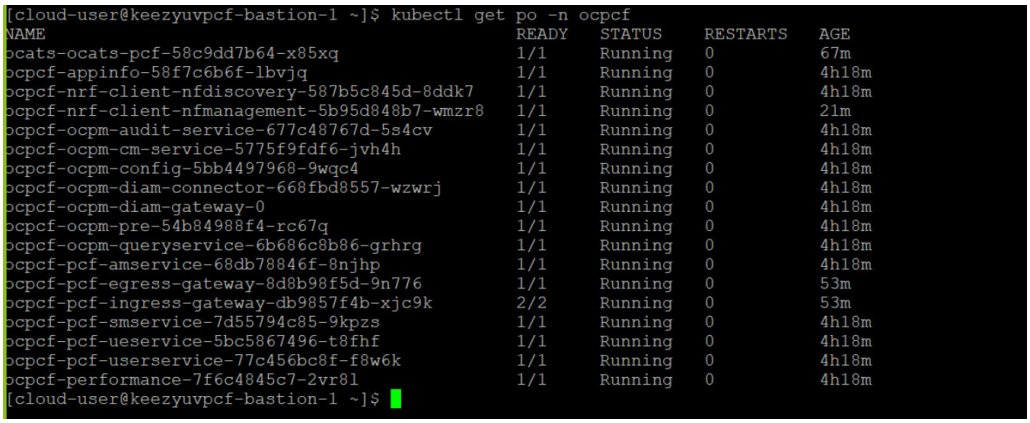


Figure 3-6 Sample Screen: PCF Namespace with PCF and ATS after Installation



```
[cloud-user@keezyuvpcf-bastion-1 ~]$ kubectl get po -n ocpcf
NAME                                          READY   STATUS    RESTARTS   AGE
ocats-ocats-pcf-58c9dd7b64-x85xq              1/1     Running   0          67m
ocpcf-appinfo-58f7c6b6f-lbvjq                 1/1     Running   0          4h18m
ocpcf-nrf-client-nfdiscovery-587b5c845d-8ddk7 1/1     Running   0          4h18m
ocpcf-nrf-client-nfmanagement-5b95d848b7-wmzr8 1/1    Running   0          21m
ocpcf-ocpm-audit-service-677c48767d-5s4cv     1/1     Running   0          4h18m
ocpcf-ocpm-cm-service-5775f9fdf6-jvh4h        1/1     Running   0          4h18m
ocpcf-ocpm-config-5bb4497968-9wqc4            1/1     Running   0          4h18m
ocpcf-ocpm-diam-connector-668fbd8557-wzwrj    1/1     Running   0          4h18m
ocpcf-ocpm-diam-gateway-0                     1/1     Running   0          4h18m
ocpcf-ocpm-pre-54b84988f4-rc67q               1/1     Running   0          4h18m
ocpcf-ocpm-queryservice-6b686c8b86-grhrg      1/1     Running   0          4h18m
ocpcf-pcf-amservice-68db78846f-8njhp          1/1     Running   0          4h18m
ocpcf-pcf-egress-gateway-8d8b98f5d-9n776      1/1     Running   0          53m
ocpcf-pcf-ingress-gateway-db9857f4b-xjc9k     2/2     Running   0          53m
ocpcf-pcf-smservice-7d55794c85-9kpzs          1/1     Running   0          4h18m
ocpcf-pcf-ueservice-5bc5867496-t8fhf          1/1     Running   0          4h18m
ocpcf-pcf-userservice-77c456bc8f-f8w6k        1/1     Running   0          4h18m
ocpcf-performance-7f6c4845c7-2vr8l            1/1     Running   0          4h18m
[cloud-user@keezyuvpcf-bastion-1 ~]$
```

SCP ATS Installation Procedure

The SCP ATS installation procedure covers two steps:

1. Locating and downloading the ATS images.
2. Deploying ATS images.

Locating and Downloading ATS Images

To locate and download ATS Images:

1. Login to My Oracle Support using the appropriate credentials.
2. Select **Patches & Updates** tab.
3. In **Patch Search** console, select **Product or Family (Advanced)** tab.
4. Enter *Oracle Communications Cloud Native Core - 5G* in **Product** field and select the product from the Product drop-down.
5. Select *Oracle Communications Cloud Native Core Service Communication Proxy <release_number>* in **Release** field.
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 on **Download**. File Download window appears.
9. Click on the **<p********_<release_number>_Tekelec>.zip** file.
10. Extract the ATS release package zip file to download the ATS images to the system where network function must be installed.
11. The `ocats-scp` directory has a following files:

```
ocats-scp-pkg-1.9.0.0.0.tgz
ocats-scp-pkg-1.9.0.0.0-readme.txt
ocats-scp-custom-configtemplates-1.9.0.0.0.zip
ocats-scp-custom-configtemplates-1.9.0.0.0-readme.txt
```

> **Note:**
>
> The `ocats-scp-custom-configtemplates-1.9.0.0.0-readme.txt` file contains all the information required for the package.

The `ocats-scp-pkg-1.9.0.0.0.tgz` file has following images and charts packaged as tar files:

```
ocats-scp-pkg-1.9.0.0.0.tgz

      |

      |_ _ _ocats-scp-pkg-1.9.0.0.0.tgz

      |        |_ _ _ _ _ _ ocats-scp-1.9.0.tgz (Helm Charts)

      |        |_ _ _ _ _ _ ocats-scp-images-1.9.0.tar (Docker
Images)

      |        |_ _ _ _ _ _ Readme.txt
```

The `ocats-scp-custom-configtemplates-1.9.0.0.0.zip` file has following images and charts packaged as tar files:

```
ocats-scp-custom-configtemplates-1.9.0.0.0.zip

            |_ _ _ _ _ _ ocats-scp-custom-serviceaccount-1.9.0.yaml
(Template to create custom service account)

            |_ _ _ _ _ _ ocats-scp-values-1.9.0.yaml (Custom values
file for installation)

            |_ _ _ _ _ ocats-scp-tests-jenkinsjobs.tgz (ocscp_tests
and jenkins jobs folder to be copied if persistent volume is
deployed)
```

The user can copy the tar file from here to their kubernetes cluster where, they want to deploy ATS.

Deploying ATS in Kuberbetes Cluster

To deploy ATS in Kubernetes Cluster:

> **Note:**
>
> Deploy ATS and SCP in the same namespace.

> **Note:**
>
> ATS is deployed with role binding by default instead of cluster role binding.

1. Execute the following command to extract the tar file content:
   `tar -xvf ocats-scp-pkg-1.9.0.0.0.tgz`

   The output of this command is:

   ```
   ocats-scp-1.9.0.tgz
   ocats-scp-images-1.9.0.tar
   Readme.txt
   ```

   The `ocats-scp-images-1.9.0.tar` file contains ocats-scp:1.9.0 (ATS Image) and ocats-gostub:1.9.0 (stub image).
2. In your cluster, execute the given command to load the ATS image and then, push it to your registry:
   `docker load --input ocats-scp-images-1.9.0.tar`

   `docker tag ocats/ocats-scp:1.9.0 <local_registry>/ocats/ocats-scp:1.9.0`

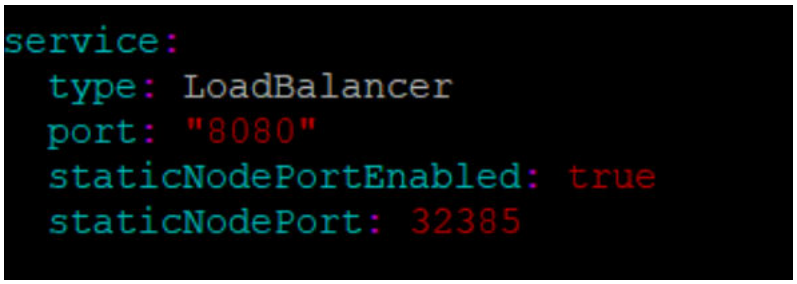
   `docker push <local_registry>/ocats/ocats-scp:1.9.0`
3. Execute the following command to extract the zip file content:

   ```
   Unzip "ocats-scp-custom-configtemplates-1.9.0.0.0.zip"
   ```

   The output of this command is:

   ```
   ocats-scp-values-1.9.0.yaml
   ocats-scp-custom-serviceaccount-1.9.0.yaml
   ocats-scp-tests-jenkinsjobs.tgz
   ```
4. Update the image name and tag in the **ocats-scp-values-1.9.0.yaml** file as required:
   For this, you need to open the `ocats-scp-values-1.9.0.yaml` file and update the `image.repository` and `image.tag`
5. ATS supports static port. By default, this feature is not available. To enable this feature:
   - In the **ocats-scp-values-1.9.0.yaml** file under service section, set the value of **staticNodePortEnabled** parameter as true and provide a valid nodePort value for **staticNodePort**.
   - A sample screen is given below:

Figure 3-7 ocats-scp-values-1.9.0.yaml- service section



> **Note:**
>
> You can enable static node port at the time of deployment.

- Add an appropriate value for the **serviceMeshCheck** parameter. Its value depends on whether ATS needs to be executed with aspen mesh or not.

6. To enable persistence storage, customer should create a PVC and associate the same to the ATS pod. The steps to enable persistent storage are:
   a. Set **PVEnabled** flag to **true**.
   b. Set **PVClaimName** to PVC that user has created for ATS.

```
deployment:
  customExtension:
    labels: {}
    annotations: {}
  PVEnabled: true
  PVClaimName: "ocats-scp-1.9.0-pvc"
```

   For more details on Persistent Volume Storage, you can refer to #unique_20.

7. Update the **lbDeployments** section of the helm deployment file in SCP ATS with the following annotations, wherein
   - 8091 port is added to fetch soothsayer pod metrics
     `traffic.sidecar.istio.io/excludeOutboundPorts: "8091"`

> **Note:**
>
> **This point is applicable only if you are planning to test ATS with service mesh. Also, do not modify this port.**

8. Execute the following command to deploy ATS:
   Using Helm 2: `helm install ocats-scp-1.9.0.tgz --name <release_name> --namespace <namespace_name> -f ocats-scp-values-1.9.0.yaml`

   Example: `helm install ocats-scp-1.9.0.tgz --name ocats-scp --namespace scpsvc-f ocats-scp-values-1.9.0.yaml`

Using Helm 3: `helm3 install <release_name> ocats-scp-1.9.0.tgz -n <namespace_name> -f ocats-scp-values-1.9.0.yaml`

Example: `helm3 install ocscp-ats ocats-scp-1.9.0.tgz -n scpsvc -f ocats-scp-values-1.9.0.yaml`

> **Note:**
>
> If there are two Helm versions on your system then, specify the version number in the Helm commands. If there is only one Helm version then there is no need to mention the version number.

9. Verify ATS deployment by executing the given command:
   `helm3 status <release_name> -n <namespace_name>`

   The following sample screen checks ATS helm release.

> **Note:**
>
> If ATS is deployed in service mesh environment, the **Ready** field for pods shows 2/2.

Figure 3-8 Checking ATS Helm Release

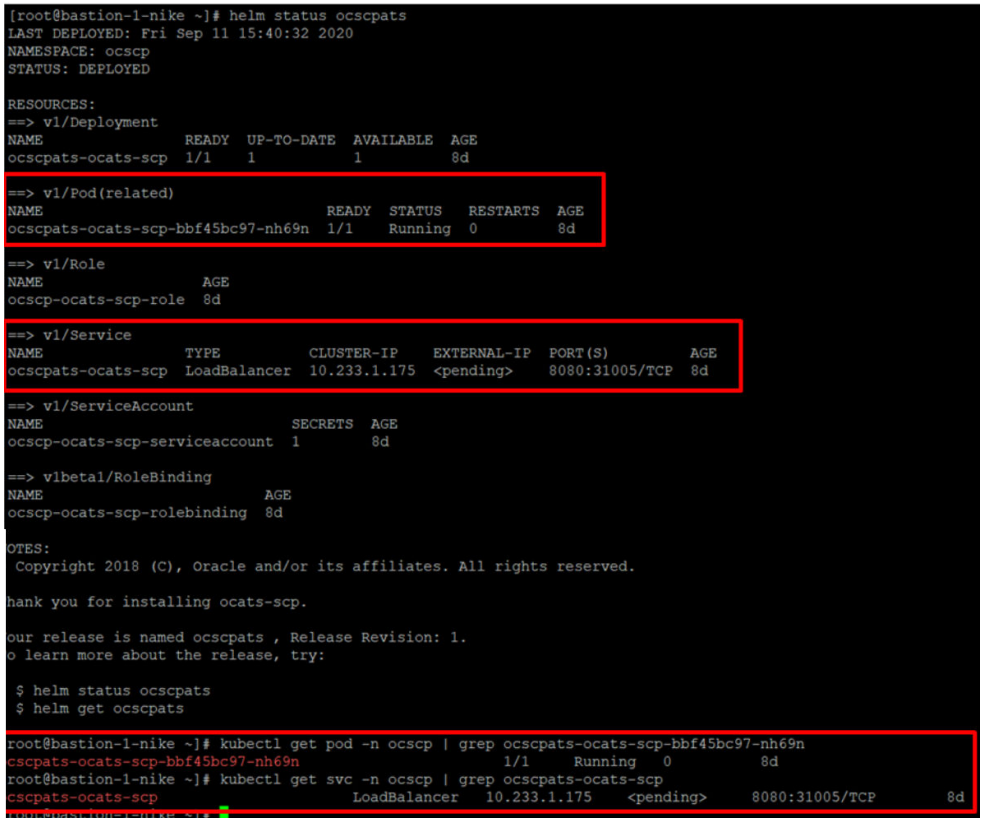
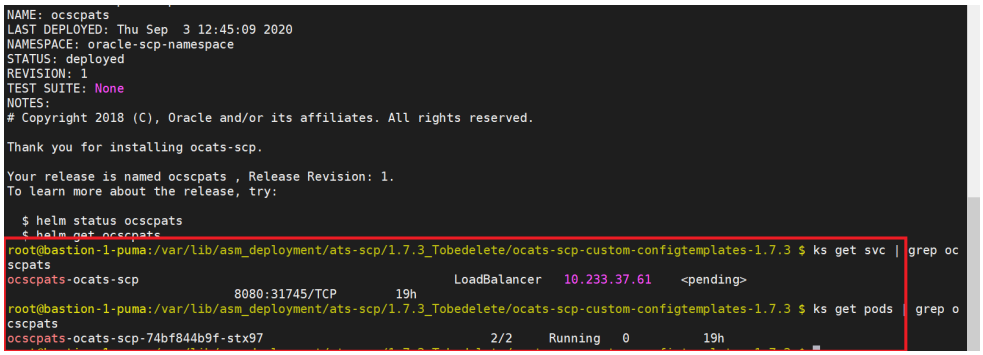


Figure 3-9 Helm Status Image



4
Executing NF Test Cases using ATS

In this chapter, you will learn to execute NF (NRF, PCF and SCP) Test Cases using ATS.

