Oracle® Communications

Diameter Signaling Router

DSR API Gateway Installation Guide

Release 8.3

E93571-01

September 2018

ORACLE®

Oracle Communications DSR API Gateway Installation Guide, Release 8.3.

CAUTION: Use only the Upgrade procedure included in the Upgrade Kit.

Before upgrading any system, please access My Oracle Support (MOS) (https://support.oracle.com) and review any Technical Service Bulletins (TSBs) that relate to this upgrade.

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

Call the CAS 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.

See more information on My Oracle Support (MOS).

Table of Contents

List of Tables

List of Figures

List of Procedures

1. Introduction

This document describes the installation procedures for a OpenStack HEAT template. This document assumes platform-related configuration has already been done. The audience for this document includes Oracle customers as well as these groups: Software System, Product Verification, Documentation, and Customer Service including Software Operations and First Office Application.

This document contains the procedures for these components of DSR APIGW:

- OCSG Database Server
- OCSG Admin Server
- OCSG Application Server

1.1 References

[1] DSR Cloud Benchmarking Guide

[2] DSR Cloud Installation Guide

[3] DSR API Gateway User Guide

1.2 Acronyms

An alphabetized list of acronyms used in the document.

Table 1: Acronyms

Acronym	Definition
APIGW	API Gateway
CLI	Command Line Interface
DSR	Diameter Signaling Router
KVM	Kernel-based Virtual Machine
OCSG	Oracle communications services Gatekeeper
OHC	Oracle Help Center
OVA	Open Virtualization Archive
OVM-M	Oracle VM Manager
OVM-S	Oracle VM Server
PEM	Privacy Enhanced Mail
SSO	Single Sign On
YAML	Yet Another Markup Language

1.3 How to Use This Document

Although this document is primarily to be used as an initial installation guide, its secondary purpose is as a reference for disaster recovery procedures. When executing this document for either purpose, there are a few points to help ensure you understand this document's intent. These points are:

1. Before beginning a procedure, completely read the instructional text (it will appear immediately after the Section heading for each procedure) and all associated procedural WARNINGS or NOTES.

2. Before execution of a STEP within a procedure, completely read the left and right columns including any STEP specific WARNINGS or NOTES.

If a procedural STEP fails to execute successfully, STOP and contact Oracle's Customer Service for assistance before attempting to continue. See Appendix G, for information on contacting Oracle Customer Support.

Figure 1 shows an example of a procedural step used in this document.

- Any sub-steps within a step are referred to as step X.Y. The example in Figure 1 shows steps 1 through 3, and step 3.1.
- GUI menu items, action links, and buttons to be clicked on are in bold Arial font.
- GUI fields and values to take note of during a step are in bold Arial font.
- Where it is necessary to explicitly identify the server on which a particular step is to be taken, the server name is given in the title box for the step (for example, "ServerX" in step 2 Figure 1).

Each step has a checkbox the user should check to keep track of the progress of the procedure.

The Title column describes the operations to perform during that step.

Each command the user enters, and any response output, is formatted in 10-point `Courier` font.

	Title	**Directive/Result Step**
1. ☐	Change directory	Change to the backout directory. `$ cd /var/TKLC/backout`
2. ☐	ServerX: Connect to the console of the server	Establish a connection to the server using cu on the terminal server/console. `$ cu –l /dev/ttyS7`
3. ☐	Verify Network Element data	View the Network Elements configuration data; verify the data; save and print report. 3. Select **Configuration > Network Elements** to view Network Elements Configuration screen.

Figure 1. Example of a Procedure Steps Used in This Document

2. Installation Overview

This section provides a brief overview of the recommended method for installing HEAT templates.

2.1 Prerequisites

These prerequisites are needed to install DSR APIGW:

1. KVM/OpenStack admin and tenant privileges.
2. OCSG Patches must be downloaded from mysupport (if required).
3. DSR APIGW OVA (will be used for Admin and Application Server intallation).
4. DSR 8.3 OVA files (will be used for Database Server intallation).
5. **.pem** file must be avaiable in Openstack.

6. DSR APIGW Database server must be configured and accessible from DSR APIGW Admin and Application VMs.
7. Following YAML files are required:
   - For DSR APIGW Admin/Application server: **dsrapigw.yml** and **dsrapigw_env.yml**.
   - For DSR APIGW Database server: dsrResources_provider.yaml and dsrResourcesNoVip_provider.yaml.
8. Qemu-img tool must be available to convert VMDK to qcom2 format, if required.

2.2 Installation Procedures

The following table illustrates the progression of the installation process by procedures. The phases outlined in are to be executed in the order they are listed.

Installation and configuration of instances can be performed either on VMware or using KVM/Openstack. On KVM/Openstack; and you can install and configure instances either manually or using HEAT templates.

2.2.1 Install and Configure Instances Using VMware

Table 2 explains the sequence to be followed using VMware:

Table 2: Install and Configure Instances on VMware

Procedure	Title	Description
Procedure 1	Import DSR APIGW Database and Admin/Application OVAs	Import both DSR APIGW database, and admin/application server OVAs
Procedure 2	Create DSR APIGW Database VMs	Create DSR APIGW database servers
Procedure 10	Configure DSR APIGW Database Server	Configure the DSR APIGW database server
Procedure 11	Configure DSR APIGW DB Server Group	Configure database server group
Procedure 12	Configure GatekeeperAuditPlugin on DSR APIGW Database Server	Configure GatekeeperAuditPlugin on database server
Procedure 3	Create DSR APIGW Admin/Application Servers	Create admin and application VMs
Procedure 13	Configure DSR APIGW Admin/Application Server	Install and configure DSR APIGW admin/application server(s)

2.2.2 Install and Configure Instances on KVM/Openstack

Table 3 explains the sequence to be followed on KVM/Openstack:

Table 3: Install And Configure Instances on KVM/Openstack Manually

Procedure	Title	Description
Procedure 4	Import DSR APIGW Database and Admin/Application OVAs (Openstack)	Import both DSR APIGW database, and admin/application server OVAs
Procedure 5	Create DSR APIGW Database VMs (Openstack)	Create DSR APIGW database servers

Procedure	Title	Description
Procedure 10	Configure DSR APIGW Database Server	Configure the DSR APIGW database server
Procedure 11	Configure DSR APIGW DB Server Group	Configure database server group
Procedure 12	Configure GatekeeperAuditPlugin on DSR APIGW Database Server	Configure GatekeeperAuditPlugin on database server
Procedure 6	Create DSR APIGW Admin/Application VMs (Openstack)	Create DSR APIGW admin/application VMs
Procedure 13	Configure DSR APIGW Admin/Application Server	Install and configure DSR APIGW admin/application server(s)

Table 4: Install And Configure Instances on KVM/Openstack Using Heat Template

Procedure	Title	Description
Procedure 7	Download Openstack HEAT Template and Parameter File	Download the template and parameter files from OHC
Procedure 8	Create DSR APIGW Database and Admin/Application Parameter File	Create parameter file based on your configuration
Procedure 9	Deploy DSR APIGW Database and Admin/Application Using HEAT Templates	Deploy the servers using HEAT template
Procedure 10	Configure DSR APIGW Database Server	Configure the DSR APIGW database server
Procedure 11	Configure DSR APIGW DB Server Group	Configure database server group
Procedure 12	Configure GatekeeperAuditPlugin on DSR APIGW Database Server	Configure GatekeeperAuditPlugin on database server
Procedure 13	Configure DSR APIGW Admin/Application Server	Install and configure DSR APIGW admin/application server(s)

2.3 Network Model

Figure 2 shows the supported network model for DSR APIGW deployments. DSR APIGW is deployed in cluster mode and one to one mapping should be maintained between the DSR site and DSR APIGW cluster. The DSR APIGW deployment model has three networks:

1. XMI — External Management Interface exposes the Administrative, Partner Management, and Partner portals. Ports 9002 are opened for management traffic on XMI. Links to portals:
   - Admin portal — https:<Admin-server-XMI-IP>:9002/console
   - Partner Management portal — https:<AppServer-XMI-IP>:9002/portal/partner-manager/index/login.html
   - Partner Portal — https://<AppServer-XMI-IP>:9002/portal/partner/index/partnerLogin.html
2. IMI — Internal Management Interface is used within DSR APIGW cluster between DSR APIGW and the database for internal communication.
3. XSI — External Signalling Interface is used to receive and send network traffic to and from app servers. Ports 10001 for http traffic, and 10002 for https traffic on XSI.

