MURAL Software Standard Installation Guide for Rack Servers

Version 3.6
MURAL Software Standard Installation Guide for Rack Servers

Published: 2015-10-12

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
THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB’s public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT WILL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY LOST REVENUE, PROFIT, OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, EXEMPLARY, OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco’s trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

MURAL Software Standard Installation Guide for Rack Servers

Copyright © All rights reserved.
# Table of Contents

**Installation Overview** ................................................................. 7

Before You Begin ........................................................................... 7

Installation Package ......................................................................... 8

Customer Information Questionnaire ........................................... 8

System Components ......................................................................... 10

Hardware Topology ......................................................................... 12

Installing Rack Servers ................................................................. 13

**Configuring and Managing the VM** ............................................. 15

Prerequisite ....................................................................................... 15

Creating VMs Using OVF Template .................................................. 15

Copying VMDK Files to VMware Host Machine ............................. 17

Attaching VMDK Images to VM ......................................................... 18

Adding Storage Disks to VM from VMWare Datastore ................. 21

Setting VM Start/Stop with Host ....................................................... 25

Assigning IP Addresses to VMs ......................................................... 27

Sharing SSH Key from Master GCR Node ....................................... 28

Collecting MAC IDs from All Nodes ................................................. 29

**Configuring the Master GMS Node** ............................................ 30

Applying Patches Manually on GMS and Other Nodes ................. 30

Setting up the GMS Server ............................................................... 30

Generating the Final XML ............................................................... 30

**Applying Configuration on MURAL Nodes** ............................... 36

Using the install appliance Command ........................................... 36
Site Specific Applications Configuration .................................................. 37
Troubleshooting Node Installation .......................................................... 37
Verifying the Status of Processes .............................................................. 37
Generating and Pushing IBs ....................................................................... 40
Configuring DCs and Gateways For All IBs .............................................. 41

**Processing the Data** ............................................................................. 49
Setting Up a New User for the ASR in the Collectors ................................. 49
Ingesting Data Into the System ................................................................. 49

**Validating Data on Nodes** ................................................................. 50
Validating Data on the Collector Nodes ..................................................... 50
Validating Data ....................................................................................... 51
Starting UI Processes and Verifying Data .................................................. 52
Updating Whitelists .................................................................................. 54

**Setting Up Offline Reports** ............................................................... 56
Uncategorized URL, UA, and TAC Reports ............................................... 56
Tethering Databases ................................................................................ 56
Tethering Reports .................................................................................... 58
APNTethering Reports ............................................................................ 59
Updating Whitelists ................................................................................ 60

**Mandatory Parameters for Incoming ASR Files** ............................... 62
Mandatory Attributes for Flow EDRs for MURAL ..................................... 62
Mandatory HTTP EDR Attributes for MURAL ......................................... 63
ASR-Side Configuration .......................................................................... 64

**Glossary** .......................................................................................... 66
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.

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.

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)

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
**System Components**

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

![MURAL System Components Diagram](image)

**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.

**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
MURAL Software Standard Installation Guide for Rack Servers

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.

![Diagram of MURAL system topology on two rack servers]

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).

**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 15.
MURAL Software Standard Installation Guide for Rack Servers

- 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 30.
   - 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 Configuration on MURAL Nodes" on page 36.

5. Configure environmental parameters specific to your environment.
   - Generate and Push IBs in "Applying Configuration on MURAL Nodes" on page 36
   - "Processing the Data " on page 1

6. Verify that the system is working as expected. See "Validating Data on Nodes" on page 50 and "Setting Up Offline Reports" on page 56.
Configuring and Managing the VM

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*.

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.

<table>
<thead>
<tr>
<th>Guest OS</th>
<th>CentOS 4/5/6 (64-bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM Version</td>
<td>8</td>
</tr>
<tr>
<td>CPU</td>
<td>20 vCPU</td>
</tr>
<tr>
<td>Memory</td>
<td>90112 MB</td>
</tr>
<tr>
<td>Memory Overhead</td>
<td>667.98 MB</td>
</tr>
<tr>
<td>VMware Tools</td>
<td>Not running (Not installed)</td>
</tr>
<tr>
<td>IP Addresses:</td>
<td></td>
</tr>
<tr>
<td>DNS Name:</td>
<td>Powered On</td>
</tr>
<tr>
<td>State:</td>
<td>On</td>
</tr>
<tr>
<td>Host:</td>
<td>localhost.localdomain</td>
</tr>
<tr>
<td>Active Tasks:</td>
<td></td>
</tr>
<tr>
<td>vSphere HA Protection:</td>
<td>N/A</td>
</tr>
</tbody>
</table>

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

**Copying VMDK Files to VMware Host Machine**

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

   ```bash
   # 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.

   ```bash
   $ 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.

