MURAL Software Standard Installation Guide for Rack Servers

Version 3.9

Copyright

Americas Headquarters

Cisco Systems, Inc.

170 West Tasman Drive

San Jose, CA 95134-1706 USA

http://www.cisco.com

Tel: 408 526-4000

         800 553-NETS (6387)

Fax: 408 527-0883

Table of Contents

Copyright

1. Installation Overview

This document describes how to install the Mobility Unified Reporting and Analytics (MURAL) application on rack servers. MURAL provides Web-based reporting and analytics for deep packet inspection (DPI) data emerging from your network. In this setup, VMware based Virtual Machines are being used to install multiple MURAL nodes into single rack server. There are two rack servers will be used to distribute the application load and achieve the high availability (HA) of mandatory components that ensures the availability of application in case of hardware failure from one rack server.

1.1 Before You Begin

This document assumes that you have a working knowledge of the following technologies:

- Linux operating system

- Cisco UCS Rack Servers like C240/C460

- VMWare

- Rack servers

Before you begin the installation, we recommend that you:

- Provision the bill of material (BOM) and Customer Inventory Questionnaire (CIQ) sheet for rack servers.

- Ensure rack servers are up and running with all necessary infrastructure support (Like hardware installation of servers in rack, power supply connections and network connectivity for DRAC access as well as NIC ports of the servers).

- One windows OS based laptop with following software installed into it:

  - VMware vSphere client version 5.5.
  - JRE 1.7 with latest updates.
  - putty/ssh terminal, scp/pscp and wget tools.

- Provision to connect this laptop/PC with rack servers over DRAC/Mgmt IPs to install MURAL system.

1.2 Installation Package

The MURAL software installation package contains the following components:

1. MURAL release media, including:

   a. OVF template file - `vmTemplate.zip`

   b. VMDK images for MURAL release software - `rootfs.vmdk` and `datafs.vmdk`

      For the exact image name and the MD5 checksum for the software image, refer to the release notes for your release.

   c. Any software patches that is available for the release. Refer to the *MURAL Release Notes, 3.6* for a complete list of patches.

   d. Scripts to share ssh keys, fetch MAC IDs and start GMS server - `expect_shell.sh`, `fetchMacIDs.sh`, `shareSSHkey.sh`, and `startGMS.sh`.

2. The CIQ Excel spreadsheet, which will be used to create hardware and the final configuration file (XML). This CIQ sheet provides network, cluster, node, and miscellaneous information that will be used to create the configuration file (XML) for installing and configuring the MURAL system.

3. Management information bases (MIBs)

1.3 Customer Information Questionnaire

The CIQ is an Excel spreadsheet of configuration settings based on a site survey that was completed before the installation process. Its worksheets include the indicated kind of information:

- **MiscInfo** — Specifies configuration file, site, hardware type, login message and NTP settings.

- **NetworkInfo** — Specifies networks, default gateway, DNS, NTP and SNMP

servers.

- **ClusterInfo** — Specifies cluster names and types as well as interfaces and VIPs (if any).

- **NodeInfo** — Specifies chassis and slot IDs, hostnames, MAC IDs and IP addresses, KVM IPs of blades and cluster names for all the blades to group into specific roles.

- **StorageInfo** — Specifies node hostnames, WWIDs and mount-points to assign LUNs from SAN storage devices.

Apart from the CIQ sheet, the following information from the site is also required during installation:

- **Connectivity**—Specifies the details for ports and connections

- **Firewall**—Identifies the firewall changes required for connectivity

- **ASR**—Specifies locations for various ASR information bases (IBs) that are required by the application

For more information about the CIQ sheet, see "Understanding the CIQ Sheet" on page 84.

1.4 System Components

The following figure shows the components of the MURAL platform, focusing on how the data flows through the system:

1.4.1 MURAL System Setup

The MURAL system installation for rack servers is as follows:

Note: Throughout this document, master nodes and standby nodes shown above are referred as master GCR node and Standby GCR node, respectively. All references to GCR, Collector and Rubix nodes are referred to as the GCR node.

The MURAL platform (Starter Pack Setup) consists of the following nodes, each hosted on blades in the UCS Chassis:

- **GCR node**—In Starter Pack Setup, GCR node hosts GMS, Collector and Rubix components all together in one server. Starter Pack setup can be used where only a few gateways are connected to MURAL system. The GCR node cluster has two servers with 1+1 redundancy.

  ○ **General Management Server (GMS) node**—Provides centralized management of the other MURAL nodes, such as remote manufacturing of blades (installing the MURAL software), patch management, monitoring of all nodes and operations, and importing and running node configurations. The GCR node cluster supports high availability.

- **Collector node**—Collects data streams pushed to the Reflex platform, interprets the exported flows, enriches them with static data, and assembles data sets. The Collector stores the raw data in the Hadoop file system (HDFS) and sends it to the Compute node. The Collector node cluster can have any number of servers, in pairs for master and standby and uses 1+1 redundancy (transparent failover between pairs of active-active nodes).

- **UI/Caching (Rubix) node**—Hosts the Rubix engine and Rubix data cache. The Rubix engine queries the Insta nodes constantly and when new data is available, it fetches it to store in the data cache, so that it can respond more quickly to requests from the UI engine. The UI/Caching node is sometimes called the Rubix node. The UI/Caching node uses N+1 redundancy in active-active mode.

- **Compute node**—Analyzes and aggregates the data, creating *data cubes*. The term data cube is a convenient shorthand for the data structure, which is actually multi-dimensional and not limited to the three dimensions of a cube. The Compute node cluster can have any number of servers, depending on the deployment, and uses N+1 redundancy.

- **Insta node**—Stores and manages the processed data in a columnar database. Manages the Insta database, which stores processed data cubes. The Insta node cluster has one node but another node can be prepared over VM in another rack server to keep the backup of databases and used as Insta fail-over node in case rack server hosting Insta VM becomes unavailable.

1.5 Hardware Topology

MURAL system can be installed using UCS two or more rack servers depending upon site-specific requirement like ASR throughput, add-on applications support, etc. In rack server based MURAL system installations, Virtual Machines (VMs) created upon Host Machine (Rack Server, here) used as MURAL nodes to host different components like GMS, Collector, UI, Compute and Insta nodes under Standard, Medium or Starter Pack setups. In this guide, two UCS rack Servers are being installed with VMware ESXi operating system (OS) image to create Virtual

Machines (VMs). The MURAL application is hosted on these VMs that consists GCR, Compute and Insta nodes, under Starter Pack configuration.

Below diagram shows an example of MURAL system topology on two rack servers.

* VM2-INSTA-Backup node is optional here

The data flows that feed the MURAL system are pushed by an ASR 5000 or ASR 5500 platform (hereafter referred to simply as an ASR).

1.6 Installing Rack Servers

The MURAL installation process includes the following steps:

Warning: Skipping a task or performing the tasks out of sequence may cause a misconfiguration that results in system failure.

1. Install VMware into rack servers. For more information, refer to the *MURAL VMware Installation Guide for Rack Servers*.

2. Create and configure the VM. For more information, see "Configuring and

Managing the VM" on page 1.

- Create VMs using OVF template, attach images/storage and update CIQ sheet for Block/SCSI IDs.

- Set auto start/stop of VMs with Host machine.

- Power on VMs and assign IP addresses.

- Share SSH key among all the VMs.

- Collect MAC IDs for all the VMs and update CIQ sheet.

3. Install MURAL system from Master GCR node. For more information, see "Configuring the Master GMS Node" on page 1.

- On Master GCR node, apply patches and configure GMS Server.

- Create hardware and the final XML.

- Apply patches to all other nodes and activate XML.

4. Install Appliance into the MURAL nodes. See "Applying Configurations on the New Nodes" on page 1.

5. Configure environmental parameters specific to your environment.

- Generate and Push IBs in "Applying Configurations on the New Nodes" on page 1

- "Processing the Data " on page 1

6. Verify that the system is working as expected. See "Validating Data on Nodes" on page 1 and "Setting Up Offline Reports" on page 1.

2. Configuring and Managing the VM

2.1 Prerequisite

Ensure that VMware system is installed on rack servers and Host Management is configured. For more information, refer to the *MURAL VMware Installation Guide for Rack Servers*.

2.2 Creating VMs Using OVF Template

1. Extract the `vmTemplate.zip` file on your machine.

2. Launch vSphere client and log into VMware host.

3. From VMware management interface, Select **File** > **Deploy OVF Template...**

4. Select the OVF template from your computer and click **Next**.

5. Verify OVF Template details and click **Next**.

---

6. Specify a name for the deployed template and click **Next**.

7. Select the radio button for the **Thick Provision Lazy Zeroed** option and click **Next**.

8. Ensure both VM network interfaces are shown and then click **Next**.

9. Click **Finish** to start VM creation.

10. Click **Close** to complete the VM creation.

11. From left pane, select newly created VM by expanding Host tree. From right pane, select **Summary** tab to see the VM configuration info.

Repeat procedures in this section (from step 1) to create all other VMs with different names.

2.3 Copying VMDK Files to VMware Host Machine

1. Log into VMware host machine using SSH and run the following commands.

```
# cd /vmfs/volumes/datastore1/
# mkdir vmdkFiles
```

2. From your computer, run the `scp/pscp` command to copy VMDK files into `/vmfs/volumes/datastore1/vmdkFiles` directory of VMware host.

```
$ ls *.vmdk
datafs.vmdk    rootfs.vmdk
$ scp *.vmdk root@<VMware Host IP>:/vmfs/volumes/datastore1
/vmdkFiles/
```

3. Now from SSH session on Host machine, confirm the size of VMDK files

properly copied into destination directory.

```
# cd /vmfs/volumes/datastore1/vmdkFiles/
# pwd
/vmfs/volumes/datastore1/vmdkFiles
# gunzip datafs.vmdk.gz
# gunzip rootfs.vmdk.gz
```

4. Now copy both the image files under the VM storage directory, please provide the correct VM directory path here.

   Note: VMDK file copy may take 5-10 minutes per VM.

```
# ls -d ../VM*
../VM1-GCR-1  ../VM1-GCR-2  ../VM2-INSTA-1  ../VM3-DN-1
../VM4-DN-2
# cp *.vmdk ../VM1-GCR-1/
# cp *.vmdk ../VM2-INSTA-1/
# cp *.vmdk ../VM3-DN-1/
# cp *.vmdk ../VM4-DN-2/
```

After successful file copy, proceed to the next section.

2.4 Attaching VMDK Images to VM

1. Select the VM, right click and select **Edit Settings**. It launches VM properties page. Click **Add**.

2. Select **Hard Disk** and click **Next**.

3. Select **Use an existing virtual disk** and click **Next**.

4. On the **Selecting Existing Disk** page, click **Browse** to find and select the VMDK image file. Find the file by navigating to **Datastores** > **datastore1** > **VM Dir**.