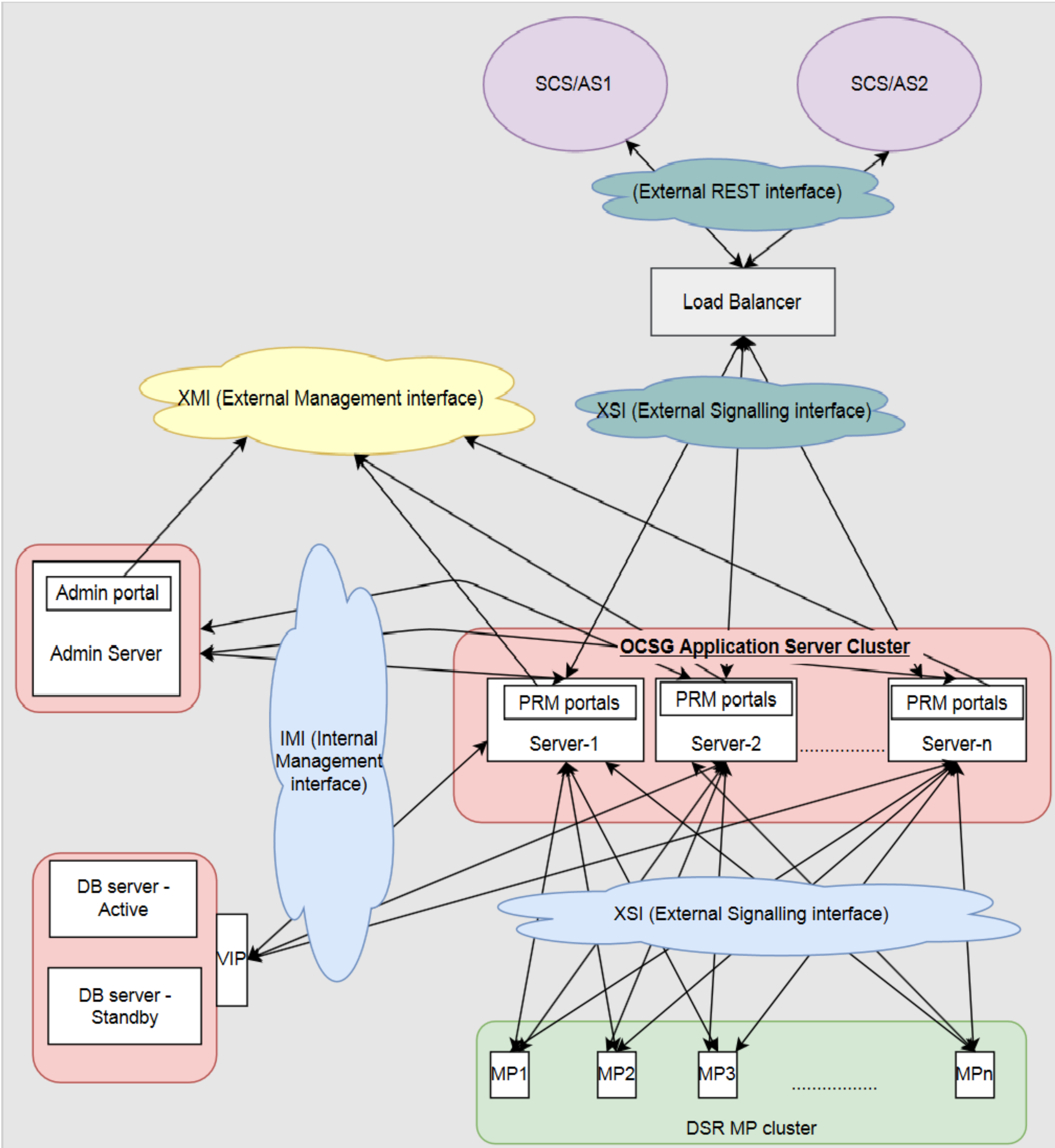


Figure 2: Network Model

3. Software Installation on VMware

The host configuration and virtual network setup is done before executing the procedures in this document. It is assumed that at this point the user has access to:

- Consoles of all guests and hosts at all sites
- ssh access to the guests at all sites
- GUI access to hosts at all sites
- A configuration station with a web browser, ssh client, and scp client
- VM manager privileges to add OVA's to catalog (VMware only)
- VMware, KVM/OpenStack admin and tenant privileges

3.1 Create Instances on VMware

3.1.1 Import DSR APIGW Database and Admin/Application OVAs (VMware)

This procedure imports the DSR APIGW database and Admin/Application OVAs to the VMware catalog or repository.

Procedure 1. Import DSR APIGW Database and Admin/Application OVAs (VMware)		
1. ☐	**VMware Client**: Add DSR APIGW database OVA image	1. Launch the VMware client of your choice. 2. Add the DSR APIGW Database OVA image to the VMware catalog or repository. Follow the instructions provided by the Cloud solutions manufacturer.
2. ☐	**VMware Client**: Add DSR APIGW Admin/Application OVA image	1. Launch the VMware client of your choice. 2. Add the DSR APIGW Admin/Application OVA image to the VMware catalog or repository. Follow the instructions provided by the Cloud solutions manufacturer.

3.1.2 Create DSR APIGW Database VMs (VMware)

This procedure creates database VMs.

Procedure 2. Create DSR APIGW Database VMs		
1. ☐	**VMware Client**: Create the DB1 VM from the OVA image	1. Browse to the library or repository where you placed the **DSR APIGW DB OVA** image. 2. Deploy the DSR APIGW DB OVA Image using vSphere Client or vSphere Web Client. 3. Name the **DB1** instances and select the data store.
2. ☐	**VMware Client**: Configure resources for the DB1 VM	Configure the **DB1** as per the NOAM resource profile from Appendix D to create the DB resource profile using the vSphere Client or vSphere Web Client.
3. ☐	**VMware Client**: Power on DB1	Use the vSphere Client or vSphere Web Client to power on the DB1 VM.

Procedure 2. Create DSR APIGW Database VMs		

4. ☐	**VMware Client**: Configure DB1	1. Access the DB1 VM console using the vSphere Client or vSphere Web Client.
		2. Login as the **admusr** user.
		3. Set the <ethX> device:
		Note: Where <ethX> is the interface associated with the XMI network.
		`$ sudo netAdm add --device=<ethX> --address=<IP Address in External management Network> --netmask=<Netmask> --onboot=yes --bootproto=none`
		4. Add the default route for ethX:
		Note: Add the gateway only to the externally routable network.
		`$ sudo netAdm add --route=default --gateway=<gateway address for the External management network> --device=<ethX>`
		5. Ping the XMI gateway for network verification.
		`$ ping -c3 <Gateway of External Management Network>`
		6. Depending on the number of instances, configure network interfaces (step 4) for each network (IMI, XSI1, XSI2, etc.)
		7. Restart network.
		`$ service network restart`

5. ☐	**VMware Client**: Configure DB2	Repeat steps 1 through 4 for the DB2 VM.

Note: For configuring databases refer to the Configure DSR APIGW Database and Configure GatekeeperAuditPlugin on DSR APIGW Database Server sections. For configuring the admin and application servers, refer to the Configure DSR APIGW Admin/Application Server section.

3.1.3 Create DSR APIGW Admin/Application VMs (VMware)

This procedure creates all admin and application servers.

Note: This procedure provides an example for creating an Admin. Follow the same steps to create other guests with their respective VM names and profiles.

Procedure 3. Create DSR APIGW Admin/Application Servers		
1. ☐	**VMware Client**: Create the Admin VM from the OVA image	1. Browse to the library or repository where you placed the **DSR APIGW OVA** image. 2. Deploy the OVA image using vSphere Client or vSphere Web Client. 3. Name the **Admin VM** and select the data store.
2. ☐	**VMware Client**: Configure resources for the Admin VM	Configure the **Admin VM** per the resource profiles defined in Appendix D for the DSR APIGW Admin server using the vSphere Client or vSphere Web Client. Interfaces must be added as described in Network Model section.

Procedure 3. Create DSR APIGW Admin/Application Servers		
3. ☐	**VMware Client**: Power on Admin VM	1. Power on the Admin VM with the vSphere Client or vSphere Web Client. 2. Monitor the vApps screen's Virtual Machines tab until the Admin VM reports **Powered On** in the Status column.

4. ☐	**VMware Client**: Configure XMI interface	1. Access the VM console using the vSphere Client or vSphere Web Client.
		2. Login as the **admusr** user.
		3. Set the <ethX> device:
		Note: Where ethX is the interface associated with the XMI network.
		`$ sudo netAdm add --device=<ethX> --address=<IP Address in External Management Network> --netmask=<Netmask> --onboot=yes --bootproto=none`
		4. Add the default route for ethX:
		Note: Add the gateway only to the externally routable network.
		`$ sudo netAdm add --route=default --gateway=<gateway address for the External management network> --device=<ethX>`
		5. Ping the XMI gateway for network verification.
		`$ ping -c3 <Gateway of External Management Network>`
		6. Depending on the number of instances, configure network interfaces (step 4) for each network (IMI, XSI1, XSI2, etc.).
		7. Restart network.
		`$ service network restart`

5. ☐	**VMware Client**: Verify network connectivity	1. Access the Admin VM console using the vSphere Client or vSphere web Client. 2. Login as the **admusr** user. 3. Ping the Admin. `$ ping -c3 <IP Address in External Management Network>`
6. ☐	**VMware Client**: Repeat for other Application VMs	Repeat steps 1 through 5 for the Application VMs. Use unique labels for the VM names.

4. Software Installation on KVM/Openstack

4.1 Create Instances on KVM/OpenStack Manually

4.1.1 Import DSR APIGW Database and Admin/Application OVAs (Openstack)

This procedure adds the DSR APIGW Admin/Application and Database OVA files to the glance image catalog.

<table>
<tr><th colspan="3">Procedure 4. Import DSR APIGW Database and Admin/Application OVAs (Openstack)</th></tr>
<tr><td>1.
☐</td><td>Openstack Controller: Preparation</td><td>Create instance flavors.</td></tr>
<tr><td>2.
☐</td><td>Openstack Controller: Add DSR APIGW database OVA image</td><td>1. Copy the DSR APIGW database OVA file from the Oracle repository to the OpenStack control node.
<code>$ scp <user_name>@<Oracle Repository server>:<path-to-OVA>/DSR-8.3.0.0.0_83.x.0.ova .</code>
2. In an empty directory, unpack the OVA file using tar.
<code>$ tar xvf DSR-x.x.x.x.x.ova</code>
3. One of the unpacked files has a .vmdk suffix. This is the VM image file that must be imported.
DSR-x.x.x.x.x-disk1.vmdk
4. Source the OpenStack admin user credentials.
<code>$. keystonerc_admin</code>
5. Select an informative name for the new image.
dsr-8.3.x.x.x-original
6. Import the image using the glance utility from the command line.
<code>openstack image create --disk-format vmdk --container-format bare --public --file dsrapigw-x.x.x.x.vmdk dsrapigw-x.x.x.x-original</code>
This process takes about 5 minutes, depending on the underlying infrastructure. If you want to convert vmdk file to qcow2 format, refer to Appendix E.
Note: This process takes about 5 minutes, depending on the underlying infrastructure.</td></tr>
<tr><td>3.
☐</td><td>Openstack Controller: Add DSR APIGW OVA</td><td>Repeat this procedure to add DSRAPIGW-8.3.0.0.0_83.x.0.ova DSR APIGW OVA.</td></tr>
</table>

4.1.2 Create DSR APIGW Database VMs (Openstack)

This procedure creates database VMs.

Procedure 5. Create DSR APIGW Database VMs (Openstack)		
1. ☐	**Openstack Controller**: Name the new VM instance	1. Create an informative name for the new instance: **DB1**. 2. Examine the interfaces described in Network Model section.

