MURAL Software Standard
Installation Guide for Rack Servers

Version 3.4.2

Published: 2015-02-03

Copyright © 2015, Cisco Systems, Inc.
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 PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHTS ARISING FROM THE USE OF THE DOCUMENT FILES OR SOFTWARE. NO LICENSING RIGHTS, IF ANY, ARE GRANTED TO THE USER.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOSS PROFITS OR LOSS OF DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, 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 © 2015, Cisco Systems, Inc. All rights reserved.
# Table of Contents

**Installation Overview** ................................................................. 9  
Before You Begin ............................................................................. 9  
Installation Package ....................................................................... 10  
Customer Information Questionnaire .............................................. 10  
System Components ....................................................................... 11  
Hardware Topology ......................................................................... 13  
Installation Process Steps ............................................................... 14  

**Verifying UCS Hardware Configuration for MURAL** ..................... 17  

**Installing VMware OS on Rack Servers** ........................................ 18  
Installing VMware OS .................................................................... 18  
VMWare Host Management ............................................................. 30  

**Configuring and Managing the VM** .............................................. 35  
Creating VMs Using OVF Template .................................................. 35  
Copying VMDK Files to VMware Host Machine ............................... 37  
Attaching VMDK Images to VM ....................................................... 38  
Adding Storage Disks to VM from VMWare Datastore ....................... 40  
Setting VM Start/Stop with Host ..................................................... 44  
Assigning IP Addresses to VMs ....................................................... 46  
Sharing SSH Key from Master GCR Node ........................................ 47  
Collecting Mac IDs from All Nodes ................................................ 49  

**Configuring the Master GMS Node** ............................................. 50  
Apply Patches on Master GCR nodes .............................................. 50  
GMS Server setup .......................................................................... 50
## MURAL Software Standard Installation Guide for Rack Servers

Selecting Application Templates for GMS .............................................. 50

### Loading and Verifying Initial Configuration Settings ................................ 53

- Accessing the GMS User Interface .................................................... 53
- Loading the Base Configuration File ................................................. 54
- Verifying XML Settings .................................................................. 55
- Validating the XML Configuration .................................................. 78

### Using GMS Lite to Automate Installation of Nodes ................................. 79

- Applying Patches on MURAL nodes ............................................... 79
- Activating the XML File .................................................................. 79
- Installing GCR Nodes .................................................................... 79
- Repairing HDFS .............................................................................. 81
- Configuring Application with Site Specifics .................................... 82
- Troubleshooting Node Installation .................................................. 85
- Tuning the System for Performance .................................................. 85

### Verifying that Processes are Running .................................................. 86

### Generating and Pushing the Information Bases .................................... 88

- Configuring IBs for EDR ................................................................. 88
- Configuring DCs and Gateways For All IBs .................................... 90
- Copying IBs to the Anomaly Feature ............................................... 93
- Synchronize the IBs on the Standby GCR Node ................................. 93

### Processing the Data ........................................................................ 95

- Setting Up a New User for the ASR in the Collectors ......................... 95
- Ingesting Data Into the System ....................................................... 96

### Validating Data on Nodes ................................................................ 97
Validating Data on the Collector Nodes .................................................. 97
Validating Data on the Compute Blades (Data Nodes) ......................... 99
Validating Bulk Stats Data on the Insta Blade ................................ 101
Starting UI Processes and Verifying Data ........................................ 102
Updating Whitelists ........................................................................... 104

**Setting Up Offline Reports** ................................................................. 105
Uncategorized URL, UA, and TAC Reports ...................................... 105
Tethering Reports .................................................................................. 105
Rule-Based Reports .............................................................................. 107
Updating Whitelists .............................................................................. 108

**Mandatory Parameters for Incoming ASR Files** ......................... 109
Mandatory Attributes for Flow EDRs for MURAL .............................. 109
Mandatory HTTP EDR Attributes for MURAL .................................. 110
ASR-Side Configuration ........................................................................ 111

**Appendix I: Setting Up Virtual Drives on Rack Server** ............. 113

**Glossary** ......................................................................................... 120
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:

- 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. VMware OS Media (.ISO) - VMware-VMvisor-Installer-5.5.0.update01-1623387.x86_64.iso
   b. OVF template file - MasterVM.ovf
   c. 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.
   d. Any software patches that is available for the release. See the Release Notes for the appropriate release for a complete list of patches.

2. Sample GMS configuration file - muralRS.xml

   This sample configuration file provides 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 you supplied in the Customer Information Questionnaire (CIQ).

