Oracle® Communications

DSR Automated Test Suite Installation and User Guide

Release 9.1.0.0.0
G20539-02
June 2025

Oracle Communications DSR Automated Test Suite Installation and User Guide, Release 9.1.0.0.0

G20539-02

Contents

What's New in this Release

This section introduces the documentation updates for release 9.1.0.0.0.

Release 9.1.0.0.0 - G20539-02, June 2025

Added the Workaround for Password Expiry section as an appendix.

Release 9.1.0.0.0 - G20539-01, February 2025

Added a note about activation or deactivation of RSA and DSA in the Prerequisites for Test Case Execution section.

1
Introduction

The Automated Test Script (ATS) is a software that is used on the system under test to check if the system is functioning as expected. This software performs testing of the features offered by OC-DSR through automation decreasing the manual test effort. This software is flexible enough that the user can create additional test cases with ease using the APIs provided by the framework.

1.1 Limitations

Only a single Multiprotocol Routing Agent (MRA) and Multimedia Policy Engine (MPE) cluster can be used in the test environment.

1.2 Acronyms

Table 1-1 Acronyms

Term	Definition
API	Application programming interface
ATS	Automated Test Suit
DSR	Diameter Signaling Router
NTP	Network Time Protocol
OS	Operating System
SDS	Subscriber Data Server
SUT	System Under Test
VNFM	Virtual Network Functions Manager
vSTP	Virtual Signaling Transfer Point

1.3 How to use this document

Read the following instructions before performing any procedure documented in this guide:

1. Read the instructional text and all associated procedural Warnings or Notes.
2. If a procedural step fails to execute, contact Oracle's Customer Service for assistance before attempting to continue. My Oracle Support for information on contacting Oracle Customer Support.

1.4 Documentation Admonishments

Admonishments are icons and text throughout this manual that alert the reader to assure personal safety, to minimize possible service interruptions, and to warn of the potential for equipment damage.

Table 1-2 Admonishments

Icon	Description
	Danger: (This icon and text indicate the possibility of personal injury.)
	Warning: (This icon and text indicate the possibility of equipment damage.)
	Caution: (This icon and text indicate the possibility of service interruption.)

1.5 Customer Training

Oracle University offers training for service providers and enterprises. Visit our web site to view, and register for, Oracle Communications training at http://education.oracle.com/communication.

To obtain contact phone numbers for countries or regions, visit the Oracle University Education web site at www.oracle.com/education/contacts.

1.6 My Oracle Support

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

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

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

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

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

1.7 Emergency Response

In the event of a critical service situation, emergency response is offered by the Customer Access Support (CAS) main number at 1-800-223-1711 (toll-free in the US), or by calling the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. The emergency response provides immediate coverage, automatic escalation, and other features to ensure that the critical situation is resolved as rapidly as possible.

A critical situation is defined as a problem with the installed equipment that severely affects service, traffic, or maintenance capabilities, and requires immediate corrective action. Critical situations affect service and/or system operation resulting in one or several of these situations:

- A total system failure that results in loss of all transaction processing capability
- Significant reduction in system capacity or traffic handling capability
- Loss of system ability to perform automatic system reconfiguration
- Inability to restart a processor or the system
- Corruption of system databases that requires service affecting corrective actions
- Loss of access for maintenance or recovery operations
- Loss of the system ability to provide any required critical or major trouble notification

Any other problem severely affecting service, capacity/traffic, billing, and maintenance capabilities may be defined as critical by prior discussion and agreement with Oracle.

2
Prerequisites

- Download the ATS Image from Oracle Software Delivery Cloud (OSDC). Example of an ATS image: `ats-9.0.2.0.0-1.0.26.tgz`.
- Ensure that ATS and DSR are in the same network.

3
ATS Server Deployment Overview

The ATS server is deployed using Virtual Network Functions Manager (VNFM). It has features for Rf_Routing, Gy_S6b_Stateless_Routing, Rx_Gateway_MCPTT, and Radius_Routing. It has a separate cleanup pipeline.

4
Deploying ATS Using VNFM

The ATS Master VNF supports dynamic and fixed IP deployment models.

> **Note:**
>
> ATS supports both IPv4 and IPv6 suites at the time of deployment.

To deploy the ATS Master VNF, you must have the following pieces of information:

- The VNF ID for a previously created ATS Master VNF instance.
- Information about the OpenStack instance on which the VNF must be deployed:
  - OpenStack Controller URI
  - User Domain Name
  - Project Domain Id
  - Username
  - Password
  - Tenant name
- The name of a public network in the selected OpenStack instance that will carry the ATS master traffic.
- The IP of an NTP server accessible by VMs within the selected OpenStack instance. The OpenStack controller that controls the selected OpenStack instance hosts an NTP server.

For more information about the list of all the inputs and possible outputs of the command `instantiate VNF`, refer to **ETSI NFV-SOL 003**, section **5.4.4.3.1**, or the **DSR VNFM Swagger specification**.

> **Note:**
>
> It is mandatory to add two XSI Networks in ATS Master to instantiate a stack.

Sample Request for instantiating ATS Master Dynamic IP deployment model:

```
URL: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/< VNF ID received
from create request>/instantiate

        Accept: application/json
        Content-Type: application/json
        X-Token: Token generated after login

        {
        "flavourId": "master",
        "instantiationLevelId": "small",
```

```
"extVirtualLinks": "extVirtualLinks",
"extManagedVirtualLinks": [],

"vimConnectionInfo":[ {
"id": "vimid",
"vimType": "OpenStack",
"interfaceInfo": {
"controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
},
"accessInfo": {
"username": "dsrci.user",
"password": "xxxxx",
"userDomain": "Default",
"projectDomain": "default",
"tenant": "DSR CI"
}

}],
"localizationLanguage": "localizationLanguage",
"additionalParams": {
"xmiNetwork": {
"name": "ext-net8",
"ipVersion": "IPv4",
"xmiSubnetName":"ext-net8-subnet"
},
"xsiNetwork": [{
"name": "ext-net7",
"ipVersion": "IPv4",
"xsiSubnetName":"ext-net7-subnet"
},
{
"name": "ext-net6",
"ipVersion": "IPv4",
"xsiSubnetName":"ext-net6-subnet"
}],
"ntpServerIp": "10.250.32.10",
"dnsServerIp": "10.250.32.10",
"atsKeyName": "atsKeypair",
"atsMasterFlavor": "ats.master",
"atsMasterImage": "ATS_BOX.qcow2",
"atsAvailabilityZone": "nova"
}
}
```

Sample request for initiating ATS Master Request for Fixed IP deployment model:

```
URL: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/< VNF ID received
from create request>/instantiate

Accept: application/json
Content-Type: application/json
X-Token: Token generated after login

{
```

```
"flavourId": "master",
"instantiationLevelId": "small",
"extVirtualLinks": "extVirtualLinks",
"extManagedVirtualLinks": [],

"vimConnectionInfo":[ {
"id": "vimid",
"vimType": "OpenStack",
"interfaceInfo": {
"controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
},
"accessInfo": {
"username": "dsrci.user",
"password": "xxxxx",
"userDomain": "Default",
"projectDomain": "default",
"tenant": "DSR CI"
}
}],
"localizationLanguage": "localizationLanguage",
"additionalParams": {
"xmiNetwork": {
"name": "ext-net8",
"ipVersion": "IPv4",
"xmiSubnetName":"ext-net8-subnet",
"fixedIps": {
"masterXmiIp":"10.75.123.16"
}
},
"xsiNetwork": [{
"name": "ext-net7",
"ipVersion": "IPv4",
"xsiSubnetName":"ext-net7-subnet",
"fixedIps":
{
"xsiIp": "10.75.195.21"
}
},
{
"name": "ext-net6",
"ipVersion": "IPv4",
"xsiSubnetName":"ext-net6-subnet",
"fixedIps":
{
"xsiIp": "10.75.195.22"
}
}],
"ntpServerIp": "10.250.32.10",
"dnsServerIp": "10.250.32.10",
"atsKeyName": "atsKeypair",
"atsMasterFlavor": "ats.master",
"atsMasterImage": "ATS_BOX.qcow2",
"atsAvailabilityZone": "nova"
}
}
```

Sample Response

Instantiating the ATS Master VNF response

```
202 Accepted

            Headers:
            {
            location: https://localhost:8443/vnflcm/v1/vnf_lcm_op_occs/lcmOp-
fb21f9d3-43ad-46cd-a03f-7220bb36a5c6
            date: Tue, 29 Jan 2019 10:39:24 GMT
            content-length: 0  content-type:
            application/xml
            }
```

The following table describes the parameters for ATS Master:

Table 4-1 ATS Master Parameters

Parameter	Definitions
`flavourId`	Identifier of the VNF deployment flavor to be instantiated.
`xmiNetwork`	Network used to provide access master VM communication.
`ntpServerIp`	IP of the NTP server.
`dnsServerIp (optional)`	IP of the DNS server. If not provided, NTP server IP will be considered as DNS server IP.
`atsKeyName`	Key pair name for ATS. To log in to the ATS instance, use same key pair.
`masterXmiIp`	In case of fixed IP scenario, the IP of master will be provided.
`xsiNetwork`	Network used for DSR signaling traffic.
`atsMasterFlavor (optional)`	Flavor used for OpenStack deploys.
`atsMasterImage (optional)`	Image used for OpenStack deploys.
`atsAvailabilityZone (optional)`	Name of logical partitioning in case of host aggregate.