2.
☐
	Openstack Controller: Create and boot the DB1 VM instance from the glance image	1. Get the following configuration values.
		The DSR APIGW Database image ID.
		$ glance image-list
		The flavor ID.
		$ nova flavor-list
		The network ID(s)
		$ neutron net-list
		An informative name for the instance.
		DB1
		2. Create and boot the VM instance.
		The instance must be owned by the DSR tenant user, not the admin user. Source the credentials of the DSR tenant user and issue this command. Use one **--nic** argument for each IP/interface. The number of IP/interfaces for each VM type must conform with the Network Model section.
		Note: IPv6 addresses should use the v6-fixed-ip argument instead of v4-fixed-ip.
		$ nova boot --image <image ID> --flavor <flavor id> --nic net-id=<first network id>,v4-fixed-ip=<first ip address> --nic net-id=<second network id>,v4-fixed-ip=<second ip address> <instance name>
		3. View the newly created instance using the nova tool.
		$ nova list --all-tenants
		The VM takes approximately 5 minutes to boot and may be accessed through the network interfaces and the Horizon console tool.

3. ☐	**Openstack Controller**: Check if interface is configured	If DHCP is enabled on the Neutron subnet, VM configures the VNIC with the IP address provided in step 2. To verify, ping the VM IP address provided with **nova boot…** command from step 2. $ ping <IP- Provided-During-Nova-Boot> If the ping is successful, ignore step 4 to configure the interface manually.

Procedure 5. Create DSR APIGW Database VMs (Openstack)		

4. ☐	**Openstack GUI**: Manually configure interface, if not already done (optional)	***Note:*** If the instance is already configured with an interface and has successfully pinged, then ignore this step to configure the interface manually.
		1. Log into the Openstack Horizon GUI.
		2. Go to the Compute/Instances section.
		3. Click the **Name** field of the newly created instance.
		4. Select the Console tab.
		5. Login as the **root** user.
		6. Configure the network interfaces.
		$ netAdm add --onboot=yes --device=eth0 --address=<xmi ip> --netmask=<xmi net mask>
		Note: Add the gateway only to the externally routable network.
		$ netAdm add --route=default --device=eth0 --gateway=<xmi gateway IP>
		7. Ping the XMI gateway for network verification.
		$ ping -c3 <XMI Gateway>
		8. Depending on the number of instances, configure network interfaces (step 6) for each network (IMI, XSI1, XSI2, etc.)
		9. Restart network.
		$ service network restart

5. ☐	**Openstack Controller**: Create DB2	Repeat steps 1 through 4 for DB2.

Note: For configuring databases refer to the Configure DSR APIGW Database and Configure GatekeeperAuditPlugin on DSR APIGW Database Server sections. For configuring the admin and application servers, refer to the Configure DSR APIGW Admin/Application Server section.

4.1.3 Create DSR APIGW Admin/Application VMs (Openstack)

This procedure configures all VMs, for example, Admin and Application servers.

Note: This procedure provides an example for creating an Admin. Follow the same steps to create other guests with their respective VM names and profiles.

Procedure 6. Create DSR APIGW Admin/Application VMs (Openstack)		
1. ☐	**Openstack Controller**: Name the new VM instance	1. Create an informative name for the new instance: **Admin**. 2. Examine the interfaces as described in Network Model section.

Procedure 6. Create DSR APIGW Admin/Application VMs (Openstack)		

2. ☐	**Openstack Controller**: Create and boot the Admin and Application VM instance from the glance image	1. Get the following configuration values.
		The DSR APIGW Database image ID.
		`$ glance image-list`
		The flavor ID.
		`$ nova flavor-list`
		The network ID(s)
		`$ neutron net-list`
		An informative name for the instance.
		• Admin
		• Application
		2. Create and boot the VM instance.
		Refer to Appendix D regarding the resource profile.
		The instance must be owned by the DSR tenant user, not the admin user. Source the credentials of the DSR tenant user and issue this command. Use one **--nic** argument for each IP/interface. The number of IP/interfaces for each VM type must conform with the DSR Network to Device Assignments defined in [1].
		Note: IPv6 addresses should use the `v6-fixed-ip` argument instead of `v4-fixed-ip`.
		• **Admin server**
		`nova boot --image <image ID> --flavor <flavor id> --nic net-id=<XMI network id>,v4-fixed-ip=<XMI ip address> --nic net-id=<IMI network id>,v4-fixed-ip=<IMI ip address> <instance name>`
		• **App server**
		`nova boot --image <image ID> --flavor <flavor id> --nic net-id=<XMI network id>,v4-fixed-ip=<XMI ip address> --nic net-id=<IMI network id>,v4-fixed-ip=<IMI ip address> --nic net-id=<XSI network id>,v4-fixed-ip=<XSI ip address> <instance name>`
		3. View the newly created instance using the nova tool.
		`$ nova list  --all-tenants`
		The VM takes approximately 5 minutes to boot and may be accessed through the network interfaces and the Horizon console tool.

3. ☐	**Openstack Controller**: Check if interface is configured	If DHCP is enabled on the Neutron subnet, VM configures the VNIC with the IP address provided in step 2. To verify, ping the VM IP address provided with **nova boot…** command from step 2. `$ ping <IP- Provided-During-Nova-Boot>` If the ping is successful, ignore step 4 to configure the interface manually.

Procedure 6. Create DSR APIGW Admin/Application VMs (Openstack)		

4. ☐	**Openstack GUI:** Manually configure interface, if not already done (Optional)	***Note:*** If the instance is already configured with an interface and has successfully pinged, then ignore this step to configure the interface manually.
		1. Log into the Openstack Horizon GUI.
		2. Go to the Compute/Instances section.
		3. Click the **Name** field of the newly created instance.
		4. Select the Console tab.
		5. Login as the **admusr** user.
		6. Configure the network interfaces, conforming with the Network Model section.
		`$ sudo netAdm add --onboot=yes --device=eth0 --address=<ip> --netmask=<net mask>`
		Note: Add the gateway only to the externally routable network.
		`$ sudo netAdm add --route=default --device=eth0 --gateway=<gateway ip>`
		7. Ping the gateway for network verification.
		`$ ping -c3 <Gateway>`
		Under some circumstances, it may be necessary to configure as many as 6 or more interfaces.
		8. Depending on the number of instances, configure network interfaces (step 6) for each network (IMI, XSI1, XSI2, etc.)
		9. Restart network.
		`$ service network restart`
		10. Reboot the Admin VM. It takes approximately 5 minutes for the VM to complete rebooting.
		`$ sudo init 6`
		The new VM should now be accessible through both the network and Horizon consoles.

5. ☐	Repeat for other application VMs	Repeat steps 1 through 4 for the other application VMs. Use unique labels for the VM names. Assign addresses to all desired network interfaces.

4.2 Create Instances on KVM/Openstack Using Heat Template

4.2.1 Download Openstack Template and Parameter Files

This procedure selects the templates and environment files needed to deploy DSR APIGW and DSR stacks.

Prerequisite: All the respective infrastructures has to be up and running.

Procedure 7. Download Openstack HEAT Template and Parameter Files		
1. ☐	Log into Oracle document repository - OHC	Log into the Oracle Document Repository at http://docs.oracle.com/en/industries/communications/diameter-signaling-router/index.html.
2. ☐	Select the DSR Release	Select the respective release folder, for example, Release 8.3.x.
3. ☐	Download HEAT templates	Download the **HEAT Templates** zip file.
4. ☐	**Openstack Controller**: Unzip the HEAT templates to a folder in Openstack	1. Log into the Openstack controller and navigate to home directory where you want to store the HEAT templates. 2. Create a new folder with any name for storing the HEAT templates under home directory, for example, **/home/heat_templates**. 3. Store the downloaded HEAT templates zip file to the folder. Example: /home/heat_templates/exampleHeat.zip 4. Unzip the downloaded heat templates. `unzip /home/heat_templates/exampleHeat.zip`
5. ☐	Determine the template and environment files	The HEAT templates contain files for all scenarios. Determine the appropriate template and parameter files with respect to your requirement. The YAML files for DSR APIGW admin/application servers are **dsrapigw.yml** and **dsrapigw_env.yml**. The YAML files for DSR APIGW database are **dsrResources_provider.yaml** and **dsrResourcesNoVip_provider.yaml**.

4.2.2 Create DSR APIGW Database and Admin/Application Parameter Files

This procedure manually creates the input parameters file needed to deploy DSR APIGW and DSR.

Prerequisite: All the respective infrastructures has to be up and running.

Procedure 8. Create DSR APIGW Database and Admin/Application Parameter Files		
1. ☐	**Openstack Controller**: Log into the Openstack controller	Log into the Openstack controller though command line.

Procedure 8. Create DSR APIGW Database and Admin/Application Parameter Files		
2. ☐	**Openstack Controller**: Create the parameter file	1. Navigate to the folder which is already created in the above procedure for storing the templates. 2. Create an empty parameter file in this folder, following the below naming convention just to identify the purpose of the file: • For DSR APIGW Database: <DSR Name>_Params.yaml Example: dsrCloudInit__Params.yaml • For DSR APIGW Admin/Application: <DSR APIGW Name>_Params.yml Example: dsrapigw_Params.yml
3. ☐	**Openstack Controller**: Sample file	Refer to Appendix A for a sample file with the values. ***Note:*** It is important to keep the Example File handy since this help understand the use of each Key Value pair described in the steps to create the parameter file.
4. ☐	**Openstack Controller**: Populate the parameters file	Refer Appendix A to create the parameter file in YAML format. ***Note:*** Follow these guidelines while working with the YAML files. • The file must end with .yaml extension. • YAML must be case-sensitive and indentation-sensitive. • YAML doesn't support the use of tabs. Instead of tabs, it uses spaces. 1. This file is in YAML format and it contains **key:value** pairs. 2. The first key should be **parameters:** and then the remaining required key/value pairs for the topology. 3. Refer to Appendix A for all required key value pairs.