**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.

**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).

<table>
<thead>
<tr>
<th>VM Role</th>
<th>LUN Size (GB)</th>
<th>SCSI ID</th>
<th>Block ID</th>
<th>Mount Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCR</td>
<td>1024</td>
<td>0:0</td>
<td>/dev/sda</td>
<td>/data/collector</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>0:1</td>
<td>/dev/sdb</td>
<td>/data/pgsql</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>0:2</td>
<td>/dev/sdc</td>
<td>/data/diskstore</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0:3</td>
<td>/dev/sdd</td>
<td>/data/rge</td>
</tr>
<tr>
<td>INSTA-DB</td>
<td>200</td>
<td>0:0</td>
<td>/dev/sda</td>
<td>/data/pgsql</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>0:1</td>
<td></td>
<td>dbroot1</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>0:2</td>
<td></td>
<td>dbroot2</td>
</tr>
<tr>
<td>DN</td>
<td>1024</td>
<td>0:0</td>
<td>/dev/sda</td>
<td>/data/hadoop-admin</td>
</tr>
</tbody>
</table>
Note: Size of the LUNs can be different as per site-specific configuration, please refer to the sizing sheet.


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 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.

**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 needed 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.

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 eth0 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
   ```

### 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>
```

**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.
Configuring the Master GMS Node

Applying Patches Manually on GMS and Other Nodes

Download the patches from the FTP server to the master GCR server in the /data/tmp_patch directory. Apply all patches on this node.

*Note:* See the *MURAL Release Notes* for software version 3.6 for a complete list of patches and installation instructions.

Node will reboot after successfully applying the patches.

Setting up the GMS Server

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
```

Generating the Final XML

To generate the final XML, perform the following steps:

1. Log in to the GMS node using ssh and run the following commands:

```
# cd /opt/etc/scripts/mmi/CIQ2GMSXML/src/
# python GMSXMLGenerator.py -f /data/Mural_CIQ_Template.xls
GMS HARDWARE XML saved as /data/configs/gms/Mural_Hardware.xml
************************************************************
SUCCESSFULLY COMPLETED GMS HARDWARE XML GENERATION
************************************************************
```

Copyright
The XML is created in the `/data/configs/gms/` folder. Execute the following command:

```bash
# ls -l /data/configs/gms/Mural_Hardware.xml
-rw-r--r-- 1 admin root 27352 Feb 23 11:16 /data/configs/gms/Mural_Hardware.xml
```

2. Check the errors in the `ciq2gmsxml.log` log file at `/data/mmi/`.

3. From the command terminal, launch Cisco Mural Deployment wizard to select the pack type, additional applications, cluster components and specify site-specific parameters.

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

General instructions to use the Cisco Mural Deployment wizard:

- Use arrow keys to navigate into wizard screen
- Use Space-bar to choose the selected item(s)
- Press Enter key after selecting Next, Previous or Exit to move into next screen, previous screen or exit from the MMI screen, respectively.
- Selected items appear with swapped colors for text and background.

By default, MURAL installs Content Analytics (CA) application. Additionally, you can also purchase the following applications to be installed. These applications have impact on resources that are required for deployment and should be enabled only if the deployment was sized for them:

- HTTP Errors
- Bulkstats and KPI
- Anomaly
- Tethering
MURAL nodes further classified and install together in a particular cluster to host specific application component. Following table shows cluster configurations for Starter Pack installation:

<table>
<thead>
<tr>
<th>Installation Type</th>
<th>GMS Cluster</th>
<th>Collector Cluster</th>
<th>UI Cluster</th>
<th>iNSTA Cluster</th>
<th>DN Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starter Pack</td>
<td>GCU (GMS + Namenode and Collector + UI)</td>
<td></td>
<td>iNSTA</td>
<td></td>
<td>Compute</td>
</tr>
</tbody>
</table>

Update global variables for application configuration parameters for site-specific requirements. The following table lists samples of such parameters. Ensure that you update the global variables specific to your site.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Meaning</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>adaptor.bulkStats.numThreads:</td>
<td>Keep 2 threads for Bulkstats and other two equal sets for flow and http, totaling 75% of collector cores.</td>
<td>2</td>
</tr>
<tr>
<td>adaptor.edrflow.numThreads:</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>adaptor.edrhttp.numThreads:</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>application.atlas.cachePersistToDisk</td>
<td>true - enable disk based caching for atlas and BS; false - disable</td>
<td>false</td>
</tr>
<tr>
<td>application.bulkstats.cachePersistToDisk</td>
<td></td>
<td>false</td>
</tr>
<tr>
<td>application.atlas.rubixInstance.1.tomcatInstanceMaxSize</td>
<td>Maximum and initial java memory for atlas process (should be &lt;= 50% of Rubix node RAM)</td>
<td>As per sizing sheet</td>
</tr>
<tr>
<td>application.atlas.rubixInstance.1.initialJavaHeapSize:</td>
<td></td>
<td>As per sizing sheet</td>
</tr>
<tr>
<td>Property Name</td>
<td>Meaning</td>
<td>Default Value</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>application.ruleEngine.rubixInstance.1.tomcatInstanceMaxSize</td>
<td>Maximum and initial java memory for ruleEngine process (keep it as default)</td>
<td>5g</td>
</tr>
<tr>
<td>application.ruleEngine.rubixInstance.1.initialJavaHeapSize</td>
<td></td>
<td></td>
</tr>
<tr>
<td>application.bulkstats.rubixInstance.1.tomcatInstanceMaxSize</td>
<td>Maximum and initial java memory for bulkstat process (should be &lt;= 10% of Rubix node RAM)</td>
<td>20g</td>
</tr>
<tr>
<td>application.bulkstats.rubixInstance.1.initialJavaHeapSize</td>
<td></td>
<td>20g</td>
</tr>
<tr>
<td>application.reportAtlas.rubixInstance.1.tomcatInstanceMaxSize</td>
<td>Maximum and initial java memory for offline report process (keep it as default)</td>
<td>10g</td>
</tr>
<tr>
<td>application.reportAtlas.rubixInstance.1.initialJavaHeapSize</td>
<td></td>
<td>5g</td>
</tr>
<tr>
<td>application.rge.rubixInstance.1.tomcatInstanceMaxSize</td>
<td>Maximum and initial java memory for report engine process (keep it as default)</td>
<td>10g</td>
</tr>
<tr>
<td>application.rge.rubixInstance.1.initialJavaHeapSize</td>
<td></td>
<td>10g</td>
</tr>
<tr>
<td>Property Name</td>
<td>Meaning</td>
<td>Default Value</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>application.httperror.rubixInstance.1.tomcatInstanceMaxSize</td>
<td>Maximum and initial java memory for http error process (should be &lt;= 10% of Rubix node RAM)</td>
<td>20g</td>
</tr>
<tr>
<td>application.httperror.rubixInstance.1.initialJavaHeapSize</td>
<td></td>
<td>20g</td>
</tr>
<tr>
<td>application.launcher.rubixInstance.1.tomcatInstanceMaxSize</td>
<td>Maximum and initial java memory for launcher process (keep it as default)</td>
<td>1g</td>
</tr>
<tr>
<td>application.launcher.rubixInstance.1.initialJavaHeapSize</td>
<td></td>
<td>1g</td>
</tr>
<tr>
<td>Timezone:</td>
<td>Sets the timezone for MR jobs and UI</td>
<td>UTC</td>
</tr>
<tr>
<td>FQDN:</td>
<td>Sets URL to access the MURAL UI</td>
<td>ucsd.cisco.com</td>
</tr>
<tr>
<td>rgemailSupport:</td>
<td><a href="mailto:dummy-support@cisco.com">dummy-support@cisco.com</a></td>
<td>Default</td>
</tr>
<tr>
<td>mailSender:</td>
<td>Sets the E-mail ID as sender of offline reports.</td>
<td><a href="mailto:admin@cisco.com">admin@cisco.com</a></td>
</tr>
<tr>
<td>mailHost:</td>
<td>Sets SMTP Server host address</td>
<td>mx1.cisco.com</td>
</tr>
<tr>
<td>mailPort:</td>
<td>Sets SMTP Server Port</td>
<td>25</td>
</tr>
</tbody>
</table>
After successful execution, the final XML is generated as illustrated in the following message:

```
Final XML successfully generated (/config/gms/Mural_Application.xml)
#
```
Applying Configuration on MURAL Nodes

Using the `install appliance` Command

To install appliance on nodes:

1. SSH to the GMS server using the management IP address and activate the XML:

   ```
   > en
   # config
   (config) # gms config Mural_Application.xml activate
   File successfully activated
   (config) #
   ```

   **Note:** If you make any changes to the XML file, for example during troubleshooting, you must run this command again to activate the changed file.

Run the `install appliance` command for various clusters in order as specified in the following table. For successful installation, the `status` command displays the “Node successfully installed” message.

<table>
<thead>
<tr>
<th>Node</th>
<th>Install Command</th>
<th>Status Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master GCR</td>
<td>(config) # install appliance cluster cluster-name &lt;GCR Cluster&gt; node</td>
<td>(config) # install appliance show</td>
</tr>
<tr>
<td></td>
<td>&lt;Hostname of Master GCR Server&gt; force-format</td>
<td>installation-status cluster &lt;GCR Cluster&gt; node</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Hostname of Master GCR Server&gt;</td>
</tr>
<tr>
<td>Standby GCR</td>
<td>(config) # install appliance cluster cluster-name &lt;GCR Cluster&gt; node</td>
<td>(config) # install appliance show</td>
</tr>
<tr>
<td></td>
<td>&lt;Hostname of Standby GCR Server&gt; force-format</td>
<td>installation-status cluster &lt;GCR Cluster&gt; node</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Hostname of Standby GCR Server&gt;</td>
</tr>
</tbody>
</table>
### Site Specific Applications Configuration

1. In case Anomaly Detection feature is enabled, run the following command from both the master and the standby Name nodes:

   ```bash
   # pmx subshell oozie
   (pm extension)# set dataset anomalyMonthly attribute path /data/output/AnomalyAggDay/%Y/%M/%D/%H
   ```

### Troubleshooting Node Installation

If the installation fails at any point, contact Technical Support.

Logs can be collected on the GMS server from the location `/data/gms-server/logs`.

### Verifying the Status of Processes

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

**Note:** After you make any modifications, wait at least 15 minutes to ensure processes have had a chance to restart before running the following commands.
1. Log in to the master Namenode and verify that all Compute nodes have joined the HDFS cluster. The output shown in the following sample command sequence indicates correct configuration.