   Select **rootfs.vmdk**and press **Open**.

   When you return to the Add Hardware wizard, click **Next**.

5. Press **Next** on **Advanced Options** page.

6. Press **Finish** on **Ready to Complete** page.

7. VM properties appear as shown below after adding the hard disk.

8. Repeat the above procedure, adding a different VMDK image file, **datafs.vmdk**, to this VM.

2.5 Adding Storage Disks to VM from VMWare Datastore

We need to add different number and size of LUNs to respective VMs based upon roles assigned to them for MURAL system, as shown in the following table (per node).

VM Role	LUN Size (GB)	SCSI ID	Block ID	Mount Point
GCR	1024	0:0	/dev/sda	/data/collector
	300	0:1	/dev/sdb	/data/pgsql
	500	0:2	/dev/sdc	/data/diskstore
	200	0:3	/dev/sdd	/data/rge
INSTA-DB	200	0:0	/dev/sda	/data/pgsql
	1024	0:1		dbroot1
	1024	0:2		dbroot2
DN	1024	0:0	/dev/sda	/data/hadoop-admin

Note: Size of the LUNs can be different as per site-specific configuration, please refer to the sizing sheet.

1. From **Virtual Machine Properties** page, press **Add**.

2. Select **Device type Hard Disk** and press **Next**.

3. Select **Create a new virtual disk** and press **Next**.

4. Apply **Capacity** and **Disk Provisioning** settings.

   a. Under **Capacity** section, select disk size.

   b. Under **Disk Provisioning** section, select **Thick Provision Laze Zeroed**.

   c. Under **Location** section, select **Specify a datastore or datastore cluster**. Press **Browse** to select the **datastore1**, as shown below.

Press **OK**, datastore will appear as shown below.

5. Press **Next**.

6. Press **Next** on the **Advanced Options** page, without changing any configurations.

7. On **Ready to complete** page, select **Finish** to complete the disk creation and go back to the **VM Properties** page.

8. Add all other required LUNs, repeating the same procedure. **VM properties** page will appear as shown below, after adding all the required LUNs.

9. Press **OK** on **VM Properties** page to finish the disk creation after all the LUNs added successfully.

10. Select **VM** > **Edit Settings** and ensure that all required disks have been attached, as shown in below example.

11. Repeat the above steps to add required LUNs for rest of the VMs. Note

Copyright

down the SCCI IDs for all the VMs and update the CIQ sheet that will be later used to update the GMS configuration for storage LUNs attach with each MURAL node.

2.6 Setting VM Start/Stop with Host

Following configuration will enable VMs to automatically start and stop with Host machine, in case of power cycle happens for the rack servers.

1. Navigate to **Host IP** > **Configuration** tab > **Software** section and select the **Virtual Machine Startup/Shutdown** link.

2. Click on **Properties** link.

3. Check box for **Allow virtual machines to start and stop automatically with the system**.

    a. Set **Startup Delay Time** – **120 seconds**

    b. Set **Shutdown Delay Time** – **60 seconds**

4. Select the **VM** and press **Move Up** and **Move Down** buttons as neededn to bring all the VMs under **Automatic Startup Order**, as shown below.

5. Press **OK** to save the changes.

6. Re-launch the vSphere client and check the Host configuration again. Navigate to **Configuration** tab > **Software** section and selecting **Virtual Machine Startup/Shutdown** link.

7. VMs should appear with **Startup** enabled as shown below.

Now VMs are ready to be power on for installation of MURAL system.

2.7 Assigning IP Addresses to VMs

Each VMs will be assigned with two IP addressed, one each from Internal and External Network.

- **Internal Network**—Private IP range used only for communication between rack servers thru eth0 ports.
- **External Network**—LAN IP range used to access the VMs and MURAL system thru eth1 ports.

Refer CIQ sheet to assign the IP Addresses, subnet and default gateway to all the VMs.

1. From vSphere client, select each VM and click **Power ON**.

2. Ensure that all VMs are in the **Powered ON** state before clicking **Console** to launch the VM console.

3. After VM is up, a login prompt will appear on the VM console, click the mouse pointer in the console area and press **Enter**.

4. Now login as admin user, assign IP addresses for both the interfaces (internal as well as external IPs) and default gateway, as shown below.

```
guavus-52970c login: admin
guavus-52970c > en
guavus-52970c # conf t
guavus-52970c (config) # interface eth0 ip address
10.10.10.110 /22
guavus-52970c (config) # interface eth1 ip address
192.168.113.130 /22
guavus-52970c (config) # ip default-gateway 192.168.112.1
guavus-52970c (config) # write memory
guavus-52970c (config) # quit
```

5. After successful IP assign, press **Ctrl**+**Alt** to close the console.

6. Set IP addresses for all other VMs using the same procedure.

   - Close vSphere client.

   - Relaunch it with VMware Host IP address and access details of another Rack Server.

   - Create and configure the VMs.

7. Ensure that all VMs can be accessible over LAN after assigning the IP address.

8. From your computer, copy (by scp, ftp, or pscp) the following script into Master GCR node at /data dir:

   ```
   expect_shell.sh fetchMacIDs.sh shareSSHkey.sh startGMS.sh
   ```

2.8 Sharing SSH Key from Master GCR Node

To share the SSH key with all other nodes from the master GCR Node, use the following steps.

1. Log in to master GCR node using ssh, as admin user and run following commands.

   ```
   > en
   # _shell
   ```

```
# cd /data

# ./shareSSHkey.sh <Comma separated IP list>
```

For example,

```
# ./shareSSHkey.sh 10.10.10.130,10.10.10.131,10.10.10.132
```

IP addresses of nodes are displayed, review them and press **Y** to proceed.

2. Ensure successful key sharing by logging into each node from Master GCR node. You must be allowed to log in to without specifying the credentials.

```
# ssh admin@<Node IP>
```

2.9 Collecting MAC IDs from All Nodes

1. Run the following script from the master GCR node to collect the MAC IDs:

```
# ./fetchMacIDs.sh <Comma separated IP list>
```

For example,

```
# ./fetchMacIDs.sh 10.10.10.130,10.10.10.131,10.10.10.132
```

IP addresses of nodes are displayed, review them and press **Y** to proceed.

2. Collect node IPs and MAC addresses. Update the CIQ sheet under **nodeInfo** worksheet with IP and MAC addresses.

3. Update all the other sections of CIQ sheet with Network, Cluster, Storage and Misc sections.

4. In the **storageInfo** Section, use device block ID such as /dev/sda for assigning LUNs to MURAL components except Insta dbroots.

5. For Insta dbroots, use SCSI IDs such as 0:0 and 0:1.

6. Copy the updated CIQ sheet into GMS node at /data directory.

3. Configuring the Master GMS Node

Log into the master GCR node using ssh as an admin user and `admin@123` as the password. Run the following commands to start the GMS Server Application.

```
> en
# _shell
# /data
# ./startGMS.sh
```

The following message is displayed on the terminal after the GMS server process is started successfully:

```
startGMS.sh:msg:tomcat process for GMS started successfully
```

3.1 Generating the Final XML

To generate the final XML, perform the following steps:

1. Copy (SCP or FTP) the updated CIQ sheet into GMS node under `/data` directory.
2. Log into the GMS node using ssh and run the following commands to prepare the final XML:

```
> en
# _shell
# mount -o remount,rw /
# cd /opt/etc/scripts/mmi/CIQ2GMSXML/src/
# python GMSXMLGenerator.py -f /data/Mural_CIQ_Template.xls -s
1
GMS HARDWARE XML saved as /data/configs/gms/Mural_Hardware.xml
**************************************************
SUCCESSFULLY COMPLETED GMS HARDWARE XML GENERATION
**************************************************
#
```

Successful execution creates an XML file under `/data/configs/gms/` folder. In case of error, details are available in the log file stored at `/data/mmi/ciq2gmsxml.log`.

```
# ls -l /data/configs/gms/Mural_Hardware.xml
-rw-r--r-- 1 admin root 32349 Feb 23 11:26
/data/configs/gms/Mural_Hardware.xml
```

3.2 Applying Patches on MURAL Nodes

Apply all the patches applicable for the respective MURAL nodes.

Note: Refer to the MURAL Release Notes for a complete list of patches and installation instructions.

4. Configuring Mural Nodes

After you have installed the MURAL software on GMS, perform the following tasks to configure the software:

Note: It is recommended to create screen and use the created screen for installation to ensure that the connection is not lost between the terminal and servers during installation process.

4.1 Creating GMS XML using Deployment Wizard

The GMS XML is created by using the deployment wizard. The XML is further used for node configuration.

To create the GMS XML, perform the following steps:

1. From the command terminal, execute the following commands to launch MURAL Deployment wizard to prepare the final hardware XML.

   ```
   # sh /opt/etc/scripts/mmi/cisco_mural_deployment_wizard.sh
   ```

   The MURAL Deployment wizard, as illustrated below, is launched on the terminal to accept various configuration inputs.

   General instructions to use the MURAL Deployment wizard:

- Use arrow keys to navigate in the Deployment wizard.

- Use Space-bar to choose the selected item(s).

- Press **Enter** after selecting **Next**, **Previous** or **Exit** to move into next screen, previous screen or exit from the Deployment wizard, respectively.

- Selected items appear with swapped colors for text and background.

By default, MURAL installs Content Analytics (CA) application and following additional application can be selected if purchased. These additional applications have impact on resources required for deployment and should be enabled only if the deployment was sized for them:

- HTTP Errors

- Bulkstats and KPI

- Anomaly

- Tethering

MURAL nodes are further classified and installed together in a particular cluster to host specific application component. The following table lists cluster configurations for various installation types and applications to be selected for these clusters.

Installation Type	GMS Cluster	Collector Cluster	UI Cluster (RGE)	iNSTA Cluster	DN Cluster
Standard Pack	GMS	Namenode & Collector	Rubix-DPI Other Rubix Apps	iNSTA	Compute
Medium Pack	GC (GMS + Namenode & Collector)		Rubix-DPI Other Rubix Apps	iNSTA	Compute

Installation Type	GMS Cluster	Collector Cluster	UI Cluster (RGE)	iNSTA Cluster	DN Cluster
Starter Pack	GCU (GMS + Namenode & Collector + UI)			iNSTA	Compute

The following image illustrates the available cluster configurations.

The following table lists the global variables for Application Configuration Parameters for site-specific requirements.