Executing NRF Test Cases using ATS

To execute NRF Test Cases using NRF ATS 1.6.1, you need to ensure that following prerequisites are fulfilled.

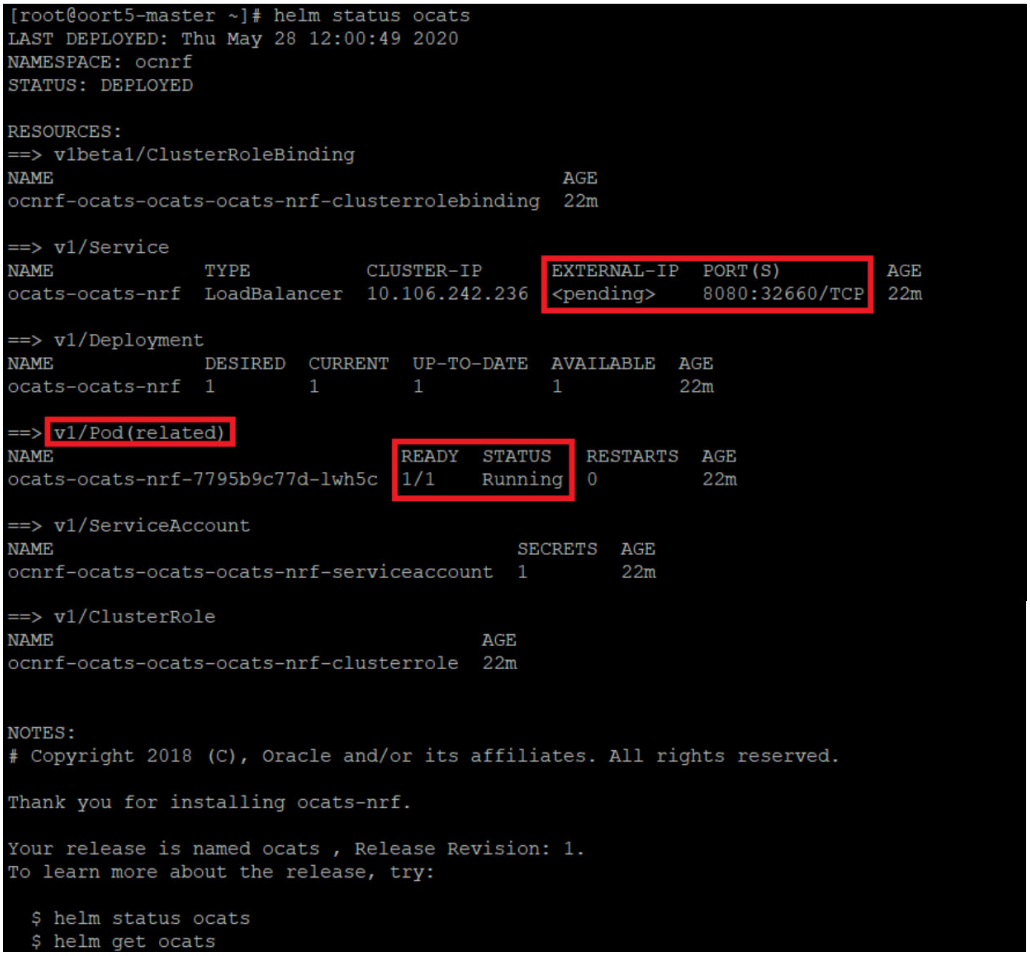
Prerequisites

- The user should create certificates/keys (public and private) for AccessToken micro-service before deploying NRF.
- The user **MUST** copy the public keys (RSA and ECDSA) created in the above step to the ATS pod at the **/var/lib/jenkins/ocnrf_tests/public_keys** location.
- Deploy NRF 1.6.1 with default helm configurations using helm charts.
- All micro-services of NRF should be up and running including Accesstoken micro-service.
- Deploy ATS using helm charts.
- Deploy Stub using helm charts.
- Deploy ATS and Stub in the same namespace as of OCNRF.
- Ensure Prometheus service is up and running.
- For NRF ATS 1.6.1, you need to deploy two stub servers for executing SLF and Forwarding functionality test cases. The service name for both the STUB servers should be **notify-stub-service** and **notify-stub-service02**.

Logging into ATS

Before logging into ATS, you need to ensure that ATS is deployed successfully using HELM charts. A sample screen is given below:

Figure 4-1 Verifying ATS Deployment



There are two ways to login to ATS Jenkins GUI.

- When an external load balancer (metalLB in case of OCCNE) is available and an external IP is provided to the ATS service, user can login to ATS GUI using `<External-IP>:8080`.
- When an external IP is not provided to the ATS service, user can open the browser and provide the external IP 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 highlighed in red as **32660**. For more details on ATS deployment, refer to NRF ATS Installation Procedure.

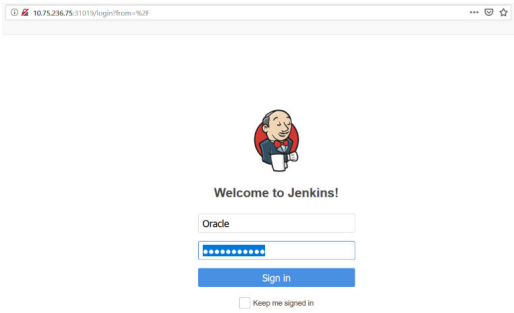
Open a browser and provide IP and port details as <Worker-Node-IP>:<Node-Port-of-ATS> (As per above example: 10.75.224.92:32660). The ATS login screen appears.

Executing ATS

To execute ATS:

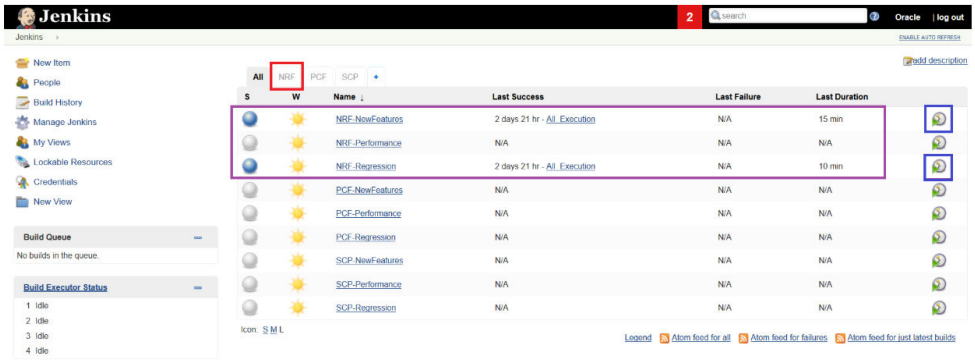
1. Enter the **username** as "Oracle" and **password** as "Welcome@123". Click **Sign in**. A sample screen is shown below.

Figure 4-2 Sample Screen: Logging into ATS GUI



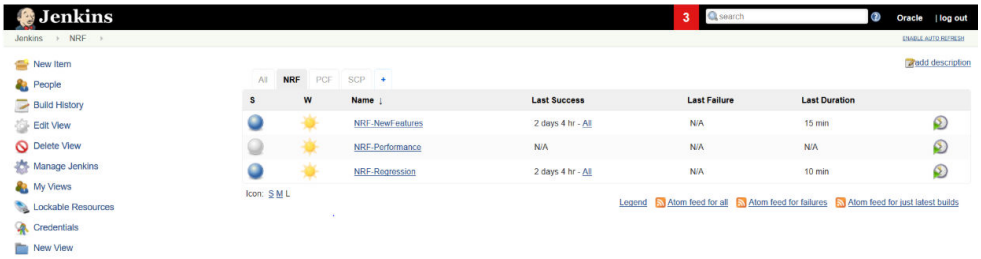
2. Following screen appears showing pre-configured pipelines for NRF, PCF and SCP individually (9 Pipelines).
   - New-Features
   - Performance
   - Regression

Figure 4-3 ATS First Logged-in Screen



By default, you are in the **All** tab. To test the NRF test cases, click the **NRF** tab. The NRF tab displays the pipelines related to NRF only.

Figure 4-4 NRF Tab - Pre-Connfigured pipelines



Jenkins ATS Pre-configured Pipelines

In the NRF tab, there are three pre-configured pipelines.

- **NRF-NewFeatures:** This pipeline has all the test cases, which are delivered as part of NRF ATS - 1.6.1
- **NRF-Performance:** This pipeline is not operational as of now. It is reserved for future releases of ATS.
- **NRF-Regression:** This pipleine has all the test cases, which were delivered in NRF ATS - 1.0.0

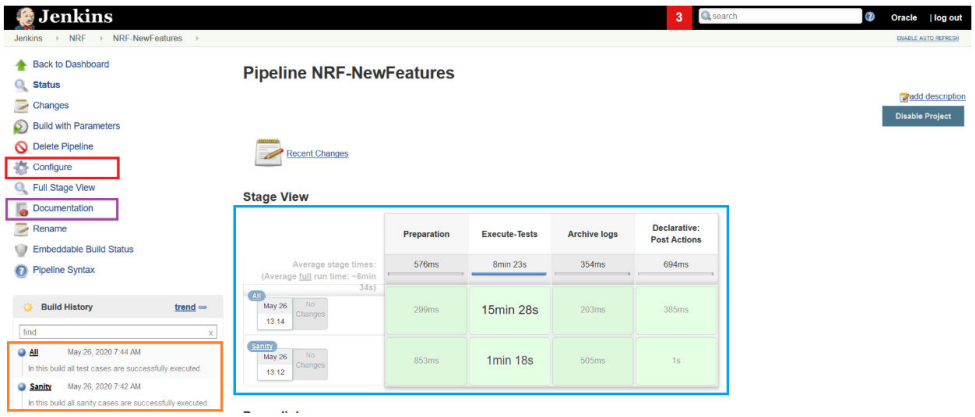
They are explained below:

NRF-NewFeatures Pipeline

After identifying the NRF pipelines, the user needs to do one-time configuration in ATS as per their SUT deployment. In this pipeline, all the new testcases related to NRF are executed. To configure its parameters:

1. Click **NRF-NewFeatures** in the Name column. Following screen appears:

Figure 4-5 NRF-NewFeatures-Configure

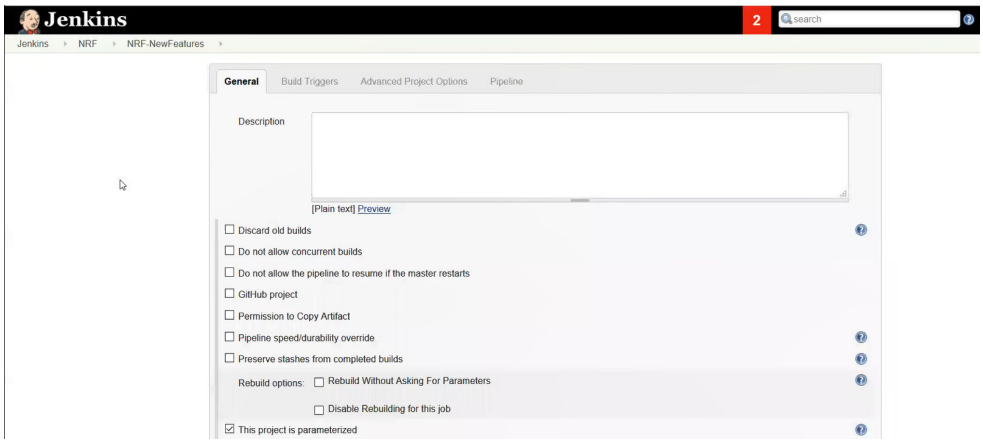


In the above screen:

- If you click **Configure**, you are navigated to the screen where configuration needs to be done.
- If you click **Documentation**, you are navigated to the screen that has documented test cases, which are part of this NRF release.
- If you click on blue dots inside **Build History** box, you are redirected to the success console logs of the "All" and "Sanity" respectively.
- The **Stage View** represents the already executed pipeline for the customer reference.