```bash
> 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_secondarnamenode
#
# cli -t "en" "conf t" "show pm process collector" | grep
status Current status: running
# hdfs dfsadmin -report 2>/dev/null | 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:

```bash
> en
(config) # _shell
```
3. Log in to the master Insta node, and run the following commands:

```bash
> en
# conf t
(config) # show pm process insta
```

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

```bash
(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 in to the standby Insta node and repeat the same commands shown in step 3.

5. Check Postgres process on both the Master and Standby nodes:

For the Standard Pack setup: GMS, Collector and iNSTA nodes

For the Standard Pack setup: GMS and iNSTA nodes

```bash
> en
# _shell
# ps -ef | grep postmaster | grep -v greppostgres
 2278  1  0 Feb13 ?    00:22:34
```
Generating and Pushing IBs

To configure the IBs for EDR:

1. Log in to the master Collector node, and update IBs from the image.

```
> en
# config (config) # pmx
pm extension> subshell aggregation_center
pm extension (aggregation center)> update all ibs from image
```

2. Now we need to add GGSN, SGSN, and APN information bases by running:

```
pm extension (aggregation center)> edit ib ib-file-name.map
add
IB IP: ib-IP-address
IB: ib-name
APN Group: group-name
```

Where `ib-file-name` is:

- ipGgsn
- ipSgsn
- apnGroup

Where the italicized words are replaced with the corresponding values from the table below for GGSN, SGSN, and APN information bases:

<table>
<thead>
<tr>
<th>IB</th>
<th>ib-file-name</th>
<th>ib-IP-address</th>
<th>ib-name</th>
<th>group-name</th>
</tr>
</thead>
<tbody>
<tr>
<td>GGSN</td>
<td>ipGgsn</td>
<td>27.23.157.1</td>
<td>GGSN1</td>
<td>(not applicable)</td>
</tr>
<tr>
<td>SGSN</td>
<td>ipSgsn</td>
<td>2.2.2.1</td>
<td>SGSN1</td>
<td>(not applicable)</td>
</tr>
<tr>
<td>APN</td>
<td>apnGroup</td>
<td>(not applicable)</td>
<td>Sushfone-1</td>
<td>Sushfone-1</td>
</tr>
</tbody>
</table>

3. Verify the IP addresses and names were successfully added by running:

```
pm extension (aggregation center)> show ib ib-file-name.map
```

Where `ib-file-name` is:
MURAL Software Standard Installation Guide for Rack Servers

- ipGgsn
- ipSgsn
- apnGroup

The resulting output may resemble:

<table>
<thead>
<tr>
<th></th>
<th>Address</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[27.23.157.1]</td>
<td>[GGSN1]</td>
</tr>
<tr>
<td>1</td>
<td>[2.2.2.1]</td>
<td>[SGSN1]</td>
</tr>
<tr>
<td>1</td>
<td>[Sushfone-1]</td>
<td>[Sushfone-1]</td>
</tr>
<tr>
<td>2</td>
<td>[Sushfone-2]</td>
<td>[Sushfone-2]</td>
</tr>
<tr>
<td>3</td>
<td>[Sushfone-3]</td>
<td>[Sushfone-3]</td>
</tr>
</tbody>
</table>

4. Exit the aggregation subshell by running `quit`.

To configure the IBs for BulkStats:

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

**Configuring DCs and Gateways For All IBs**

Guidelines for adding 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. We recommend EDR file size to be about 10MB, each.

- Ensure that you 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 have file extensions of .gz or .txt, then you must provide ".*" in the file name format configuration when adding the gateway.

- All collector (internal) IPs can be provided with comma (,) separated values such as 10.10.10.133,10.10.10.134.

- The following table lists some guidelines and samples to help in configuring filename patterns and collector configurations:

<table>
<thead>
<tr>
<th>Filename pattern</th>
<th>Example</th>
<th>Timestamp</th>
<th>Regex in Wrapper CLI</th>
<th>Regex in Collector Config</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway-name_ (multiple strings separated by underscore or hyphen or both) <em>flow</em> timestamp_str4.gz</td>
<td>Gateway-name_ str1_ str2_ str3_ flow_ timestamp_str4.gz</td>
<td>MMDDYYYY hhmmss</td>
<td><em>.</em>.*.<em>MM%DD %YYYY%hh%mm%ss_</em>.gz</td>
<td>%DC_.*_.<em>MM%DD%YYYY%hh%mm%ss</em>.gz</td>
</tr>
<tr>
<td>Filename pattern</td>
<td>Example</td>
<td>Timestamp</td>
<td>Regex in Wrapper CLI</td>
<td>Regex in Collector Config</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>-----------</td>
<td>----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Gateway name_ (multiple strings separated by underscore or hyphen or both)_flow_timestamp_string_timestamp.gz</td>
<td>Gateway name_ hhmmss</td>
<td>MMDDYYYY hhmmss</td>
<td><em>_</em><em>*.MM%DD%YYYY%hh%mm%ss</em><em>.</em>.gz</td>
<td>%DC_<em>_</em>.%MM%DD%YYYY%hh%mm%ss*.gz</td>
</tr>
<tr>
<td>Gateway name_ (multiple strings separated by underscore or hyphen or both)_flow_timestamp_string_timestamp.gz</td>
<td>Gateway name_ hhmmss</td>
<td>MMDDYYYY hhmmss</td>
<td><em>_</em><em>*..%MM%DD%YYYY%hh%mm%ss</em><em>.</em>.gz</td>
<td>%DC_<em>_</em>..%MM%DD%YYYY%hh%mm%ss*.gz</td>
</tr>
<tr>
<td>Filename pattern</td>
<td>Example</td>
<td>Timestamp</td>
<td>Regex in Wrapper CLI</td>
<td>Regex in Collector Config</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Gateway name_ (multiple strings separated by underscore or hyphen or both)</td>
<td>Gateway name_str1_str2_str3_flow_str4_timestamp_str5_str6.gz</td>
<td>MMDDYYYYhhmmss</td>
<td><em>.<em>.</em></em><em>.<em>%MM%DD%YYYY-%hh%mm%ss</em>.</em>_.*.gz</td>
<td>%DC_<em>.</em><em>*.</em>%MM%DD%YYYY%hh%mm%ss*.gz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><em>%DC</em><em>.</em><em>*.</em>%MM%DD%YYYY%hh%mm%ss*.gz</td>
<td></td>
</tr>
<tr>
<td>Filename pattern</td>
<td>Example</td>
<td>Timestamp</td>
<td>Regex in Wrapper CLI</td>
<td>Regex in Collector Config</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>-----------</td>
<td>----------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Gateway-name_</td>
<td>Gateway-name_</td>
<td>YYYYMMDD</td>
<td><em>*</em>*_</td>
<td>%DC_<em>_</em></td>
</tr>
<tr>
<td>(multiple strings separated by underscore or hyphen or both)</td>
<td>hhmmss</td>
<td>_%YYYY%MM%DD-</td>
<td>_%YYYY%MM%DD-</td>
<td><em>%YYYY%MM%DD-</em></td>
</tr>
<tr>
<td>flow_</td>
<td>flow_</td>
<td>%hh%mm%ss_*</td>
<td>%hh%mm%ss_*</td>
<td>hh%mm%ss*.gz</td>
</tr>
<tr>
<td>string_</td>
<td>string_</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>timestamp_</td>
<td>timestamp_</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>string.gz</td>
<td>string.gz</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Filename Pattern

<table>
<thead>
<tr>
<th>Gateway name (multiple strings separated by underscore or hyphen or both)</th>
<th>Gateway name_</th>
<th>Timestamp</th>
<th>Regex in Wrapper CLI</th>
<th>Regex in Collector Config</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway name_ str1_ str2_ str3_ flow-string_timestamp_string.gz</td>
<td>Gateway name_</td>
<td>MMDDYYYY hhmmss</td>
<td><em>_.</em><em>.*.</em>%MM%DD%YYYY%hh%mm%ss_* (*.gz</td>
<td>%DC_<em>.</em>_ .<em>%MM%DD%YYYY%hh%mm%ss</em>.gz</td>
</tr>
</tbody>
</table>

* It is mandatory to send gateway name as the first substring in EDR files, followed by an underscore (_) character. It can have 'http' in place of 'flow'.

To configure gateways:

1. Add gateway information for each gateway (refer to guidelines above for clarification about input parameters):

   pm extension> subshell aggregation_center
   pm extension (aggregation center)> add gateway name <gateway-name> region <region-name> location <location-name> schema_version <version> ip <Gateway IP address> timezone <ASR timezone> edr-filename-pattern <EDR filename pattern sent by ASR> bulkstat-filename-pattern <BS filename pattern sent by ASR> edr-collector-filename-pattern <EDR filename pattern in collector> bulkstat-collector-filename-pattern <BS filename>
pattern in collector> collector-ip <IP addresses of collector nodes> type HA edr-file-path <EDR files input dir> bulkstat-file-path <BS files input dir>

For example,