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:

- **Contacts**—Site personnel and their responsibilities
- **Space_Power Req**—Space and power requirements
- **IP Survey**—Physical network connections, virtual LANs (VLANs),
interfaces, slot assignments, Simple Network Management Protocol (SNMP) and Simple Mail Transfer Protocol (SMTP) settings, and so forth

- **Network Diagrams**—Physical connections between system components
- **Connectivity**—Details for ports and connections
- **Firewall**—Firewall changes required for connectivity
- **Alarms**—All supported SNMP traps
- **ASR**—Locations for required ASR information bases (IBs)

**System Components**

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

**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. GMS HA supported by GCR node cluster.
• **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.

![MURAL System Topology Diagram](image)

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).

**Installation Process Steps**

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. Verify that the UCS hardware is correctly set up and configured for the MURAL system. See "Verifying UCS Hardware Configuration for MURAL" on page 1

2. Prepare VMware host on UCS rack servers. See RAID Configuration for all
the hard drives:

- RAID Configuration for all the hard drives to create single virtual drive. See “Creating RAID Groups and the LUN for the Master GMS Node” on page 1.
- Install VMware OS into rack servers. See VMware OS installation
- VMware Host level configurations. See VMware Host Management

3. VM creation and configuration. See VM Configuration and Management.

- Create VMs using OVF template and attach images/storage and update CIQ sheet for Block/SCSI IDs. See Creation of VMs using OVF template.
- Set auto start/stop of VMs with Host machine. See Set VM start/stop with Host (Server).
- Power ON VMs and assign IP addresses. See Assign IP Addresses to the VMs.
- Share SSH key among all the VMs. See SSH key sharing from Master GCR node to all other nodes.
- Collect MAC IDs for all the VMs and update CIQ sheet. See Collect MAC IDs from all the node.

4. Install MURAL system from Master GCR node. See Configure Master GMS Server.

- On Master GCR node, apply patches and configure GMS Server. See Configure Master GMS Server.
- Verify and update GMS configuration file. See Defining the Deployment Topology Using GMS.
- Apply patches to all other nodes and activate XML. See Apply Patches on all other MURAL nodes.

5. Run Install Appliance for all the nodes, in order. See MURAL SYSTEM INSTALLTION.

6. Configure environmental parameters specific to your environment.
MURAL Software Standard Installation Guide for Rack Servers

- “Troubleshooting Node Installation” on page 72
- "Generate and Push the Information Bases" on page 74
- "Processing the Data " on page 78

7. Verify that the system is working as expected. See "Validating the System Installations" on page 79.
Verifying UCS Hardware Configuration for MURAL

The Cisco Unified Computing System (UCS) C240/C460 Rack Servers hosts the VMWare VMs (nodes) of the MURAL platform. This includes:

- UCS hardware configuration details to verify— Server Installations in rack, power supply, and network ports connectivity with switch for Management port as well as LAN ports (eth0/eth1).

- Initial configuration of the Rack Servers—Setting the CIMC configuration – admin password, setting up the management port IP address (DRAC IP Access)

- Both the rack servers can be accessible from laptop/PC oven LAN for CIMC login.

Before you begin, verify that you have all items listed in the bill of materials (BOM).
Installing VMware OS on Rack Servers

VMware ESXi 5.5.0 operating system software needs to install upon rack servers to configure them as VMware hosts. We need to configure the available hard disks under RAID-5 disk group and make this single virtual disk available to install the VMware ESXi OS. To configure storage that will be used for ESXi OS installation and Datastore for Virtual Machines (VMs), see APPENDIX-I: Virtual Drive Setup on Rack Servers.

From Web browser, access rack servers thru Cisco Integrated Management Controller (CIMC) WebUI, to manage and configure Server, Storage and start KVM console.

**Note:** Below steps are written for one rack server, please follow the same steps for another rack server as well to create all the required VMs for installation of MURAL system in dual rack server topology.

Installing VMware OS

To install the VMware image on the rack servers:

1. Log into CIMC using admin username and password:

2. Ensure vKVM properties below are enabled:
a. Select **Server** tab > **Remote Presence** > **Virtual KVM** tab and check the box for **Enabled** under vKVM Properties.

![Cisco Integrated Management Controller](image1.png)

b. Select the **Virtual Media** tab, check **Enabled** box under vKVM Console Based vMedia Properties.

![Remote Presence](image2.png)

c. Press **Save Changes** button at right bottom of the page.
d. From **Server** tab > **Summary** > **Launch KVM Console**.

![Server Summary](image)

e. Select **Virtual Media** tab from KVM Console screen.

![Virtual Media](image)

f. Go to **Add Image** the select an ISO image from client PC/Laptop and
3. From **Server** tab > **Summary** > **Power Cycle Server**. When prompted, press **OK** to reboot the server.

g. After successfully adding the image, check **Mapped** box.
When the screen is green, it means the Rack Server power cycle has started. Wait for boot options.