Property Name	Description	Value
adaptor.bulkStats.numThreads	Keep 2 threads for BS always and other two equal sets to flow and http, totaling 75% of collector cores.	2
adaptor.edrflow.numThreads:		8
adaptor.edrhttp.numThreads:		8

Property Name	Description	Value
adaptor.edrflow.profile: adaptor.edrhttp.profile:	For Anonymisation, replace `EdrFlowParquetReader-NoAnoConfig.xml` file with `EdrFlowParquetReader-Config.xml` and `EdrHttpParquetReader-NoAnoConfig.xml` file With `EdrHttpParquetReader-Config.xml`	No Anonymis-ation
application.atlas.cachePersistToDisk application.bulkstats.cachePersistTo-Disk	true - enable disk based caching for atlas and BS; false - disable	true true
application.atlas.rubixInstance.1.tomcatInstanceMaxSize application.atlas.rubixInstance.1.initialJavaHeapSize:	Maximum and initial java memory for atlas process (should be <= 50% of Rubix node RAM)	90g 90g
application.ruleEngine.rubixInstance.1.tomcatInstanceMaxSize application.ruleEngine.rubixInstance.1.initialJavaHeapSize	Maximum and initial java memory for ruleEngine process (keep it as default)	5g 5g
application.bulkstats.rubixInstance.1.tomcatInstanceMaxSize application.bulkstats.rubixInstance.1.initialJavaHeapSize	Maximum and initial java memory for bulkstat process (should be <= 10% of Rubix node RAM)	20g 20g

Property Name	Description	Value
application.reportAtlas.rubixInstance.1.tomcatInstanceMaxSize application.reportAtlas.rubixInstance.1.initialJavaHeapSize	Maximum and initial java memory for offline report process (keep it as default)	10g 5g
application.rge.rubixInstance.1.tomcatInstanceMaxSize application.rge.rubixInstance.1.initialJavaHeapSize	Maximum and initial java memory for report engine process (keep it as default)	10g 10g
application.httperror.rubixInstance.1.tomcatInstanceMaxSize application.httperror.rubixInstance.1.initialJavaHeapSize	Maximum and initial java memory for http error process (should be <= 10% of Rubix node RAM)	20g 20g
application.launcher.rubixInstance.1.tomcatInstanceMaxSize application.launcher.rubixInstance.1.initialJavaHeapSize	Maximum and initial java memory for launcher process (keep it as default)	1g 1g
Timezone:	Sets the timezone for deployment	UTC
FQDN:	Sets URL to access the MURAL UI	ucsd.cisco.com
rgemailSupport:	dummy-support@cisco.com	As Default
mailSender:	Sets the E-mail ID as sender of offline reports.	admin@cisco.com
mailHost:	Sets SMTP Server host address	mx1.cisco.com
mailPort:	Sets SMTP Server Port	25

2. Select **Generate & Exit**, and press the **Enter** key to create the final XML.

After successful execution final XML (`/config/gms/Mural_
Application.xml`) is generated.

4.2 Assigning Role to MURAL Nodes

After all the nodes are manufactured and GMS XML is created, the nodes are assigned with roles to be performed.

1. Log into the master GMS server using the management IP address and start the configuration

   Execute the `postPxeGmsFlow.py` script on the GMS server

   ```
   > en
   #_shell
   #cd /opt/etc/scripts/GMS_FlowAutomation
   # python postPxeGmsFlow.py
   ```

2. The script installs appliance on all the nodes.

   Provide the name of the Application XML and Management network name. If the name of the Application XML is same as the value displayed for default, press Enter.

   The following sample may resemble the output:

   ```
   Logging started in log file :  install_gms_20160711.log
   Enter the Application Installation xml generated [default =
   Mural_Application.xml]:
   Enter the Management Network name in Mural_Application.xml
   Application xml : MgmtNW
   Initializing the GMS Node Class ...
   Cluster = rge-clu :
   rge-1    => 192.168.188.53
   rge-2    => 192.168.188.55
   Cluster = gc-clu :
   gcn-2    => 192.168.188.59
   gcn-1    => 192.168.188.58
   Cluster = ins-clu :
   ```

```
ins-2    => 192.168.188.54
ins-1    => 192.168.188.52
Cluster = cmp-clu :
dn-1     => 192.168.188.60
dn-2     => 192.168.188.61
Cluster = ui-clu :
ui-1     => 192.168.188.57
ui-2     => 192.168.188.62


Stopping PM Process:  pgsqld
Current state of pgsqld : stopped
Stopping PM Process:  gms_server
Current state of gms_server : stopped
GMS Version Mode set:  full-version
Starting pgsql process
Starting gms_server process
Launched GMS in full-version
Waiting web services to come up
Current process state Pgsql :  running
Current process state GMS :   running
Activating GMS Install XML:  /data/configs/gms/Mural_
Application.xml
GMS XML Activation Done : True
Conjugate cluster node [standby 192.168.188.56]
Installation started on Cluster: [gms-clu] Node [gms-1]
Installation started on Cluster: [gms-clu] Node [gms-2]
Installation started on Cluster: [col-clu]
Installation started on Cluster: [cmp-clu]
Installation started on Cluster: [ins-clu]
Installation started on Cluster: [ui-clu]
Installation started on Cluster: [rge-clu]


Node installation commands executed.
```

```
**************************************************
dn-1 : Node successfully installed.      DateTime: 2016-10-06
13:53:06.788538 Config-Version 1
dn-2 : Node successfully installed.      DateTime: 2016-10-06
13:53:21.014069 Config-Version 1
col-1 : Node successfully installed.     DateTime: 2016-10-06
13:16:11.987805 Config-Version 1
col-2 : Node successfully installed.     DateTime: 2016-10-06
13:12:54.077793 Config-Version 1
gms-1 : Node successfully installed.     DateTime: 2016-10-06
12:04:30.191348 Config-Version 1
gms-2 : Node successfully installed.     DateTime: 2016-10-06
12:13:03.730950 Config-Version 1
ins-1 : Node successfully installed.     DateTime: 2016-10-06
14:29:20.633255 Config-Version 1
ins-2 : Node successfully installed.     DateTime: 2016-10-06
14:05:30.575501 Config-Version 1
ui-1 : Node successfully installed.      DateTime: 2016-10-06
15:12:41.938352 Config-Version 1
ui-2 : Node successfully installed.      DateTime: 2016-10-06
15:10:37.865661 Config-Version 1
rge-1 : Node successfully installed.     DateTime: 2016-10-06
15:25:32.565343 Config-Version 1
rge-2 : Node successfully installed.     DateTime: 2016-10-06
15:25:37.715186 Config-Version 1
**************************************************


Execution Complete !!!
```

From output messages of `postPxeGmsFlow.py` script, make sure that all
the nodes have been successfully installed, before moving to next step.

3. Run the following command on pmx subshell of the master Name node

```
# pmx subshell hadoop_yarn repair hdfs
```

4.3 Configuring Site-Specific Applications

If anonymization feature is enabled, run the following commands on both the master and standby Name nodes

```
# cli -m config
(config) # collector anonymization key <key> iv <iv_key> encode
(config) # write memory
(config) # pm process collector restart
(config) # quit
```

For example:

```
(config) # collector anonymization key
0123456789012345678901234567890 iv 0123456789012345 encode
```

<key> and <iv_key> string shall be provided by customer during installation.

4.4 Verifying the Setup

After making any performance related modifications, verify that the processes are running.

Note: After you make any modifications, wait for atleast 15 minutes to ensure processes have had a chance to restart before running the following commands.

4.4.1 Verifying Processes

Perform the following steps to verify that the required processes are running:

1. Log in to the master Name node and execute the following commands to verify that all Compute nodes have joined the HDFS cluster. The output shown in the following sample command sequence indicates correct configuration.

```
> en
# _shell
# ps -ef | grep java | grep Dproc | grep -v Dproc_dfs | awk '
{print $9}'
-Dproc_historyserver
-Dproc_journalnode
-Dproc_namenode
-Dproc_resourcemanager
-Dproc_namenode
-Dproc_datanode
-Dproc_secondarynamenode
#
# hdfs dfsadmin -report | egrep "available|Name|Status"
Datanodes available: 3 (3 total, 0 dead)   Name:
10.10.2.13:50010
Decommission Status : Normal
Name: 10.10.2.14:50010
Decommission Status : Normal
Name: 10.10.2.17:50010
Decommission Status : Normal
```

Note: The IP addresses mentioned in the sample code above are internal IP addresses.

2. Run the following commands on the standby Name node:

```
> en
(config) # _shell
# ps -ef | grep java | grep Dproc | grep -v Dproc_dfs | awk '
{print $9}'
-Dproc_namenode
-Dproc_namenode
-Dproc_datanode
```

```
-Dproc_secondarynamenode

-Dproc_journalnode
```

3. Log into the master Insta node, and execute the following commands:

```
> en

# conf t

(config) # show pm process insta
```

If the process is running, the command output is `Current status:`
`running`.

Execute the following command to verify the status of the Insta node.

```
(config) # insta infinidb get-status-info
```

The output must show all modules in ACTIVE state. It must also list all the instances and Adaptor status as RUNNING.

4. Log into the standby Insta node and repeat the same commands shown in step 3.

5. Execute the following commands to check status of Postgres process on both the Master and Standby nodes.

For the Standard Pack setup, check the status of Postgress on GMS, Collector and iNSTA nodes.

For the Medium Pack setup, check the status of Postgress on GMS and iNSTA nodes.

```
> en

# _shell

# ps -ef | grep postmaster | grep -v grep

2278     1  0 Feb13 ?        00:22:34
/usr/pgsql-9.3/bin/postmaster -p 5432 -D /data/pgsql/9.3/data
```

4.4.2 Verifying Installation with Dummy Data

you can verify the installation and identify configuration issues by running the custom utility. For more information, see"Verifying Installation with Dummy

Data" on page 62.

5. Configuring ASR and IBs

This topic provides information about creating and configuring the ASR and IBs. It also provides information about verification of the collector processes.

To configure your system for the data specific for your environment, you must update the information bases (IBs) and fetch them. In some cases, you might need to also manually modify the IBs to add or reclassify entries.

5.1 Updating IBs

The following table shows a sample data set for setting up the IBs.

Sample Data Set													
DC	Gurgaon	GGSNIP	27.23.157.1	SGSNIP	2.2.2.1	SGSN	CYBERCITY	APN	AIRTEL-NH8	GROUP	AIRTEL	RATID	1	RATTYPE	CDMA
Region	SEZ	GGSN	GURGAON-GGSN	SGSNIP	2.2.2.2	SGSN	UBICITY	APN	AIRTEL-OGN	GROUP	AIRTEL	RATID	2	RATTYPE	GSM
Area	ITPARKS							APN	VDF-NH8	GROUP	VODAFONE	RATID	3	RATTYPE	WIMAX
								APN	Sushfone-1	GROUP	VODAFONE	RATID	4	RATTYPE	LTE

Note: Refer to the preceding table as a sample. Use the data that matches your environment. For example, for GGSN, you may want to use GGSN, PGW, or HA. In this case, GGSNIP is the management IP address. For SGSN, you may want to use SGSN, SGW, HSGW, or PDSN. In this case, SSGNIP is the service IP address.