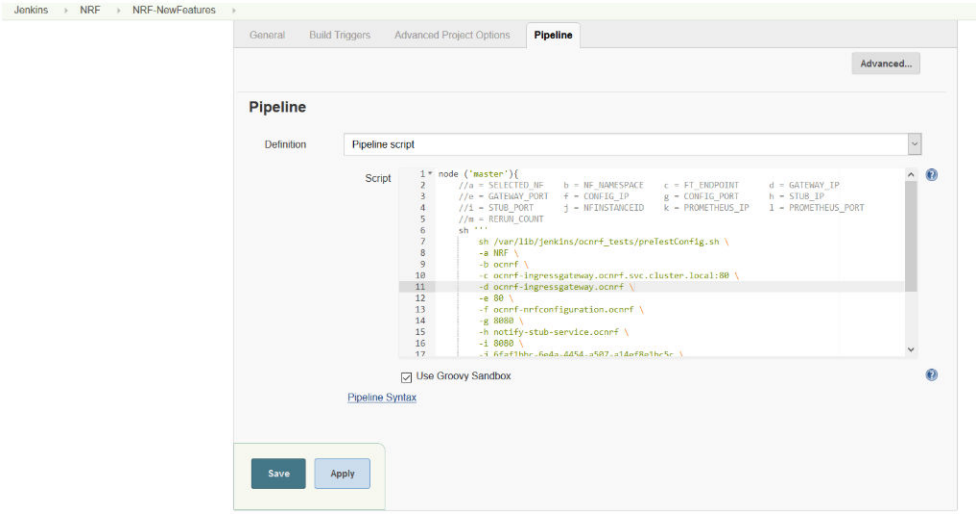
2. Click **Configure**. The following screen appears:

Figure 4-6 NRF-NewFeatures - On Configure



3. User **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:
MAKE SURE THAT THE SCREEN SHOWN ABOVE LOADS COMPLETELY BEFORE YOU PERFORM ANY ACTION ON IT. ALSO, DO NOT MODIFY ANY CONFIGURATION OTHER THAN DISCUSSED BELOW.

Figure 4-7 Pipeline Script



In the above screen, the values of the '**Pipeline script**' needs to be changed. The content of the pipeline script is as follows:

```
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
    sh '''
```

```
        sh /var/lib/jenkins/ocnrf_tests/preTestConfig.sh \
        -a NRF \
        -b ocnrf \
        -c ocnrf-ingressgateway.ocnrf.svc.cluster.local:80 \
        -d ocnrf-ingressgateway.ocnrf \
        -e 80 \
        -f ocnrf-nrfconfiguration.ocnrf \
        -g 8080 \
        -h notify-stub-service.ocnrf \
        -i 8080 \
        -j 6faf1bbc-6e4a-4454-a507-a14ef8e1bc5c \
        -k occne-prometheus-server.occne-infra \
        -l 80 \
        -m 2
    '''
    load "/var/lib/jenkins/ocnrf_tests/jenkinsData/Jenkinsfile-
NewFeatures"
}
```

> **Note:**
>
> The User **MUST NOT** change any other value apart from **line number 8** to **line 20**.

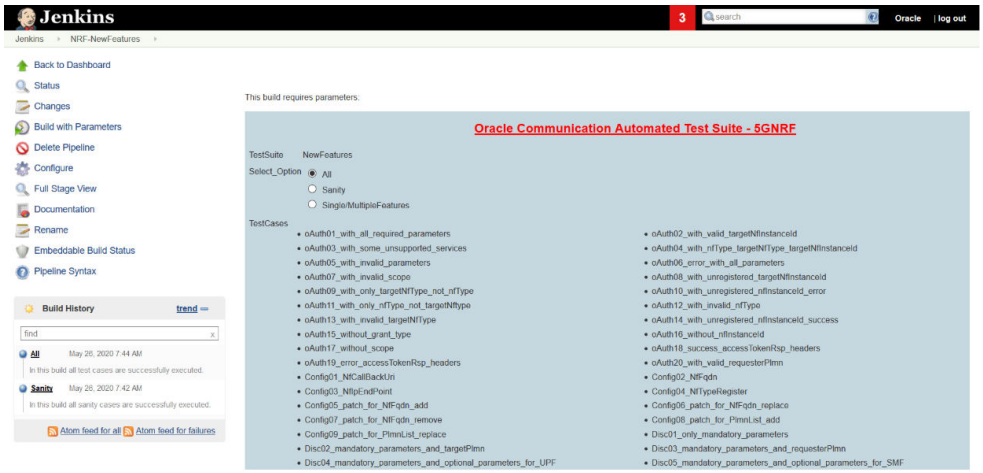
The parameters marked as **"a"** to **"m"** are only that you need to change as per the user requirement. The details about these parameters are provided as comments in **line number 2 - to -5**.

- **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 the NRF using the helm chart
- **d** - Name_of_NRF_ingressgateway_service.namespace (ocnrf-nrfconfiguration.ocnrf) - this is also known as as cluster_domain.
- **e** - Port of ingressgateway service (80)
- **f** - Name_of_NRF_configuration_service.namespace (ocnrf-nrfconfiguration.ocnrf)
- **g** - Port of configuration service (8080)
- **h** - Name_of_stub_service.namespace (notify-stub-service.ocnrf)
- **i** - Port of stub service (8080)
- **j** - NRF_Instance ID (6faf1bbc-6e4a-4454-a507-a14ef8e1bc5c)
- **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).

> **Note:**
>
> You do not have to change any value if OCCNE cluster is used and NRF, ATS and STUB are deployed in ocnrf namespace.

Click **Save** after making neccesary changes. You are navigated back to the NRF-NewFeatures screen. Click **Build with Parameters**. Following screen appears:

Figure 4-8 NRF-NewFeatures - Build with Parameters



Executing NRF Test Cases

To execute NRF test cases:

1. Click the **Schedule a Build with parameters for NRF-NewFeatures** icon present in extreme right column corresponding to **NRF-NewFeatures** row as shown below.

Figure 4-9 New-Features Pipeline Icon

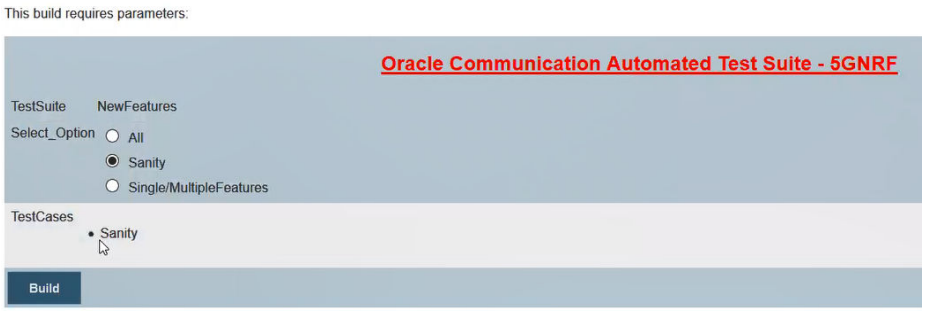
2. The following screen appears.

Figure 4-10 Build Requires Parameters

In the above screen, there are three **Select_Option**(s), which are:

- **All:** By default, all the NRF test cases are selected for execution. User just need to scroll down and click **Build** to execute all the test cases.
- **Sanity:** It is recommended to execute Sanity before executing any test case. This helps to ensure that all the deployments are done properly or not. When you select Sanity, the following screen appears:

Figure 4-11 Build Requires Parameters - Sanity



Click **Build** to execute all the sanity test cases.

- **Single/MultipleFeatures:** This option allows you to select any number of test cases that you want to execute from the list of total test cases available for execution. After selecting the test cases, scroll-down and click **Build**. The selected NRF test cases are executed.

The NRF testcases are divided into NRF Service operations as follows:

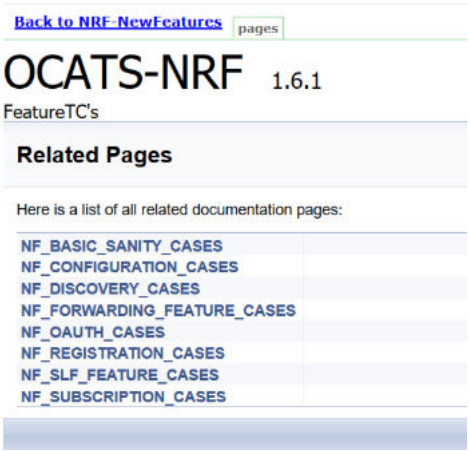
- **AccessToken** - These feature files are listed with a prefix as **"oAuth"**.
- **Configuration** - These feature files are lited with a prefix as **"Config"**.
- **Discovery** - These feature files are lited with a prefix as **"Disc"**.

- **NRF Forwarding** - These feature files are lited with a prefix as **"Forwarding"**.
- **Registration** - These feature files are lited with a prefix as **"Upd"**. These are related to update operation of registered profiles.
- **NRF SLF** - These feature files are lited with a prefix as **"SLF"**.
- **NRF Sanity** - This feature file contains all the basic sanity cases for NRF ATS 1.6.1.
- **Subscription** - These feature files are lited with a prefix as **"Subs"**.

NewFeatures - Documentation

To view NRF functionalities, go to NRF-NewFeatures pipeline and click **Documentation** link in the left navigation pane. The following screen appears:

Figure 4-12 NRF-NewFeatures - Documentation

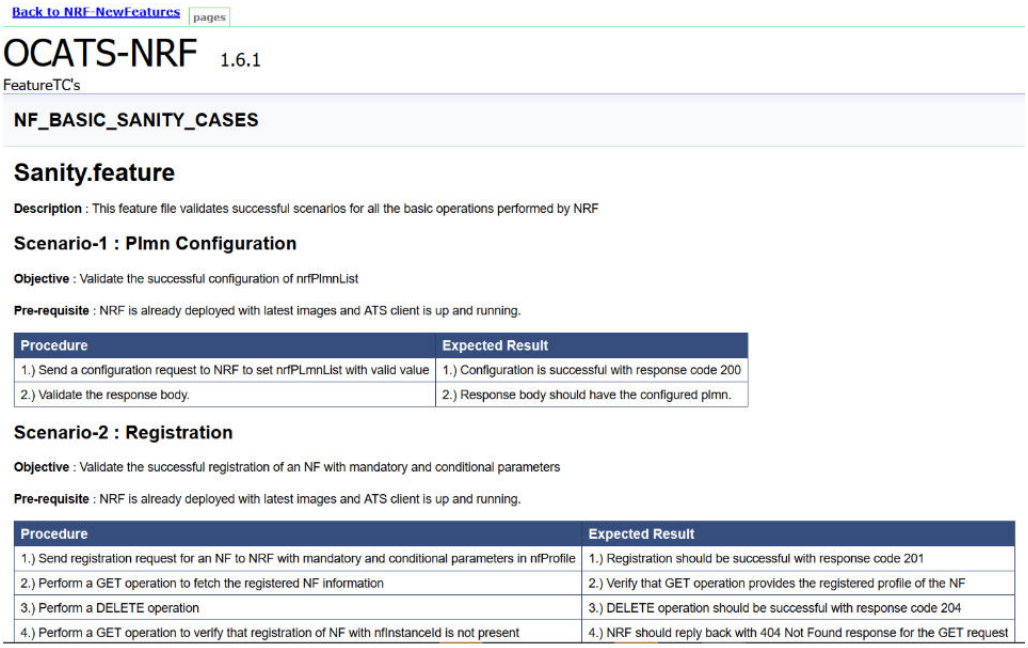


Each one of the documentation feature is described below:

- **NF_BASIC_SANITY_CASES** - Lists all the sanity cases, which are useful to identify whether all the NRF functionality works fine.
- **NF_CONFIGURATION_CASES** - Lists all the cases related to NRF configuration.
- **NF_DISCOVERY_CASES** - Lists all the discovery microservice related cases.
- **NF_FORWARDING_FEATURE_CASES** - Lists all the forwarding related cases.
- **NF_OUTH_CASES** - Lists all the accesstoken related cases.
- **NF_REGISTRATION_CASES** - Lists all the registration related cases.
- **NF_SLF_FEATURE_CASES** - Lists all the SLF related cases.
- **NF_SUBSCRIPTION_CASES** - Lists all subscription related cases.

You can click any functionality to view its test cases and scenarios of each test case. A sample screen is given below:

Figure 4-13 NRF - Feature Detail



Back to NRF-NewFeatures pages

OCATS-NRF 1.6.1

FeatureTC's

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

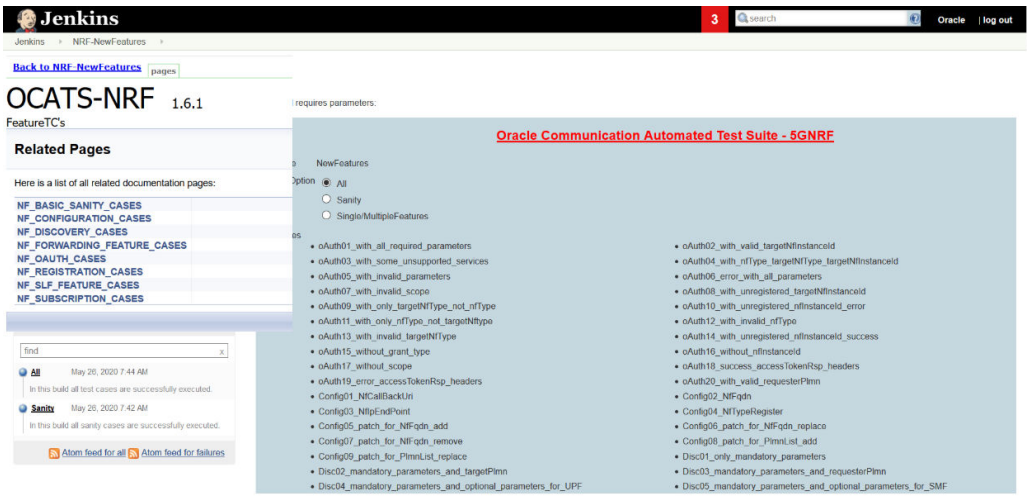
Objective : Validate the successful registration of an NF with mandatory and conditional parameters

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
3.) Perform a DELETE operation	3.) DELETE operation should be successful with response code 204
4.) Perform a GET operation to verify that registration of NF with nfInstanceId is not present	4.) NRF should reply back with 404 Not Found response for the GET request

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 **Back to NRF-NewFeatures** link available on top left corner of the screen.