4. During power on process, above screen appears on console. Press **F6** to enter the Boot Menu.

5. From the Boot Menu, select **Cisco vKVM-Mapped vDVD1.22** as boot device and press **ENTER**.
The server starts booting from ESXi image.

6. As the server boots from the ESXi image, it loads:
a. ESXi installer packages

![ESXi installer packages](image1)

b. Library packages

![Library packages](image2)

7. After the server completes loading these packages, follow the below steps to install the OS.

a. Press **Enter** when this screen appears.

![Install screen](image3)

b. Press **F11** to accept license agreement and continue.
c. Select a local drive for OS installation and press **Enter** to continue.

![Select a Disk to Install or Upgrade](image)

- Select a local drive for OS installation and press **Enter** to continue.

- When the installation is in progress, the screen will show,

```plaintext
Enter a root password

_root password: ******** 
_confirm password: ********

Passwords match.
```

f. When the installation is in progress, the screen will show,
g. At this screen, press **F11** to confirm the installation.

```
Confirm Install

The installer is configured to install ESXi 5.5.0 on:
naa.600685b0071fb3901b886a723f9d6b4a.

Warning: This disk will be repartitioned.

(Esc) Cancel (F9) Back (F11) Install
```

h. Once the OS Installation is in progress, it takes 20-30 minutes to complete.

8. When the installation is completed, the next screen provides instructions to remove the installation media from KVM before pressing **Enter** key to reboot.

```
Installation Complete

ESXi 5.5.0 has been successfully installed.

ESXi 5.5.0 will operate in evaluation mode for 60 days. To use ESXi 5.5.0 after the evaluation period, you must register for a VMware product license. To administer your server, use the vSphere Client or the Direct Control User Interface.

Remove the installation disc before rebooting.
Reboot the server to start using ESXi 5.5.0.

(Enter) Reboot
```

9. Select **Virtual Media** tab from KVM console and uncheck the **Mapped** box.
   It should appear as shown below after removing the media from KVM.
10. Select the **KVM Console** tab and press **Enter** to reboot the server. It may take 3-5 minutes.

   Wait until the below screen appears on console after reboot and press **F2** to set Host Management configuration.

11. Press **F2** key to set Host Management configuration and apply the following configurations.

   a. Log into the Host Management Interface using the root password you configured earlier.
b. From the Host Management Interface, select **Configure Management Network** and press **Enter**.

c. Select **Network Adapters** and press **Enter**.

d. Select appropriate NIC port to connect host server with network, press the spacebar to toggle the selection. Press **Enter**.

e. After successful configuration, correct NIC port appears to the right of
Network Adaptors.

f. Select **IP Configuration** to set IP Addresses, subnet mask, default gateway and DNS IP, as shown below.

   - Set IP Configuration and press **Enter**.

   ![IP Configuration](image)

   - Set DNS Configuration and press **Enter**.

   ![DNS Configuration](image)

g. Press **ESC** key to exit Host Management Network configuration.

h. Press **Y** to save the configuration changes.

i. Select **Troubleshooting Options** and enable ESXi shell and SSH login, as shown below.

   ![Troubleshooting Options](image)

**Note:** Press **Enter** to toggle the settings after selecting the item.

j. Select **Test Management Network** and press **Enter**.
VMware host is now ready to use for VM creation.

**VMWare Host Management**

All the required Virtual Machines (VM) will be created using master OVF file and two VMDK image files provided with MURAL release deliverables.

1. Start VMware vSphere client application from laptop/PC and open VMware Management interface to configure host (Bare Metal node) and guest (VM) servers.
2. Enter the rack server IP address, username, and password to open the VM management interface shown below.

MURAL Software Standard Installation Guide for Rack Servers

a. From left pane, select the Host (in this case 192.168.113.122) > **Configuration** tab > **Hardware** > **Networking**.

![Configuration tab screenshot](image)

b. Click on the **Add Networking** link to add a vnic0 interface.

c. Select **Virtual Machine** option and press **Next**.

![Add Network Wizard](image)

d. Select **vmnic0** box and press **Next**.
e. Enter **VM Network 0** for Network Label and press **Next**.

f. Press **Finish** on summary page to finish the network port creation.
g. Network interfaces appear as shown in below screenshot after successful creation.
Configuring and Managing the VM

Creating VMs Using OVF Template

1. Launch vSphere client and log into VMware host.

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

3. Select the OVF template from your computer and click Next.

4. Verify OVF Template details and click Next.

5. Specify a name for the deployed template and click Next.

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

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

8. Press **Finish** to start VM creation.

9. Press **Close** to complete the VM creation.

10. 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.

**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 vmdkFiles/
   # pwd
   /vmfs/volumes/datastore1/vmdkFiles
   # ls -l
   total 13729792
   -rwxr--r-- 1 root root 7458586624 Aug 21 13:06
   ```
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.