```
pm extension (aggregation center)> add gateway name pnst40ems5 region WEST location Pune schema_version 16 ip 10.132.69.121 timezone UTC edr-filename-pattern *_*_*_*%MM%DD%YYYY%hh%mm%ss_*_.gz bulkstat-filename-pattern *%YYY%MM%DD%hh%mm%ss edr-collector-filename-pattern %DC_*_*_*_*.gz bulkstat-collector-filename-pattern *%YYYY%MM%DD%hh%mm%ss collector-ip 10.10.10.101,10.10.10.102 type HA edr-file-path /GMPLAB1/edr/ bulkstat-file-path /GMPLAB1/bs/
```

2. Verify the new gateway has been added:

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

3. Verify the IP addresses for all Collectors:

```
pm extension (aggregation center)> show collector IPs
```

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:
7. Generate and push IBs on Bulkstats:

```
pm extension> subshell bulkstats
pm extension (bulk stats)> generate all ibs
pm extension (bulk stats)> push all ibs
```

8. Execute the following commands if the Anomaly Detection feature 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
```
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.

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

Setting Up a New User for the ASR in the Collectors

To set up a new user for the ASR in the Collectors:

1. Log on to the master Collector node and create the user:

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

   **Note:** The username and password should be the same ones configured on the ASR for EDR and BulkStats file transfers.

2. Repeat step 1 on the standby Collector node.

Ingesting Data Into the System

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 Nodes

Validating Data on the Collector Nodes

1. Log in to the master Collector node and go to _shell.

```
> en
# _shell
```

2. Run the indicated **hadoop** 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.

```
# hadoop dfs -ls /data/collector/1/output/edrflow/YYYY/MM/DD/HH/mm/* 2>/dev/null
# hadoop dfs -ls /data/collector/1/output/edrhttp/YYYY/MM/DD/HH/mm/* 2>/dev/null
# hadoop dfs -ls /data/collector/1/output/bulkStats/YYYY/MM/DD/HH/mm/* 2>/dev/null
```

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.

Setting the Data Start Time

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

1. Log in to the master GCR node and make the / file system writable.

```
# 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 "Validating Data on Nodes" above.
For example, if EDR and Bulkstats data starts coming into the Hadoop system from April 1, 2013, 06:00 onwards, run the following scripts with the start_time value as "2013-04-01T06:00Z":

```
# ./setOozieTime --dataStartTime 2013-04-01T06:00Z --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.

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

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

**Starting the Data Processing**

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.

**Validating Data**

To validate data ingestion into system before starting the UI application:

1. Log in to the master Collector or Name node and run the following command to validate collector data ingestion and availability in Hadoop:
MURAL Software Standard Installation Guide for Rack Servers

For EDR:

```bash
# hdfs dfs -du /data/collector/1/output/edr*/2015/02/22
```

For Bulkstat:

```bash
# hdfs dfs -du /data/collector/1/output/bulkStats/2015/02/22
```

2. Login into Master insta node and run the following command to validate EDR data availability in Insta database:

To display the oldest available Data Time:

```bash
# date -d@\`idbmysql database_mural -e "select mints from bin_metatable where aggregationinterval=-1:" | tail -1`
Fri Feb  6 06:00:00 UTC 2015
```

To display latest available Data Time:

```bash
# date -d@\`idbmysql database_mural -e "select maxts from bin_metatable where aggregationinterval=-1:" | tail -1`
Mon Feb 23 05:00:00 UTC 2015
```

Starting UI Processes and Verifying Data

Starting 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 UI 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 in to the master UI node.

   ```bash
   > en
   # conf t
   (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
```

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.

```
(config)# pm process rubix restart
(config)# rubix modify-app ApplicationName enable
(config)# rubix modify-app ApplicationName modify-instance 1 enable
```

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

- bulkstats
- reportAtlas
- rge
- ruleEngine (if Anomaly or BulkStat is enabled)
- httperror
- launcher (if HTTP Error is enabled)

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 applications are started properly without any exception.

4. Log in to the standby UI node and repeat steps 1 through 3.

5. Access the UIs by going to the URL `https://domain-name:21443/` through your browser.
MURAL Software Standard Installation Guide for Rack Servers

The domain name to be used is the one, which was provided as FQDN at the time of final XML creation using Cisco MURAL deployment wizard. For example:

<table>
<thead>
<tr>
<th><a href="https://demo.sanmateo.com:21443/">https://demo.sanmateo.com:21443/</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Username: admin</td>
</tr>
<tr>
<td>Password: admin123</td>
</tr>
</tbody>
</table>

**Note:** Certificate from a recognized certificate authority is required for accessing the UI. If certificate is not installed, type the following URLs for the BulkStats and RGE in your browser and accept certificates before opening the actual URL.