Note:

The atsKeyName pair is created dynamically through VNFM. The same public key is put into all the ATS instances (master, core & tools), and the private key is in the ATS master stack output. Use the same private key to log in to the ATS instance (master, core & tools) by executing the following command:

```
ssh -i <ats private key> <username>@<ats master Ip>
```

Example: `ssh -i atskey.pem cloud-user@10.75.189.120`

4.1 Custom Folder Implementation

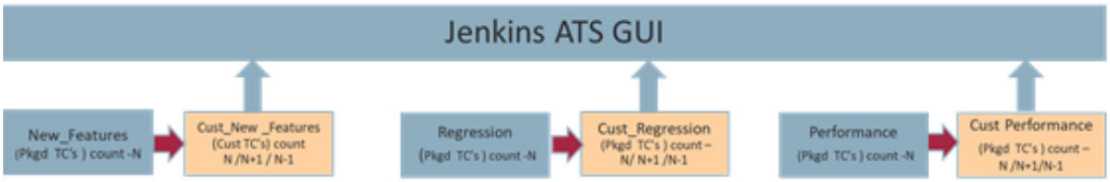
New custom test cases folders (**cust_newfeatures**, **cust_regression** and **cust_performance**) have been introduced to accommodate the customization's to original product packaged test cases. These folders carry the customized test cases (any new test cases added by customers or subset of test cases from the original product supplied test cases or modified test cases).

Initially when packaged and released, both the product test case folders (newfeatures, regression and performance) and the custom test case folders (cust_newfeatures, cust_regression and cust_performance) carries same set of test cases. Subsequently, customers can use the custom test case folders to carry out any customization's from their side (updates, additions, or deletions of test cases) without disturbing the original product packaged test cases or folders. Jenkins always pick the test cases from the custom test cases folders.

Custom Folder Structure is implemented in the Performance Job where Performance-Suite and Cust-Performance-Suite contain the same set of test cases. Customization, such as updates, additions, or deletions of test cases, without disturbing the original product packaged test cases or folders can be done in Cust-Performance-Suite.

Custom Folder Structure is implemented in the Health-Check Job where HealthCheck and Cust-HealthCheck contain the same set of test cases. Customization, such as updates, additions, or deletions of test cases, without disturbing the original product packaged test cases or folders can be done in Cust-HealthCheck.

Figure 4-1 Summary of Custom Folder Implementation



5
Enhancing Security using HTTPS

With the introduction of this feature, HTTPS adds a layer of encryption that helps the user to secure connection between server and clients.

When accessing a website enabled with HTTPS, users can trust that their connection is secure, ensuring the confidentiality of their data. For Jenkins to operate over HTTPS, it requires SSL certificate.

This certificate is converted into Public-Key Cryptography Standards (PKCS) 12 industry standard format and then to `JKS` format, which Jenkins readily accepts. The `JKS` format certificate is then stored in the Jenkins path, enabling the certificate to operate over HTTPS.

Prerequisites

The user needs to obtain an SSL certificate and a key from a certificate authority.

Uploading SSL Certificate

There are two methods to upload SSL certificate for running Jenkins over HTTPS:

- Upload SSL certificate in DSR NOAM
- Upload SSL certificate in ATS

An SSL certificate can be obtained from a certificate authority. After obtaining the certificate, perform the following steps to upload the certificate:

1. Log in to the **DSR NOAM** GUI.
2. Navigate to Administration, then Access Control and Certificate Management.
3. Upload the SSL certificate and key.

Configuring JKS.YAML Properties

Configure the following properties as listed below:

- `Certificate_uploaded_DSR`: To select the model, enter either 1 or 0 for yes or no.
- `Certificate_uploaded_ATS`: To select the model, enter either 1 or 0 for yes or no.
- `Dsrcertfilename`: Provide the certificate file name located in the `DSR NOAM /usr/TKLC/appworks/etc/ssl/ path`. Mandatory if Certificate_uploaded_DSR is selected.
- `Dsrkeyfilename`: Provide the key file name located in `DSR NOAM /usr/TKLC/appworks/etc/ssl/` path. Mandatory if Certificate_uploaded_DSR is selected.
- `Atscertfilename`: Provide the file name of the ATS certificate that has been uploaded. Mandatory if Certificate_uploaded_ATS is selected
- `Atskeyfilename`: Provide the file name of the ATS key that has been uploaded. Mandatory if Certificate_uploaded_ATS is selected
- `Atscertstorepath`: Provide the folder path where the SSL certificate and private key has been uploaded in ATS.

- `Dsrnoip`: Provide the DSR NOAM IP address (if the certificate has been uploaded in DSR). Mandatory if Certificate_uploaded_DSR is selected.
- `Dsrusername`: Provide the DSR NOAM CLI username (Default is "admusr"). Mandatory if Certificate_uploaded_DSR is selected.
- `Dsrpassword`: Provide the DSR NOAM CLI password (Default is "Dukw1@m?") User can change the password, after which it will be encrypted again. Mandatory if Certificate_uploaded_DSR is selected.
- `httpsKeyStorePassword`: Provide the password required for the certificate file (Default is "Welcome@123"). It will be stored in the encrypted form.

Figure 5-1 jks.yaml file

```
#Please enter either 1 or 0 in the models field : 1 for True/Yes 0 for False/No
#Please ensure that only one model runs at a time
MODELS:
  Certificate_uploaded_DSR: 0
  Certificate_uploaded_ATS: 1
FILES:
  dsrcertfilename: wildcard.crt
  dsrkeyfilename: wildcard.pem
  atscertfilename: wildcard.crt
  atskeyfilename: wildcard.pem
  atscertstorepath: /home/cloud-user/
dsrnoip: 10.75.250.54
dsrusername: admusr
dsrpassword: +UmXAjbVtnN8BJll+S3v7A== P2oeFgDK8T6BP874 mJE+VQxE2eR6JoGfD4kzSg==
```

Running the JKS.PY File

Run the `jks.py` file by using the following command:

```
python jks.py
```

> **Note:**
>
> The user cannot select yes for both Certificate_uploaded_DSR and Certificate_uploaded_ATS at the same time.

Figure 5-2 python jks.py

```
[cloud-user@mavvsskn-ats-jenkins ~]$ python jks.py
Running Pre-validation checks...
!!!PRE-VALIDATION CHECK SUCCESSFUL!!!
Files /home/cloud-user/wildcard.crt and /home/cloud-user/wildcard.pem already exists i.e. True True
File /var/lib/jenkins/jenkinsserver.jks already exists i.e. True
Importing keystore /home/cloud-user/certificate.p12 to /var/lib/jenkins/jenkinsserver.jks...
Entry for alias 1 successfully imported.
Import command completed:  1 entries successfully imported, 0 entries failed or cancelled

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "k
eytool -importkeystore -srckeystore /var/lib/jenkins/jenkinsserver.jks -destkeystore /var/lib/jenkins/jenkinsserver.jks -deststo
retype pkcs12".

Stop and restart of jenkins
/var/lib/jenkins
[cloud-user@mavvsskn-ats-jenkins ~]$
```

Configuring HTTPS File

The files `https_config`, `jks.yaml` and `jks.py` are required for running Jenkins and must not be deleted under any circumstances.

- `httpPort`: The current value of this parameter is `-1`, which should not be altered to run Jenkins through HTTPS.
- `httpsPort`: The current value of this parameter is 8443, which should not be altered to run Jenkins through HTTPS.
- `httpsKeyStore`: The current default value of this parameter is `/var/lib/jenkins/jenkinsserver.jks` and is advised not to be changed unless the `.jks` certificate file is relocated.
- `httpsKeyStorePassword`: This parameter should remain the same as the password in `jks.yaml` file and it needs to be modified only in the `jks.yaml`.

Results after Running JKS.PY File

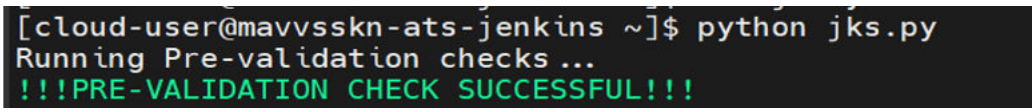
The python jks.py file generates output in 6 stages, which include:

- Prevalidation checks
- Checking the availability of ATS certificate and key files
- Checking the availability of `Jenkinsserver.jks` file
- Conversion to JKS format
- Warning for PKCS 12 format
- Restart of Jenkins

Prevalidation Checks

The console displays **PRE-VALIDATION CHECK SUCCESSFUL!!!** when the parameters in the model selection within the `jks.yaml` are in the correct format.

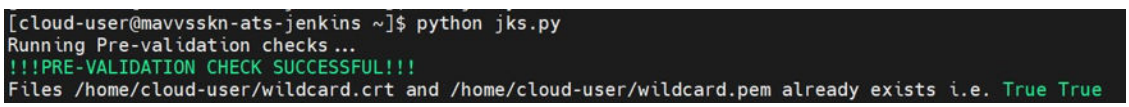
Figure 5-3 Prevalidation



Checking the availability of ATS certificate and key files

The result specifically focuses on the certificate uploaded in the ATS model. The script checks whether the ATS uploaded certificate and key files exist in the ATS uploaded folder path provided in the `jks.yaml` file.

Figure 5-4 ATS certificate



Checking the availability of `Jenkinsserver.jks` file

The script confirms the availability of the `jenkinsserver.jks file` and removes it from the Jenkins home path to place the newly created `jks file`.

Figure 5-5 Jenkinsserver.jks

```
[cloud-user@mavvsskn-ats-jenkins ~]$ python jks.py
Running Pre-validation checks ...
!!!PRE-VALIDATION CHECK SUCCESSFUL!!!
Files /home/cloud-user/wildcard.crt and /home/cloud-user/wildcard.pem already exists i.e. True True
File /var/lib/jenkins/jenkinsserver.jks already exists i.e. True
```

Conversion to JKS format

The script generates a confirmation message indicating the creation of the `jks` format file:

Figure 5-6 JKS format conversion

```
Importing keystore /home/cloud-user/certificate.p12 to /var/lib/jenkins/jenkinsserver.jks ...
Entry for alias 1 successfully imported.
Import command completed:  1 entries successfully imported, 0 entries failed or cancelled
```

> **Note:**
>
> The console displays a warning message during the conversion to `jks` format, as PKCS 12 is the recommended industry standard. It can be ignored, as `jks` format is required for Jenkins to run over HTTPS.