```bash
# 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 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 below table (per node).
### MURAL Software Standard Installation Guide for Rack Servers

#### Table: LUN Size and Mount Points

<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>200</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/rge</td>
</tr>
<tr>
<td>INSTA-DB</td>
<td>1024</td>
<td>0:0</td>
<td></td>
<td>dbroot1</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>0:1</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>
<tr>
<td>INSTA-BACKUP</td>
<td>1200</td>
<td>0:0</td>
<td></td>
<td>dbroot1</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>0:1</td>
<td></td>
<td>dbroot2</td>
</tr>
</tbody>
</table>

* INSTA-BACKUP VM is optional. It shall not be created in the server that hosts VM for INSTA-DB VM.

**Note:** Size of the LUNs can be different as per site-specific configuration, please refer 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 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) # ping -c 4 192.168.112.1
PING 192.168.112.1 (192.168.112.1) 56(84) bytes of data.
64 bytes from 192.168.112.1: icmp_seq=1 ttl=255 time=78.2 ms
64 bytes from 192.168.112.1: icmp_seq=2 ttl=255 time=0.760 ms
64 bytes from 192.168.112.1: icmp_seq=3 ttl=255 time=76.2 ms
64 bytes from 192.168.112.1: icmp_seq=4 ttl=255 time=0.779 ms
--- 192.168.112.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.760/38.998/78.213/38.235 ms
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.

7. Ensure that all VMs can be accessible over LAN after IP assign.

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. Login into master GCR node using ssh, as admin user and run following commands.

> en
# conf t
# ssh client user admin identity dsa2 generate
# write memory
# show ssh client

The resulting output may resemble:

```
(config) # show ssh client
SSH client Strict Hostkey Checking: no
No SSH global known hosts configured.
User Identities:
User admin:
DSA v2 Public key:
ssh-dss AAAAB3NzaC1kc3[...]*C.eF4ukzCBf50M=
DSA v2 Private key:
------BEGIN DSA PRIVATE KEY------
...
```

Where [...] indicates text was removed for brevity.

2. Copy DSAv2 Public key from above command share with all other nodes, using following CLI command:

```
ssh client user admin authorized-key sshv2 "DSA-v2-Public-key"
```

For example, log into another node using SSH as an admin user.

```
guavus-c37469 > en
guavus-c37469 # conf t
guavus-c37469 (config) # ssh client user admin authorized-key sshv2 "ssh-dss AAAAB3NzaC1kc3[...]*C.eF4ukzCBf50M=
guavus-c37469 (config) # no username root disable
guavus-c37469 (config) # cmc client enable
guavus-c37469 (config) # cmc client available
guavus-c37469 (config) # license install LK2-RESTRICTED_CMDS-88A4-FNLG-XCAU-U
guavus-c37469 (config) # no username root disable
```

```
guavus-c37469 (config) # username root password root@123
```
guavus-c37469 (config) # username admin password admin@123

guavus-c37469 (config) # write memory

guavus-c37469 (config) # quit

3. Run above set of commands for all the VMs one by one. If key sharing was successful, you should be able to log into each node from the master GCR Node without being prompted for the admin password.

> en
# _shell
# ssh admin@Node-IP-address

Where Node-IP-Address is the ip address of the VM that received the SSH key.

**Collecting Mac IDs from All Nodes**

Run the following commands from master GCR node to collect the MAC IDs. Repeat these commands for all nodes and update the CIQ sheet columns for MAC addresses to be used in the next section to update GMS configuration XML.

```bash
# ssh root@10.10.10.110 'echo "show interface eth0 br" | /opt/tms/bin/cli -m config | /bin/grep -e "IP address:" -e "HW address:"'
IP address: 10.10.10.110
HW address: 00:0C:29:B9:2C:91

# ssh root@10.10.10.110 'echo "show interface eth1 br" | /opt/tms/bin/cli -m config | /bin/grep -e "IP address:" -e "HW address:"'
IP address: 192.168.113.130
HW address: 00:0C:29:B9:2C:9B
```
Configuring the Master GMS Node

Apply Patches on Master GCR nodes

Download the patches from the FTP server to the master GCR server in the /data directory. Apply all patches applicable for GMS node.

Note: See the release notes for software version 3.4 for a complete list of patches and installation instructions.

Node will reboot after successfully applying the patches.