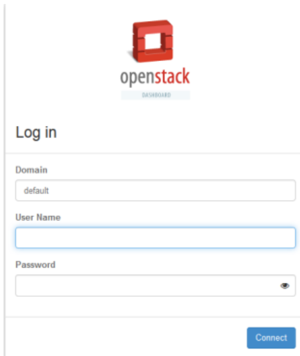
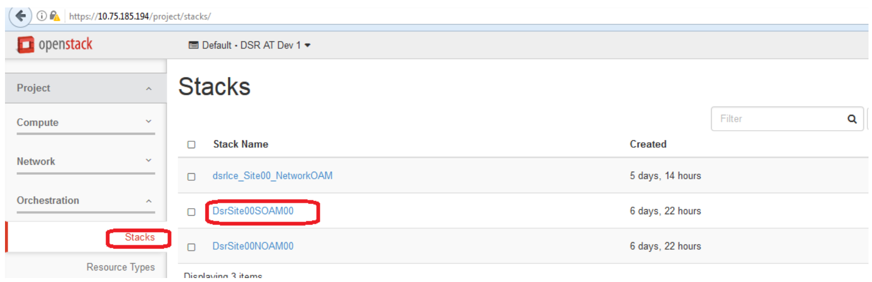
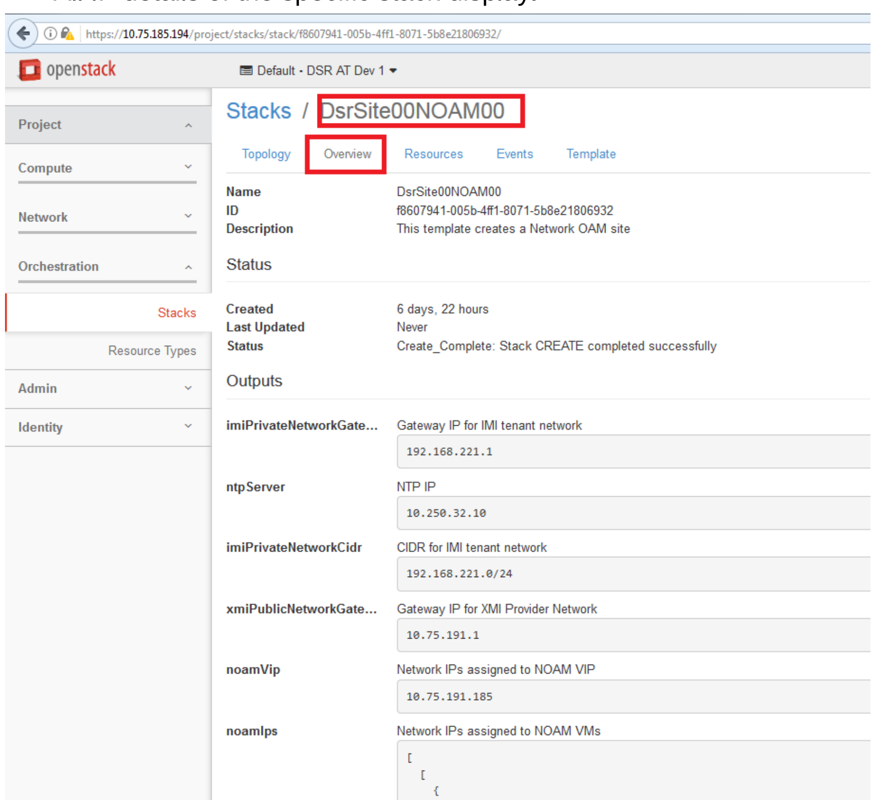
4.2.3 Deploy DSR APIGW Database and Admin/Application using HEAT Templates

This procedure deploys the HEAT templates to create the DSR APIGW admin and application stacks.

Prerequisite: All the respective infrastructures has to be up and running. The required input files are all available.

Procedure 9. Deploy DSR APIGW Database and Admin/Application Using HEAT Templates		
1. ☐	**Openstack Controller**: Log into OpenStack server CLI	If not already done, log into the OpenStack CLI.
2. ☐	**Openstack Controller**: Prepare the input files required for the deployment	Gather the information needed for the parameter file to deploy the HEAT templates and create the DSR APIGW admin and application stacks.

<table>
<tr><th colspan="3">Procedure 9. Deploy DSR APIGW Database and Admin/Application Using HEAT Templates</th></tr>
<tr><td>3. ☐</td><td>Openstack Controller: Deploy DSR APIGW stack</td><td>Execute the OpenStack command to create DSR APIGW admin and application stack, passing the three input files. Make sure the Template and Parameter files are selected with respect to DSR APIGW admin and application stack.<pre>openstack stack create -e <ParameterFile.yaml> -t
<TemplateFile></pre></td></tr>
<tr><td>4. ☐</td><td>Openstack Controller: Verify the stack creation status</td><td>1. After the OpenStack create commands are executed, execute the command to see the stack creation status:<pre>$ openstack stack show <stackname></pre><pre>+----------------------------------+-----------+--------------------+--------------------+
| ID | Name | Status | Created |
+----------------------------------+-----------+--------------------+--------------------+
| (uuid) | teststack | CREATE_IN_PROGRESS | (timestamp) |
+----------------------------------+-----------+--------------------+--------------------+</pre>2. It takes approximately two minutes to complete the creation. Execute the command again to verify the status.<pre>$ openstack stack show <stackname></pre><pre>+--------------------------------------+-----------------+-----------------
| ID | Stack Name | Stack Status
+--------------------------------------+-----------------+-----------------
| 950ed51a-cca7-478a-81e4-3d61562c045d | teststack | CREATE_COMPLETE</pre></td></tr>
</table>

Procedure 9. Deploy DSR APIGW Database and Admin/Application Using HEAT Templates		
5. ☐	**Openstack Controller**: Retrieve required IPs from created stacks	1. Log into Openstack GUI with valid credentials.  2. Navigate to **Project > Orchestration** and click **Stacks**.  3. Click on the stack you created (<stackname>) and click **Overview**. All IP details of the specific stack display. 

Procedure 9. Deploy DSR APIGW Database and Admin/Application Using HEAT Templates		

6. ☐	**Openstack GUI**: Manually configure interface, if not already done (Optional)	1. Log into the openstack Horizon GUI.
		2. Go to the Compute/Instances section.
		3. Click the **Name** field of the newly created instance.
		4. Select the Console tab.
		5. Login as the **root** user.
		6. Configure the network interfaces.
		`$ netAdm add --onboot=yes --device=eth0  --address=<xmi ip> --netmask=<xmi net mask>`
		Note: Add the gateway only to the externally routable network.
		`$ netAdm add --route=default --device=eth0 --gateway=<xmi gateway IP>`
		7. Ping the XMI gateway for network verification.
		`$ ping -c3 <XMI Gateway>`
		8. Depending on the number of instances, configure network interfaces (step 6) for each network (IMI, XSI1, XSI2, etc.)
		9. Restart network.
		`$ service network restart`

Note: For configuring databases refer to the Configure DSR APIGW Database and Configure GatekeeperAuditPlugin on DSR APIGW Database Server sections. For configuring the admin and application servers, refer to the Configure DSR APIGW Admin/Application Server section.

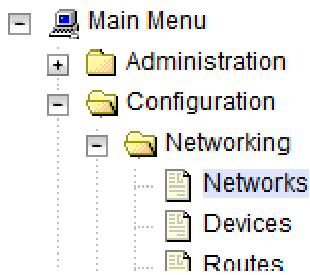
5. Configure DSR APIGW Database

5.1 Configure DSR APIGW Database Server

This procedure configures the DSR APIGW database server.

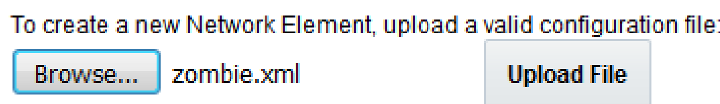
Procedure 10. Configure DSR APIGW Database Server		
1. ☐	**DB GUI**: Login	Establish a GUI session on the DB server using VM console. Login as the **guiadmin** user. If prompted by a security warming, click **Continue to this Website** to proceed.

<table>
<tr><th colspan="3">Procedure 10. Configure DSR APIGW Database Server</th></tr>
<tr>
<td>2.
☐</td>
<td>DB GUI: Create the DB network element using the XML file</td>
<td>
1. Navigate to Configuration > Networking > Networks.



2. Click Browse and type the pathname of the DB network XML file.

3. Click Upload File to upload the XML file.



4. From the examples in Appendix F, configure the DSR APIGW DB network element.
5. Once the data has been uploaded, you should see a tabs display with the name of your network element. Click on the tabs with the configured individual networks.
</td>
</tr>
</table>

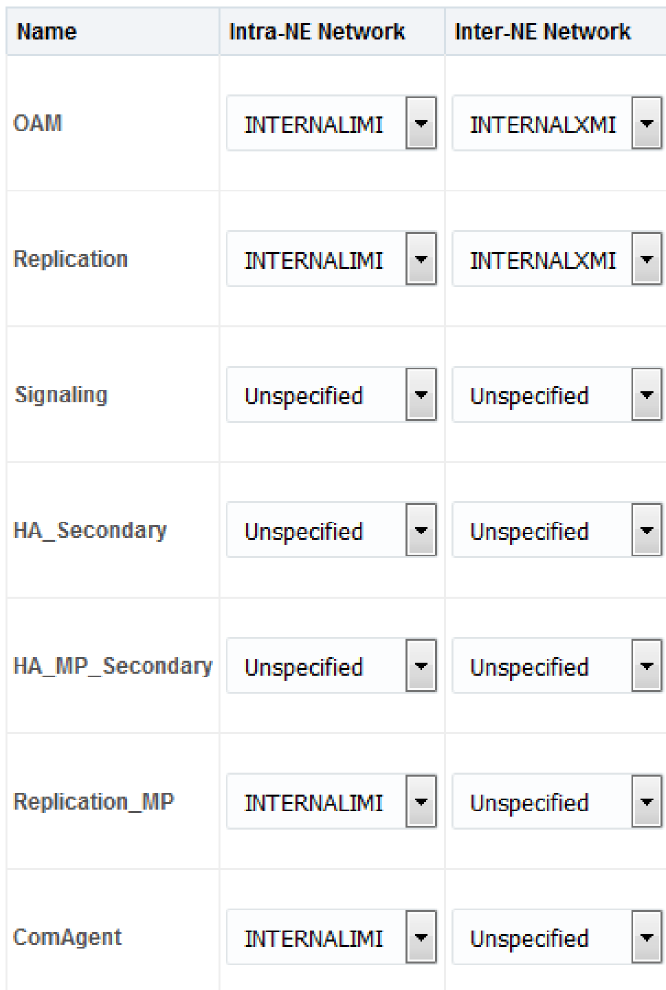
Procedure 10. Configure DSR APIGW Database Server