Figure 5-7 Warning

```
Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "k
eytool -importkeystore -srckeystore /var/lib/jenkins/jenkinsserver.jks -destkeystore /var/lib/jenkins/jenkinsserver.jks -deststo
retype pkcs12".
```

Restart of Jenkins

Restarting Jenkins results in displaying the Jenkins home path.

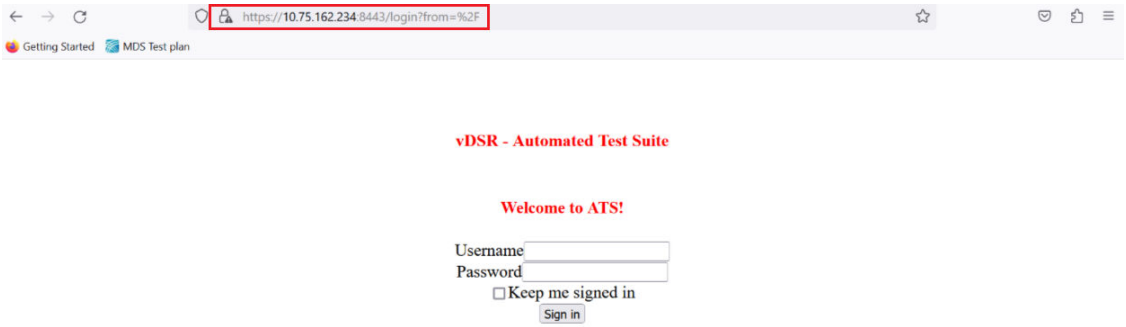
Figure 5-8 Restart of jenkins

```
Stop and restart of jenkins
/var/lib/jenkins
```

Conclusion

By configuring SSL certification, Jenkins can operate over HTTPS, ensuring secure communication between client and the Jenkins web interface. The Jenkins server is securely accessible through HTTPS port using the following URL: `https://<ATS IP>:8443`.

Figure 5-9 HTTPS Enabled Log in Screen



6
Test Case Execution

6.1 Prerequisites for Test Case Execution

This section provides information about the prerequisites that must be met in the following sequence before running the test cases:

1. Ensure no hyphen is present in the stack name of NOAM and SOAM while deploying the VDSR stack through VNFM.
2. **Location of the Test Cases**
   - The Cust-Roaming-Suite directory path is `/var/lib/jenkins/workspace/Regression/Cust-Roaming-Suite`.
   - The Cust-Core-DSR directory path is `/var/lib/jenkins/workspace/Regression/Cust-Core-DSR`.
   - The New-Features are located in `/var/lib/jenkins/workspace/New-Features`
   - The Performance test cases are located in `/var/lib/jenkins/workspace/Performance/Radius_Traffic`
   - All Cleanup pipeline test cases are located in `/var/lib/jenkins/workspace/Cleanup/Cust-Cleanup-DSR`
   - All DSA stateless and stateful pipeline test cases are located in the file `/var/lib/jenkins/workspace/DSA`. The only option for all DSA stateless and stateful pipeline is to run the full suite.
   - All vSTP test cases are divided into four different suites:
     `/var/lib/jenkins/workspace/vSTP_Regression/behave_test_framework/vSTP-Suite1/`

     `/var/lib/jenkins/workspace/vSTP_Regression/behave_test_framework/Cust-vSTP-Suite2/`

     `/var/lib/jenkins/workspace/vSTP_Regression/behave_test_framework/Cust-vSTP-Suite3/`

     `/var/lib/jenkins/workspace/vSTP_Regression/behave_test_framework/Cust-vSTP-Suite4/`
3. **SUT Requirements**

Table 6-1 DSR and SDS SUT Details

Server	Quantity
DSR SUT	
DSR NOAM Active	1
DSR NOAM Standby	1
DSR Signaling SOAM Active	1
DSR Signaling SOAM Standby	1

Table 6-1 (Cont.) DSR and SDS SUT Details

Server	Quantity
DA-MP	2
IPFE	2
SDS SUT	
SDS NOAM Active	1
SDS NOAM Standby	1
Query Server	1
SDS Signaling SOAM Active	1
SDS Signaling SOAM Standby	1
DP Server	1

Table 6-2 vSTP SUT Details

Server	Quantity
vSTP SUT	
vSTP NOAM Active	1
vSTP NOAM Standby	1
vSTP Signaling SOAM Active	1
vSTP Signaling SOAM Standby	1
MP	2

> **Note:**
>
> When the SUT is created using VNFM, ensure that Mediation, FABR, and RBAR features are enabled.

4. **Update SUT Information in ATS**
Following are the mandatory steps for Cleanup, New-Features, Performance, Regression, VDSR-HealthCheck suites:

   a. Update `/home/cloud-user/Verizon-drop1/dsr-atsV2/dut.yaml` with the SUT details. The same will be automatically copied to the required location when the execution starts from Jenkins.
   Update the `dut.yaml` file by referring to the following file:

```
DSRVIP:
 - name: DSRNOVIP
    IP: 10.75.191.136
  - name: DSRSOVIP
    IP: 10.75.191.222

    #Pick the DSR XMI/XSI IPv6/ipv4 addresses from Main Menu:
Configuration -> Networking -> Devices GUI screen

    SIGNALING_IPs:
      - IP: 10.196.14.175
        type: LocalIp
      - IP: 10.196.15.170
```

```
        type: LocalIp
      - IP: 10.196.14.124
        type: IpfeTsa
      - IP: 10.196.14.124
        type: IpfeTsa
    ipfeInitiatorDampIp:
      - IP: 10.196.14.175

    MP_XMI:

      - IP: 10.75.191.133
        type: LocalIp
      - IP: 10.75.191.115
        type: LocalIp

    MP_IMI:
      - IP: 192.167.1.125
        type: LocalIp
      - IP: 192.167.1.203
        type: LocalIp

# If DSR ips are ipv6, entire dut file should be updated with ipv6 ips.
# If Ipv6 SDS is not available, make sure to comment each parameter in
SDS or remove ipv4
  ips from yaml file.

    SDSVIP:

    - name: SDSNOVIP
    IP: 10.75.191.130
    - name: SDSSOVIP
    IP: 10.75.191.45
    - name: SDSQS
    IP: 10.75.191.168

    SDS_IMI:
    - name: SDP00imi
      IP: 192.167.1.4
    - name: SDP01imi
      IP: 192.167.1.108

#DSR/SDS NOAM host and SOAM host IPs should be VIPs.
#If host IPs are in IPv6 format, IP address should be enclosed with [].

    UI_data:
  - name: UIData
    PassWordUI: ########
    UserNameUI: ########
    UDR: https://10.75.157.242
    noamHost: http://10.75.191.136
    soamHost: http://10.75.191.222
    StandBysoamHost: http://10.75.191.33
    StandBynoamHost: http://10.75.191.92
    sdsnoamHost: http://10.75.191.130
    sdssoamHost: http://10.75.191.45
```

```
StandBysdssoamHost: http://10.75.191.85
StandBysdsnoamHost: http://10.75.191.85
```

b. Update `/home/cloud-user/Verizon-drop1/dsr-atsV2/auth.yaml`. The same will be automatically copied to the required location when the execution starts from Jenkins. Server credentials, such as username and password, displayed in the following image can be updated if required.

c. Verify the following command to check if the jenkins is running:

```
ps -eaf | grep jenkins
```

Output:

Figure 6-1 Output

If not, then run the following command:

```
./jenkins_start.sh
```

d. The rerun functionality in the Roaming suite can be changed by updating the following values in `dut.yaml` file:

- RERUN_COUNT: 1
- SDS_ENABLE: N

Following are mandatory tasks for vSTP-Regression suite:

a. Update `/home/cloud-user/Verizon-drop1/vSTP-ats/vstp_signalling_ips.yaml` with the SUT details for vSTP test cases. The same will be automatically copied to the required location when the execution starts from Jenkins. Edit the `vstp_signalling_ips.yaml` file. Update the `vstp_signalling_ips.yaml` file by referring to the following file:

```
VSTP_SIGNALLING_IP:
# MP XSI IP on which traffic is to be run
  - name: XSI1
    IP: 121.131.152.209
MEAT_SIGNALLING_IP:
# MEAT XSI IP from which traffic is to be run
  - name: meat1
    IP: 121.131.152.207
ACTIVE_SO_IPS:
#Current active SO XMI IP
  - name: so_sg1
    IP : 10.75.162.138
  - name: so_sg2
    IP : 10.75.162.245
ACTIVE_NO_IP:
#Current active NO XMI IP
```

```
    IP: 10.75.162.199
VSTP_TPCs:
#TPC not to be changed to be kept as such
  - vstp_tpc_itui: 3-45-4
    vstp_tpc_itun: '8734'
    vstp_tpc_ansi: 5-44-8
# XMI IP of meat machine from which traffic is to run
MEAT_IP: 10.75.162.228
UDR:
#UDR IP for future use
  - name: udr1
    IP: 10.75.218.250
NUM_MP_PER_SITE: 2
ALL_SITE_XSI1_IP:
# Name and XMI IP of all MP's present
  - name: so1mp1
    IP: 121.131.152.209
  - name: so1mp2
    IP: 121.131.152.140
```

b. Update `/home/cloud-user/Verizon-drop1/vSTP-ats/passwords/auth.yaml`. The same will be automatically copied to the required location when the execution starts from Jenkins. Server credentials, such as username and password, displayed in the following image can be updated if required.

c. Verify using the command:

```
ps -eaf | grep jenkins
```

Output:

Figure 6-2 Output

If not, then run the command:

```
./jenkins_start.sh
```

d. Disable firewall from an active SOAM of vSTP from the path shown in below figure.

e. To access the MPs from the ATS machine, you need to run the VSTP suite run, the following command for all MPs. <stp ip> to be replaced by XMI IP of one MP at a time:

```
cat ~/.ssh/id_rsa.pub | ssh admusr@<stp ip> "mkdir -p ~/.ssh && chmod
700 ~/.ssh && cat >> ~/.ssh/authorized_keys"
```

5. **Enabling the Feature on SUT**
Following is a list of the Suites in ATS 9.0.2 onwards and their respective mandatory DSR features that are required to be activated:

Table 6-3 list of the suites in ATS 9.0.2

Suit Name	DSR's Features Required to be activated
Regression	Custom Roaming RBAR, Mediation Custom- Core FABR, Mediation, RBAR
New -Features	RSA, RBAR
DSA	DCA
Performance	FABR

> **Note:**
>
> Before activating and deactivating RSA and DSA using Jenkins GUI, delete both the files /root/.ssh/known_hosts and /home/cloud-user/known_hosts files.