GMS Server setup

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.

```plaintext
> en
  # conf t
  # psql dbroot /data/pgsql
  # psql mode external
  # pm process psqlld restart
  # pm process gms_server restart
  # write memory
  # show pm process gms_server
  # quit
```

Selecting Application Templates for GMS

Run from shell prompt #

```plaintext
  cd /config/gms/Profiles/Custom/pwd
  ln -s /opt/deployment/GMS_Templates/collector/Feed/bulkstats/collector_adaptor_bulkstats_template.xml.
  ln -s /opt/deployment/GMS_Templates/collector/Feed/edrhttp_edrflow/collector_adaptor_edrhttp_edrflow_template.xml.
  ln -s /opt/deployment/GMS_Templates/hadoop/dfs_template.xml.
```
ln -s /opt/deployment/GMS_Templates/hive/hive_mural.xml.
ln -s /opt/deployment/GMS_Templates/insta/App/bulkstats/insta_mural_bulkstats_flat.xml.
ln -s /opt/deployment/GMS_Templates/insta/App/dpi/insta_mural_dpi_flat.xml.
ln -s /opt/deployment/GMS_Templates/insta/App/httperror/insta_httperror_flat.xml.
ln -s /opt/deployment/GMS_Templates/oozie/App/anomalyDetection/workflow_anomaly_with_timeout_jobs.xml.
ln -s /opt/deployment/GMS_Templates/oozie/App/bulkstats/workflow_bulksat_mural_with_timeout.xml.
ln -s /opt/deployment/GMS_Templates/oozie/App/dpi/workflow_dpi_with_timeout_jobs.xml.
ln -s /opt/deployment/GMS_Templates/oozie/App/dynamicWhitelisting/workflow_dynamic_whitelisting_with_timeout.xml.
ln -s /opt/deployment/GMS_Templates/oozie/App/httperror/workflow_httperror_with_timeout.xml.
ln -s /opt/deployment/GMS_Templates/oozie/App/rulebase/workflow_rulebase_apn_with_timeout.xml.
ln -s /opt/deployment/GMS_Templates/oozie/App/tethering/workflow_tethering_with_timeout.xml.
ln -s /opt/deployment/GMS_Templates/oozie/Feed/edrhttp_edrfflow/workflow_edrhttp_edrfflow_with_timeout.xml.
ln -s /opt/deployment/GMS_Templates/oozie/Feed/edrhttp_edrfflow/workflow_hive_edrhttp_edrfflow_with_timeout.xml.
ln -s /opt/deployment/GMS_Templates/postgres/postgresql_mural.xml.
ln -s /opt/deployment/GMS_Templates/rubix/anomaly/rubix_anomaly_mural.xml.
ln -s /opt/deployment/GMS_Templates/rubix/bulkstats/rubix_bulkstats_mural.xml.
ln -s /opt/deployment/GMS_Templates/rubix/dpi/rubix_atlas_
distributed_mural.xml.

ln -s /opt/deployment/GMS_Templates/rubix/httperror/rubix_httperror_mural.xml.
ln -s /opt/deployment/GMS_Templates/rubix/launcher/rubix_launcher_mural.xml.
ln -s /opt/deployment/GMS_Templates/rubix/rge/rubix_rge_mural.xml.
ln -s /opt/deployment/GMS_Templates/solution/solution_mural.xml.

Wait for 2 minutes for tomcat server to start and check

```
# ps -eaf | grep tomcat
```

From laptop/PC connected with MURAL nodes over LAN, open web browser and launch GMS interface using following URL: http://<Master-GCR-Node-IP>/configure

Create the XML that will be used to manufacture and configure MURAL system on all the VMs, using the next section.
Loading and Verifying Initial Configuration Settings

This topic explains how to define the deployment topology using the General Management System (GMS) instead of having to install and configure software manually on each blade. GMS is a node that enables installation and monitoring of the software.

Notes:

- The GMS user interface requires Java Plug-in 10.40.2.43 or above, and JRE version 1.7.0_40-b43 or above.

Before you begin, locate your Customer Information Questionnaire (CIQ)—for some of the steps in this procedure, you will need to refer to the information on the IP Survey tab.

Accessing the GMS User Interface

To access the GMS user interface (UI), perform the following steps:

1. In a browser, access the following URL:

   http:// IP-Address-GMS-Lite/configure

   Note: The preferred browser for opening the GMS Applet is Mozilla Firefox v28.0 (or above) with Java Applet plugin v7 update 51.

2. Log in as admin using the password you defined in "Setting Up the Master GMS Node" on page 1 (the recommended password is admin@123).

3. In the Java applet pop-up window, enter user name and password:
4. Once you have logged in, the GMS UI displays a series of screens, where you enter configuration information. Proceed as outlined in the following sections.

**Loading the Base Configuration File**

A sample configuration file is provided with the installation package. This file, `muralRS.xml`, is configured specifically for the software version and can be changed for the local set up according to the bill of materials (BoM). In this case, it includes a logical grouping of the nodes into two chassis (rack servers) and configuration settings for all of the system components.

**Warning:** Do not modify the file directly in a text editor. Use the GMS UI to make any modifications to the `muralRS.xml` file.