To configure the IBs :

1. Log into the master collector node and execute the following commands:

```
[admin@collector-1 ~]# pmx
Welcome to pmx configuration environment.
pm extension> subshell aggregation_center
pm extension (aggregation center)> update all ibs from image
```

2. Enter the GGSN IPs and GGSN names:

```
pm extension (aggregation center)> edit ib ipGgsn.map add
GGSN IP: 27.23.157.1
GGSN: GGSN1
pm extension (aggregation center)> show ib ipGgsn.map
```

```
1[27.23.157.1][GGSN1]
pm extension (aggregation center)>
```

3. Enter the SGSN IPs and SGSN name:

```
pm extension (aggregation center)> edit ib ipSgsn.map add
SGSN IP: 2.2.2.1
SGSN: SGSN1
pm extension (aggregation center)> show ib ipSgsn.map
1[2.2.2.1][SGSN1]
pm extension (aggregation center)>
```

4. Enter the APN name and corresponding APN group:

```
pm extension (aggregation center)> edit ib apnGroup.map add
APN: Sushfone-1APN Group: Sushfone-1
pm extension (aggregation center)> show ib apnGroup.map
1[Sushfone-1][Sushfone-1]
2[Sushfone-2][Sushfone-2]
3[Sushfone-3][Sushfone-3]pm extension (aggregation center)> quit
```

For performing the cell sector annotation see Modifying Behaviour for Cell Sector Annotation.

Execute the following commands to configure the IBs for BulkStats:

```
pm extension> subshell bulkstats
pm extension (bulk stats)> update all ibs from image
pm extension (bulk stats)> quit
```

5.2 Adding ASR Gateways

Add all new DC or gateways to the configuration for the system. DC, ASR, and Gateway are synonymous terms and can be used interchangeably. Gateway/DC name is a unique key for a gateway configuration. In the BulkStat hierarchy, the gateway name and DC name are the same.

The following section lists guidelines for adding new gateways:

- All input directories are created under the /data/collector path. Hence, in the example below, the ASR should send EDR files to `data/collector/California/edr1` and send the Bulkstats file to `/data/collector/California/bs1`.

  Important: Confirm the actual input directories with the Cisco Technical Support team for your ASR platform.

- Log into master Name-node and create a CSV file (under /data dir) for adding the gateways into system as per below prescribed format and example:

```
gateway,dc,region,location,schema-version,timezone,edr-filename-
pattern,bulkstat-filename-pattern,edr-collector-filename-
pattern,bulkstat-collector-filename-pattern,collector-ip,edr-
file-path,bulkstat-file-path
```

Example of EDR with Bulkstat.

```
DC1,DC1,REGION1,AREA1,19,UTC,*_MURAL-edr-
*_%MM%DD%YYYY%hh%mm%ss*,*_%YYYY%MM%DD%hh%mm%ss,*_*_
*_%MM%DD%YYYY%hh%mm%ss*.gz,*_%YYYY%MM%DD%hh%mm%ss,192.168.192.130
|192.168.192.131,/DC1/edr,/DC1/bs
```

Example of EDR with without Bulkstat.

```
DC1,DC1,REGION1,AREA1,,UTC,*_MURAL-edr-
*_%MM%DD%YYYY%hh%mm%ss*,,*_*_
*_%MM%DD%YYYY%hh%mm%ss*.gz,,192.168.192.130|192.168.192.131,/DC1/
edr,
```

Note: Use internal physical IPs for collector IPs and separate them using vertical bar i.e. "|" for multiple entries.

For adding gateway without Bulkstat feed, keep the fields related to Bulkstat empty.

For EDR filename patterns, see "File Pattern Regular Expression" on page 75

To configure gateways:

1. Add gateway information for each gateway based on the guidelines provided in the preceding table:

```
> en
# conf t
(config)# pmx subshell aggregation_center
pm extension (aggregation center)> Welcome to pmx configuration
environment.
pm extension (aggregation center)> add bulk gateways
Please enter bulk file path: /data/gw.csv
```

2. Verify that the new gateway has been added successfully:

```
pm extension (aggregation center)> show gateways
```

3. Verify the IP addresses for all Collectors:

```
pm extension (aggregation center)> show collector IPs
 192.168.103.110, 192.168.103.113
pm extension (aggregation center)>
```

4. Set the BulkStats timezone to UTC in gateway.conf for every gateway. The reason for this is that the ASR internally changes the time zone to GMT for the BulkStats file. Edit gateway.conf for every BulkStats source at the path:

```
/data/configs/gateway/gateway.conf "timezone": "UTC"
```

5. Push the gateway configuration to all the Collectors:

```
pm extension (aggregation center)> push gateway configuration
```

6. Generate and push all IBs:

```
pm extension (aggregation center)> generate urlcat ib from image
pm extension (aggregation center)> show urlcat version
pm extension (aggregation center)> generate all ibs
pm extension (aggregation center)> push all ibs
pm extension (aggregation center)> quit
```

For Bulkstats:

```
pm extension> subshell bulkstats

pm extension (bulk stats)> generate all ibs

pm extension (bulk stats)> push all ibs
```

Run the following commands if Anomaly Detection is enabled:

```
[admin@collector-1 ~]# pmx

Welcome to pmx configuration environment.

pm extension> subshell anomaly

pm extension (anomaly)> update all ibs

pm extension (anomaly)> quit
```

5.3 Modifying Behaviour for Cell Sector Annotation

You can categorize the Cell Sectors, which are not present in the IB configuration into Other cells,and and GGSN and SGSN which are not present in the IB configuration into Miscellaneous.

The values of Cell Sectors, GGSN and SGSN to be retained in data are specified in the following IBs:

- **cellLocationInfo.list** for Cell Sectors
- **ipGGSN.map** for GGSN
- **ipSGSN.map** for SGSN

Perform the following steps for categorizing the Cell Sectors, GGSN and SGSN:

1. Execute the following command to create the backup of *EDR.json*:

```
# cp -p /opt/etc/oozie/EDR/app/EDR.json
/opt/etc/oozie/EDR/app/EDR.json.org
```

2. Execute the following command to check the current values assigned to **groupSGSNAsMisc**, **groupGGSNAsMisc** and **emitUsingMCCMNC**:

```
# grep -e "groupSGSNAsMisc" -e "groupGGSNAsMisc" -e
"emitUsingMCCMNC" /opt/etc/oozie/EDR/app/EDR.json
        "groupGGSNAsMisc":"false"
```

```
        "groupSGSNAsMisc":"false"

        "emitUsingMCCMNC":"false"
```

3. Execute the following command to set the value of groupGGSNAsMisc as true:

```
# sed -i
's/"groupGGSNAsMisc":"false"/"groupGGSNAsMisc":"true"/'
/opt/etc/oozie/EDR/app/EDR.json
```

4. Execute the following command to set the value of groupSGSNAsMisc as true:

```
# sed -i
's/"groupSGSNAsMisc":"false"/"groupSGSNAsMisc":"true"/'
/opt/etc/oozie/EDR/app/EDR.json
```

5. Execute the following command to set the value of emitUsingMCCMNC as true:

```
# sed -i
's/"emitUsingMCCMNC":"false"/"emitUsingMCCMNC":"true"/'
/opt/etc/oozie/EDR/app/EDR.json
```

6. Execute the following command to validate the value of **groupSGSNAsMisc**, **groupGGSNAsMisc** and **emitUsingMCCMNC**:

```
# grep -e "groupSGSNAsMisc" -e "groupGGSNAsMisc" -e
"emitUsingMCCMNC" /opt/etc/oozie/EDR/app/EDR.json
        "groupGGSNAsMisc":"true"
        "groupSGSNAsMisc":"true"
        "emitUsingMCCMNC":"true"
```

5.3.1 Adding the Cell Sectors, GGSN and SGSN in IBs

If you want to retain the values of Cell Sectors, GGSN and SGSN, add their values in cellLocationInfo.list, ipGGSN.map and ipSGSN.map respectively.

To add the values, perform the perform the following steps:

1. To add the values of Cell Sectors, GGSN and SGSN to be retained incellLocationInfo.list, ipGGSN.map and ipSGSN.map respectively, execute the following commands:

```
> en
# conf t
(config) # pmx subshell aggregation_center
pm extension (aggregation center)> edit ib
cellLocationInfo.list add
MCC,MNC: 101,01
pm extension (aggregation center)> edit ib ipGgsn.map add
GGSN IP: 27.23.157.1
GGSN: GGSN1
pm extension (aggregation center)> edit ib ipSgsn.map add
SGSN IP: 2.2.2.1
SGSN: SGSN1
```

NOTE: If the IB is empty, all the cell sectors will be emitted as "Other Cells".

2. To update the IBs, execute the following commands:

```
> en
# conf t
(config) # pmx subshell aggregation_center
pm extension (aggregation center)> generate all ibs
pm extension (aggregation center)> push all ibs
```

5.4 Verifying Collector Process

To verify the collector process, run the following command on both the master and standby Collector:

```
# cli -t "en" "conf t" "show pm process collector" | grep status
Current status:  running
```

6. Processing the Data

This section includes information for setting up an ASR user in the Collector nodes and sending the EDR and BulkStats data feeds to the MURAL platform, setting the data start time, and running the data processing commands.

6.1 Verifying Incoming Data

This section includes information for setting up an ASR user in the Collector nodes and sending the EDR and BulkStats data feeds to the MURAL platform, setting the data start time, and running the data processing commands.

Use one of the ASR data feed methods to send data to the MURAL platform.

6.1.1 Creating a New User for the ASR in the Collectors

Perform the following step to set up a new user for the ASR in the Collectors.

1. Log into the master Collector node and execute the following commands create the user:

```
# en
> conf t
(config)> username username password password
(config)> write memory
(config)> _shell
```

Note: The username and password must be same as configured on the ASR for EDR and BulkStats file transfers.

6.1.2 Configuring the SSH Session Limit

Perform the following steps to configure the SSH session limits:

1. Execute the following commands to set the concurrent SSH limit to 50:

```
> en
# conf t
(config) # ssh server max-startups 50
```

```
(config) # ssh server max-sessions 50
(config) # write memory
(config) # pm process sshd restart
```

2. Repeat steps 1 through 2 on the standby Collector node.

6.1.3 Verifying the Input Data Feed

Start sending the EDR and BulkStats data feeds to the MURAL platform. If the ASR is used as an input node, the start time from the filename is created in the incoming EDR flow directory.

The file name has the timestamp, which can be used for job scheduling in the following process.

Notes:

- MURAL processes data that is received after the deployment is completed. MURAL is not configured to process the historical data.

- MURAL recommends ASR files to be of 10MB compressed size for optimum performance.

Validating Data on the Collector Nodes

1. Log into the master CollectorNamenode and go to _shell.

```
> en
# _shell
```

2. Run the indicated **hdfs** commands for the *mm* variable (minutes) in the timestamp), specify a multiple of 5 (**05**, **10**, **15**, and so on) up to **55**.