Activation through automation

Before starting the execution of any test suites, it's essential to activate RBAR, FABR, and Mediation. The activation process for these features should be carried out through the Regression Suite. You can either choose the *All* option as shown below or, you can choose to run each Feature (RBAR, FABR, and Mediation) individually.

Figure 6-3 Choosing a suite

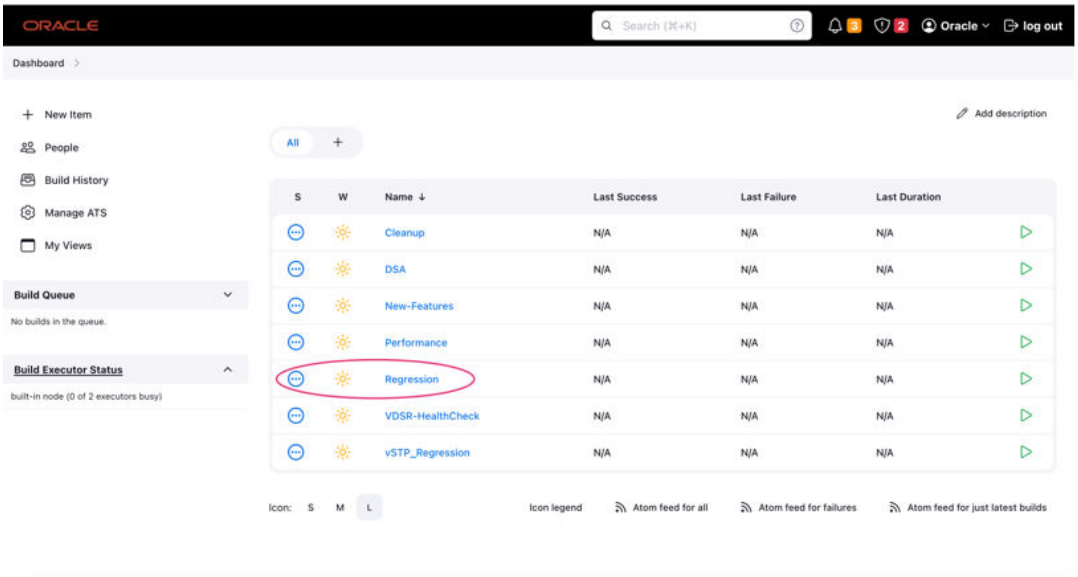
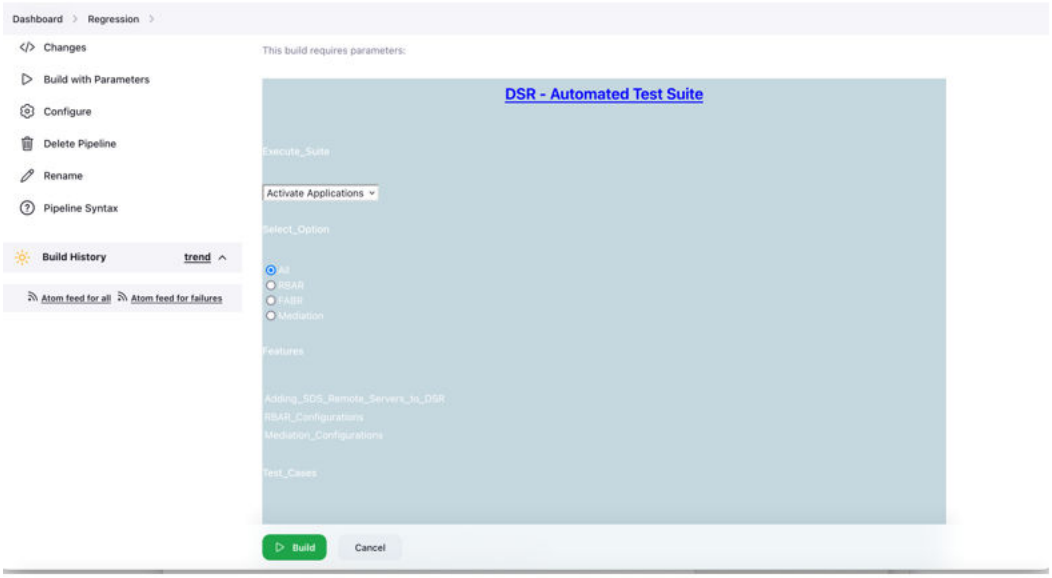


Figure 6-4 Activation of Applications



Location of all the activation related feature files:

- RBAR, FABR and Mediation
  - The feature Activation file is in the following directory path `/var/lib/jenkins/workspace/Regression/Cust-Roaming-Suite/Activation.feature`
- RSA
  - The RSA activation file is the following directory path:

```
/var/lib/jenkins/workspace/New-Features/Activate_RSA.feature
```

- DSA
  - File: The DSA activation file is the following directory path: `/var/lib/jenkins/workspace/DSA/Activate_DSA.feature`

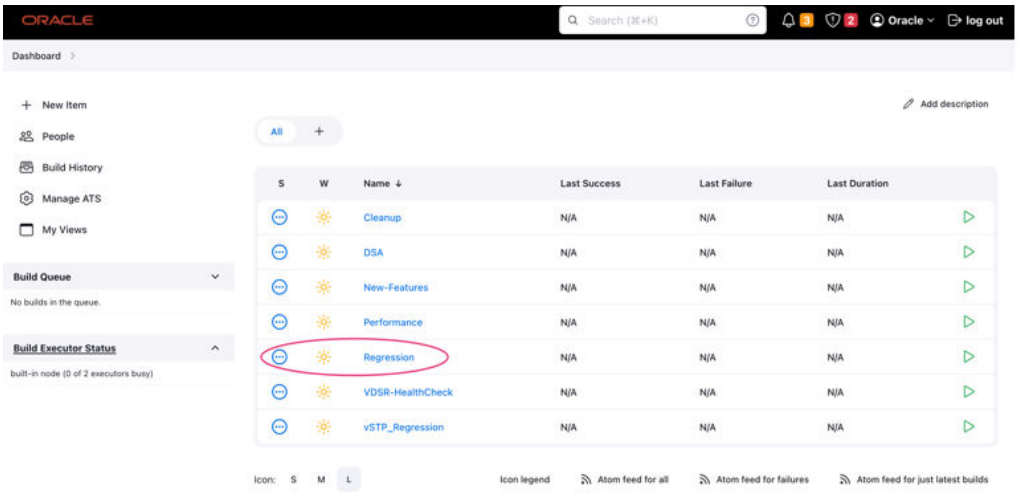
Activating the features

RBAR FABR Mediation:

a. Log in to ATS Jenkins GUI `https://<ATS IP>:8443`

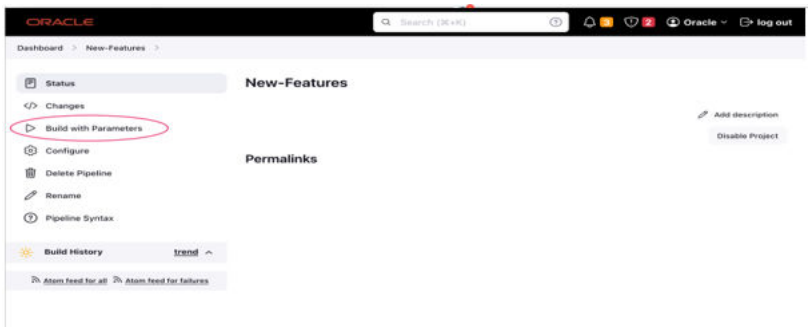
b. To activate ATS Jenkins GUI, click **Regression**, as shown in the following image:

Figure 6-5 Regression



c. Click **Build with Parameters** to build the parameters required for the activation.