To upload the configuration file in the GMS UI:

1. Click the file navigation button (…) to the right of the first text field.
2. Navigate to and select the configuration file.
3. Select the **Load Config File** button.
4. When the configuration file is loaded, the GMS UI appears.

5. Configuration information is loaded from the `muralRS.xml` file, populating fields on each tab with values that are specific to your deployment.

   **Note:** In many cases, all you need to do is verify the settings are correct. If you find incorrect values, you can Proceed with the initial configuration as outlined in "Verifying XML Settings " below. You need to check the XML against values specified as necessary for your deployment.

**Verifying XML Settings**

Click on the name of a chassis to see more details and verify settings

The hardware type, chassis number and chassis logical name are displayed, as well as the list of nodes (blades) in each slot.

**Note:** In most cases you only need to verify settings in the subsequent sections. You might need to modify settings if, for example, the slot numbers for blades have changed or some information was not yet known when the CIQ was completed for your deployment.

Proceed with verifying settings applied by the configuration file as outlined below.

**Understanding LUN SCSI and Block IDs**

The number of LUNs, and therefore the number of SCSI/Block IDs, assigned to each node depends on the type of node:
MURAL Software Standard Installation Guide for Rack Servers

- GCR nodes—Each GCR node has three assigned WWIDs:
  - GMS—pgsql database to store GMS and Zabbix data, HET and MURAL UI users/Rubix-related data
  - Collector—EDR/Bulkstat raw data files received from ASR
  - Rubix—storing offline reports

- Compute nodes—Each node has one WWID assigned

- Insta nodes—Each Insta node (being clustered) has two assigned WWIDs—one assigned to `dbroot1` of both Insta 1 and Insta 2, one assigned to `dbroot2` of both Insta 1 and Insta 2.

**Important:** Ensure that SCSI IDs (like 0:0, 0:1, etc) will only be used for Insta node, all other nodes will used device Block IDs (like /dev/sda, /dev/sdb, etc created corresponding to SCSI IDs 0:0, 0:1 and so on), as shown in below screenshots:

If you are at a stage where UCS configuration is not yet fully set up, then you can use a dummy WWID for each blade. However, if you have configured both UCS and EMC, you must enter the actual WWID, which you can find as described below.

**Verifying Server Details Tab**

In this section you will update the Chassis info, MAC addresses of interface bindings and SCSI/Block IDs for the storage allocated to each node. Refer the updated CIQ sheet to update all these fields on master and standby GCR nodes, all Compute nodes, and Insta node. During the initial configuration process, you must do this for each node in each of the two logical chassis. Update the nodes' other details per site-specific configurations.

The figures in the following sections show a sample configuration. Your configuration may be different. Verify in the GMS user interface (UI) that all the nodes in your deployments have been configured.

**Note:** Throughout this section, if the value is incorrect, click on Add to create a new entry and set the correct value. Then select the incorrect entry and click Delete to remove it.
1. Open the **Server Details** tab in the GMS UI.

2. In the **Chassis** box, select the chassis.

3. The **Slot** section lists the nodes (blades) in each slot. Select a node to verify or modify its WWID values.

4. Repeat steps 1 through 3 for each of the following nodes in Chassis 1:
   - First GCR node
   - Second GCR node
   - First Compute node
   - Second Compute node, and so on
   - First Insta node

   **Note:** Ensure that the same WWID is assigned to dbroot1 and dbroot2 for both Insta 1 and Insta 2 in the GMS configuration.

5. If using a dual chassis, repeat steps 1 through 4.
6. In the **Interface Members** box, verify that the MAC address of the interface binding for the node matches the value written in your CIQ.

7. In the **Storage** section, use the following Block IDs as indicated:

   a. For non-Insta nodes:
b. For Insta nodes:

![Image of Insta node network configuration]

**Verifying Networks Tab**

To define the networks of the blades, perform the following steps:

1. Under the **Networks** tab, verify the following fields for the:
   
   - Network Name
   - Network IP Address: prefix and subnet
   - Interface Type
   - Network Interface
   - Network Range

![Image of Network configuration details]
Verifying Global Settings Tab

Verify the global settings for the system and IP addresses configured for each node cluster.

To verify the network IP address, global settings, and IP addresses of the nodes:

Under the **Global Settings** tab, verify the following information for the MURAL system:

1. DNS server
2. Default gateway
3. NTP server IP address
4. NTP version

5. SNMP Details

Set the GCR IP address as the SNMP Server IP.

Set Passwords for Pre-defined users and add other users required to send...
EDR data into MURAL system or collect MURAL reports.

**Verifying Nodes Tab**

Define the internal and external IP address and subnet mask of each node.