Figure 4-14 Build Requires Parameters in sync with Documentation



Jenkins

Jenkins › NRF-NewFeatures

Back to NRF-NewFeatures pages

OCATS-NRF 1.6.1

FeatureTC's

Related Pages

Here is a list of all related documentation pages:

NF_BASIC_SANITY_CASES
NF_CONFIGURATION_CASES
NF_DISCOVERY_CASES
NF_FORWARDING_FEATURE_CASES
NF_OAUTH_CASES
NF_REGISTRATION_CASES
NF_SLF_FEATURE_CASES
NF_SUBSCRIPTION_CASES

requires parameters:

Oracle Communication Automated Test Suite - 5GNRF

NewFeatures

Option All / Sanity / Single/MultipleFeatures

- oAuth01_with_all_required_parameters
- oAuth02_with_valid_targetNfInstanceId
- oAuth03_with_some_unsupported_services
- oAuth04_with_nfType_targetNfType_targetNfInstanceId
- oAuth05_with_invalid_parameters
- oAuth06_error_with_all_parameters
- oAuth07_with_invalid_scope
- oAuth08_with_unregistered_targetNfInstanceId
- oAuth09_with_only_targetNfType_not_nfType
- oAuth10_with_unregistered_nfInstanceId_error
- oAuth11_with_only_nfType_not_targetNftype
- oAuth12_with_invalid_nfType
- oAuth13_with_invalid_targetNfType
- oAuth14_with_unregistered_nfInstanceId_success
- oAuth15_without_grant_type
- oAuth16_without_nfInstanceId
- oAuth17_without_scope
- oAuth18_success_accessTokenRsp_headers
- oAuth19_error_accessTokenRsp_headers
- oAuth20_with_valid_requesterPlmn
- Config01_NfCallBackUri
- Config02_NfFqdn
- Config03_NfIpEndPoint
- Config04_NfTypeRegister
- Config05_patch_for_NfFqdn_add
- Config06_patch_for_NfFqdn_replace
- Config07_patch_for_NfFqdn_remove
- Config08_patch_for_PlmnList_add
- Config09_patch_for_PlmnList_replace
- Disc01_only_mandatory_parameters
- Disc02_mandatory_parameters_and_targetPlmn
- Disc03_mandatory_parameters_and_requesterPlmn
- Disc04_mandatory_parameters_and_optional_parameters_for_UPF
- Disc05_mandatory_parameters_and_optional_parameters_for_SMF

In the above screen, you can notice that the initials of test cases are similar to the functionalities listed in the contents of Documentation screen. The details are as follows:

- **oAuth01 to 0Auth20:** These 20 feature files belong to the NF_OAUTH_CASES functionality.

- **Config01 to Config09:** These 9 feature files belong to the NF_CONFIGURATION_CASES functionality.
- **Disc01 to Disc20:** These 20 feature files belong to the NF_DISCOVERY_CASES functionality.
- **Forwarding01 to Forwarding05:** These 5 feature files belong to the NF_FORWARDING_FEATURE_CASES functionality.
- **Upd01 to Upd20:** These 20 feature files belong to the NF_REGISTRATION_CASES functionality.
- **SLF01 to SLF02:** These 2 feature files belong to the NF_SLF_FEATURE_CASES functionality.
- **Sanity:** This feature file belongs to the NF_BASIC_SANITY_CASES functionality.
- **Subs21 to Subs39:** These 19 feature files belong to the NF_SUBSCRIPTION_CASES functionality.

> **Note:**
>
> Here, the initials starts from Subs21 because the first 20 feature files covering Subs01 to Subs20 were delivered in ATS Release 1.0.0

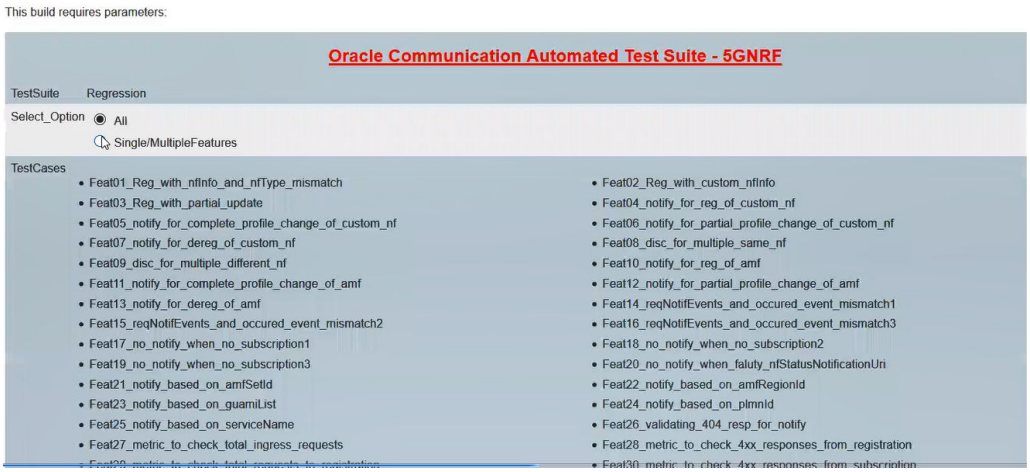
- **SubsPatch01 to SubsPatch04:** These 4 test cases also belong to the NF_SUBSCRIPTION_CASES functionality.

NRF-Regression Pipeline

This pre-configured pipeline has all the test cases of previous releases like ATS Release 1.0.0. The configuration method and parameters are same as the **NewFeatures** pipeline. Only difference in this pipeline is that it does not have **"Sanity"** option. This is so because these cases are already executed as part of previous release.

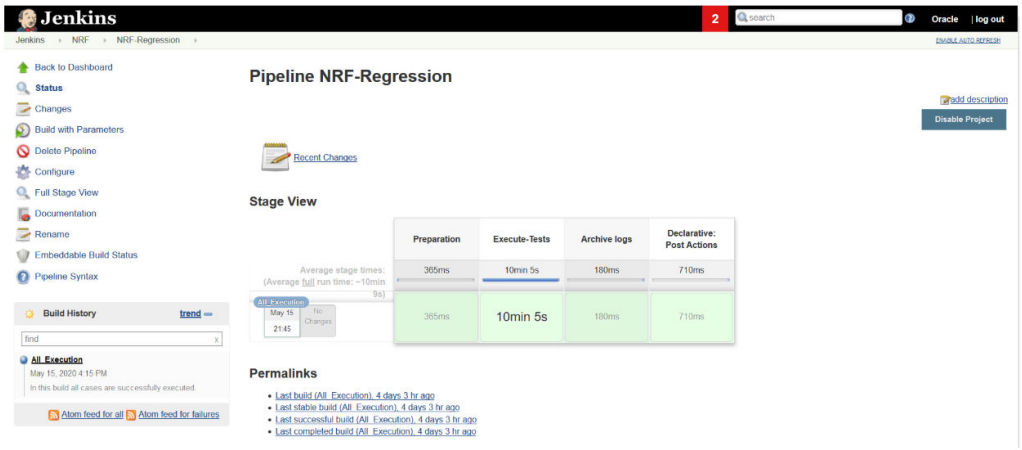
To view Regression pipeline details, click **Build with Parameters** in the left navigation pane. The following screen appears:

Figure 4-15 Regression - Build with Parameters



A sample screen of full successful execution is provided below as part of ATS image.

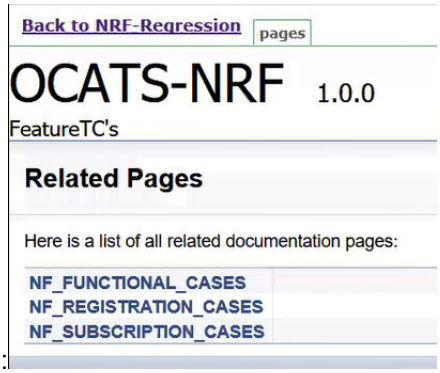
Figure 4-16 NRF Regression



Regression - Documentation

Click **Documentation** in the left navigation pane of the NRF-Regression pipeline.

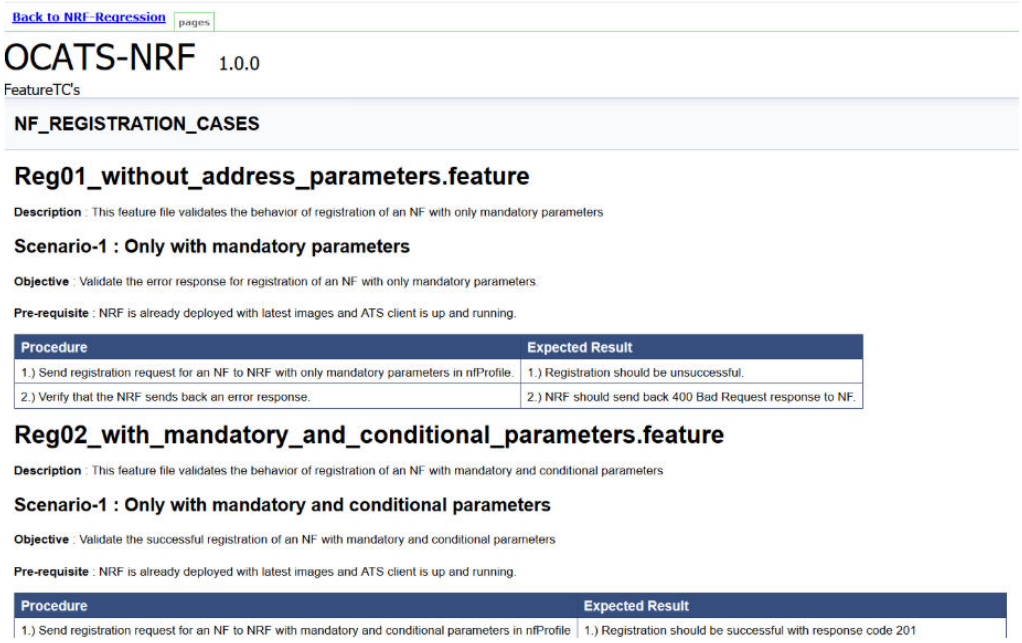
Following screen appears:



This screen shows only those functionalities whose test cases were released in previous releases.

A sample screen showing documentation for NRF ATS - 1.6.1 is given below:

Figure 4-17 Sample Screen: Documentation for NRF ATS


Back to NRF-Regression pages

OCATS-NRF 1.0.0

FeatureTC's

NF_REGISTRATION_CASES

Reg01_without_address_parameters.feature

Description : This feature file validates the behavior of registration of an NF with only mandatory parameters

Scenario-1 : Only with mandatory parameters

Objective : Validate the error response for registration of an NF with only mandatory 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 only mandatory parameters in nfProfile.	1.) Registration should be unsuccessful.
2.) Verify that the NRF sends back an error response.	2.) NRF should send back 400 Bad Request response to NF.

Reg02_with_mandatory_and_conditional_parameters.feature

Description : This feature file validates the behavior of registration of an NF with mandatory and conditional parameters

Scenario-1 : Only with mandatory and conditional parameters

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

Executing PCF Test Cases using ATS

To execute PCF Test Cases, you need to ensure that following prerequisites are fulfilled.

Prerequisites

- ATS 1.1.0 is compatible with PCF 1.6.0 and 1.6.1 for both TLS (only server side) enabled and disabled.
  - PCF with TLS disabled. In the PCF's custom values file, check if the following parameters are configured with the respective values:

    ```
    ingress-gateway:
    enableIncomingHttps: false
    egress-gateway:
    enableOutgoingHttps: false
    ```

  - If TLS is enabled, then you first need to enable the **Https** support for Egress and Ingress gateway. For more details, refer to the **Enabling Https support for Egress and Ingress Gateway** section in this topic.
- ATS is deployed using helm charts in the same namespace where PCF is deployed.
- Prometheus server is installed with Loadbalancer as the pod type.
- To get all configmaps in your namespace, execute the following command.
  `kubectl get configmaps -n <pcf_namespace>`

  In the **application-config** configmap, configure the following parameters with the respective values:
  `primaryNrfApiRoot=http://nf1stub.<stub_namespace>.svc:8080`

Example: If stubs are deployed in "ocats" namespace,

```
primaryNrfApiRoot=http://nf1stub.ocats.svc:8080

nrfClientSubscribeTypes=UDR,CHF
```

- Database cluster should be in a running state with all the required tables. You need to ensure that there are no previous entries in database before executing test cases.

Enabling TLS in ATS Pod

You can enable TLS in ATS pod after successful deployment of PCF (TLS enabled server side) and ATS. To enable TLS in ATS Pod:

1. Execute the following command to copy the caroot.cer generated while PCF deployment to ATS pod in "cert" directory.

   ```
   kubectl cp  <path_to_file>/caroot.cer <namespace>/<ATS-Pod-name>:
   /var/lib/jenkins/cert/ -n <namespace>
   ```

   Example:

   ```
   kubcetl cp cert/caroot.cer ocpcf/ocpcf-ocats-pcf-56754b9568-rkj8z:
         /var/lib/jenkins/cert/
   ```

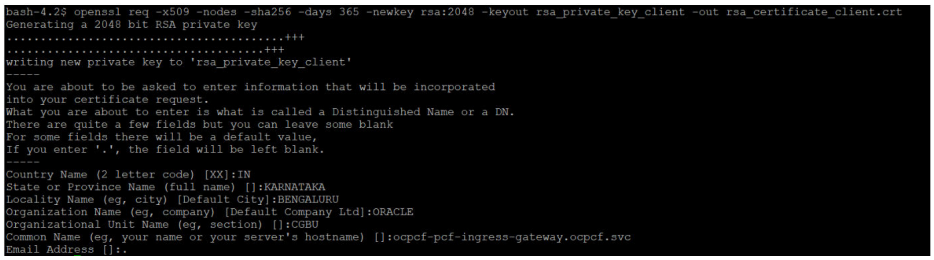
2. Execute the following command to login to your ATS Pod.