Figure 6-6 Build with Parameters



d. You can either select the *All* option as shown below or, you can select to run each feature (RBAR, FABR, and Mediation) individually.

e. Click **Build** to **Activate Applications.**

Figure 6-7 Build

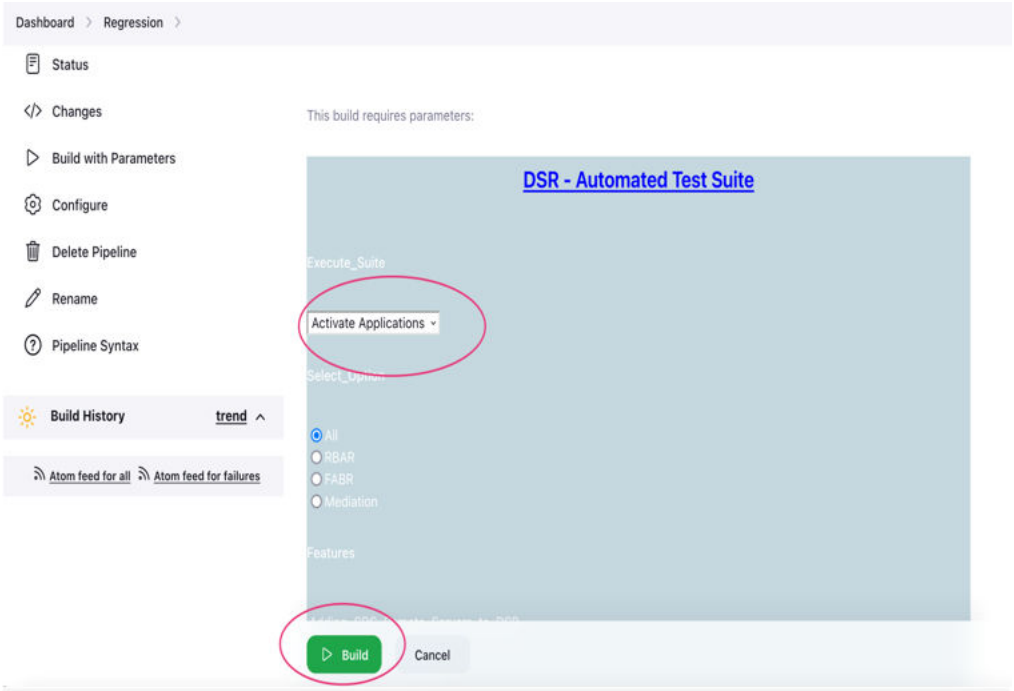
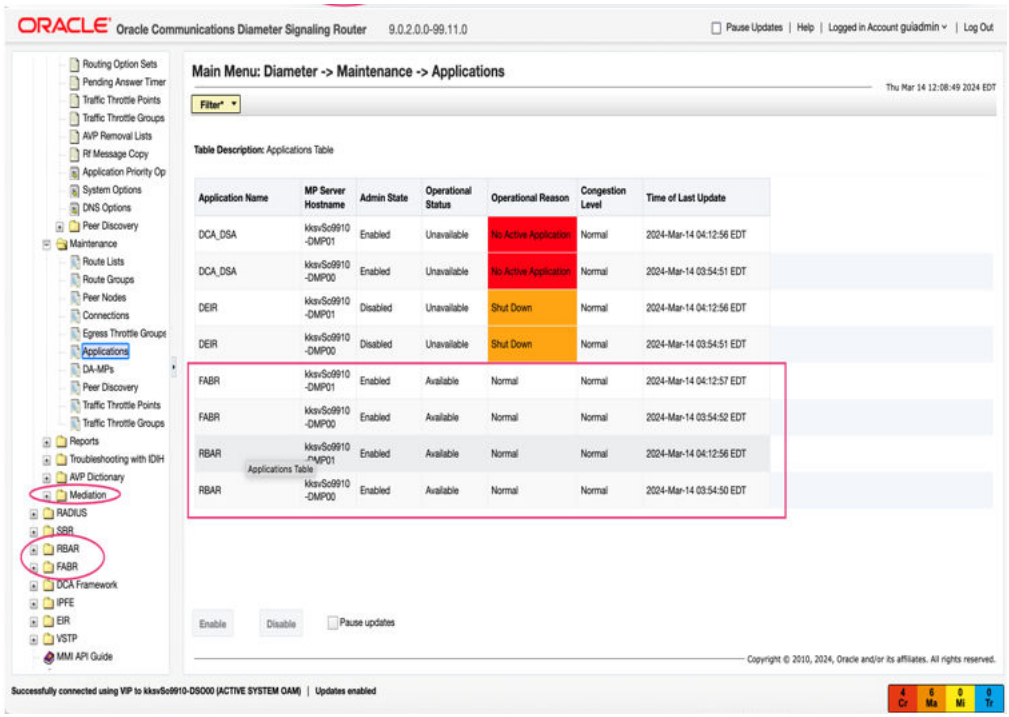


Figure 6-8 Applications



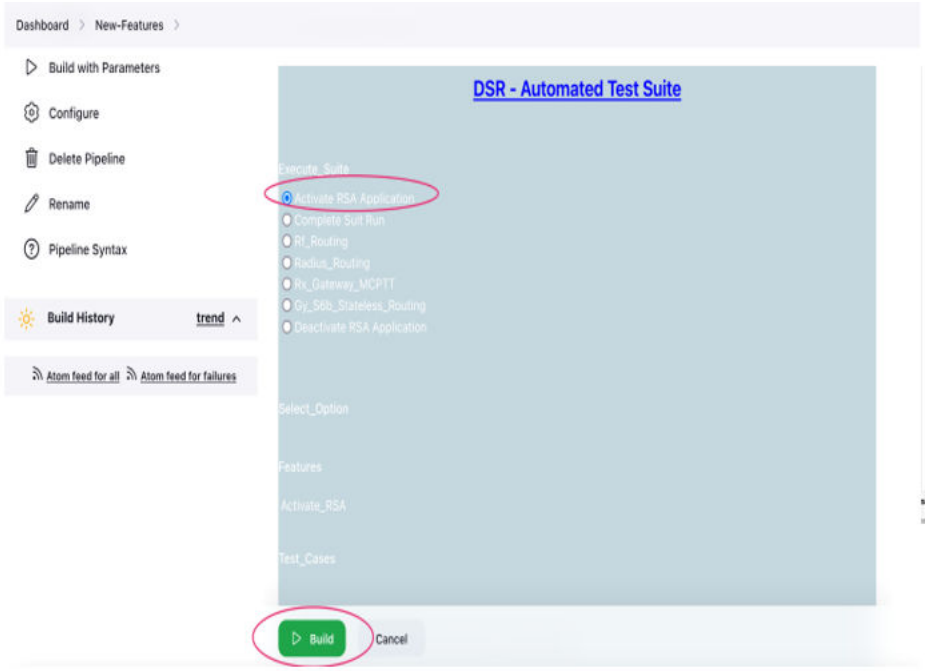
f. RSA:

   i. Log in to ATS's Jenkins GUI `https://<ATS IP>:8443`.

   ii. Select **New-Features**, and then click **Build with Parameters** .

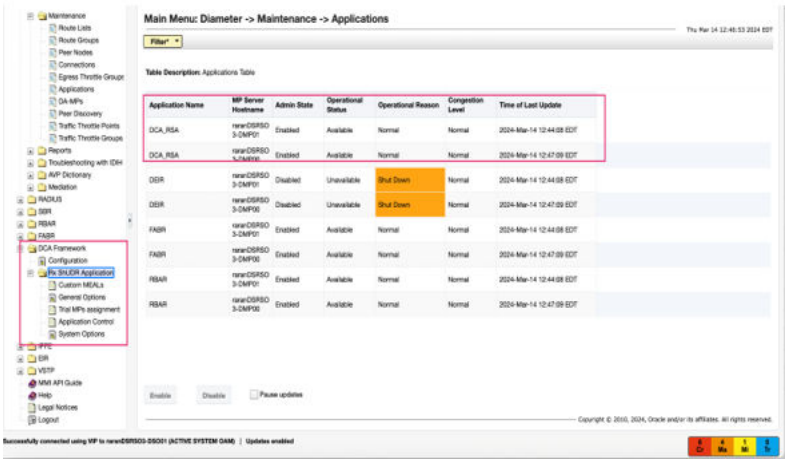
g. Select **Activate RSA Application**, and click **Build**.

Figure 6-9 Activate RSA



h. The following image shows the result

Figure 6-10 Result



i. To deactivate, select **Deactivate RSA Application**, and click **Build.**

Figure 6-11 Deactivate

j. To configure DSA:

i. Log in to ATS Jenkins GUI `https://<ATS IP>:8443`.

ii. Select **DSA**, as shown in the following image, and then click **Build with Parameters**