    Note: Specify the year, month day, hour, and minute for which data is being sent to the MURAL system.

```
# hdfs dfs -ls /data/collector/1/output/edrflow/YYYY/MM/DD/HH/mm/*
# hdfs dfs -ls /data/collector/1/output/edrhttp/YYYY/MM/DD/HH/mm/*
# hdfs dfs -ls /data/collector/1/output/bulkStats/YYYY/MM/DD/HH/mm/*
```

If the Collector node is receiving data in the expected format, it retains the data in HDFS. These directories and files are updated continuously as the data keeps coming in.

6.2 Configuring Oozie Jobs

This topics provides information about setting the start time of the oozie jobs and starting the oozie jobs after configuring the start time.

6.2.1 Setting the Start Time of Oozie jobs

To set the data start time in the configuration, perform the following steps:

1. Log into the master Name node and make the the root "**/"** file system writable.

```
> en
# _shell
# mount -o remount,rw /
# cd /opt/deployment/Mural_setStartTime/
# ./setOozieTime --dataStartTime data-start-time --node collector-mgmt-
IP --password admin-password
```

2. Execute the **setOozieTime** script to set the time at which EDR and BulkStats data starts coming into the Hadoop directories listed in "Configuring Oozie Jobs" above.

   For example, if EDR and Bulkstats data starts coming into the Hadoop system from April 1, 2015, 06:00 onwards, run the following scripts with the `start_time` value as "2015-04-01T06:00Z":

```
# ./setOozieTime --dataStartTime  2015-04-01T06:00Z --node
192.168.147.11 --password admin@123
```

Note:

- Enter minutes as a multiple of 5. For example, "2013-04-01T06:00Z." Ensure that there is a continuous flow of data into the Hadoop without any gaps since the specified time.

- If the starttime is not provided in the command, system takes the current time of system as start time.
- If the starttime of the setoozie script is set as the last day of the month, update the job timing of **AnomalyHiveMonthly** manually after executing the script. The new job starttime to be updated should be last hour of the next month **+/- offset** of based on timezone. For example, if data starttime for the system is 31st march 5:00 and timezone is europe/budapest, then set the job **AnomalyHiveMonthly** starttime as 30 april 22:00.

Execute the following commands to update the start time:

```
pmx subshell oozie
pm extension (oozie)> set job AnomalyHiveMonthly
attribute jobStart 2016-04-30T22:00
```

3. Execute the Set Job Time Script for both the master and standby Namenodes.

   Note: This script may take upto 30 minutes to complete for one node. Therefore, wait until it completes and returns to the prompt.

6.2.2 Starting the Oozie Jobs

Log into the master Collector/Name node and run the data processing commands from the Oozie subshell:

```
> en
# conf t
(config)# pmx
Welcome to pmx configuration environment.
pm extension> subshell oozie
pm extension (oozie)> run job all
```

The command output shows all the jobs that were initiated and if the jobs started successfully or not.

Note: It may take approximately 20 minutes to start all the jobs, depending upon what all applications are enabled.

6.3 Validating Data on Nodes

This section includes instructions for validating data after completing the system installations.

6.3.1 Validating Data on the Compute Blades (Data Nodes)

This section includes the steps required to validate data on the Compute blades (Data nodes) following the installation process.

Caution: Wait two hours after completing the steps in "Validating Data on Nodes" above. This allows sufficient time for the jobs that process the collector data to start, and the `done.txt` files to be updated. If you do not wait, the validation may fail.

6.3.2 Validating EDR Data

1. Log into the master Collector node and go to the `_shell`.

   ```
   > en
   # _shell
   ```

2. Check the last timestamp for the Core job.

   ```
   # hdfs dfs -text /data/CoreJob/done.txt
   ```

3. Check the last timestamp for the EDR data cubes being generated by the job.

   ```
   # hdfs dfs -text /data/EDR/done.txt
   ```

4. Check the last timestamp for CubeExporter data cubes.

   ```
   # hdfs dfs -text /data/CubeExporter/done.txt
   ```

5. Check the last timestamp for generated and exported Bulkstats data cubes.

   ```
   # hdfs dfs -text /data/BulkStat/done.txt
   # hdfs dfs -text /data/BSAgg15min/done.txt
   # hdfs dfs -text /data/BulkStatExporter_15min/done.txt
   ```

6.3.3 Validating Insta Data

1. Log into the master Insta node and check the name of the database configured for EDRHET:

```
> en
# _shell
# cli -t "en" "conf t" "show runn full" |
grep "insta instance 01 cubes-database" | awk -F ' ' '{print $5}'
database_muralHET
```

2. Open the **idbmysql** user interface and specify **database_mural** asselect the database.

```
# idbmysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
...
mysql> use DATABASE_MURALHttperror MAP;
Database changed
```

3. Display the values in the **mints** and **maxts** columns for the 60-minute bin class and -1 aggregation level (shown in the first row of the following example).

4. Run the **date** command to convert the values from the **mints** and **maxts** columns to human-readable format.

6.3.4 Validating Bulk Stats Data on the Insta Blade

1. Use SSH to log into the master Insta node and check the name of the database configured for EDR:

```
> en
# _shell
# cli -t "en" "conf t" "show runn full" |
grep "insta instance 1 cubes-database" | awk -F ' ' '{print $5}'
bulkstats
```

2. Open the **idbmysql** user interface and select **bulkStats** as the database.

```
# idbmysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
...
mysql> use bulkStats;
Database changed
```

3. Display the values in the **mints** and **maxts** columns for the 900 aggregation interval (shown in the second row in the example).

```
mysql> select * from bin_metatable;
+----------+--------------------+------------+------------+---------+
| binclass | aggregationinterval | mints      | maxts      | binType |
+----------+--------------------+------------+------------+---------+
| 5min     |                 -1 |          0 |          0 | NULL    |
| 5min     |                900 | 1406713500 | 1409301900 | NULL    |
| 5min     |               3600 |          0 |          0 | NULL    |
| 5min     |              86400 |          0 |          0 | NULL    |
| 5min     |             604800 |          0 |          0 | NULL    |
| 5min     |            2419200 |          0 |          0 | NULL    |
+----------+--------------------+------------+------------+---------+
6 rows in set (12.18 sec)
mysql> quit
```

4. Convert the date format. Run the **date** command with the value of **maxts** (captured from the step above) for the row which shows **aggregationinterval** as 900.

    The following example indicates that data was processed between 09:45 on July 30 and 08:45 on August 29.

```
# date -d @1406713500
Wed Jul 30 09:45:00 UTC 2014
# date -d @1409301900
Fri Aug 29 08:45:00 UTC 2014
```

6.3.5 Configuring System Monitoring Interface

System Monitoring Interface (SMI) is a dashboard that can be used to display MURAL's system-related health information for the rolling 24-hours period.

Benefits of SMI:

- The system administrator can check the health of the MURAL system and contact Technical Support if a condition is not being met.
- The system administrator can use the dashboard to provide assurance to customer that MURAL services are working fine as per the benchmarking.

To configure SMI, perform the following steps:

1. Log into the RGE node master node.

2. Execute the following command to copy the active GMS file from the GMS node to the RGE master node:

```
# cd /var/home/root
# scp admin@<GMS Master IP/Hostname>:/config/gms/active-
configuration/resolved_active.xml .
```

3. Execute the following commands to install SMI on the RGE master node:

```
# cd /var/home/root
# tar -xvzf pkg_dashboard.tgz
#  ./install_dashboard.sh
```

The output may resemble as follows:

```
    ----------------------------------------------------------
--------------------
    install_dashboard.sh:msg: Usage.......
    ----------------------------------------------------------
--------------------
    install_dashboard.sh:input: Enter TAR Filename(with full
path) : /var/home/root/dashboard.tgz
    install_dashboard.sh:input: Enter GMS XML Filename(with
full path) : /var/home/root/resolved_active.xml
    install_dashboard.sh:input: Enter Management Network Name
used in above XML[MgmtNW]: MgmtNW
    install_dashboard.sh:msg: Network Name is : MgmtNW
```

```
    ------------------------------------------------------------
--------------------

    install_dashboard.sh:msg: Extracting tar ball into node
...
    .
    .
    .
    .
    .


    Stopping pm:
[  OK  ]


    Starting pm:
[  OK  ]


    install_dashboard.sh:msg: Dashboard configuration is done
successfully!!!!


    install_dashboard.sh:msg: Start Dashboard from UI by
http://<This node IP>:15443


    #
```

Note: Provide the value for "Management Network Name" attribute as specified in the resolved_active.xml file.

4. Execute the following commands to check if SMI processes have started successfully:

   - To check if grafana is up:

   ```
   # ps -ef|grep grafana
   ```

The output may resemble as follows:

```
admin       4300   2653   0 14:43 ?           00:00:00
/opt/grafana/bin/grafana-server -homepath=/opt/grafana

admin       6669 31484   0 14:44 pts/1     00:00:00 grep
grafana
```

- To check if graphite and carbon-cache are up:

```
# ps -ef|grep graph
```

The output may resemble as follows:

```
admin       3313   2653   0 14:43 ?           00:00:00
/usr/local/bin/python /opt/graphite/bin/carbon-cache.py -
-nodaemon --logdir=/var/log --pidfile=/var/run/carbon-
cache.pid start

admin       5546   2653   1 14:43 ?           00:00:00
/usr/local/bin/python /usr/local/bin/gunicorn_django --
bind=127.0.0.1:12040 --log-file=/var/log/gunicorn-
graphite/gunicorn.log --preload --
pythonpath=/opt/graphite/webapp/graphite --
settings=settings --workers=3 --pid=/var/run/gunicorn-
graphite/gunicorn-graphite.pid

admin       5753   5546   0 14:43 ?           00:00:00
/usr/local/bin/python /usr/local/bin/gunicorn_django --
bind=127.0.0.1:12040 --log-file=/var/log/gunicorn-
graphite/gunicorn.log --preload --
pythonpath=/opt/graphite/webapp/graphite --
settings=settings --workers=3 --pid=/var/run/gunicorn-
```

```
graphite/gunicorn-graphite.pid

admin      5773  5546  0 14:43 ?          00:00:00
/usr/local/bin/python /usr/local/bin/gunicorn_django --
bind=127.0.0.1:12040 --log-file=/var/log/gunicorn-
graphite/gunicorn.log --preload --
pythonpath=/opt/graphite/webapp/graphite --
settings=settings --workers=3 --pid=/var/run/gunicorn-
graphite/gunicorn-graphite.pid

admin      5779  5546  0 14:43 ?          00:00:00
/usr/local/bin/python /usr/local/bin/gunicorn_django --
bind=127.0.0.1:12040 --log-file=/var/log/gunicorn-
graphite/gunicorn.log --preload --
pythonpath=/opt/graphite/webapp/graphite --
settings=settings --workers=3 --pid=/var/run/gunicorn-
graphite/gunicorn-graphite.pid

admin      6702 31484  0 14:44 pts/1    00:00:00 grep
graph
```

5. Execute the following commands:

```
# cli -m config
(config)# cluster master self
(config)# quit
#
```

6.4 Starting UI Processes and Verifying Data

Start the UI processes and verify UI data. Ensure that the URL is set up in the DNS for the production system.