   ```
   kubectl exec -it <ATS-Pod-name> bash -n <namespace>
   ```

3. Execute the following commands from cert directory to create private key and certificates:

   a. `openssl req -x509 -nodes -sha256 -days 365 -newkey rsa:2048 -keyout rsa_private_key_client -out rsa_certificate_client.crt`

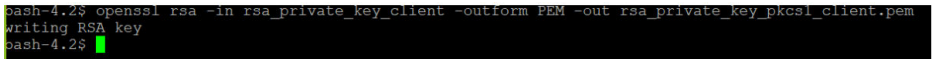
   Figure 4-18 Command 1



> **Note:**
>
> You need to provide appropriate values and specify fqdn of PCF Ingress Gateway service i.e. <ingress-servicename>.<pcf_namespace>.svc in Common Name.

b. `openssl rsa -in rsa_private_key_client -outform PEM -out rsa_private_key_pkcs1_client.pem`

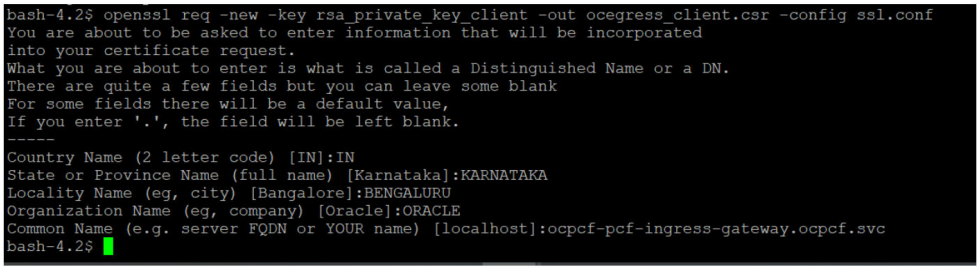
Figure 4-19 Command 2



c. `openssl req -new -key rsa_private_key_client -out ocegress_client.csr -config ssl.conf`

> **Note:**
>
> You can either use or copy the ssl.conf file, which was used while deploying PCF to ATS pod for this step.

Figure 4-20 Command 3



4. Execute the following command to copy the ocegress_client.csr to the bastion.

```
openssl x509 -CA caroot.cer -CAkey cakey.pem -CAserial serial.txt
-req
      -in ocegress_client.csr -out ocegress_client.cer -days 365
-extfile
    ssl.conf -extensions req_ext
```

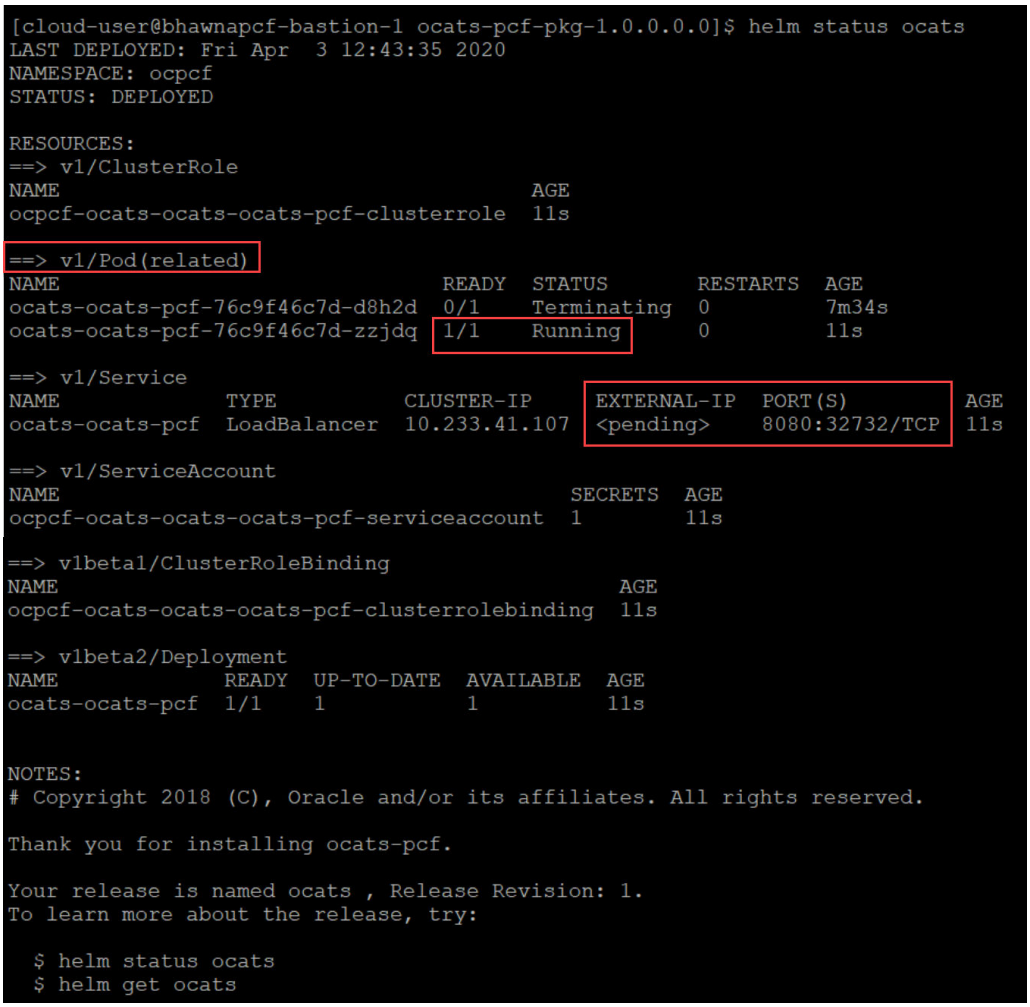
Figure 4-21 Copying ocegress_client.csr to bastion

5. Copy the **ocegress_client.cer** from Bastion to the ATS Pod.
6. Restart the ingress and egress gateway pods from the Bastion.

Logging into ATS

Before logging into ATS, you need to ensure that ATS is deployed successfully using HELM charts. A sample screen is given below:

Figure 4-22 Sample Screen: Verifying ATS Deployment



```
[cloud-user@bhawnapcf-bastion-1 ocats-pcf-pkg-1.0.0.0.0]$ helm status ocats
LAST DEPLOYED: Fri Apr  3 12:43:35 2020
NAMESPACE: ocpcf
STATUS: DEPLOYED

RESOURCES:
==> v1/ClusterRole
NAME                                      AGE
ocpcf-ocats-ocats-ocats-pcf-clusterrole   11s

==> v1/Pod(related)
NAME                                READY  STATUS       RESTARTS  AGE
ocats-ocats-pcf-76c9f46c7d-d8h2d    0/1    Terminating  0         7m34s
ocats-ocats-pcf-76c9f46c7d-zzjdq    1/1    Running      0         11s

==> v1/Service
NAME             TYPE          CLUSTER-IP     EXTERNAL-IP  PORT(S)          AGE
ocats-ocats-pcf  LoadBalancer  10.233.41.107  <pending>    8080:32732/TCP   11s

==> v1/ServiceAccount
NAME                                        SECRETS  AGE
ocpcf-ocats-ocats-ocats-pcf-serviceaccount  1        11s

==> v1beta1/ClusterRoleBinding
NAME                                             AGE
ocpcf-ocats-ocats-ocats-pcf-clusterrolebinding   11s

==> v1beta2/Deployment
NAME             READY  UP-TO-DATE  AVAILABLE  AGE
ocats-ocats-pcf  1/1    1           1          11s


NOTES:
# Copyright 2018 (C), Oracle and/or its affiliates. All rights reserved.

Thank you for installing ocats-pcf.

Your release is named ocats , Release Revision: 1.
To learn more about the release, try:

  $ helm status ocats
  $ helm get ocats
```

There are two ways to login to ATS Jenkins GUI.

- When an external load balancer is available and an external IP is provided to the ATS service, user can login to ATS GUI using `<External-IP>:8080`.
- When an external IP is not provided to the ATS service, user can open the browser and provide the external IP of the worker node and nodeport of the ATS service to login to ATS GUI.
  `<Worker-Node-IP>:<Node-Port-of-ATS>`

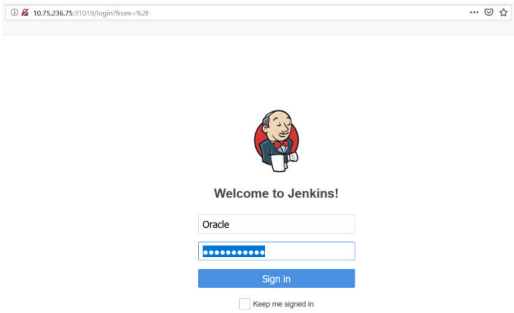
> **Note:**
>
> In the **Verifying ATS Deployment** screen, the nodeport, '32732' is highlighted in red. For more information on ATS deployment in PCF, refer to Policy ATS Installation Procedure.

Executing ATS

To execute ATS:

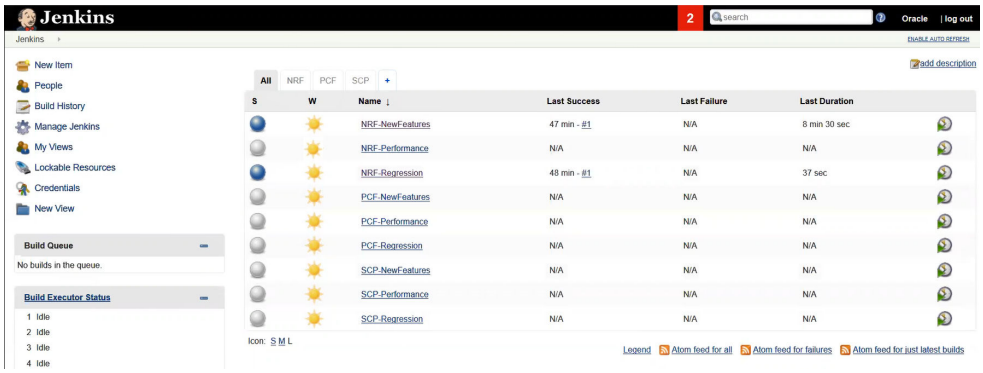
1. Enter the **username** as "Oracle" and **password** as "Welcome@123". Click **Sign in**. A sample screen is shown below.

Figure 4-23 Sample Screen: Logging into ATS GUI



2. Following screen appears showing pre-configured pipelines for NRF, PCF and SCP individually (9 Pipelines).
   - New-Features
   - Performance
   - Regression

Figure 4-24 Jenkins First Screen on Login



By default, you are in the **All** tab. To test the PCF test cases, click the **PCF** tab. The PCF tab displays the pipelines related to PCF only.

Figure 4-25 PCF Tab

Jenkins ATS Pre-configured Pipelines

In the PCF tab, there are three pre-configured pipelines.

- **PCF-NewFeatures**
- **PCF-Performance:** This pipeline is not operational as of now. It is reserved for future releases of ATS.
- **PCF-Regression**

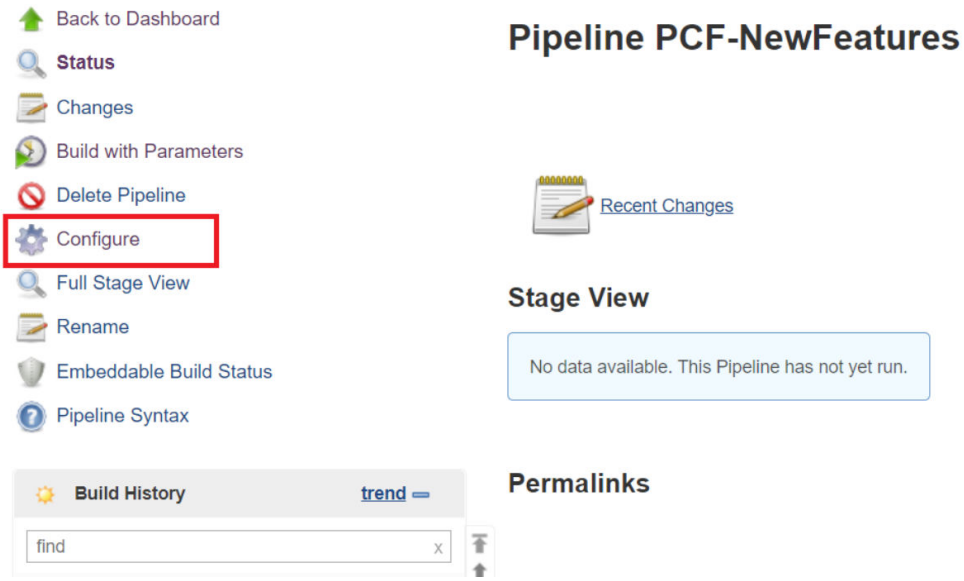
They are explained below:

PCF-New Features Pipeline

This is a pre-configured pipeline where all the PCF test cases are executed. To configure its parameters, which is a one time activity:

1. Click **PCF-NewFeatures** in the Name column and then, click **Configure** in the left navigation pane as shown below:

Figure 4-26 PCF-NewFeatures - Configure



2. The PCF-NewFeatures, **General** tab appears. Make sure that the screen loads completely.
3. Scroll-down to the end. The control moves from **General** tab to the **Advanced Project Options** tab as shown below:

Figure 4-27 Advanced Project Options

In the **Script** area of the Pipeline section:

- Change parameter "b" to update the namespace where PCF was deployed in the bastion.
- Change parameter "c" for prometheus server service name.
- Change parameter 'd' option for gostub namespace.
- If you intend to run ATS in TLS disabled mode, then you need to set parameter "e" as "unsecure", else set parameter e as "secure".
- Parameter "f" is to set nfInstanceId provided in application configmap of PCF.
- Parameter "g" should not be set less than 20 secs. This is because the default time for the nrf-client-management pod to come up on restart is set to 20 secs and every TC requires nrf-client-management pod to be restarted.
- Parameter "h" should not be set less than 60 secs. This is the default wait time given for the configured policy to be added to the Database.
- Parameter "i" should not be set less than 140 secs. The default wait time for Nf_Notification Test Cases is set to 140 secs. You can modify this as per requirement.
- DO NOT MODIFY ANYTHING OTHER THAN PARAMETERS VALUE.
- Press **Save** after updating the parameters value.