Figure 6-12 DSA feature

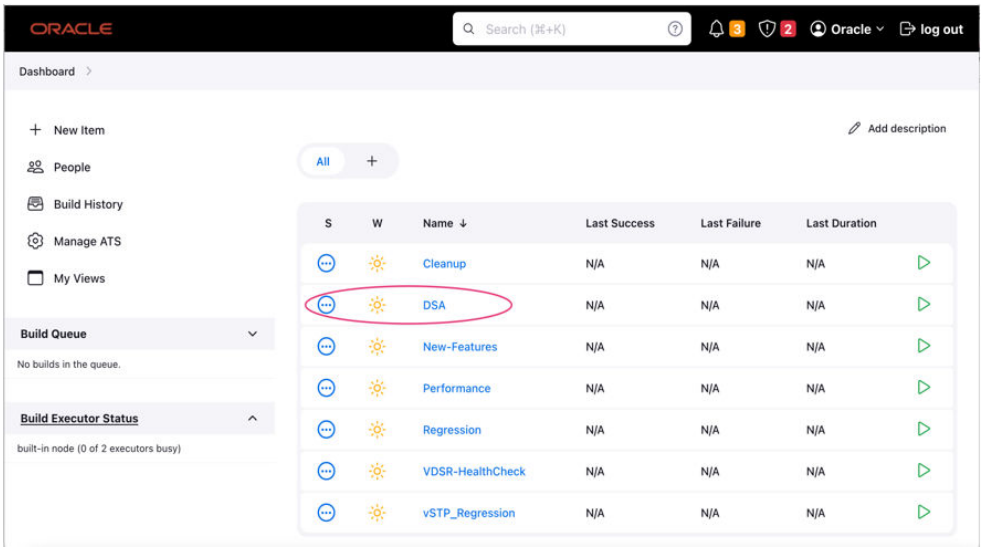
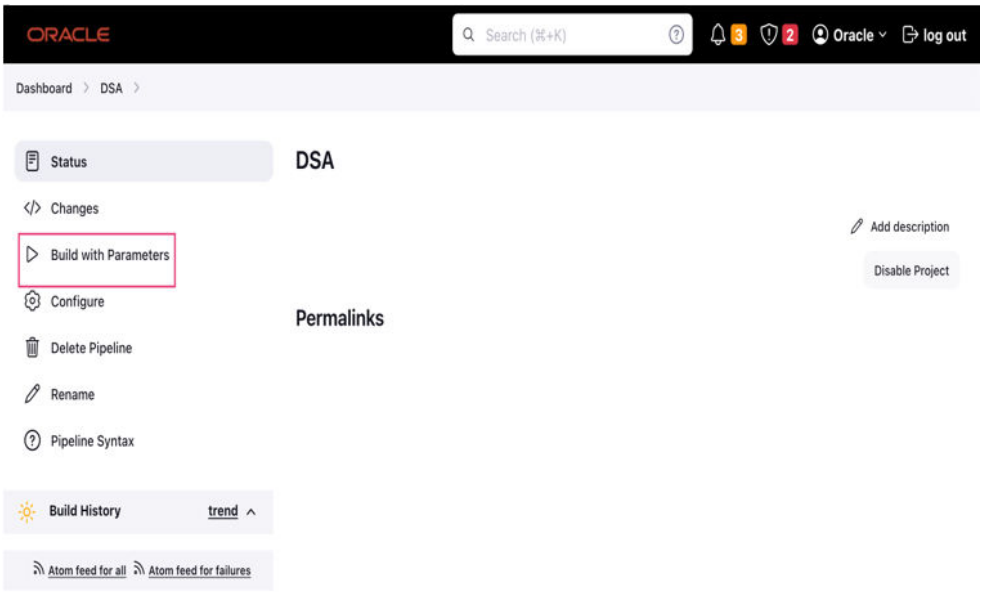
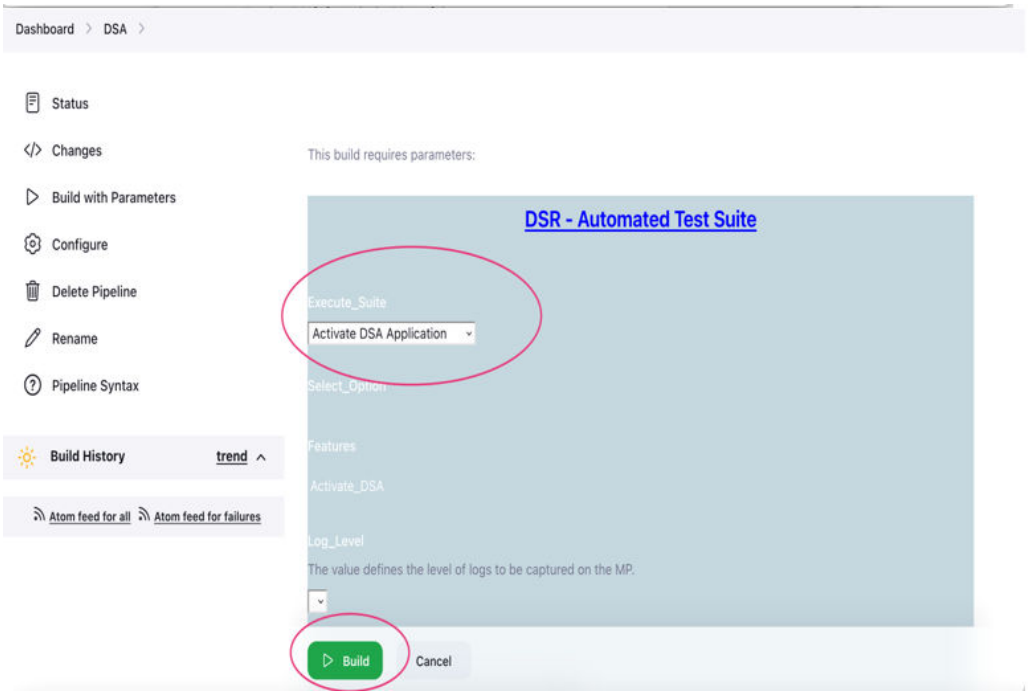


Figure 6-13 DSA



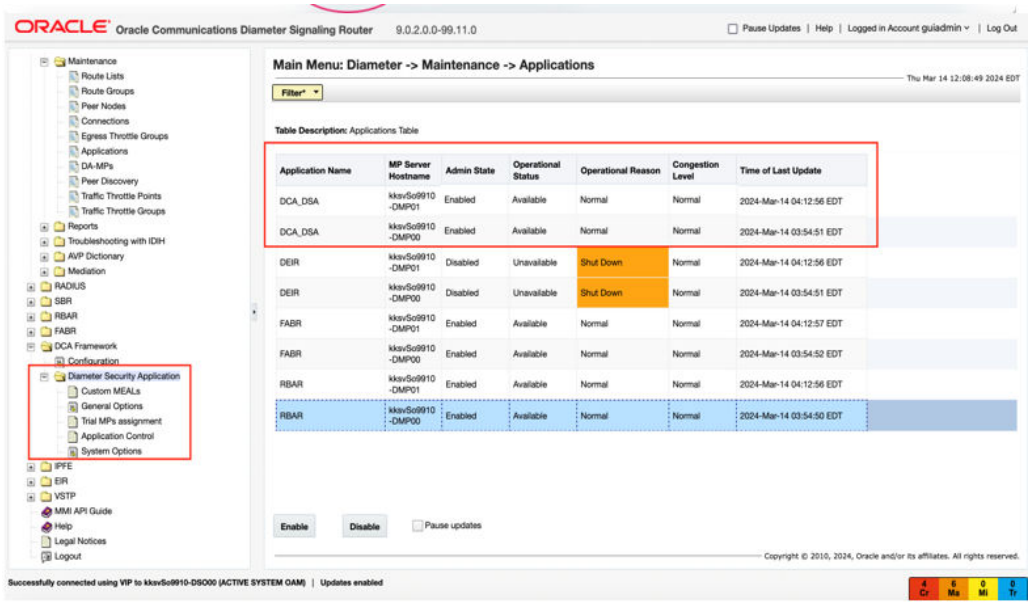
k. To activate, select **Activate DSA Application**, and then click **Build**.

Figure 6-14 Activate DSA



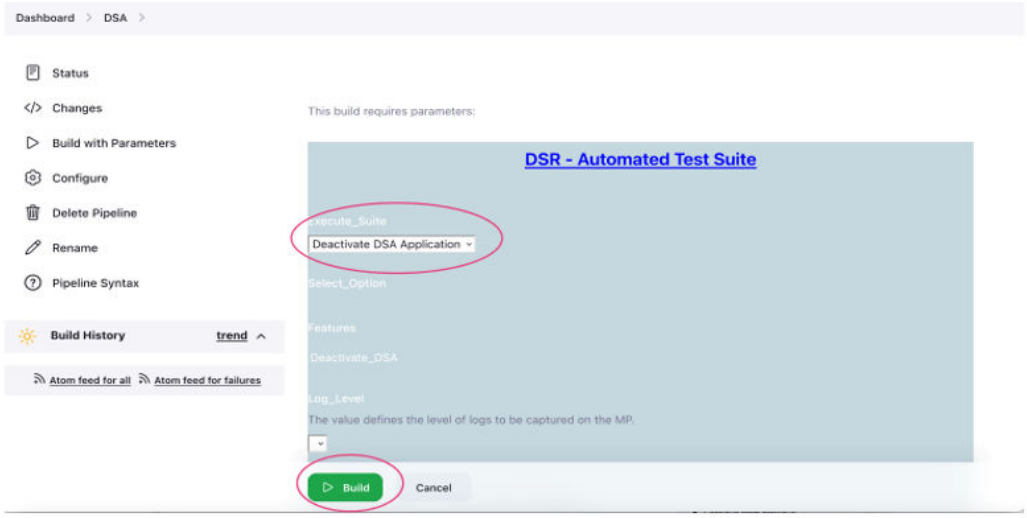
l. Following is the result shown in the image:

Figure 6-15 Result



m. To deactivate, select **Deactivate DSA Application**, and then click **Build**.

Figure 6-16 Deactivate DSA application



Things to remember:

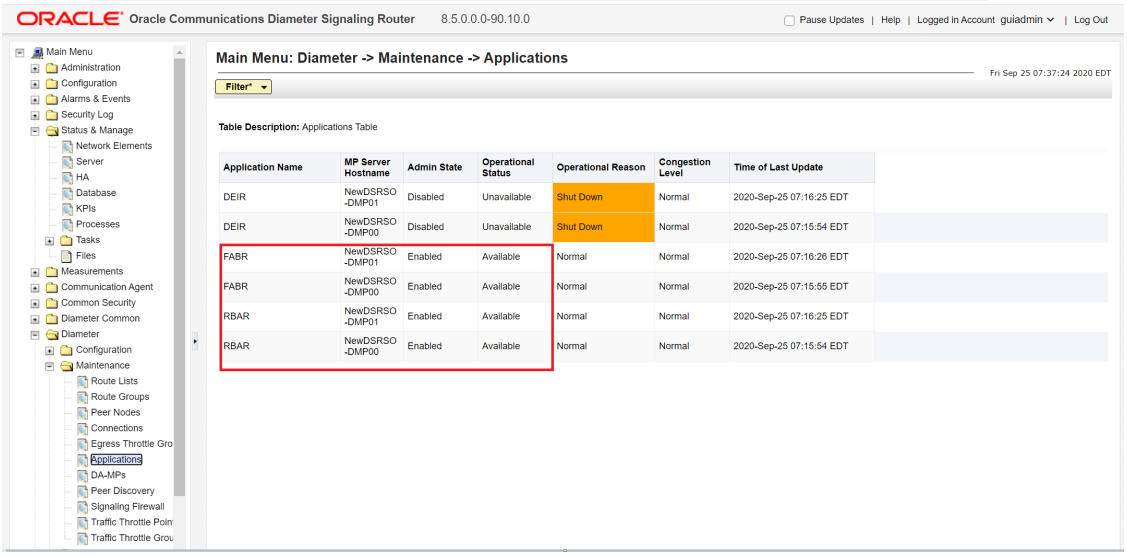
- Deactivate DSA before Activating RSA, and vice versa.
- Activation and Deactivation option for respective suites are available in DSA and New-Features suites, respectively.
- For FABR, RBAR, and Mediation, there is no Deactivate option, since they serve as prerequisites for multiple suites.