**Verifying Clusters Tab**

The `muralRS.xml` configuration file included with your installation package includes configurations of the node clusters. The figures below show an example—the actual configurations for your setup might be different depending on how much traffic is being sent to the MURAL system.

During initial configuration, you must create clusters for each type of node and add the member nodes to each cluster.

To configure a cluster, perform the following steps:
1. Go to the **Clusters** tab.

2. From the **Cluster Type** list, select the cluster you want to configure or verify.

3. Select **Cluster Type** from the drop-down list, enter the **Cluster Name**, and **Cluster Interface**, and click **Add**. For example, the following figure shows a GCR cluster called **GCR-CLUS**:

4. Select the newly created cluster, and under **Cluster VIPs**, click **Add** and enter the virtual IP address of the new cluster.

5. Repeat steps 2 through 4 for the rest of the nodes.
Verifying Applications Tab and Specifying Component Profiles

This section describes how to load profiles for each component. These profiles set the configurations for the components. You can then change individual settings to adjust the configuration for your environment.

Templates for GMS, Postgres, HDFS, Solution, Collector, Hive, Workflow and Rubix need to be properly defined for GCR cluster.

Configuration requirements for this deployment are provided here. For guidance on how to apply these settings, skip to "Applying Profile Settings" on page 77.

- GCR with the following values:

  Attach the following profiles to GCR cluster as required by features that are enabled for a site. Ensure that Rubix Atlas and Launcher profiles are added to the application instance 1 and the remaining profiles are added to the application instance 2.

<table>
<thead>
<tr>
<th>App Name</th>
<th>App Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFS</td>
<td>dfs_custom_template</td>
</tr>
</tbody>
</table>

  **Application Instance = 1**
  **Cluster = GCR-CLUS**
  **Note:** Verify the name is the correct one for your setup

  hadoop.mapred.min.split.size = 134217728 (i.e. 128 MB)
  hadoop.mapred.tasktracker.map.tasks.maximum (Default value is 8)
<table>
<thead>
<tr>
<th>App Name</th>
<th>App Profile</th>
<th>Settings</th>
</tr>
</thead>
</table>
| solution | solution_custom_template | **Cluster** = GCR-CLUS  
**Expected start date** = DataStartTime  
**Client name** = cisco  
**Property** is **StarterPack** = true  
**timezone** = Refer to the table for the valid values. The selected value is applied automatically to all the jobs. |

| Workflow | workflow_custom_rulebase_apn_with_timeout | Update time-zone properties for the following jobs and datasets if the time-zone needs to be set as anything other than UTC and Rule base reporting is enabled:  
Jobs that require TZ Configuration:  
1. RulebaseAggregator  
2. RulebaseCSVFormatter  
3. RulebaseDataTransfer  
4. CleanupRulebase  
Datasets that require TZ Configuration:  
1. RulebaseAggregatorInput  
2. RulebaseAggregatorOutput  
3. RulebaseCSVFormatterOutput  
**Note:** Please keep the time-zone value same as used in UI templates. |
<table>
<thead>
<tr>
<th>App Name</th>
<th>App Profile</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow</td>
<td>workflow_custom_anomaly_with_timeout_jobs</td>
<td>Update time-zone properties for the following jobs and datasets if the time-zone needs to be set as anything other than UTC and Anomaly detection is enabled:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. AnomalyCoreJob_output_anomaly</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. anomaly_hive_hourly</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. anomalyDaily</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. anomaly_Agg_daily</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. anomalyMonthly</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. anomalyHourlyAggregate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7. anomalyDailyAggregate</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Please keep the time-zone value same as used in UI templates.</td>
<td></td>
</tr>
</tbody>
</table>

| Workflow | Profiles for Content Analytics:                   | workflow_custom_edrhttp_edrflow_with_timeout                              |
|          |                                                  | workflow_custom_dpi_with_timeout_jobs                                     |
|          |                                                  | workflow_custom_dynamic_whitelisting_with_timeout                          |
|          | Profiles for Bulkstats:                          | workflow_custom_bulkstat_mural_with_timeout                               |
|          | Profiles for Tethering:                          | workflow_custom_tethering_with_timeout                                    |
|          | Profiles for HTTP Error Tracking:                | workflow_custom_httperror_with_timeout                                   |
|          | Profiles for Anomaly:                           | workflow_custom_hive_edrhttp_edrflow_with_timeout                        |

**Application Instance = 1**  
**Cluster = GCR-CLUS**
<table>
<thead>
<tr>
<th>App Name</th>
<th>App Profile</th>
<th>Settings</th>
</tr>
</thead>
</table>
| HIVE     | hive_custom_mural | **Application Instance = 1**
|          |              | **Cluster = GCR-CLUS** |
| GMS      | gms_default_without_nbi_template | **Cluster = GCR-CLUS**
|          |              | **logclean.maxSize = 100000, read-write, integer**
|          |              | **logclean.maxAge = 30, read-write, integer**
|          |              | **zabbix.AdminPassword = zabbix, read-write, string**
|          |              | **logcollection.enable = false, read-write, boolean** |
| PostgreSQL | postgresql_custom_mural | **Cluster = GCR-CLUS** |
### Collector

**Cluster name** = GCR-CLUS

- **adaptor.bulkStats.input.fileConfig.bulkStatsFile1**
  - **inputDirectory** = /data/collector/bulkstats_files/gateway_name

  Where `gateway_name` is replaced with the name of the gateway where the ASR is located. For example, GMPLAB1.