Starting the Rubix Tomcat Instance on Both GCR Nodes

Note: You should only start UI Tomcat instances after at least two hours of data has been pushed into the Insta node.

1. Log into the master GCR node.

```
> en
# conf t
```

   If RGE and DPI UI are running on same node, execute the following command:

```
(config)# rubix modify-app rge set adv-attribute
rubixReportQueueMax value 1
(config)# write memory
```

2. Run the following commands to start the EDR process.

```
(config)# pm process rubix restart
(config)# rubix modify-app atlas enable
(config)# rubix modify-app atlas modify-instance 1 enable
(config)# write memory
```

   Check the tomcat process status using command:

```
(config)# rubix status application atlas
```

   Note: Ensure the running status of above service as `Current Status: running` before proceeding to start next process.

3. Run the following commands to start the other processes:

```
> en
# conf t
(config)# pm process rubix restart
(config)# rubix modify-app ApplicationName enable
(config)# rubix modify-app ApplicationName modify-instance 1 enable
(config)# write memory
```

   Where *ApplicationName* is replaced by the following applications in the same order:

   - bulkstats
   - reportAtlas

- rge

- ruleEngine (if Anomaly or BulkStat is enabled)

- httperror

- launcher

Check the tomcat process status using command:

```
(config)# rubix status ApplicationName atlas
```

Note: Ensure the running status of above service as `Current Status: running` before proceeding to start next process. Check the corresponding rubix.log file to ensure that Atlas and HET applications are started properly without any exception.

4. Log into the standby GCR node and repeat Steps 2 and 3.

5. Access the UIs by going to the URL **https://*domain-name*:21443/** through your browser.

The domain name to be used is the one which was provided at the time of initial configuration through GMS for the UI nodes configuration details. For example:

```
URL: https://demo.cisco.com:21443/
Username: admin
Password: admin123
```

Note: Since the common certificate installation procedure is not finalized, you must click the same URL for the BulkStats and RGE ports once before opening the actual URL.

Visit the following ports once and accept the certificates:

```
https://domainName:20443/
https://domainName:30443/
```

For example:

If the Anomaly feature is enabled, also run the command on port 50443. For example,

```
https://demo.cisco.com:50443/
```

Note: Once the installation is completed, ensure to back up the configurations. Refer to the *Operations and Troubleshooting Guide* for more information.

6.5 Updating Whitelists

After running MURAL system for two-three hours, run the following command from master GCR node to generate updated whitelists:

```
# pmx subshell aggregation_center
pm extension (aggregation center)> generate whitelist
Triggered Whitelist Generation; Check status via 'show whitelist
status' command
pm extension (aggregation center)> quit
pm extension> quit
#
```

Observe the categorization in UI after two hours to see the effects of whitelists update.

6.6 Setting Up Offline Reports

This topic provides information related to the configuration required for offline reports.

6.6.1 Uncategorized URL, UA, and TAC Reports

Create a file named `serverFile_uncatReports` on the master GCR node containing the destination information, to which the uncategorized URL, UA, and TAC reports would be copied.

1. The `serverFile_uncatReports` file contains the entries for the data transfer destination location. This file has the following format:

   ```
   IP, username, password, location-to-copy-reports
   ```

   For example,

```
192.168.156.96, admin, admin@123, /data/offline_uncat_reports
```

Note: The delimiter in this file must be ", " (comma followed by a space).

2. Log into the master GCR node and navigate to the `/data/work` subdirectory:

```
> en
# _shell
# cd /data
# cd work
```

3. Create the `serverFile_uncatReports` file:

```
# vi /data/work/serverFile_uncatReports
192.168.156.96, admin, admin@123, /data/offline_uncat_reports
```

Note: For the SCP protocol, the destination path should be the destination server. The destination path is not required for SFTP.

4. Create the same file on the standby GCR node.

6.6.2 Tethering Reports

Create two files called `serverFile_tethering` and `serverFile_tethering_subscribers_report` with details of the ASR gateways, where the TAC, OS or UA databases, and subscriber reports created as a result of tethering processing, need to be pushed.

1. The `serverFile_tethering` file contains the entries for the data transfer destination location. This file has the following format:

```
Gateway-IP, gateway-username, gateway-password, location-to-
copy-reports
```

Where:

- *Gateway-IP* is the ASR5K gateway IP address
- *gateway-username* is the username for logging into ASR5K Gateway
- *gateway-password* is the corresponding password to the username

- *location-to-copy-reports* is the location on the ASR5K Gateway machine where databases need to be copied

2. Log in to the master GCR node:

```
> en
# _shell
```

3. Go to the data directory and create a sub-directory named work:

```
# cd /data
# mkdir work
```

4. Go to the work subdirectory and create the files:

```
# cd work
# vi /data/work/serverFile_tethering
192.168.156.96, admin, admin@123, /data/tethering_ibs
# vi  /data/work/serverFile_tethering_subscribers_report
192.168.156.96, admin, admin@123, /data/tethering_subs
```

Note: The delimiter in this file must be ", " (comma followed by a space).

5. Create the same file on the standby GCR node as well.

For the SCP protocol, the destination path should be present at the destination server. This is not required for SFTP.

This file can have multiple rows of this kind.

6.6.3 Rule Base Reports

Create a file with details of ASR IP addresses, access the details and report the destination paths.

1. Log in to the master Name node and run the following commands:

```
> en
# _shell
# cd /data
# mkdir work
```

```
# cd work
# vi /data/work/serverfile_Rulebase
192.168.156.96, admin, admin@123, /data/ruleBase_reports
```

2. Create a similar file on the standby Name node as well.

7. Verifying Installation with Dummy Data

If real input feed is not available due to any reason, you can verify the installation and identify configuration issues by running the custom utility.

This custom utility performs the following functions:

- Data Generator

    - Generates EDR and Bulkstats feeds with the configurable parameters including input header and input file name.

    - Sets up configuration required for collector.

    - Sets up IB specific configuration including Bulkstats.

    - Creates new iNSTA database.

- Job Configurations

    - Stops all jobs and removes HDFS .done file.

    - Configures the job with parameters defined in the config.txt file.

    - Configures Rubix by enabling atlas application on Rubix servers.

    - Runs and monitors the jobs.

    - Checks disk cache size.

- Cleanup

    - Deletes the content from the `/data/ib/` folder on all the servers.

    - Deletes the content from the `/data/atlasData/` folder on Rubix servers.

    - Removes data from the `/data/collector/edrflow_backup`, `/data/collector/edrhttp_backup`, and `/data/collector/bulkstats_files_backup/test/` folders on all the namenodes.

    - Removes the HDFS-related contents from the `/IB/`,

`/local/IB/,/data/ib/,` and `/data/output/` folders.

- Removes the HDFS-related content from the `/data/EDR_ EDRSubcrBytesAgg_lastSuccess,` `/data/EDR_ SubscriberDeviceMPH_lastSuccess,` `/data/EDR_EDRSubcrDev_ lastSuccess,` and `/data/EDR_EDRTopN_lastSuccess` folders.

- Removes HDFS-related content from the `/data/TEST/edrhttp/,` and `/data/TEST/edrflow/` folders.

- Removes disk cache (`/data/diskstore/UI-EDR`) from all the Rubix servers.

- Removes Bulkstats cache (`/data/diskstore/bulkstats- <hostname>`) from all the RGE nodes (applicable only if bulkstat is enabled).

- Reverts the system to its original configurations (settings made before executing the utility).

7.1 Prerequisite

The following table lists the parameters and their default values that are required to be filled in the config.txt file present at *opt/etc/scripts/DummyData* .

Group	Parameter Name	Default Value	Description
COMMON_ CONF	jobName	CoreJob, EDR, and CubeExporter	Jobs that are configured, initiated, and monitored by the utility.
	edrFlowPath	/data/TEST/edrflow/%Y/%M/%D/%H/-%mi	Path where collector puts the dummy data output for edrflow.
	edrHttpPath	/data/TEST/edrhttp/%Y/%M/%D/%H/-%mi	Path where collector puts the dummy data output for edrhttp.

Group	Parameter Name	Default Value	Description
	isBulkstatEnabled	true	Flag indicating if bulkstats need to be configured by the utility to initiate bulkstats pipeline. **Note**: The version of the Bulkstats file is 19.
	bulkstatInputPath	/data/TEST/bulkStats/%Y/%M/%D/%H/%mi	Path where collector puts the dummy data output for bulkstats.
	gmsVip	None	Virtual IP address of the GMS node for parsing the active XML.

Group	Parameter Name	Default Value	Description
	edrFilePattern	None Note: Ensure that this value is same as the value set for collector. To obtain the collector value, execute the command, `echo show running-config full | cli -m config | grep edr | grep filename-format | awk '{print $NF}' | sort -nu.`	edr file pattern in which input data is required.
GENERATOR_CONF	edrflow_header	\<default header>	Input edrflow file header required for data generation. Ensure to update it based on business requirement.
	edrhttp_header	\<default header>	Input edrhttp file header required for data generation. Ensure to update it based on business requirement.

7.2 Assumptions

Refer to the following assumptions before using this utility:

- The bulkstats generator version 19 is used.

- The anomaly jobs are not configured.

- The setup is freshly manufactured with no data in collector input/output paths and empty disk cache.

- After successful execution of the full pipe, separate script is run to clean the data from the system for all modules as follows:

  a. Collector— input local path and output hdfs path
  b. Jobs output and intermediate path (for example, last successful path for EDR actions)
  c. iNSTA— database used for dummy execution
  d. Rubix— diskcache and /data/atlasData ibs
  e. IBs that are updated or generated as part of the process

- The database used in iNSTA is hardcoded as TEST_MURAL and TEST_ BULKSTATS. The default database is hardcoded as DATABASE_MURAL and BULKSTATS.

- The time range and data sizes are hardcoded.

- The cleanup IB procedure reverts any changes done in the IB manually.

- The setOozieTime script is executed after the cleanup action.

- The config.txt is correctly filled. Refer to the table in the "Prerequisite" on page 63 section.

- gmsVip in config.txt is correctly configured.

- If the tool is stopped in between due to any reason, you must run the dummy script with option cleanup first before proceeding further.

- The whole pipeline is executed in the live mode.