Activation Manually (As done in release 9.0.1 and before Excluding DSA)

> **Note:**
>
> Ensure that there are no backup files in the /var/TKLC/db/filemgmt/backup location while executing the database restore test case on SOAM. If there are backup files, then the Mediation feature must be enabled in each backup file. Otherwise, sometimes the Mediation feature might get disabled while execution of this test case.

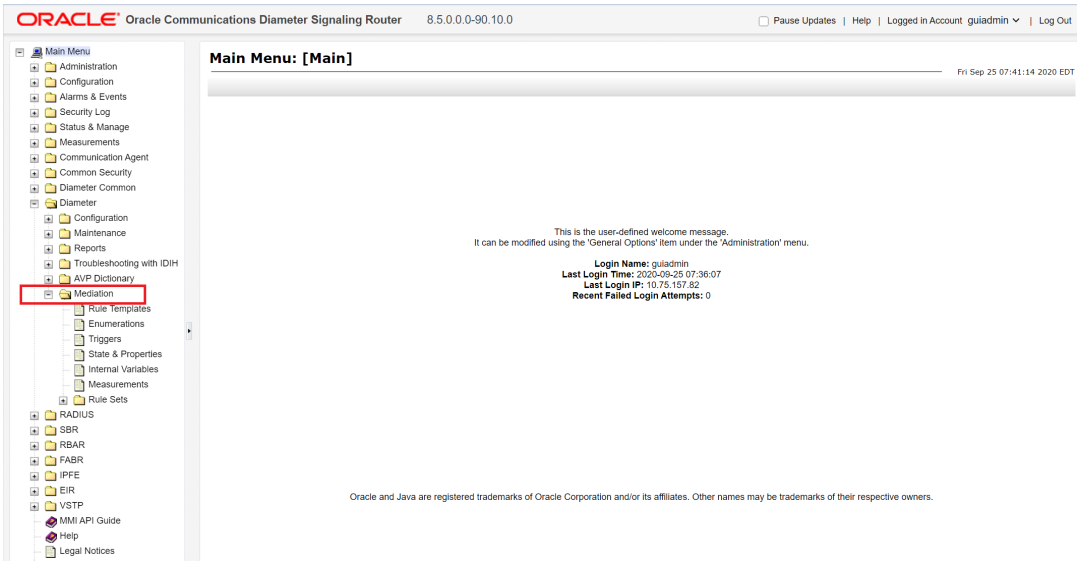
Ensure that RBAR and FABR are enabled as displayed in the following image:

Figure 6-17 RBAR and FABR Enabled on the DSR GUI



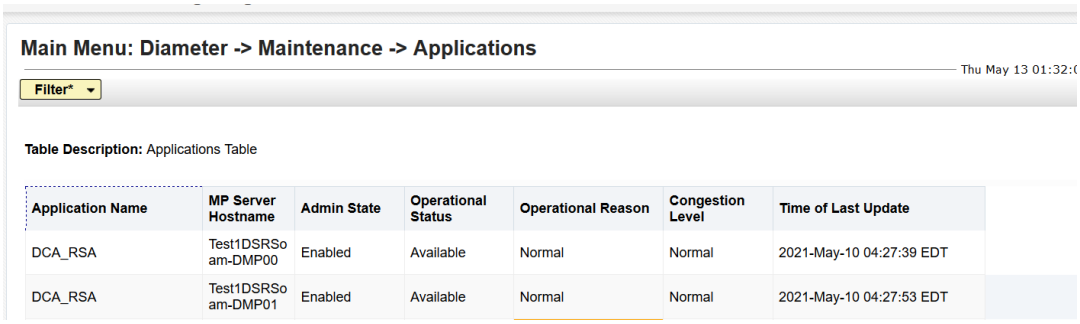
- Ensure that Mediation is enabled as displayed in the following image:

Figure 6-18 Mediation Enabled on the DSR GUI



- Ensure that DCA is enabled as displayed in the following image:

Figure 6-19 DCA Enabled on the DSR GUI



- Configure ComAgent connections on DSR by referring to the *Diameter Signaling Router Cloud Installation Guide*.

> **Note:**
>
> The DSR BUG 29035530 can cause ATS GUI case failure due to the "Security Violation" error when you perform any common GUI operation. This can be identified in `/var/TKLC/appw/logs/Process/AppWorksGui.log` by searching for the `Security violation by a user` keyword.

6.2 Test Case Execution Process

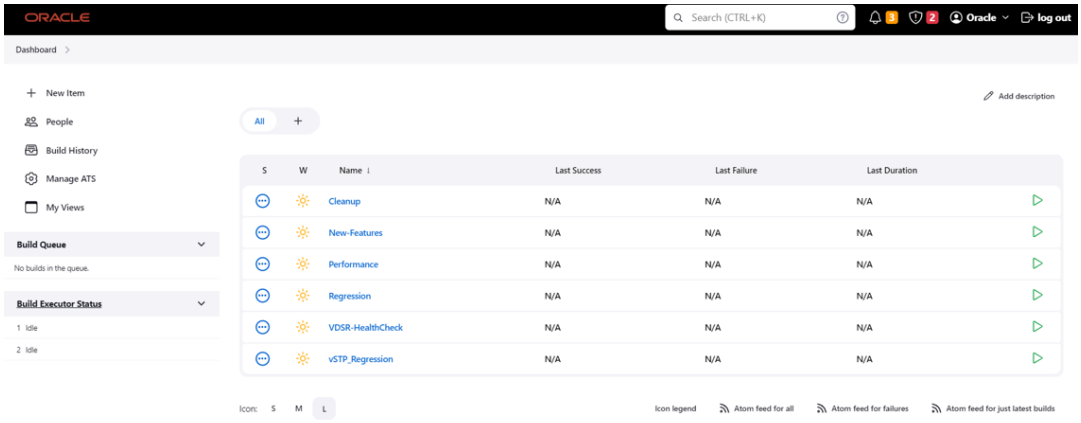
Perform the following procedure to run the test cases and check the VDSR health status.

Complete the tasks described in Prerequisites for Test Case Execution.

1. Go to `https://<ATS_IP>:8443/`.
2. Log in to the Jenkins GUI using your log in credentials.

The system displays the Jenkins GUI.

Figure 6-20 Jenkins GUI



3. To run the required test cases, in the Fav column, click the corresponding

- **vSTP_Regression**: This suite runs vSTP related regression test cases. It has four in suites which are sets of features related to vSTP functionality. If a single feature must be run, it can be run only through CLI.
- **New-Features**: This suite contains the following new features:
  - Rf_Routing
  - Radius_Routing
  - Rx_Gateway_MCPTT
  - Gy_S6b_Stateless_Routing
- **Performance**: This suite checks whether the performance testcases are passed on the current DSR build. It runs the Relay and FABR traffic. This suite consists of Diameter_Traffic and Radius_Traffic execution suites.
- **Regression**: This suite consists of all the Roaming and Core testcases. It contains all the testcases as per the requirement document.
- **VDSR-HealthCheck**: This suite checks the status of VDSR. This suite checks whether all the prerequisites are complete or not.
- **Cleanup**: This suite consists of cleanup feature to perform cleanup on SUT.
- **DSA**: Contains two suites of DSA stateful and stateless test cases.

> **Note:**
>
> You can run these suites in any sequence, however, it is recommended to run the **VDSR-HealthCheck** suite first.

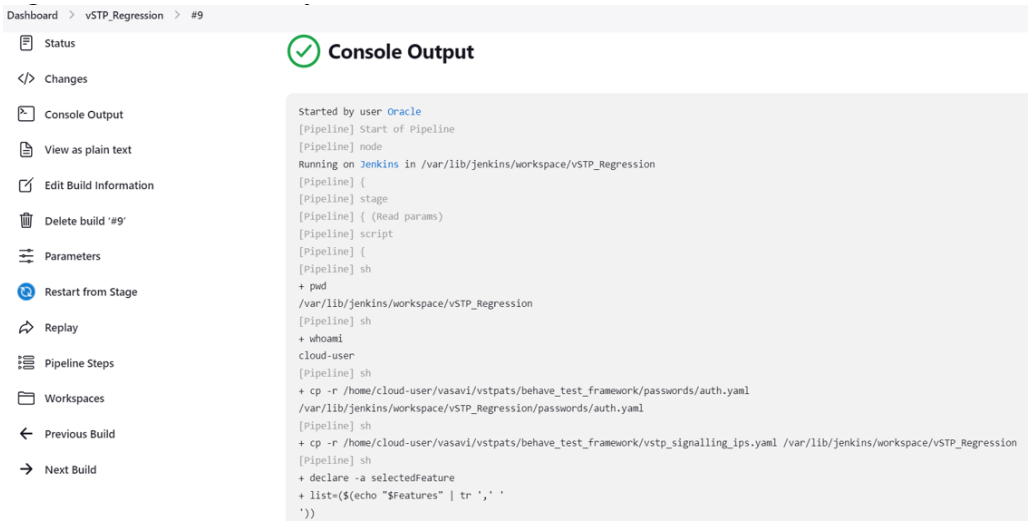
4. To perform the VDSR health check, click the corresponding  button.

a. In the lower-left corner of the GUI, in the Build History area, click  or  to check the log in Console Output.

If the build has failed, then the  icon appears red (). This indicates that the log contains DSR alarms.

The following image provides an example of a console output:

Figure 6-21 Console Output



b. If the log contains DSR alarms, clear the alarm and then perform the VDSR health check again by clicking the corresponding health check button (

).