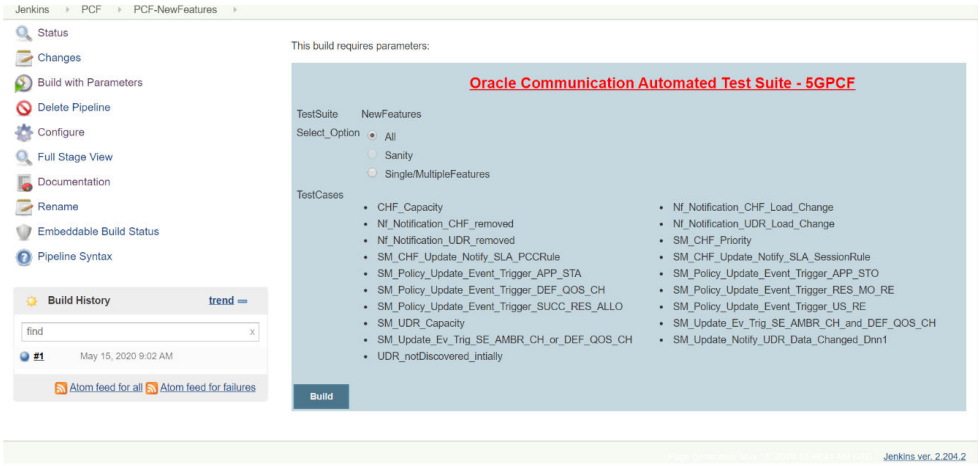
> **Note:**
>
> It is recommended to save the pipeline scipt in your local machine. This would be helpful in the event of ats pod restarts.

Executing PCF Test Cases

To execute PCF test cases:

1. Click the **Build with Parameters**. The following screen appears.

Figure 4-28 PCF - Build with Parameters



In the above screen, there are three **Select_Option**(s), which are:

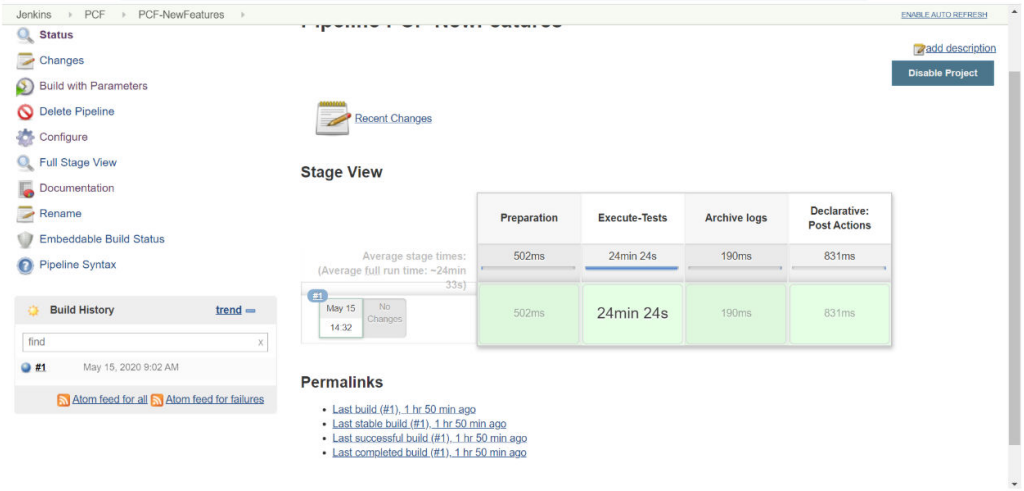
- **All:** By default, all the PCF test cases are selected for execution. User just need to scroll down and click **Build** to execute all the test cases.
- **Sanity:** This option is **NOT AVAILABLE** for PCF.
- **Single/MultipleFeatures:** This option allows you to select any number of test cases that you want to execute from the list of total test cases available for execution. After selecting the test cases, scroll-down and click **Build**. The selected PCF test cases are executed.

Figure 4-29 Sample Test Output in Console

```
2020-03-11 13:09:58,408□[32m     INFO LOG.CleanUP:343 □[0m| □[32mDeleting CONFIGURATIONS pcf.smservice.cfg/pcf.smservice.cfg□[0m
2020-03-11 13:09:58,436□[32m     INFO LOG.Pcf_restapi:174 □[0m| □[32m[+] DELETE 200 @http://pcf-ocpm-cm-
service.ocpcf:5808/ppi/config/pcf.smservice.cfg/pcf.smservice.cfg□[0m

10 features passed, 0 failed, 0 skipped
10 scenarios passed, 0 failed, 0 skipped
442 steps passed, 0 failed, 0 skipped, 0 undefined
Took 25m13.139s
[Pipeline] echo
cd /var/lib/jenkins/ocpcf_tests;unset http_proxy https_proxy;export gostub=true;export PCF_NAMESPACE=ocpcf;export PROMSVCNAME=occne-prometheus-server;sh
re-run.sh 1
[Pipeline] sh
+ cd /var/lib/jenkins/ocpcf_tests
+ unset http_proxy https_proxy
```

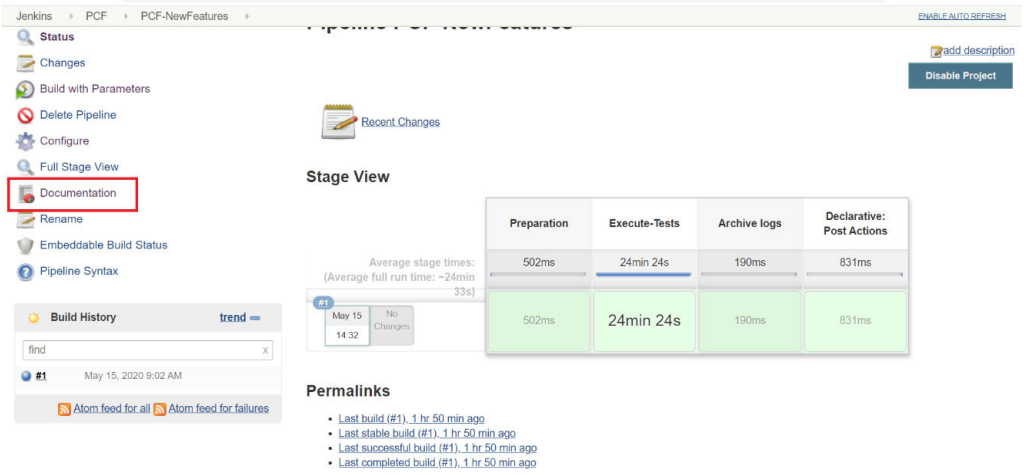
Figure 4-30 Sample output of build status - Jenkins PCF-NewFeatures Pipeline



NewFeatures - Documentation

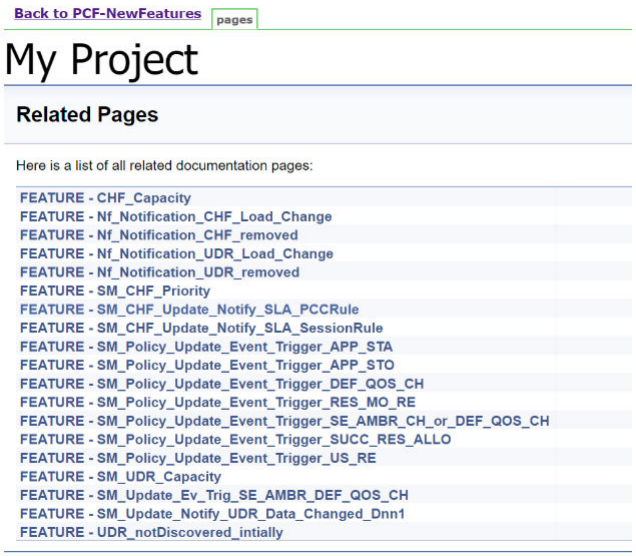
To view PCF functionalities, go to PCF-NewFeatures pipeline and click **Documentation** link in the left navigation pane as shown below:

Figure 4-31 PCF Documentation Option



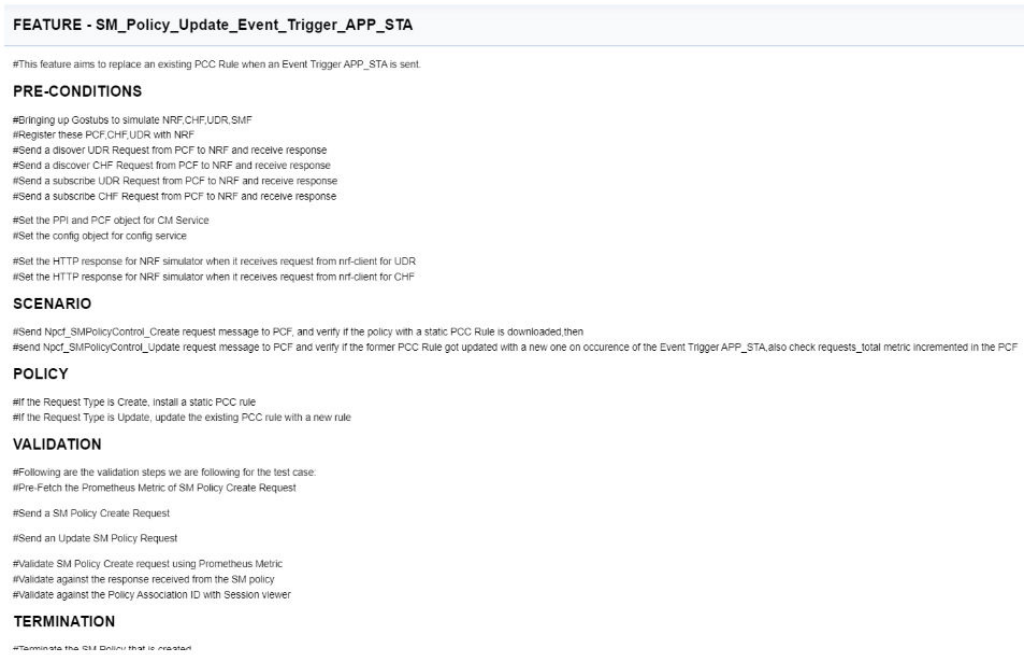
The following screen appears:

Figure 4-32 PCF-NewFeatures - Documentation



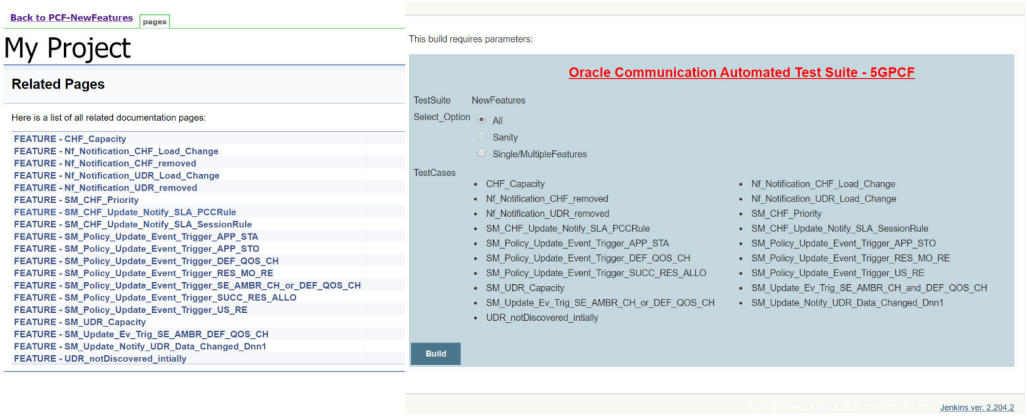
You can click any functionality to view its test cases and scenarios of each test case. For example, on click of SM_Update_Event_Trigger_APP_STA, the following test description appears:

Figure 4-33 SM_Update_Event_Trigger_APP_STA Description



Based on the functionalities covered under Documentation, the **Build Requires Parameters** screen displays test cases. To navigate back to the Pipeline PCF-NewFeatures screen, click **Back to PCF-NewFeatures** link available on top left corner of the screen.