7.3 Run the Complete Pipe

1. Login to the Master Namenode.
2. Update config.txt file present in */opt/etc/scripts/DummyData* as described in "Prerequisite" on page 63.
3. Execute the script as follows:

   Note: It is recommended to execute the script in screen.

```
[admin@VM-MUR-GCN1 DummyData]# cd /opt/etc/scripts/DummyData
[admin@VM-MUR-GCN1 DummyData]# python DummyData.py
*************************************
Welcome to the Dummy Data tool :
This tool gives you the option of creating dummy data and run
multiple jobs on it so that UI can be seen with dummy data
Once dummy data is populated, this tool also gives the option
of cleaning up the dummy data and revert the system to its
original state
 Input required : /opt/etc/scripts/DummyData/config.txt
 Logs Path: /opt/etc/scripts/DummyData/log.txt
 *************************************
Enter valid input from below options :
 1. Run complete pipe
 2. Cleanup the system
 3. Exit
1
Have you entered the proper values in config.txt file . Enter
yes or no
yes
Please check /opt/etc/scripts/DummyData/log.txt for logs


Welcome to Mural 3.9
Execution finished . Now check UI for data
```

Refer to the following images that illustrate data in the MURAL UI after running the complete pipeline.

Note: Actual data in the UI may vary by two percent.

Data on the Network tab

Data on the Radio Access tab

Data on the Device tab

Data on the Subscriber tab

Data on the Content tab

Data on the Bulkstats tab

Data on the KPI tab

7.4 Clean up the Pipe

Execute the script as follows:

```
[admin@VM-MUR-GCN1 DummyData]# cd /opt/etc/scripts/DummyData
[admin@VM-MUR-GCN1 DummyData]# python DummyData.py
```

```
*************************************
Welcome to the Dummy Data tool :
 This tool gives you the option of creating dummy data and run
multiple jobs on it so that UI can be seen with dummy data
 Once dummy data is populated, this tool also gives the option of
cleaning up the dummy data and revert the system to its original
state
 Input required : /opt/etc/scripts/DummyData/config.txt
 Logs Path: /opt/etc/scripts/DummyData/log.txt
 *************************************
Enter valid input from below options :
 1. Run complete pipe
 2. Cleanup the system
 3. Exit
2
Have you entered the proper values in config.txt file . Enter yes
or no
yes
The Cleanup will do the following things :
========================================================
 1. It will delete all folder except /data/CoreJob/config from
directory /data/CoreJob on local file system from all namenodes
  2. It will delete the /data/ib/ contents on local file system
from all servers
 3. It will delete the contents of /data/atlasData/ on local file
system from rubix servers
 4. It will remove data from /data/collector1/bulkstats_files_
backup/GMPLAB4/, /data/collector/bulkstats_files_backup/DC1/,
/data/collector/bulkstats_files_backup/DC2/,
/data/collector/bulkstats_files_backup/DC1/,
/data/collector/bulkstats_files_backup/GMPLAB1/,
/data/collector/bulkstats_files_backup/delhi/,
/data/collector/bulkstats_files_backup/DC2/,
```
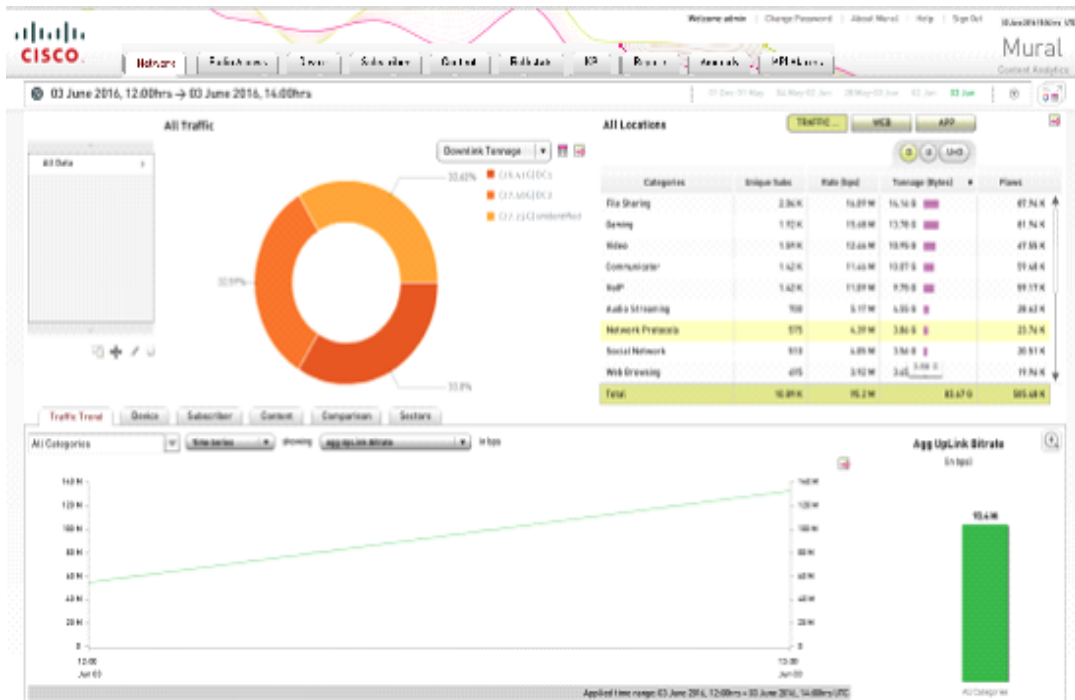
```
/data/collector/bulkstats_files_backup/DC1/,
/data/collector/bulkstats_files_backup/DC2/,
/data/collector/bulkstats_files_backup/DC2/,
/data/collector/bulkstats_files_backup/DC1/,
/data/collector/bulkstats_files_backup/DC2/,
/data/collector1/edrflow_backup, /data/collector1/edrhttp_backup on
local file system from all namenodes
 5. It will remove the hdfs contents from the path : /IB/,
/local/IB/,/data/ib/, /data/output/
 6. It will remove the hdfs content from the path:
/data/CoreJob,/data/EDR_EDRSubcrBytesAgg_lastSuccess,/data/EDR_
SubscriberDeviceMPH_lastSuccess,/data/EDR_EDRSubcrDev_
lastSuccess,/data/EDR_EDRTopN_lastSuccess
 7. It will remove hdfs content from the path
/data/TEST/edrhttp/,/data/TEST/edrflow/ and /data/TEST/bulkStats/
(if bulkstat is enabled)
 8. It will remove diskcache (/data/diskstore/UI-EDR) from all
rubix servers
 9. It will remove bulkstats cache (/data/diskstore/bulkstats-
<hostname>) from all rge nodes (applicable only if bulkstat is
enabled)
=========================================================


Do you want to proceed : yes or no yes



Welcome to Mural 3.9
resolved_active.xml                                     100% 1618KB
1.6MB/s   1.6MB/s   00:00


Please check /opt/etc/scripts/DummyData/log.txt for logs
Cleanup finished Successfully
```

8. File Pattern Regular Expression

The topic provides information about the regular expression for the EDR and collector files.

The following list provides detailed description for the using the regular expression for EDR and Collector file names:

- Ensure to provide different input paths for each new gateway being added.
- The /edr-file-path and /bulkstat-file-path should always start with a forward slash (/).
- The ASR should send the gateway name in place of %DC, as specified in the file name pattern in the Collector configurations.
- If the filenames will have file extensions of .gz or .txt, then you must provide ".*" in the file name format configuration when adding the gateway.
- All incoming files should contain the string as per their type in the file name; that is, flow EDR files should contain the string "flow" delimited by an underscore (_) or hyphen (-) and similarly HTTP EDR files must contain string "http" delimited by an underscore (_) or hyphen (-) (so combinations would also work , like "_flow-" or "–http_").
- All collector (internal) IPs can be provided with comma (,) separated values like 10.10.10.133,10.10.10.134

Following are the guidelines and samples to help in configuring filename patterns and collector configurations:

MURAL Software Standard Installation Guide for Rack Servers

FileName e pattern	Example	TimeStamp p	Regex for EDR filename	Regex for collector Filename
Gateway-name_ (multiple strings separate-d by undersco-re or hyphen or both)_ flow_ timestam-p_str4.gz	Gateway-name_ str1_ str2_ str3_ flow_ timestam-p_str4.gz	MMDDYYYY-hhmmss	*_*_ *_%MM%DD%YYYY%-hh%mm%ss_*.gz	%DC_*_ *_%MM%DD%YYYY%hh-%mm%ss*.gz
Gateway-name_ (multiple strings separate-d by undersco-re or hyphen or both)_ flow_ timestam-p_str4_ str5.gz	Gateway-name_ str1_ str2_ str3_ flow_ timestam-p_str4_ str5.gz	MMDDYYYY-hhmmss	*_*_ *_%MM%DD%YYYY%-hh%mm%ss_*_*.gz	%DC_*_ *_%MM%DD%YYYY%hh-%mm%ss*.gz

76   Copyright

FileName pattern	Example	TimeStamp	Regex for EDR filename	Regex for collector Filename
Gateway-name_ (multiple strings separated by underscore or hyphen or both)_ flow_ string_ timestamp_ string.gz	Gateway-name_ str1_ str2_ str3_ flow_ str4_ timestamp_str5.gz	MMDDYYYY-hhmmss	*_*_*_ *_%MM%DD%YYYY%-hh%mm%ss_*.gz	%DC_*_*_ *_%MM%DD%YYYY%hh-%mm%ss*.gz
Gateway-name_ (multiple strings separated by underscore or hyphen or both)_ flow_ string_ timestamp_string_ string.gz	Gateway-name_ str1_ str2_ str3_ flow_ str4_ timestamp_str5_ str6.gz	MMDDYYYY-hhmmss	*_*_*_ *_%MM%DD%YYYY%-hh%mm%ss_*_*.gz	%DC_*_*_ *_%MM%DD%YYYY%hh-%mm%ss*.gz

FileName e pattern	Example	TimeStam p	Regex for EDR filename	Regex for collector Filename
Gateway-name_ (multiple strings separate-d by undersco-re or hyphen or both)_ flow_ string_ timestam-p_string_ string.gz	Gateway-name_ str1_ str2_ str3_ flow_ str4_ timestam-p_str5_ str6.gz	YYYYMMDD-hhmmss	*_*_*_ *_%YYYY%MM%DD%-hh%mm%ss_*_*.gz	%DC_*_*_ *_%YYYY%MM%DD%hh-%mm%ss*.gz
Gateway-name_ (multiple strings separate-d by undersco-re or hyphen or both)_ flow-string_ timestam-p_string_ string.gz	Gateway-name_ str1_ str2_ str3_ flow-str4_ timestam-p_str5_ str6.gz	MMDDYYYY-hhmmss	*_*_*-*_%MM%DD%YYYY%-hh%mm%ss_*_*.gz	%DC_*_*-*_%MM%DD%YYYY%hh-%mm%ss*.gz

- It is mandatory to send gateway name as the first substring in EDR files, followed by an underscore (_) character.
- 'flow' can be replaced with 'http'.
- If timestamp is in MMDDYYYYhhmmss then set %MM%DD%YYYY%hh%mm%ss in Regex pattern.
- If timestamp is in YYYYMMDDhhmmss then set %YYYY%MM%DD%hh%mm%ss in Regex pattern.