Visit the following ports and accept the certificates:

<table>
<thead>
<tr>
<th><a href="https://domainName:20443/">https://domainName:20443/</a></th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://domainName:30443/">https://domainName:30443/</a></td>
</tr>
</tbody>
</table>

For example:

<table>
<thead>
<tr>
<th><a href="https://demo.sanmateo.com:20443/">https://demo.sanmateo.com:20443/</a></th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://demo.sanmateo.com:30443/">https://demo.sanmateo.com:30443/</a></td>
</tr>
</tbody>
</table>

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

| https://demo.sanmateo.com:50443/ |

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

**Updating Whitelists**

After running MURAL system for 2-3 hours, run the following command from master GCR node to generate updated whitelists:

```bash
# 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
```
Observe the categorization in UI after 2 hours to see the effects of whitelists update.
Setting Up Offline Reports

Uncategorized URL, UA, and TAC Reports

Create a file named `serverFile_uncatReports` on the master GCR GCN Name 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 GCN Name 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 GCN Namer node.

Tethering Databases

Create a file called `serverFile_tethering` with details of the ASR gateways, where the TAC, OS or UA databases, created as a result of tethering processing,
need to be pushed and **offline_datatransfer_mapping_file** with format details of input and output files that will be transferred to ASR gateways.

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 ASR gateway IP address
- **gateway-username** is the username for logging into ASR Gateway
- **gateway-password** is the corresponding password to the username
- **location-to-copy-reports** is the location on the ASR Gateway machine where databases need to be copied

2. Log in to the master GCR GCN 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
```

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

5. Create the same file on the standby GCR GCN 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.
The offline_datatransfer_mapping_file contains format for original filename and for modified filename that has to be shared with ASR gateway.

The format of entry in this file is as follows:

```
<Input filename regex>:<output filename>
```

Here,

- `<Input filename regex>` is the regex for matching input filename
- `<Output filename>` is the name of the shared file.

Perform the following steps:

1. Go to the data directory:

```
# cd /data
```

2. Create offline_datatransfer_mapping_file with the following entries:

```
# vi /data/offline_datatransfer_mapping_file
os-db:new-os-db
tac-db:new-tac-db
ua-db:new-ua-db
```

Note: The delimiter in the file must be ":".

3. Create the same file on the standby Namenode as well.

**Tethering Reports**

Create a file called serverFile_tethering_subscribers_report with details of the destination server, where the subscriber reports created as a result of tethering processing, need to be pushed.

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

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

Where:

- *IP* is the destination host IP address
- *username* is the username for logging into the destination server
• password is the corresponding password to the destination server
• location-to-copy-reports is the location on the destination machine machine where databases need to be copied

Perform the following steps:

1. Log in to the master GCR GCN node:

   
   ```
   > en
   # _shell
   ```

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

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

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

   ```
   # cd work
   # vi /data/work/serverFile_tethering_subscribers_reports
   192.168.156.96, admin, admin@123, /data/tethering_subs
   ```

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

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

### APNTethering Reports

Create a file called `serverFile_tethering_ApnDc_report` with details of the destination server, where the subscriber reports created as a result of tethering processing, need to be pushed.

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

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

Where:
• *IP* is the destination host IP address
• *username* is the username for logging into the destination server
• *password* is the corresponding password to the destination server
• *location-to-copy-reports* is the location on the destination machine machine where databases need to be copied

Perform the following steps:

1. Log in to the master GCR GCN node:

   ```
   > en
   # _shell
   ```

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

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

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

   ```
   # cd work
   # vi /data/work/serverFile_tethering_ApnDc_reports
   192.168.156.96, admin, admin@123, /data/tethering_apn
   ```

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

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

**Updating Whitelists**

After running MURAL system for 2-3 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 2 hours to see the effects of whitelists update.
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 the optimum performance.

**Mandatory Attributes for Flow EDRs for MURAL**

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

- sn-flow-end-time
- sn-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
- bearer-3gpp imei
- tethered

**Sample:**

```
MURAL Software Standard Installation Guide for Rack Servers

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, , , , 2.2.1.27.23.157.2, 1381518337, 1381518310, Sushfone-2, 231-10-1073-10065, 43769, 1985, rb31, , , 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:**

```plaintext
```
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
Glossary

A

ASR Cisco ASR 5000 or ASR 5500 Series platforms. The routing platform that pushes the data flows to the MURAL system.

B

Big data A collection of data that is too large to manage, store, or analyze by traditional methods.

C

Caching node Hosts the Rubix engine and 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 Caching node is sometimes called the Rubix node. The Caching node uses N+1 redundancy in active-active mode.

CIQ An Excel spreadsheet containing a site survey that you completed before beginning the installation process.

Collector The cluster of nodes consisting of the Collector nodes in active/standby High Availability clustering.

Collector node Collects data streams pushed to the MURAL 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).

Compute cluster The cluster consisting of the master and standby Compute nodes.
Compute node  Analyzes and aggregates the data, creating *data cubes*. 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 implementation, and uses N+1 redundancy.