Figure 4-34 Build Requires Parameters in sync with Documentation



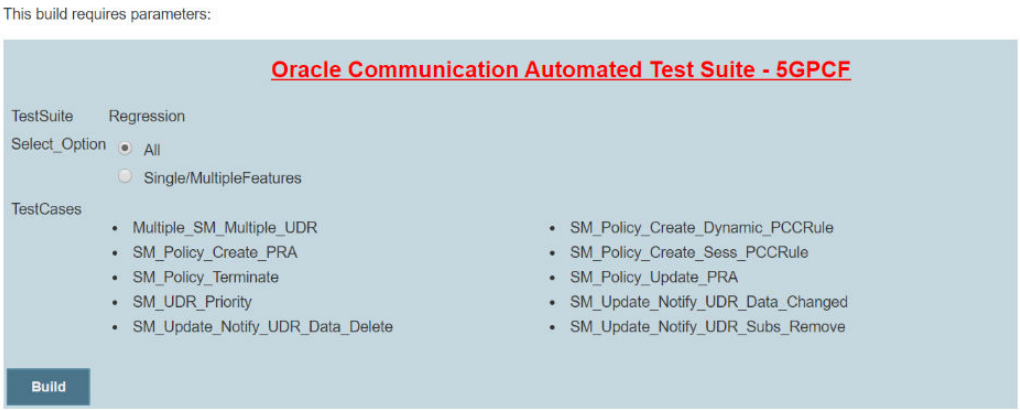
In the above screen, you can notice that the initials of test cases are similar to the functionalities listed in the contents of Documentation screen.

PCF-Regression Pipeline

This pre-configured pipeline has all the test cases of previous releases. For example, as part of Release 1.1.0, this pipeline has all the test cases that were released as part of release 1.0.0

To view Regression pipeline details, click **Build with Parameters** in the left navigation pane. The following screen appears:

Figure 4-35 PCF - Regression



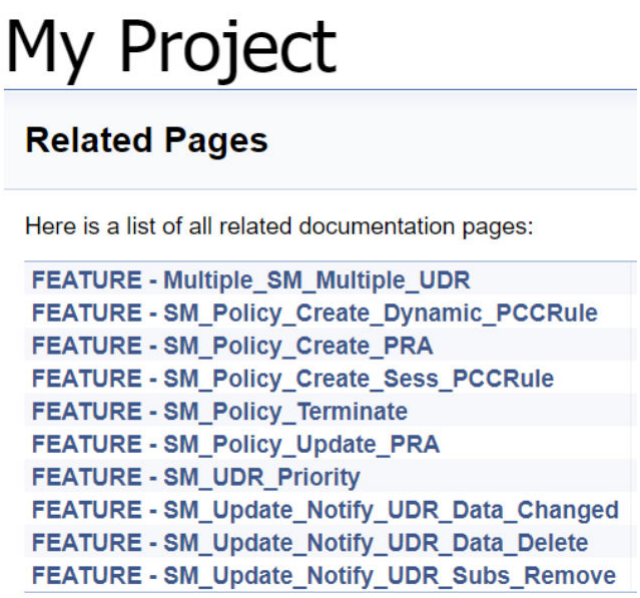
Note:

The regression pipeline does not have any sanity option. However, you should perform all the steps as performed in NewFeatures pipeline. Configure the pipeline script changes to provide environment variables.

Regression - Documentation

Click **Documentation** in the left navigation pane of the PCF-Regression pipeline. Following screen appears:

Figure 4-36 PCF Regression Documentation



This screen shows only those functionalities whose test cases were released in previous releases.

Executing SCP Test Cases using ATS

To execute SCP Test Cases, you need to ensure that following prerequisites are fulfilled.

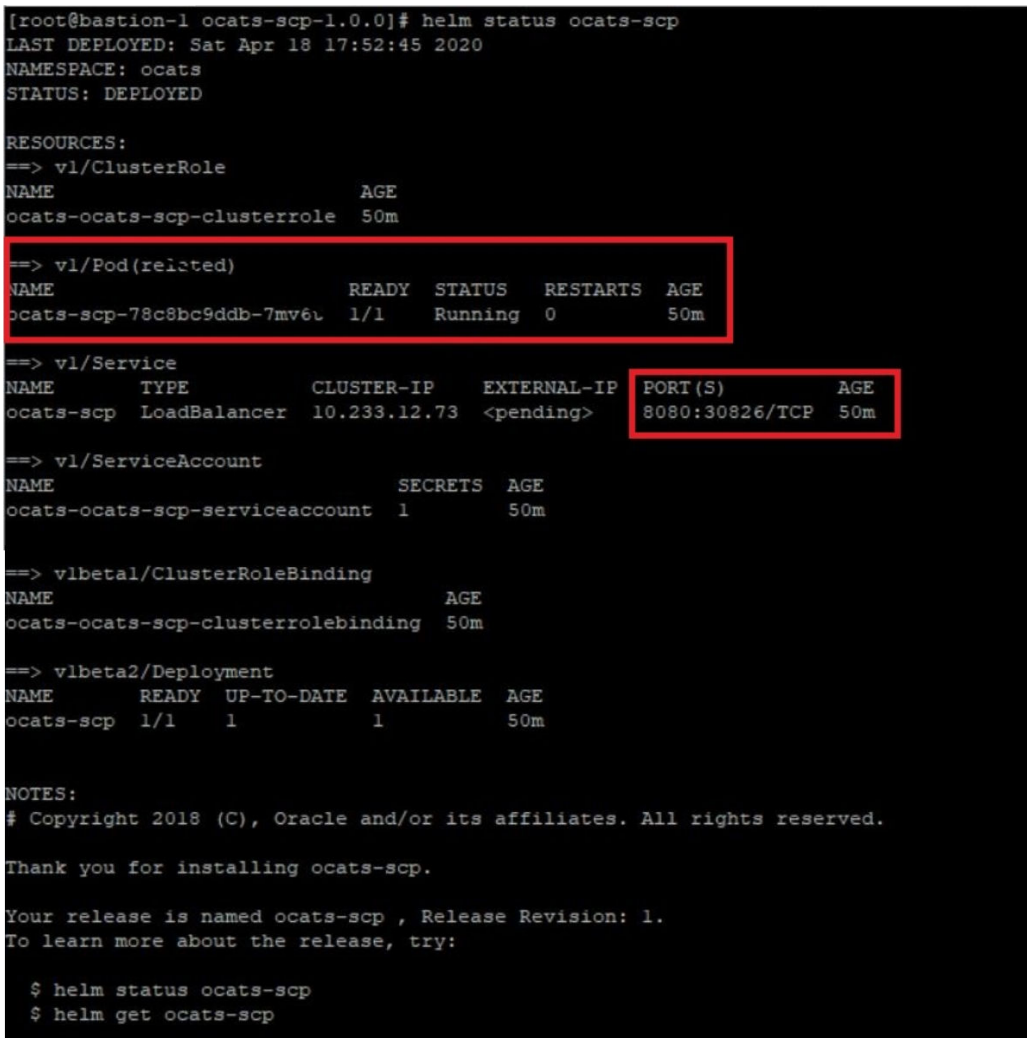
Prerequisites

- Deploy SCP 1.6.0 with following custom values in deployment file.
  - As you can provide NRF information only at time of deployment, Stub NRF details like nrf1svc and nrf2svc should also be provided at the time of deployment before executing these cases. **For Example:** If teststub namespace is ocats then SCP should have been deployed with primary nrf as *nrf1svc.ocats.svc.<clusterDomain>* and secondary nrf as *nrf2svc.ocats.svc.<clusterDomain>* for NRF test cases to work.
  - In the SCP deployment file, **servingScope** must have Reg1 and **servingLocalities** must have USEast and Loc9. In addition, the recommended auditInterval is 120 and guardTime is 10.
  - For ATS executio, you should deploy SCP with SCP-Worker replicas set to 1.
- Deploy ATS using helm charts.

Logging into ATS

Before logging into ATS, you need to ensure that ATS is deployed successfully using HELM charts. A sample screen is given below:

Figure 4-37 Verifying ATS Deployment



There are two ways to login to ATS Jenkins GUI.

- If a metalLB server is available and an external IP is provided to ATS service, user can login to ATS GUI using **<External-IP>:8080**
- When an external IP is not provided to the ATS service, user can open the browser and provide the external IP 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, the ATS nodeport is highlighed in red as **30826**. For more details on ATS deployment, refer to SCP ATS Installation Guide.

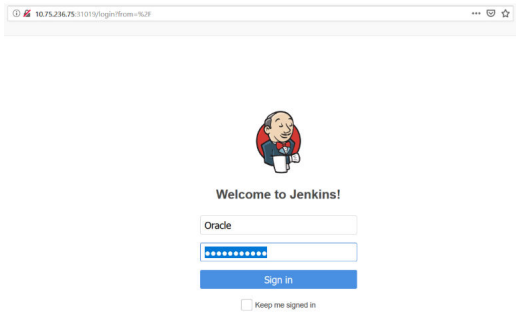
Open a browser and provide IP and port details as <Worker-Node-IP>:<Node-Port-of-ATS> to get the ATS login page.

Executing ATS

To execute ATS:

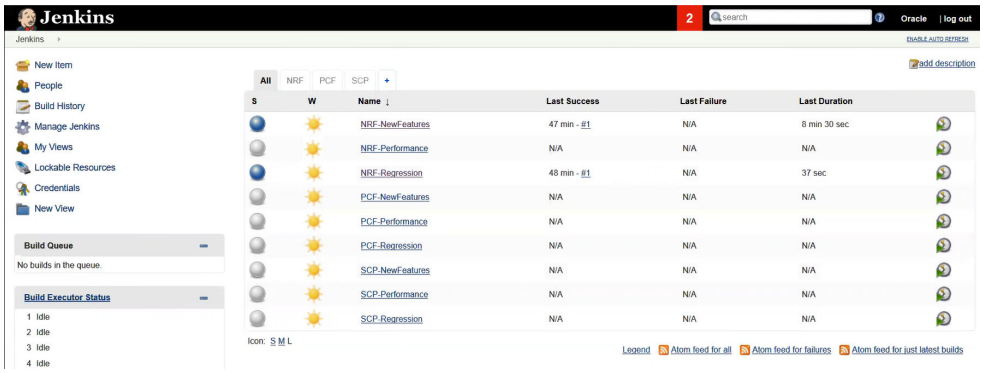
1. Enter the **username** as "Oracle" and **password** as "Welcome@123". Click **Sign in**. A sample screen is shown below.

Figure 4-38 Sample Screen: Logging into ATS GUI



2. Following screen appears showing pre-configured pipelines for NRF, PCF and SCP individually (9 Pipelines).
   - New-Features
   - Performance
   - Regression

Figure 4-39 Jenkins Logged-in First Screen



By default, you are in the **All** tab. To test the SCP test cases, click the **SCP** tab. The SCP tab displays the pipelines related to SCP only.

Jenkins ATS Pre-configured Pipelines

In the SCP tab, there are three pre-configured pipelines.

- **SCP-NewFeatures**
- **SCP-Performance:** This pipeline is not operational as of now. It is reserved for future releases of ATS.

- **SCP-Regression**

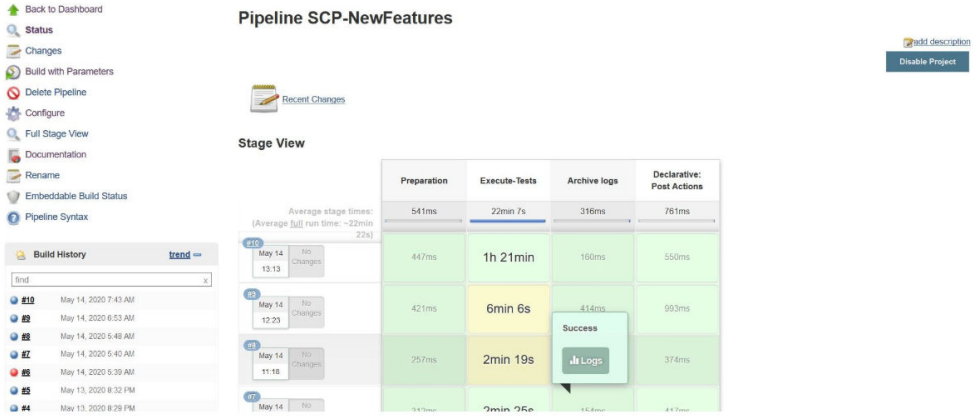
They are explained below:

SCP-NewFeatures Pipeline

This is a pre-configured pipeline where all the SCP test cases are executed. To configure its parameters, which is a one time activity:

1. Click **SCP-NewFeatures** in the Name column. The following screen appears:

Figure 4-40 SCP-NewFeatures

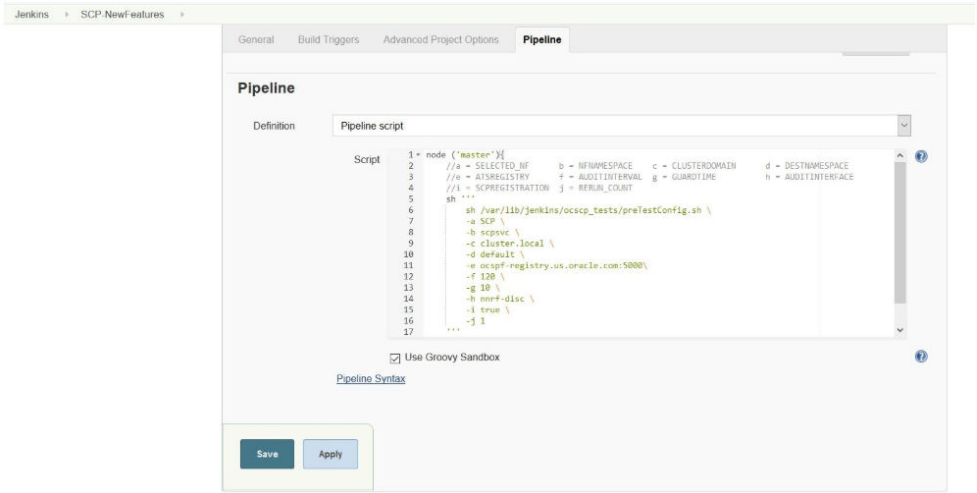


2. Click **Configure** in the left navigation pane to provide input parameters. The SCP-NewFeatures Configure - General tab appears.

```
Make sure that the screen shown above loads completely before you perform any
action on it.
```

3. Scroll-down to the end. The control moves from **General** tab to the **Advanced Project Options** tab as shown below:

Figure 4-41 Advanced Project Options



> **Note:**
>
> The description of each parameter 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
> - -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 - Audit Interface SCP is deployed with - nnrf-nfm or nnrf-disc
> - -i - Flag for SCP to Register with NRF in default region Reg1 - true/ false
> - -j - re-run count

You can modify script pipeline parameters on the basis of your deployment environment and click **Save**.