9. Mandatory Parameters for Incoming ASR Files

The following is the list of mandatory headers that need to be present in files coming from the ASR so that the MURAL system can deduce meaningful information.

Note: MURAL recommends ASR files to be of 10MB compressed size for optimum performance.

9.1 Mandatory Attributes for Flow EDRs for MURAL

Flow EDR data sent by the ASR platform to the MURAL system must contain the following attributes:

- flow-end-time

- flow-start-time

- radius-calling-station-id

- sn-app-protocol

- p2p-protocol

- sn-server-port

- sn-volume-amt-ip-bytes-downlink

- sn-volume-amt-ip-pkts-uplink

- sn-volume-amt-ip-pkts-downlink

- sn-volume-amt-ip-bytes-uplink

- tcp-os-signature

- bearere-3gpp imei

Sample:

```
#sn-start-time,sn-end-time,radius-calling-station-id,sn-server-
port,sn-app-protocol,sn-volume-amt-ip-bytes-uplink,sn-volume-amt-
ip-bytes-downlink,sn-volume-amt-ip-pkts-uplink,sn-volume-amt-ip-
pkts-downlink,p2p-protocol,ip-server-ip-address,bearer-3gpp rat-
```

```
type,voip-duration,sn-direction,traffic-type,bearer-3gpp
imei,bearer-3gpp sgsn-address,bearer-ggsn-address,sn-flow-end-
time,sn-flow-start-time,radius-called-station-id,bearer-3gpp user-
location-information,sn-subscriber-port,ip-protocol,sn-
rulebase,tcp-os-signature,bearer-3gpp charging-id

1381518310,1381518337,1000000018,70000,29,20000,20000,182,36,iax,27
.9.126.155,2,1,FromMobile,,,2.2.2.1,27.23.157.2,1381518337,13815183
10,Sushfone-2,231-10-1073-10065,43769,1985,rb31,,2
```

9.2 Mandatory HTTP EDR Attributes for MURAL

HTTP EDR data sent to the MURAL system must contain the following attributes:

- sn-start-time

- sn-end-time

- transaction-downlink-packets

- transaction-uplink-packets

- transaction-downlink-bytes

- transaction-uplink-bytes

- http-content type

- radius-calling-station-id

- http-User-Agent

- http-URL

- http-host

- http-reply code

- tcp-os-signature

- bearer-3gpp imei

Sample:

```
#sn-start-time,sn-end-time,radius-calling-station-id,transaction-
uplink-bytes,transaction-downlink-bytes,ip-subscriber-ip-
address,ip-server-ip-address,http-host,http-content type,http-
url,voip-duration,traffic-type,transaction-downlink-
packets,transaction-uplink-packets,bearer-3gpp rat-type,radius-
called-station-id,tcp-os-signature,bearer-3gpp imei,http-request
method,http-reply code,http-user-agent


1381518310,1381518338,1000000019,15000,15000,1.1.1.1,27.2.248.155,i
mages.craigslist.org,image/png,images.craigslist.org,11,,60,1,1,Sus
hfone-1,,,GET,506 Variant Also Negotiates,"Dalvik/1.6.0 (Linux; U;
Android 4.0.3; Galaxy Nexus Build/ICL53F)"
```

9.3 ASR-Side Configuration

The corresponding configuration on the side of the ASR platform is as follows:

```
     edr-format edr-flow-format
attribute sn-start-time format seconds priority 10
attribute sn-end-time format seconds priority 20
attribute radius-calling-station-id priority 30
rule-variable bearer 3gpp imsi priority 35
attribute radius-called-station-id priority 40
attribute sn-volume-amt ip bytes uplink priority 50
attribute sn-volume-amt ip bytes downlink priority 60
attribute sn-volume-amt ip pkts uplink priority 70
attribute sn-volume-amt ip pkts downlink priority 80
rule-variable bearer 3gpp imei priority 90
rule-variable bearer 3gpp rat-type priority 100
rule-variable p2p protocol priority 110
attribute sn-app-protocol priority 120
attribute sn-parent-protocol priority 130
rule-variable ip protocol priority 140
rule-variable traffic-type priority 150
attribute sn-direction priority 160
```

```
rule-variable ip server-ip-address priority 170
attribute sn-server-port priority 180
rule-variable ip subscriber-ip-address priority 190
attribute sn-subscriber-port priority 200
rule-variable bearer 3gpp sgsn-address priority 210
rule-variable bearer ggsn-address priority 220
rule-variable bearer 3gpp user-location-information priority 230
rule-variable bearer 3gpp2 bsid priority 240
attribute sn-flow-start-time format seconds priority 260
attribute sn-flow-end-time format seconds priority 270
rule-variable tcp os-signature priority 290
rule-variable tethered priority 300
attribute sn-rulebase priority 310
#exit

edr-format edr-http-format
attribute  sn-start-time format seconds priority 10
attribute  sn-end-time format seconds priority 20
attribute radius-calling-station-id priority 30
attribute radius-called-station-id priority 40
rule-variable http host  priority 70
rule-variable http content type  priority 80
attribute transaction-downlink-bytes priority 90
attribute transaction-uplink-bytes priority 100
attribute transaction-downlink-packets priority 110
attribute transaction-uplink-packets priority 120
rule-variable bearer 3gpp imei  priority 130
rule-variable bearer 3gpp rat-type  priority 140
rule-variable http reply code priority 150
rule-variable http url priority 160
rule-variable http referer priority 170
rule-variable http user-agent priority 180
#exit
```

10. Understanding the CIQ Sheet

Customer Information Questionnaire (CIQ) sheet provides key information related to configuration of the network and storage. CIQ sheet serves as a reference document for the various values related to network, clusters, nodes and storage. The CIQ sheet is has following tabs:

- "MiscInfo" below
- "NetworkInfo" on the next page
- "ClusterInfo" on page 86
- "NodeInfo" on page 86
- "StorageInfo" on page 87

10.1 MiscInfo

The MiscInfo tab provides miscellaneous information, for example site name. The information provided in this tab is used for providing the names of various XMLs generated during the installation process.

The following table describes fields in the MiscInfo tab.

Field Name	Description	Example
XML File Name	Name of the XML file to be created by system after uploading the CIQ sheet.	Mural_Hardware_ StandardPack.xml
Site Name	Name of the site.	InterNap
HW Type	The type of hardware used at the site.	Cisco
Banner	The welcome text to be displayed on the console	Welcome to Mural Standard Pack
Set GSM as NTP	Whether to use GSM server as NTP server.	FALSE
CIQ Template Version	The current version of the CIQ sheet.	1.0

10.2 NetworkInfo

The NetworkInfo tab provides information related to various network parameters that are required to be configured. The network information is categorized into VLAN, UCS Management Interfaces and Misc Network.

The following table describes fields in the NetworkInfo tab.

Field Name	Description	Example
VLAN		
Network Name	Name of network, whether public or management.	MgmtNW
Interface	The name of the interface.	eth0
Network Type	The type of network, whether normal or management.	Mgmt/Data
Network Prefix	The network prefix for the IP.	192.168.188.0
Subnet	The subnet or the value of subnet.	24
Vlan ID	The Vlan ID.	Not in use
No of Addresses	Number of IP addresses to be used.	Not in use
Remark	The cell is provided to be used for providing and additional information.	Not in use
UCS Management Interfaces		
Fabric A IP	Fabric IP of interface A.	Not in use
Fabric B IP	Fabric IP of interface B.	Not in use
Cluster VIP	UCS manager IP.	17.10.10.9
Domain	domain name used for login to UCS.	native
Username	username to login to the UCS manager.	username
Password	Password to login to the UCS manager.	password
Misc Network		
Default Gateway	The default gateway IP.	172.30.0.1
DNS	The DNS IP.	103.14.2.35
NTP IP	NTP server IP.	103.14.2.37
NTP Version	The version of the NTP.	4

Field Name	Description	Example
SNMP Community Name	Name of the SNMP used.	cisco
SNMP Receiver IP	The IP of the SNMP server.	172.30.17.75
SNMP Port	The port of the SNMP server.	162

10.3 ClusterInfo

The ClusterInfo tab provides information related to various clusters to be created on the hardware. It provides information such as Clustername, Cluster interface, cluster VIP and so on.

The following table describes fields in the ClusterInfo tab.

Field Name	Description	Example
Cluster Name	Name of the cluster.	cmp-clu
Cluster Type	Type of the cluster setup.	1+1 HA
Cluster Interface	Name of the interface connection for the cluster.	eth0
Eth0 VIP	VIP of eth0 interface.	192.168.188.75
Eth1 VIP	VIP of eth1 interface.	172.30.17.75
Eth2 VIP	VIP of the eth2 interface.	172.30.17.67

10.4 NodeInfo

The NodeInfo provides detailed information related to each node in the system. It provides information like chassis number, slot number, hostname, cluster name etc. These information are required to uniquely identify each node present in the system.

The following table describes fields in the NodeInfo tab.

Field Name	Description	Example
Chassis Number	The chassis number in which serverin installed.	c1
Slot Number	The slot number where server is installed.	1
Hostname	The host name of the sever.	gms-1
Eth0 MAC	The MAC address of the eth0 interface.	00:A0:D7:42:08:7E
Eth0 IP	The MAC IP of the eth0 interface.	192.168.188.66
Eth1 MAC	The MAC address of the eth1 interface.	00:A0:D7:42:08:6E
Eth1 IP	The MAC IP of the eth1 interface.	172.30.17.66
Eth2 MAC	The MAC address of the eth2 interface.	00:A0:D7:42:08:7E
Eth2 IP	The MAC IP of the eth2 interface.	172.30.17.98
KVM IP	The IP of the KVM assiciated with the server.	17.18.10.0
Cluster Name	The name of the cluster.	gms-clu

10.5 StorageInfo

The StorageInfo provides information related to various parameters of the storage. The storage information provided are classified into two storage information, which provides general storage information and EMC interfaces, which provides information related to the EMC interfaces for the storage.

The following table describes fields in the StorgaeInfo tab.

Field Name	Description	Example
Storage Information		
Hostname	Host name of the server.	gms-1
WWID	The WWID of the storage associated with the server.	36006016085003600ead331eb56f7e311
Component	The name of the components using the assigned storage.	PGSQL

Field Name	Description	Example
LUN name	The name of LUN associated storage.	only for information
LUN size (GB)	The LUN size of associated storage	only for information
RAID type	The RAID type used for storage.	only for information
EMC Interfaces		
EMC a	EMC interface a.	not in use
EMC b	EMC interface b.	not in use
username	Username to connect with EMC interface.	not in use
password	Password corresponding to the username to connect with the EMC interface.	not in use