3. ☐ **DB GUI**: Map services to networks

1. Navigate to **Configuration > Networking > Services**.
2. Click **Edit** and set the services as shown in this table:

Name	Intra-NE Network	Inter-NE Network
OAM	<IMI Network>	<XMI Network>
Replication	<IMI Network>	<XMI Network>
Signaling	Unspecified	Unspecified
HA_Secondary	Unspecified	Unspecified
HA_MP_Secondary	Unspecified	Unspecified
Replication_MP	<IMI Network>	Unspecified
ComAgent	<IMI Network>	Unspecified

For example, if your IMI network is named IMI and your XMI network is named XMI, then your services configuration should look like this:

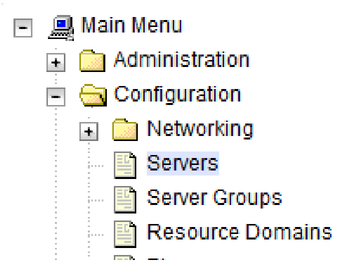


3. Click **OK** to apply the Service-to-Network selections. Dismiss any possible popup notifications.

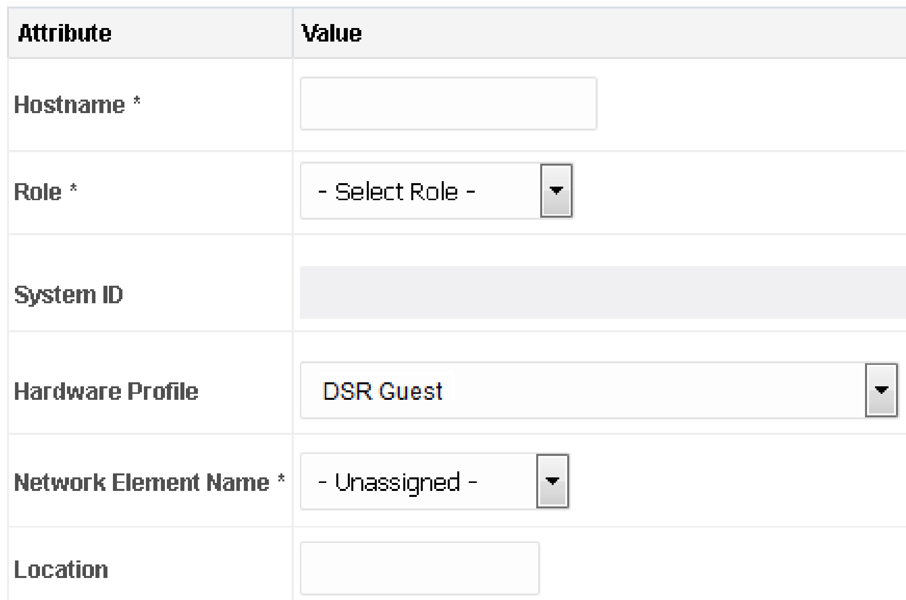
Procedure 10. Configure DSR APIGW Database Server

4. ☐ **DB GUI**: Insert the DB1 VM

1. Navigate to **Configuration > Servers**.



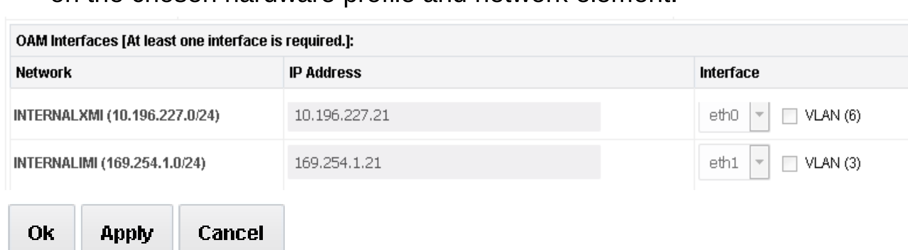
2. Click **Insert** to insert the new DB server into servers table.



3. Fill in the fields as follows:

   Hostname: <Hostname>
   Role: `NETWORK OAM&P`
   System ID: <Site System ID>
   Hardware Profile: `DSR Guest`
   Network Element Name: [Select **NE** from list]

   The network interface fields are now available with selection choices based on the chosen hardware profile and network element.



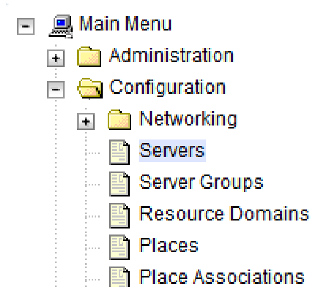
4. Type the server IP addresses for the XMI network. Select **ethX** for the interface. Leave the **VLAN** checkbox unchecked.

<table>
<tr><th colspan="3">Procedure 10. Configure DSR APIGW Database Server</th></tr>
<tr><td></td><td></td><td>5. Type the server IP addresses for the IMI network. Select ethX for the interface. Leave the VLAN checkbox unchecked.

6. Add the following NTP servers:

<table>
<tr><th>NTP Server</th><th>Preferred?</th></tr>
<tr><td>Valid NTP Server</td><td>Yes</td></tr>
<tr><td>Valid NTP Server</td><td>No</td></tr>
<tr><td>Valid NTP Server</td><td>No</td></tr>
</table>
7. Click OK when you have completed entering all the server data.

Note: Properly configure the NTP on the controller node to reference lower stratum NTP servers.</td></tr>
<tr><td>5.
☐</td><td>DB GUI: Export the initial configuration</td><td>1. Navigate to Configuration > Servers.



2. From the GUI screen, select the DB server and click Export to generate the initial configuration data for that server. Go to the Info tab to confirm the file has been created.


</td></tr>
<tr><td>6.
☐</td><td>DB Server: Copy configuration file to DB1</td><td>1. Obtain a terminal window to the DB1 server, logging in as the admusr user.

2. Copy the configuration file from the /var/TKLC/db/filemgmt directory on the DB1 to the /var/tmp directory. The configuration file has a filename like TKLCConfigData.<hostname>.sh. For example:

<code>$ sudo cp</code>
<code>/var/TKLC/db/filemgmt/TKLCConfigData.<hostname>.sh</code>
<code>/var/tmp/TKLCConfigData.sh</code></td></tr>
<tr><td>7.
☐</td><td>DB Server: Wait for configuration to complete</td><td>The automatic configuration daemon looks for the file named TKLCConfigData.sh in the /var/tmp directory, implements the configuration in the file, and asks the user to reboot the server.

If you are on the console, wait to be prompted to reboot the server, but DO NOT reboot the server, it is rebooted later in this procedure.

Verify the script completed successfully by checking the following file.

<code>$ sudo cat /var/TKLC/appw/logs/Process/install.log</code>

Note: Ignore the warning about removing the USB key since no USB key is present. No response occurs until the reboot prompt is issued.</td></tr>
</table>

Procedure 10. Configure DSR APIGW Database Server		
8. ☐	**DB1 Server**: Verify server health	1. Log into the DB1 as the **admusr** user. 2. Run syscheck and make sure no errors are returned: $ sudo syscheck Running modules in class hardware... OK Running modules in class disk... OK Running modules in class net... OK Running modules in class system... OK Running modules in class proc... OK LOG LOCATION: /var/TKLC/log/syscheck/fail_log
9. ☐	**DB1 GUI**: Login	1. If not already done, establish a GUI session on the DB1 server by using the IP address of the DB1 server. Open the web browser and type **http://<DB1_IP_Address>** as the URL. 2. Login as the **guiadmin** user.
10. ☐	**DB1 Server**: Configure DB2	Repeat the step from 4. to 8. to configure database (DB2). ***Note:*** To copy configuration file to DB2 use the following command: $ sudo scp /var/TKLC/db/filemgmt/TKLCConfigData.<hostname>.sh admusr@<ipaddr>:/var/tmp/TKLCConfigData.sh **Note**: <ipaddr> is the DB2 XMI IP address.

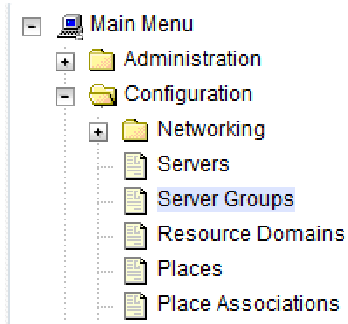
5.2 Configure DSR APIGW Database Server Group

This procedure configures the database server group.

Procedure 11. Configure DSR APIGW DB Server Group		
1. ☐	**DB1 GUI**: Login	Establish a GUI session on the DB1 server using the VM console. Login as the **guiadmin** user. If prompted by a security warming, click **Continue to this Website** to proceed.

<table>
<tr><th colspan="3">Procedure 11. Configure DSR APIGW DB Server Group</th></tr>
<tr>
<td>2.
☐</td>
<td>DB1 GUI: Add DB1 server to server group</td>
<td>

1. Navigate to **Configuration > Server Groups**.



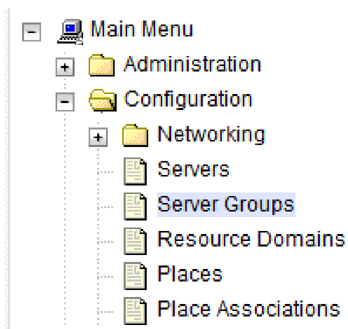
2. Click **Insert** and fill in the following fields:

Server Group Name: [Enter Server Group Name]
Level: A
Parent: None
Function: DSR (Active/Standby Pair)
WAN Replication Connection Count: Use Default Value

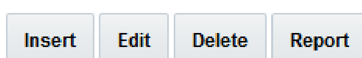
3. Click **OK** when all fields are filled in.