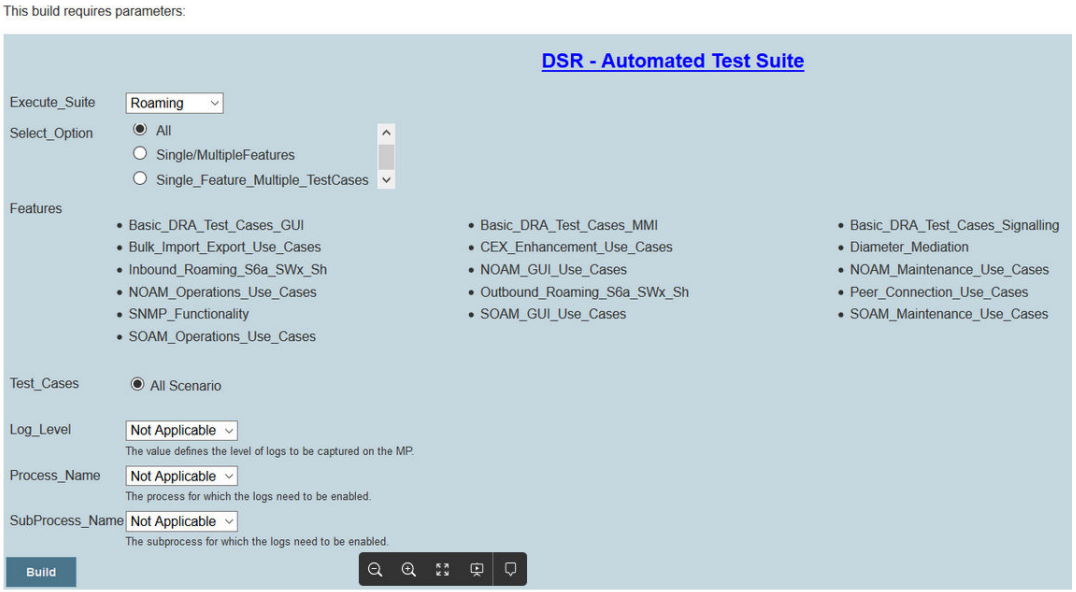
If the build is successful, in the Build History area, the  turns  .

5. To perform Regression, click the corresponding

button.

The DSR - Automated Test Suite page appears.

Figure 6-22 Regression Parameters



a. Configure the parameters as described in Regression Parameters.

   You can change the rerun count in the `/var/lib/jenkins/workspace/Regression` directory.

b. Click **Build**.

c. In the lower-left corner of the GUI, in the Build History area, click  or  to check the log in Console Output.

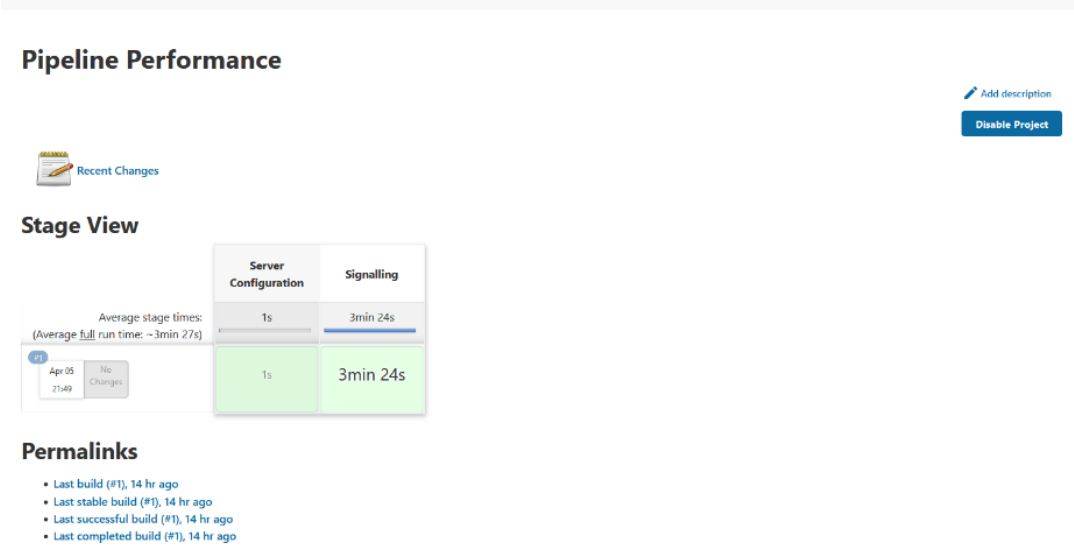
6. To check the Performance, click the corresponding

   button.

   The following image provides an example of a Performance build.

Figure 6-23 Performance Build



- In the lower-left corner of the GUI, in the Build History area, click  or  to check the log in Console Output.

6.2.1 Regression Parameters

The following table describes regression build parameters:

Table 6-4 Regression Build Parameters

Parameter	Description
`Execute_Suite`	By default, the value of this parameter is **Roaming**. You can run either **Roaming** or **Core_DSR** suite.
`Select_Option`	This parameter has three radio buttons to perform the following tasks: • **All**: To run all the Roaming cases. • **Single/MultipleFeatures**: To run multiple feature files together but not all. You must enable the check box of the required features to be executed under the `Features` parameter. • **Single_Feature_Multiple_TestCases**: To run single or multiple testcases within the same feature file. You must enable the radio button of the required features to be executed under the `Features` parameter. When it is completed, select the check box of the desired testcase to be executed under the `Test_Cases` parameter on the Jenkins GUI.

Table 6-4 (Cont.) Regression Build Parameters

Parameter	Description
Log_Level	This parameter defines the log level of DSR that can be enabled on the MP. It provides a drop-down of pre-defined log levels. The default value is **Not Applicable**. This parameter works only when the Single_Feature_Multiple_TestCases option is selected.
Process_Name	This parameter allows users to define the process name for which the logs are being enabled. It provides a drop-down of pre-defined processes in DSR. The default value is **Not Applicable**. This parameter works only when the Single_Feature_Multiple_TestCases option is selected.
SubProcess_Name	This is a string parameter. Users can parse the value of sub process for which the logs are enabled, for example, DRL, DCL, FBR, RBR, and so on. The default value is **Not Applicable**. This parameter works only when the Single_Feature_Multiple_TestCases option is selected.

A

Workaround for Password Expiry

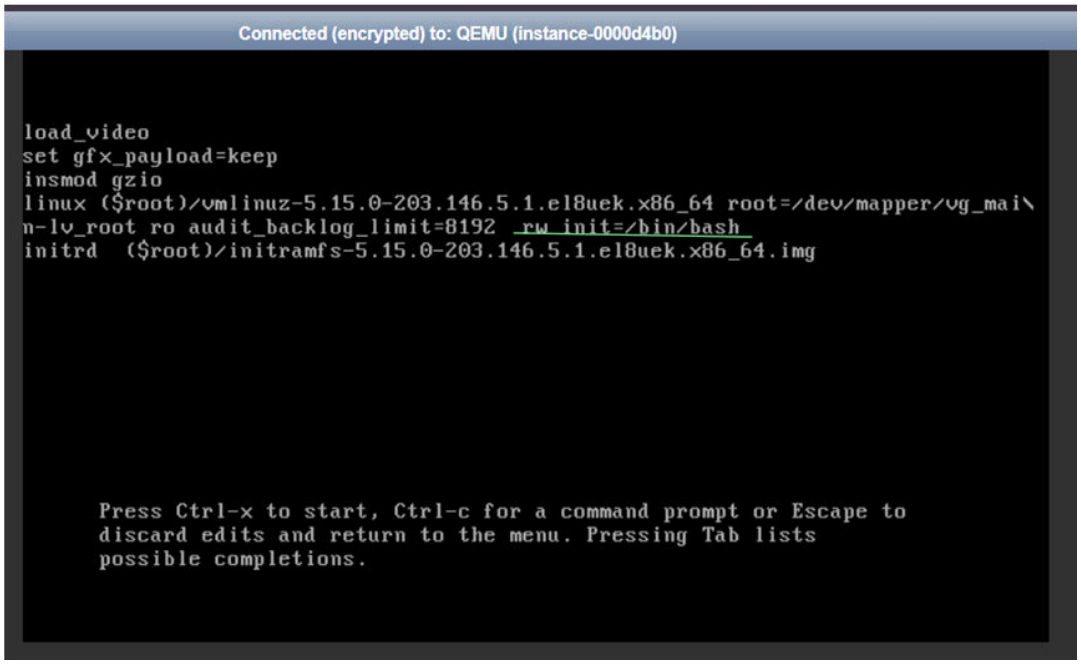
Perform the following workaround if there is an issue observed during log in:

1. Open the OpenStack console and click **Send CtrlAltDel**.

Figure A-1 Console

2. Press "e" continuously and add the `rw init=/bin/bash` when the following screen appears:
Below is a sample screenshot:

Figure A-2 Command bash



3. To reboot, press Ctrl X or F10.
4. Wait for the # prompt, then remount the filesystem in read or write mode using the following command:

```
sudo mount -o remount,rw /
```

Figure A-3 Remounting the Filesystem

5. Run the following command to verify the existing settings of password expiry:

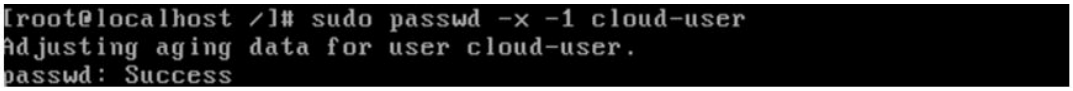
```
sudo chage -l cloud-user
```

Figure A-4 Root password

6. Run the following command to set the cloud user's password expiry to **never**:

```
sudo passwd -x -1 cloud-user
```

Figure A-5 Cloud user password



7. Run the following command to verify the settings of the password expiry:

```
sudo chage -l cloud-user
```

Figure A-6 Expiry date change

8. If SELinux is enabled, run the following command to relabel the entire filesystem:

```
sudo touch /.autorelabel
```

9. Run the following command to restart the system:

```
sudo reboot -f
```

Figure A-7 SElinux Relableing