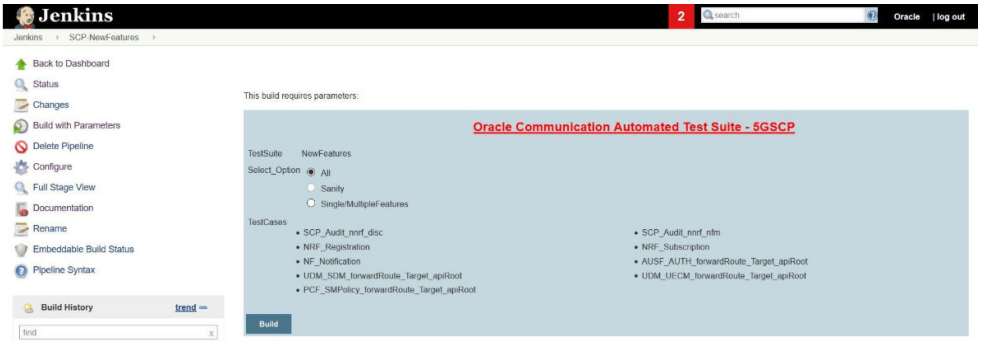
> **Note:**
>
> DO NOT MODIFY ANYTHING OTHER THAN THESE PARAMETERS.

Executing SCP Test Cases

To execute SCP test cases:

1. Click the **Schedule a Build with parameters for SCP-NewFeatures** icon present in extreme right column corresponding to **New-Features** row. Following screen appears:

Figure 4-42 SCP TestCases



In the above screen, there are three **Select_Option**(s), which are:

- **All:** By default, all the SCP test cases are selected for execution. User just need to scroll down and click **Build** to execute all the test cases.

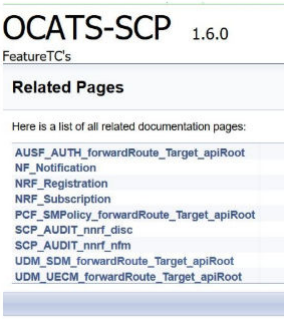
  > **Note:**
  >
  > There is an exception to All option. Test cases, **SCP_Audit_nnrf_disc** and **SCP_Audit_nnrf_nfm** are based on the input provided for AUDITINTERFACE parameter and **NRF_Registration** test case executes only if SCPREGISTRATION flag is set to true in input parameters of configure pipeline.

- **Sanity:** This option is **NOT AVAILABLE** for SCP.
- **Single/MultipleFeatures:** This option allows you to select any number of test cases that you want to execute from the list of total test cases available for execution. After selecting the test cases, scroll-down and click **Build**. The selected SCP test cases are executed.

NewFeatures - Documentation

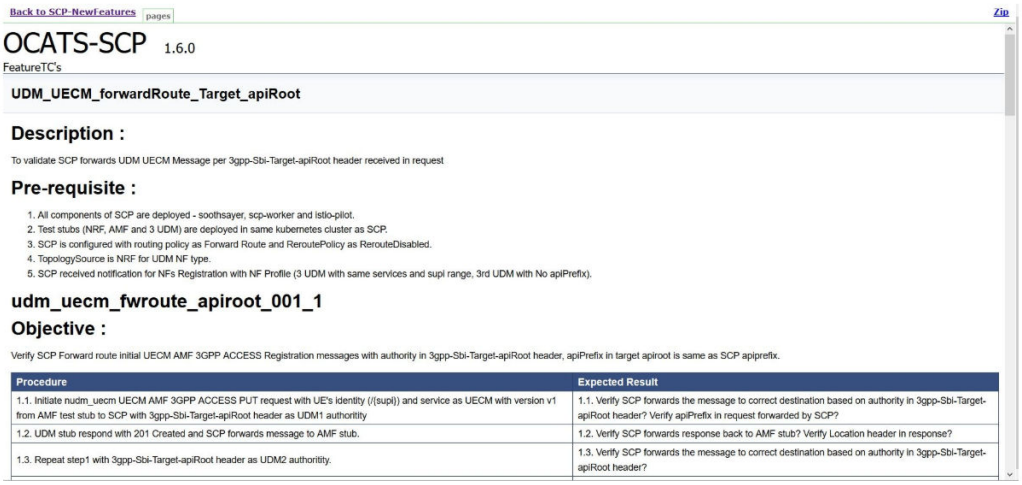
To view SCP functionalities, go to SCP-NewFeatures pipeline and click **Documentation** link in the left navigation pane. The following screen appears:

Figure 4-43 SCP-NewFeatures Documentation



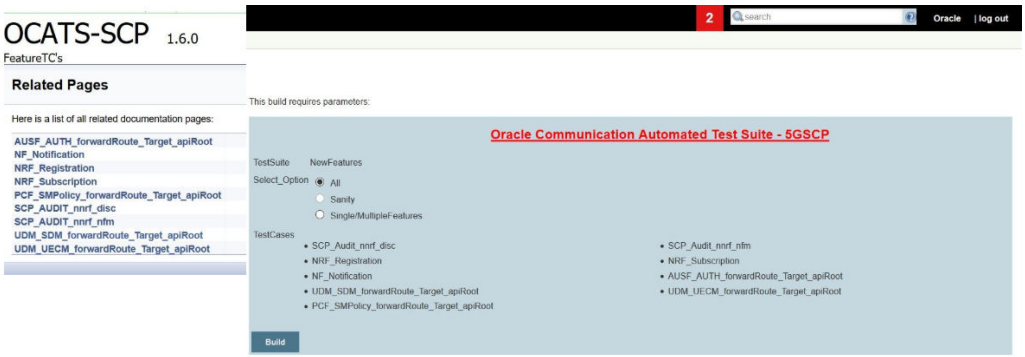
You can click any functionality to view its test cases and scenarios for each test case. For example, on click of udm_uecm_fwroute_apiroot_001_1, following screen appears:

Figure 4-44 Sample: SCP Functionality



Based on the functionalities covered under Documentation, the **Build Requires Parameters** screen displays test cases. To navigate back to the Pipeline SCP-NewFeatures screen, click **Back to SCP-NewFeatures** link available on top left corner of the screen.

Figure 4-45 Build Requires Parameters in sync with Documentation



In the above screen, you can notice that the initials of test cases are similar to the functionalities listed in the contents of Documentation screen.

SCP-Regression Pipeline

This pre-configured pipeline has all the test cases of previous releases. As this is the first release of SCP-ATS, this pipeline does not show any previous release testcases.

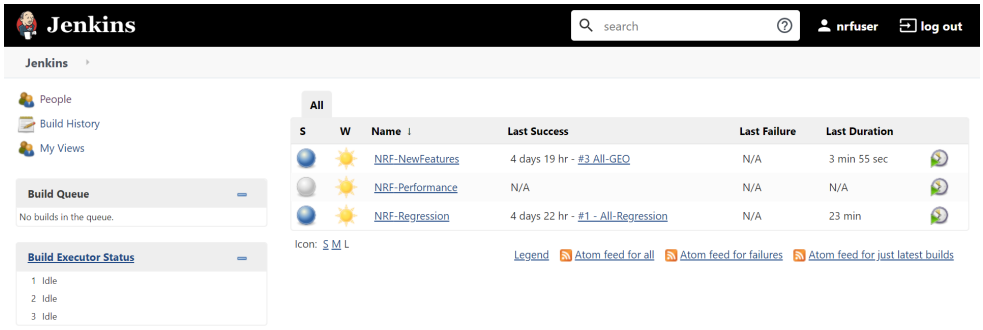
A
Modifying Login Password

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

If the user wants to modify its login password, the ATS application allows to do so. To modify login password:

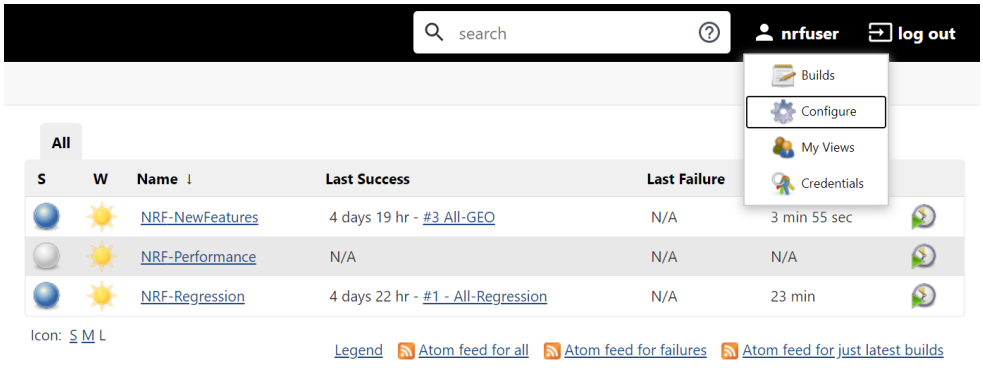
1. Login to ATS application using default login credentials. The home screen of respective NF appears showing its pre-configured pipelines.

Figure A-1 Sample: NRF Home Screen



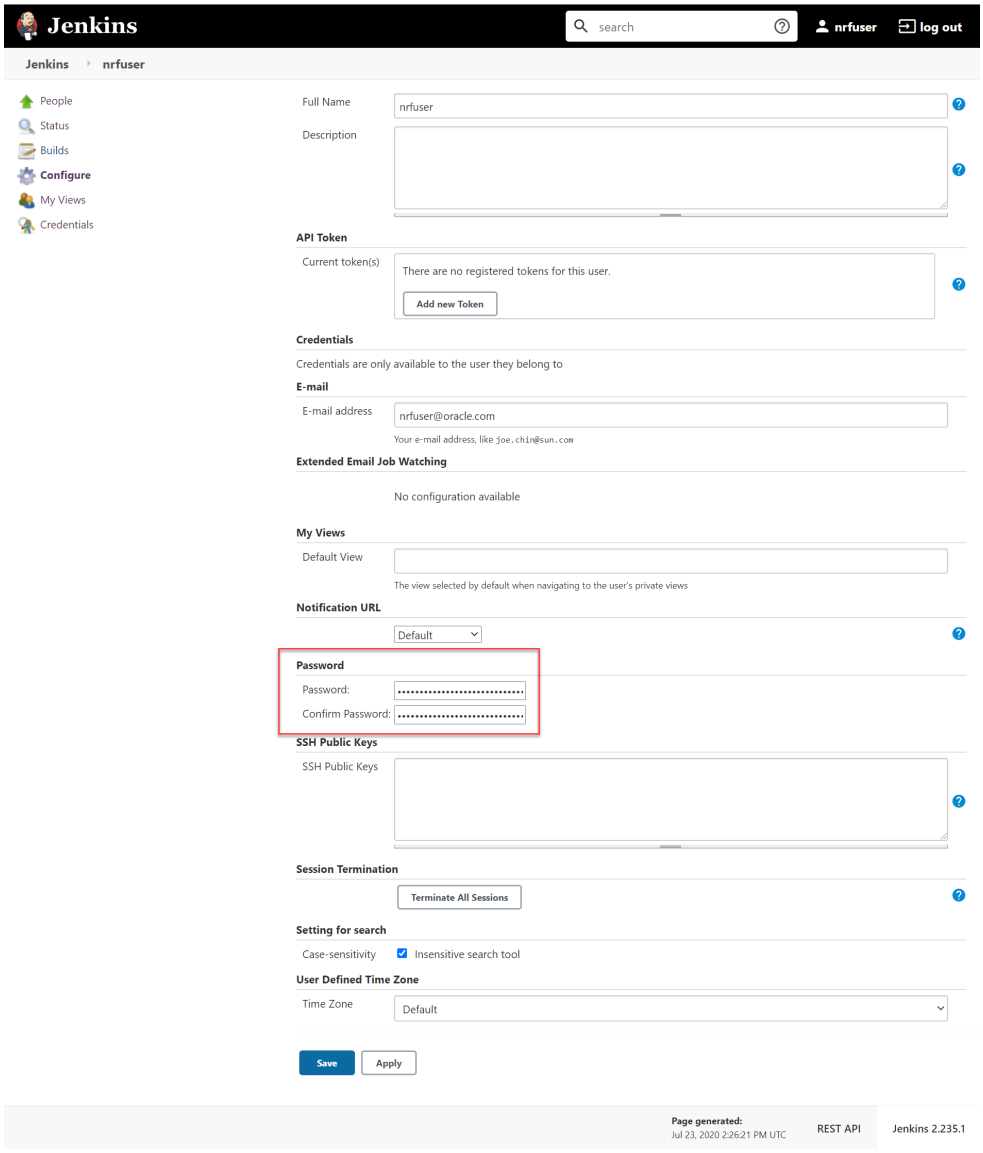
2. Hover-over logged-in user name and click the down arrow. Click **Configure** as shown below.

Figure A-2 Configure Option



3. The following screen appears.

Figure A-3 Logged-in User Detail



4. In the **Password** section, enter the new password in the **Password** and **Confirm Password** fields and click **Save**.

Thus, a new password is set for the user.