</td>
</tr>
<tr>
<td>3.
☐</td>
<td>DB1 GUI: Edit the DB1 Server Group</td>
<td>

1. Navigate to **Configuration > Server Groups**.



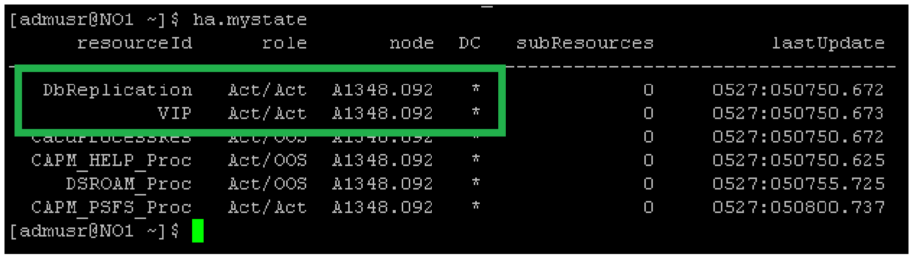
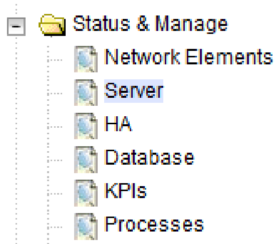
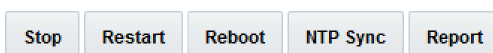
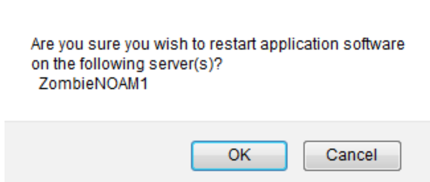
2. Select the new server group and click **Edit**.

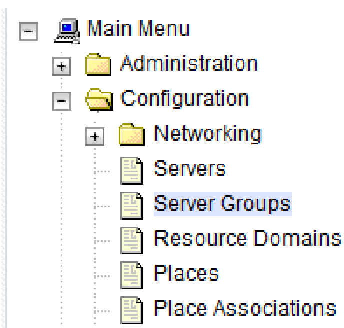


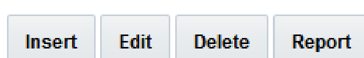
3. Select the network element that represents the DB.

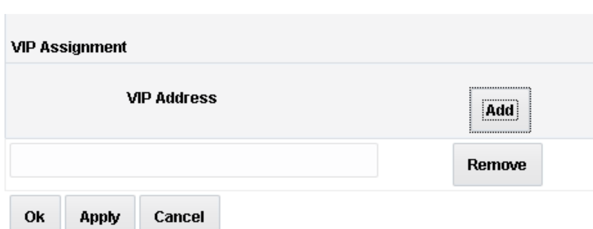
4. In the portion of the screen that lists the servers for the server group, find the DB server being configured. Mark the **Include in SG** checkbox.
5. Leave other boxes unchecked.
6. Click **OK**.

</td>
</tr>
</table>

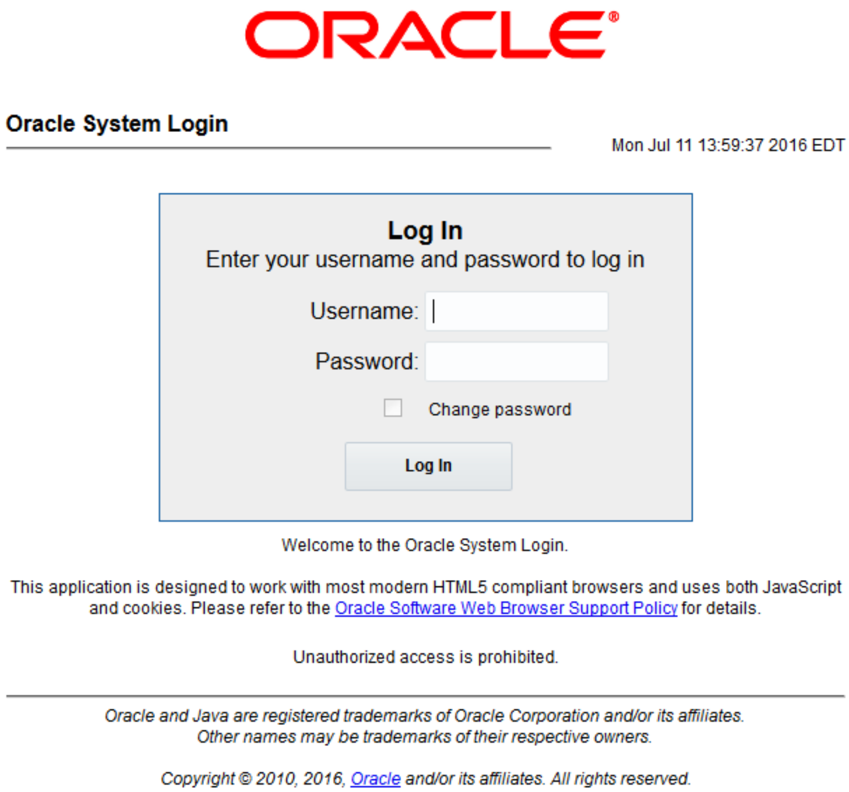
Procedure 11. Configure DSR APIGW DB Server Group		
4. ☐	**DB1 Server**: Verify DB1 VM role	1. From console window of the DB1 VM, execute the `ha.mystate` command. Verify the DbReplication and VIP items under the **resourceId** column have a value of **Active** under the role column. You may have to wait a few minutes for it to be in that state. 2. Press **Ctrl+C** to exit. Example: 
5. ☐	**DB1 GUI**: Restart DB1 VM	1. From the DB GUI, navigate to **Status & Manage > Server**.  2. Select the DB1 server. Click **Restart**.  3. Click **OK** on the confirmation screen and wait for restart to complete. 

<table>
<tr><th colspan="3">Procedure 11. Configure DSR APIGW DB Server Group</th></tr>
<tr><td>6.
☐</td><td>DB1 Server: Set sysmetric thresholds for VMs</td><td>From console window of the DB1 VM, execute the iset commands as admusr.
<code>$ sudo iset -feventNumber='-1' SysMetricThreshold where "metricId='RoutingMsgRate' and function='DIAM'"</code>
<code>$ sudo iset -feventNumber='-1' SysMetricThreshold where "metricId='RxRbarMsgRate' and function='RBAR'"</code>
<code>$ sudo iset -feventNumber='-1' SysMetricThreshold where "metricId='RxFabrMsgRate' and function='FABR'"</code>
<code>$ sudo iset -feventNumber='-1' SysMetricThreshold where "metricId='RxCpaMsgRate' and function='CPA'"</code>
<code>$ sudo iset -feventNumber='-1' SysMetricThreshold where "metricId='RxDmiwfMsgRate' and function='DM-IWF'"</code>
<code>$ sudo iset -feventNumber='-1' SysMetricThreshold where "metricId='RxMdIwfIngressMsgRate' and function='MD-IWF'"</code>
Note: These commands disable the message rate threshold alarms.</td></tr>
<tr><td>7.
☐</td><td>DB1 GUI: Add DB2 server to the Server Group</td><td>1. From the GUI session on the DB1 server, navigate to Configuration > Server Groups.


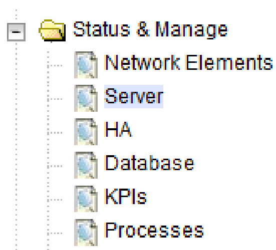
2. Select the DB server group and click Edit.


3. Add the DB2 server to the server group by marking the Include in SG checkbox for the DB2 server. Click Apply.
4. Click Add to add DB XMI and IMI VIP address. Type the VIP Address and click OK.

</td></tr>
</table>

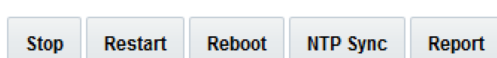
<table>
<tr><th colspan="3">Procedure 11. Configure DSR APIGW DB Server Group</th></tr>
<tr><td>8.
☐</td><td>Establish GUI session on the DB VIP</td><td>Establish a GUI session on the DB server using the DB VIP address. Login as the guiadmin user.



</td></tr>
<tr><td>9.
☐</td><td>Wait for remote database alarm to clear</td><td>Wait for the alarm ID 10200 Remote Database re-initialization in progress to clear on the Alarms & Events > View Active screen.</td></tr>
<tr><td>10.
☐</td><td>DB GUI: Restart DB2 VM</td><td>1. Navigate to Status & Manage > Server and select the DB2 server.



2. Click Restart.



3. Click OK on the confirmation screen.

Wait approximately 3-5 minutes before proceeding to allow the system to stabilize indicated by having the Appl State as Enabled.</td></tr>
</table>

5.3 Configure GatekeeperAuditPlugin on DSR APIGW Database Server

This procedure configures the GatekeeperAuditPlugin. All the specified commands are executed using **admusr** on both the database servers, that is, both active and standby servers.

Prerequisite: All the respective infrastructures has to be up and running.

Procedure 12. Configure GatekeeperAuditPlugin on DSR APIGW Database Server		
1. ☐	**Database Server**: Log into DSR APIGW	Log into the DSR APIGW using CLI.

2. ☐	**Database Server**: Verify the output show only IDB plugin	Execute this command:
		`sudo sed -ne '/^plugin-load/p' /var/TKLC/rundb/run/mysql/mysql.conf /usr/TKLC/appworks/prod/bin/Imysql.opts`
		Expected output:
		`[admusr@dsrapigwapp-0 ~]$ sudo sed -ne '/^plugin-load/p' /var/TKLC/rundb/run/mysql/mysql.conf /usr/TKLC/appworks/prod/bin/Imysql.opts`
		`plugin-load=idb=libidbstorage.so`
		`plugin-load=idb=libidbstorage.so`

3. ☐	**Database Server**: Update the IDB files to configure GatekeeperAduitPlugin	Execute this command and verify the output shows a 0 value:
		`sudo sed -i '/^plugin-load/ s/$/;GatekeeperAuditPlugin=libGatekeeperAuditPlugin.so;/' /var/TKLC/rundb/run/mysql/mysql.conf /usr/TKLC/appworks/prod/bin/Imysql.opts && echo $?`
		Expected output:
		`[admusr@dsrapigwapp-0 ~]$ sudo sed -i '/^plugin-load/ s/$/;GatekeeperAuditPlugin=libGatekeeperAuditPlugin.so;/' /var/TKLC/rundb/run/mysql/mysql.conf /usr/TKLC/appworks/prod/bin/Imysql.opts && echo $?`
		`0`

4. ☐	**Database Server**: Verify that both IDB and GatekeeperAuditPlugin are configured	Execute this command and verify the output shows both plugins:
		`sudo sed -ne '/^plugin-load/p' /var/TKLC/rundb/run/mysql/mysql.conf /usr/TKLC/appworks/prod/bin/Imysql.opts`
		Expected output:
		`admusr@dsrapigwapp-0 ~]$ sudo sed -ne '/^plugin-load/p' /var/TKLC/rundb/run/mysql/mysql.conf /usr/TKLC/appworks/prod/bin/Imysql.opts`
		`plugin-load=idb=libidbstorage.so;GatekeeperAuditPlugin=libGatekeeperAuditPlugin.so;`
		`plugin-load=idb=libidbstorage.so;GatekeeperAuditPlugin=libGatekeeperAuditPlugin.so;`

Procedure 12. Configure GatekeeperAuditPlugin on DSR APIGW Database Server		

5. ☐	**Database Server**: Enable MySql port	Navigate to **/usr/TKLC/appworks/etc/** and update the **apwSoapServer.cfg** configuration file for the **mysql_where** parameter:
		Existing:
		`mysql_where = QS`
		Expected :
		`mysql_where = OAM`

6. ☐	**Database Server**: Restart Imysqld	Kill the Imysqld process so that once the Imysqld process is started, the plugin is enabled. `sudo pm.kill Imysqld`
7. ☐	**Database Server**: Restart server	Restart the server. `sudo init 6`

6. Configure DSR APIGW Admin/Application Server

This procedure installs and configures DSR APIGW admin and application servers.