Cube engine  A process that is hosted on the UI/Caching node. The Cube engine forwards requests from the UI engine to the Insta node. It also prefetches data and locally caches it so that if the requested data is in the local cache, it can return the response directly without querying the Insta node.

D

Direct-Attached SAN  A feature on Cisco UCS that enables you to attach a fiber-channel SAN directly to the Fabric Interconnects.

Downlink Rate  The average bytes received by the mobile device from the Internet during a selected interval.

Downlink Tonnage  The total amount of data received by the mobile device from the Internet.

DPI  Deep packet inspection-an advanced form of packet filtering that makes it possible to examine more than just the packet header.

F

Fabric Interconnect  Part of the Cisco UCS platform that provides both network connectivity and management capabilities to all attached blades and chassis.

FC Switch Mode  Fibre channel switch mode, a mode on the Fabric Interconnects.

FCID  A 24-bit field used to route frames through a FC network.

flogi database  Fabric login database.

G
**GMS node**  General Guavus Management System node. This node provides centralized management of the MURAL platform nodes, such as remote manufacturing of blades, patch management, monitoring of all nodes and operations, and importing and running node configurations.

**H**

**Hadoop**  Open-source software that supports running applications on large clusters of hardware. See http://hadoop.apache.org/

**I**

**Image**  Comprised of the operating system plus the application bundle.

**Insta node**  Stores and manages the processed data in a columnar database, the Insta database, a columnar database that stores the processed data cubes, commonly three to six years' worth. The Insta node cluster has two servers with 1+1 redundancy.

**M**

**Manufacture**  Manufacturing a blade is the installing the MURAL operating system on the blade.

**MIBs**  A database used for managing the entities in a communications network; associated with the Simple Network Management Protocol (SNMP).

**MURAL**  Cisco Mobility Unified Reporting and Analytics (MURAL) application, which provides Web-based reporting and analytics abilities for deep packet inspection (DPI) data emerging from the network.

**mural.xml file**  Contains sample configuration settings for your setup, including a logical grouping of the nodes into two chassis and configuration settings for all of the system components. These settings are based on the details of your network that were supplied in the Customer Information Questionaire (CIQ).
MURAL Software Standard Installation Guide for Rack Servers

**NX-OS**  
Cisco Nexus Operating System

**O**  
Oozie  
A workflow scheduler designed by Apache to manage Hadoop jobs. Oozie is bundled on the system and hosted on the Collector nodes.

**P**  
PGSQL disks  
On a MURAL installation, these are storage partitions for the Postgres database and will store Rubix-related data.

**R**  
Rate (bps)  
Amount of data sent and received per second between the device and the Internet. (Bytes per Second – bps)

RG engine  
Report Generation engine, which serves as the HTTP request server. In Cisco MURAL, it is hosted on the same blade as the UI engine.

Rubix engine  
See Cube engine.

Rubix node  
See Caching node.

**S**  
SAN  
Storage area network. A high-speed network of storage devices that also connects those storage devices with servers. It provides block-level storage that can be accessed by the applications running on any networked servers.

**T**  
TAC  
Type Allocation Code – The initial 8 digit portion of the 15 digit IMEI code used to uniquely identify wireless devices.

Tonnage (MB)  
Total volume amount of data sent and received (Megabytes – MB)

**U**  
UCS  
UCS 5108 Blade Server, connected to UCS 6248 Fabric Interconnects, hosts the MURAL application.

UI node  
See Caching node.
**Uplink Rate**  The average bytes sent from the mobile device out to the Internet during a selected interval.

**Uplink Tonnage**  The total amount of data sent from the mobile device out to the Internet.

**V**

**vHBA Initiator Group**  vHBA initiator groups determine the fiber channel zoning configuration for all vHBAs in a service profile. Cisco UCS Manager does not include any default vHBA initiator groups. You must create vHBA initiator groups in any service profile that is to be assigned to servers included in a zone.

**VSAN**  Virtual storage area network (SAN). A collection of ports from a set of connected Fiber Channel switches that form a virtual fabric that interconnects hosts and storage devices. A VSAN allows you to have logically separate fabrics without having to have different physical fabrics.

**W**

**WWN**  World Wide Name. A unique identifier that is assigned to a manufacturer that is hard-coded into a Fiber Channel (FC) device. A unique 16-digit hexadecimal number, such as 21-00-00-30-D9-00-12-34. This identifier is hard-coded into every FC host bus adapter (HBA).

**WWNN**  World Wide Name Node. A world wide name assigned to a port in a Fibre Channel fabric, in this context, a port on a MURAL node.

**WWPN**  World Wide Name assigned to a port in a Fiber Channel fabric to uniquely identify it in the network.

**Z**

**Zoneset**  A container for one or more zones in the fabric. Zones need to be a member of a zoneset in order to be used. Only one zoneset can be active at one time in any given VSAN.