Prerequisite: All the respective infrastructures has to be up and running.

Procedure 13. Configure DSR APIGW Admin/Application Server		
1. ☐	**Openstack Controller**: Copy the .pem file (key-pair) used to create the VMs to Admin server in any location.	1. Log into Openstack controller console. 2. Copy the PEM file from the Opentack controller to the Admin server in any location. `$ scp -i /root/dsr-keypair.pem  /root/ dsr-keypair.pem admusr@<aminserverip>:/u02` ***Note:*** PEM certificates are frequently used for web servers since they can be easily translated into readable data using a simple text editor. Generally, when a PEM encoded file is opened in a text editor, it contains very distinct headers and footers. Refer to Appendix C for creating a PEM file.
2. ☐	**Admin Server**: Log into the Admin server and fill in the ocsg.properties file with all required input data for the script	1. Log into **Admin** server. 2. Navigate to /u02/app/oracle/scripts/. `$ cd /u02/app/oracle/scripts/` 3. Edit the file **ocsg.propertie**.with respective property values in the file. Refer to Appendix B for more information on properties and its parameters.
3. ☐	**Admin Server**: Execute the script	1. Log into the **Admin** server. 2. Navigate to **/u02/app/oracle/scripts**. 3. Execute python `configureOCSGSingleTier.py`.

<table>
<tr><th colspan="3">Procedure 13. Configure DSR APIGW Admin/Application Server</th></tr>
<tr><td>4.
☐</td><td>Admin Server: Monitor the screen and verify the log file for success</td><td>1. Log into Admin server.
2. Navigate to /u02/app/oracle/scripts.
3. Execute <code>vim ocsg_install.log</code>.
Note: The log file name is configured in the ocsg.properties file.
Installation takes few mins and once the installation is complete, the DSR APIGW Configuration Successful message displays.
<pre>######################### DSR API Gateway Configuration Succesful! ########################

Installation folder : /u03/app/oracle/ocsg-18.2.5/
Admin GUI Interface can be accessed at   http://10.75.242.246:7001/console/
Partner GUI Interface can be accessed at http://10.75.242.247:8001/portal/partner-manager/index/

######################### DSR API Gateway Configuration Succesful! #########################

[admusr@dsrapigwapp-0 scripts]$</pre></td></tr>
<tr><td>5.
☐</td><td>Admin server: Verify the interface accessibility</td><td>Verify the interface accessibility by opening the GUI Interface IP in a browser window. Refer to Network Model for the port information.</td></tr>
</table>

Appendix A. Example Parameter file

A.1. Guidelines to create parameter file

Basic guidelines to follow while working with YAML files:

- The file must be ended with .yaml extension.
- YAML must be case-sensitive and indentation-sensitive.
- YAML does not support the use of tabs. Instead of tabs, it uses spaces.

YAML is a human-friendly data serialization standard for all programming languages.

The values of the **key:value** can be broadly classified into the following types:

Type	Description	Examples
string	A literal string.	“String param”
number	An integer or float.	“2”; “0.2”
comma_delimited_list	An array of literal strings that are separated by commas. The total number of strings should be one more than the total number of commas.	[“one”, “two”]; “one, two”; ***Note:*** “one, two” returns [“one”, “two”]
json	A JSON-formatted map or list.	{“key”: “value”}
boolean	Boolean type value, which can be equal “t”, “true”, “on”, “y”, “yes”, or “1” for true value and “f”, “false”, “off”, “n”, “no”, or “0” for false value.	“on”; “n”

A.2. Parameter File for DSR APIGW Database

The parameter file defines the topology details. This includes all VM details such as the number of VMs, flavors, network names, etc. It is a list of key/value pairs. By referring to the **parameters** definition section in the template file, the initialization of the parameters has to be done in this section.

File Naming Convention

It is not mandatory to have a specific name for the file, but to provide a self-explanatory name for the file, it is recommended to follow this convention:

<DSR Name>_<Site Name>_<NetworkOam >_Params.yaml

For example:

dsrCloudInit_Site00_NetworkOam_Params.yaml

Sample File

Network OAM params file

```
parameters:
    numPrimaryNoams: 1
    numNoams: 1
    noamImage: DSR-60147
    noamFlavor: dsr.noam
    primaryNoamVmNames: ["DsrSite00NOAM00"]
    noamVmNames: ["DsrSite00NOAM01"]
    noamAZ: nova
    xmiPublicNetwork: ext-net
    imiPrivateNetwork: imi
    imiPrivateSubnet: imi-sub
    imiPrivateSubnetCidr: 192.168.321.0/24
    ntpServer: 10.250.32.10
    noamSG: Site00_NOAM_SG
```

Network OAM params file (Fixed IP)

```
parameters:
    numPrimaryNoams: 1
    numNoams: 1
    noamImage: DSR-8.3.0.0.0_83.x.0.vmdk
    noamFlavor: dsr.noam
    primaryNoamVmNames: ["DsrSite00NOAM00"]
    noamVmNames: ["DsrSite00NOAM01"]
    noamAZ: nova
    primaryNoamXmiIps: ["10.196.12.83"]
    noamXmiIps: ["10.196.12.84"]
    noamVip: 10.196.12.85
    xmiPublicNetwork: ext-net3
    imiPrivateNetwork: imi
    imiPrivateSubnet: imi-sub
    imiPrivateSubnetCidr: 192.168.321.0/24
    ntpServer: 10.75.185.194
    noamSG: Site00_NOAM_SG
```

A.3. Parameter file for DSR APIGW Admin/Application

The HEAT template files are:

- dsrapigw.yml
- dsrapigw_env.yml

Table 5 lists the parameters used to configure DSR APIGW Admin/Application stack.

Table 5: Example Parameter File

Parameter Category	Parameter Name	Type	Description
Common parameters	key_name	String	Name of key-pair to be used for compute instance
	image_id	String	Oracle Linux image to be used for compute instance
Number of VMs	num_app	Number	Number of AT servers to be confirgured as per the requirement
VM flavors	flavor_admin	String	Admin server VM profile
	flavor_app	String	AT server VM profile
IP Network	networks_admin	Json	List of networks (one or more) on admin server
	networks_app	Json	List of networks (one or more) on application server
hostname	hostname_admin	String	Hostname of the admin server
	user_name	String	User name of the admin server
	password	String	Password for the admin server

Appendix B. OCSG Properties file

Following table lists the user data to be filled in OCSG properties file.

Table 6: OCSG Properties File

Section	Parameter Name	Description
Admin	servers	IMI address of Admin Server. servers = ["AdminServer: xxx.xxx.xxx.xxx "] ***Note:*** It is mandatory to follow the name of Admin server as **AdminServer**. This is the DSRAPIGW DB server address where data is backed up. DR procedure uses this data.
Admin	xmiInterface	XMI Interface address of Admin Server. `xmiInterface = ["AdminServer: xxx.xxx.xxx.xxx "]`
Admin	backupServer	Provide the IMI VIP of DSR API GW database. Admin server should have access to this server using the key/pem file. This is the location in the DSRAPIGW DB server where the data should be backed up. For example: `backupServer = xxx.xxx.xxx.xxx`

Section	Parameter Name	Description
Admin	backupDomain	Full path including the DSR API GW domain folder name to where the DSR API GW files need to be backed up on backup server. For example: `backupDomain = /var/TKLC/db/filemgmt/backup/services-gatekeeper-domain`
App	servers	Add App server name and IP. Add comma separated entries for multiple servers. For example: `servers = ["AppServer1:xxx.xxx.xxx.xxx", "AppServer2:xxx.xxx.xxx.xxx"]` ***Note:*** It is mandatory to follow the name of App servers as 'AppServer1', 'AppServer2' etc.
App	xmiInterfaces	XMI address for all AppServers in [“Ip1”,”Ip2”…] format. For example: `xmiInterfaces = ["AppServer1: xxx.xxx.xxx.xxx ", "AppServer2: xxx.xxx.xxx.xxx "]`

App	xsiInterfaces	XSI address for all AppServers in [“Ip1”,”Ip2”…] format. For example:
		`xsiInterfaces = ["AppServer1: xxx.xxx.xxx.xxx ", "AppServer2: xxx.xxx.xxx.xxx "]`
		To add multiple XSIs to each AppServer the format should be:
		`[“AppServer1:XSI1-IP”,”AppServer2:XSI2”,”AppServer2:XSI1-IP”,”AppServer2:XSI2”]`

App	exteralLoadbalancerIP	IP used to publish T8 APIs. This IP is used when displaying T8 API access URLs in the Partner and API management Portal. `exteralLoadbalancerIP = xxx.xxx.xxx.xxx`
Servers	cleanUpBeforeInstall	If the script failed to execute while running, the server will be in bad shape for a fresh install. Keeping cleanUpBeforeInstall as **yes** cleans up the server and makes it ready for script re-run.
Servers	ntp	Provide NTP server IP: `ntp = xxx.xxx.xxx.xxx`
Servers	mtu	Maximum transmission unit. The script copies multiple files from the Admin server to App server. Before copying the MTU has to be set. Recommended value is **9000**. `mtu = 9000`
Servers	apiroot	This variable is part of the API creation. <apiroot> is prefixed to the context URI of the APIs exposed. For example, the API name of Device triggering is **apiroot-dt**.
Servers	dsrMpList	Provide DSR MP XSI Ip list in format: `MP1-XSI-IP:port,MP2-XSI1-IP:port`
Files	pemfile	Provide the .pem file location: `pemfile = /u02/software/ocsg-db-key.pem`

Section	Parameter Name	Description
Files	logfile	Custom log file for Installation. Change log file name if required. `logfile = ocsg_install.log`
Files	presentFolder	The scripts is in this location. This property should not be changed. `presentFolder = /u02`
Files	targetFolder	The scripts are copied to this location. This property should not be changed. `targetFolder = /u03`
Files	targetPath	Provide the location of the scripts. This property should not be changed. `targetPath = /app/oracle/`
Files	scripts	Provide the folder name where scripts need to be stored. This property should not be changed. `scripts = scripts`
Files	extendWizard	Custom scripts are present here. This property should not be changed. `extendWizard = extend_wizard/`
Files	SCEFPackage_EAR	Default EAR file name. This property should not be changed. `SCEFPackage_EAR = SCEFHandlers.ear`
Files	nodemgr	Node manager service file name. This property should not be changed. `nodemgr = nodemgr`
Files	DefaultJar	Location of ocsg_generic_jar. This property should not be changed. `defaultJar = /usr/TKLC/dsrapigw/ocsg_generic_jar`
Files	volumeName	Provide the Volume name. This property should not be changed. `volumeName = ocsgv`
Files	volumeSize	Volume size in GB. Script creates a new volume of this size. This field should not be changed. `volumeSize = 10`
Files	inventoryLoc	Inventory log location of OCSG. This property should not be changed. `inventoryLoc = /u02/inventory`

Credentials	mysqlJdbcServerUrl	MySQL DB credentials. Provide IMI VIP of the DSR API GW database setup;
		`jdbc:mysql://<db-server-ip>:15616/gatekeeper`
		For Example,
		`mysqlJdbcServerUrl = jdbc:mysql://30.30.30.17:15616/gatekeeper`

Section	Parameter Name	Description
Credentials	mysqlUserName	This property should not be changed: `mysqlUserName          = awadmin` ***Note:*** MySQL password is the default Comcol password. It is in the dsrapigw_default_params.rsp file.

Credentials	weblogicUser	Provide the DSR API GW Admin portal credentials:
		`weblogicUser           = weblogic`
		`weblogicPassword       = tekelec123`

Credentials	weblogicPassword	

Credentials	nodeManagerUser	Provide the Node Manager credentials to use in all Admin and AppServers:
		`nodeManagerUser        = nodemanager`
		`nodeManagerPassword = tekelec123`

Credentials	nodeManagerPassword	

Credentials	operatorUser	A new operator is created with this detail to access partner relationship management portal:
		`operatorUser           = oracleop3`
		`operatorPassword       = tekelec123`

Credentials	operatorPassword	

Credentials	adminServerUser	The ssh user name in Admin and AppServers.
		`adminServerUser        = admusr`
		`appServerUser          = admusr`

Credentials	appServerUser	

Ports	adminListenPort
	appListenPort
	appListenPortSSL
		These are the default ports opened on IMI network should not be changed. These ports are used only for internal communication.
		`adminListenPort        = 7001`
		`appListenPort          = 8001`
		`appListenPortSSL       = 8002`

Ports	adminIMIPorts
	adminXMIPorts
		Ports to be enabled in IP Firewall on Admin server:
		`adminIMIPorts = 7001,5556,7002,9876,8050,3075,9090,7`
		`adminXMIPorts          = 9002`

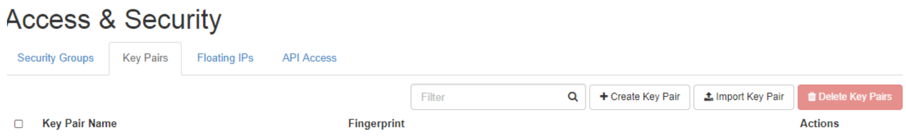
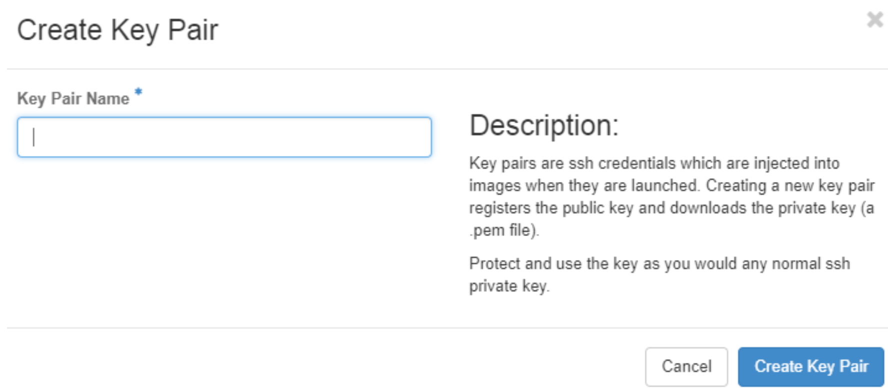
Ports	appIMIPorts
	appXMIPorts
	appXSIPorts
		Ports to be enabled in IP Firewall on AppServers:
		`appIMIPorts            = 8001,8002,9876,5556,8050,3075,9090,7`
		`appXMIPorts            = 9002`
		`appXSIPorts            = 10001,10002`

Appendix C. Create PEM file for Openstack

This procedure creates a PEM file for Openstack.

Prerequisite: All the respective infrastructures has to be up and running.

Procedure 14. Create PEM File for Openstack		
1. ☐	Log into Openstack	Log into the Openstack.

Procedure 14. Create PEM File for Openstack		
2. ☐	Go to Create Key Pair option	1. Navigate to **Project > Compute > Access and Security > Key Pairs**. 2. Click **Create Key Pair**. 
3. ☐	Create Key Pair	Enter the required **Key Pair Name** and click **Create Key Pair**. 

Appendix D. Resource Profile for DSR APIGW Database and Admin/Application

Following table provides list of resource provide for DSR APIGW Database, Admin and Application servers.

Table 7: Resource Profile for DSR APIGW

DSR APIGW	vCPU	RAM (GB)	Disk (GB)	Network Interfaces
Database Server	4	6	70	2
Admin Server	4	6	70	2
Application Server	12	16	70	3*

Note: Multiple XSI Network interfaces are supported for App servers. Maximum 16 network XSI interfaces are supported.

Appendix E. Convert vmdk to qcom2 Format

This procedure describes how to convert vmdk to qcom2 format.

Prerequisite: All the respective infrastructures has to be up and running.

Procedure 15. Convert vmdk to qcom2 Format		
1. ☐	Log into Qemu-img tool	Log into the Qemu-img tool.

Procedure 15. Convert vmdk to qcom2 Format		

2. ☐	Convert the file format	1. Convert vmdk to qcow2 format
		2. Use the qemu-img tool to create a qcow2 image file using this command:
		`qemu-img convert -f vmdk -O qcow2 <VMDK filename> <QCOW2 filename>`
		Example:
		`qemu-img convert -f vmdk -O qcow2 DSR-82_12_0.vmdk DSR-82_12_0.qcow2`
		Note: Install the qemu-img tool (if not already installed) using this yum command:
		`sudo yum install qemu-img`
		3. Import the converted qcow2 image using the “glance” utility from the command line.
		`$ glance image-create --name dsr-x.x.x-original --is-public True --is-protected False --progress --container-format bare --disk-format qcow2 --file DSR-x.x.x-disk1.qcow2`

Appendix F. Sample Network Element and Hardware Profiles

To enter all the network information for a network element into an AppWorks-based system, a specially formatted XML file needs to be filled out with the required network information. The network information is needed to configure both the NOAM and any SOAM network elements.

It is expected that the maintainer/creator of this file has networking knowledge of this product and the customer site at which it is being installed. The following is an example of a network element XML file.

The SOAM network element XML file needs to have same network names for the networks as the NOAM network element XML file has. It is easy to accidentally create different network names for NOAM and SOAM network elements, and then the mapping of services to networks are not possible.

```
<?xml version="1.0"?>
<networkelement>
    <name>NE</name>
    <networks>
        <network>
            <name>XMI</name>
            <vlanId>3</vlanId>
            <ip>10.2.0.0</ip>
            <mask>255.255.255.0</mask>
            <gateway>10.2.0.1</gateway>
            <isDefault>true</isDefault>
        </network>
        <network>
            <name>IMI</name>
            <vlanId>4</vlanId>
```

```
            <ip>10.3.0.0</ip>
            <mask>255.255.255.0</mask>
            <nonRoutable>true</nonRoutable>
        </network>
    </networks>
</networkelement>
```

Note: The Network Element Name should be unique when creating multiple Network Elements.

Appendix G. My Oracle Support (MOS)

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

Call the CAS 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 MOS, select 2.

You are connected to a live agent who can assist you with MOS registration and opening a support ticket. MOS is available 24 hours a day, 7 days a week, 365 days a year.

Emergency Response

In the event of a critical service situation, emergency response is offered by the 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 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 the system's 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.

Locate Product Documentation on the Oracle Help Center

Oracle Communications customer documentation is available on the web at the Oracle Help Center (OHC) site, http://docs.oracle.com. You do not have to register to access these documents. Viewing these files requires Adobe Acrobat Reader, which can be downloaded at http://www.adobe.com.

1. Access the Oracle Help Center site at http://docs.oracle.com.
2. Click Industries.
3. Under the **Oracle Communications** subheading, click the **Oracle Communications documentation** link. The Communications Documentation page appears. Most products covered by these documentation sets display under the headings N**etwork Session Delivery and Control Infrastructure** or **Platforms**.

   Click on your Product and then the Release Number. A list of the entire documentation set for the selected product and release displays. To download a file to your location, right-click the PDF link, select `Save target as` (or similar command based on your browser), and save to a local folder.