- **adaptor.bulkStats.input.fileConfig.bulkStatsFile1**
  - **backupDirectory** = /data/collector/bulkstats_files/gateway-name

- **adaptor.bulkStats.input.fileConfig.bulkStatsFile1**
  - **fileNameFormat** = *_%YYYY%MM%DD%hh%mm%ss

- **adaptor.edrflow.output.directory** =
  /data/collector/1/output/edrflow/%y/%m/%d/%h/%mi/gateway-name

- **adaptor.edrhttp.output.directory** =
  /data/collector/1/output/edrhttp/%y/%m/%d/%h/%mi/gateway-name

  Where `%y/%m/%d/%h/%mi` indicates the format of the time value.

Refer to the [table](#) for guidelines on setting the gateway.

**Note:** The workflow profile contains configuration settings for the MapReduce jobs.
When you set the value for `timeZone`, ensure that the value is the same as what was used in the UI templates. The following table lists the different `timeZone` values that can be used.

<table>
<thead>
<tr>
<th>TimeZone String</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>US/Central</td>
<td>United States Central Standard Time</td>
</tr>
<tr>
<td>US/Eastern</td>
<td>United States Eastern Standard Time</td>
</tr>
<tr>
<td>US/Pacific</td>
<td>United States Pacific Standard Time</td>
</tr>
<tr>
<td>America/Lima</td>
<td>Peru</td>
</tr>
<tr>
<td>Chile/EasterIsland</td>
<td>Chile</td>
</tr>
<tr>
<td>Africa/Johannesburg</td>
<td>South Africa</td>
</tr>
<tr>
<td>Asia/Manila</td>
<td>Philippines</td>
</tr>
<tr>
<td>Egypt</td>
<td>Egypt</td>
</tr>
<tr>
<td>Europe/Amsterdam</td>
<td>Netherlands</td>
</tr>
<tr>
<td>Europe/Dublin</td>
<td>Ireland</td>
</tr>
<tr>
<td>UTC</td>
<td>Universal Co-ordinated Time</td>
</tr>
</tbody>
</table>

When you set up Gateway names for the Collector, refer to the following table that lists certain guidelines and samples:

<table>
<thead>
<tr>
<th>Filename Pattern</th>
<th>Regex in Wrapper CLI</th>
<th>Regex in Collector Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gatewayname_(multiple strings separated by underscore or hyphen or both)_flow_timestamp_str4.gz</td>
<td><em>_</em><em>*</em>%MM%DD%YYYY%hh%mm%ss_*_.gz</td>
<td>%DC_<em>_</em>_%MM%DD% YYYY%hh%mm%ss*.gz</td>
</tr>
<tr>
<td><strong>Example</strong>: Gatewayname_str1_str2_str3_flow_timestamp_str4.gz</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Timestamp</strong>: MMDDYYYYhhmmss</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filename Pattern</td>
<td>Regex in Wrapper CLI</td>
<td>Regex in Collector Configuration</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------------</td>
<td>---------------------------------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>Gatewayname__(multiple strings separated by underscore or hyphen or both)_flow_timestamp_str4_str5.gz</td>
<td>* * * * %MM%DD%YYYY%hh%mm%ss_*_.gz</td>
<td>%DC_* _ _ %MM%DD%YYYY%hh%mm%ss*.gz</td>
</tr>
<tr>
<td><strong>Example:</strong> Gateway name_str1_str2_str3_flow_timestamp_str4_str5.gz</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Timestamp:</strong> GMDDY YYYhhmmmss</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gatewayname__(multiple strings separated by underscore or hyphen or both)_flow_string_timestamp_string.gz</td>
<td>* * * * %MM%DD%YYYY%hh%mm%ss_*_.gz</td>
<td>%DC_* _ _ %MM%DD%YYYY%hh%mm%ss*.gz</td>
</tr>
<tr>
<td>Example: Gatewayname_str1_str2_str3_flow_string_timestamp_str5.gz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timestamp: MMDDYYYYhhmmmss</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gatewayname__(multiple strings separated by underscore or hyphen or both)_flow_string_timestamp_string_string.gz</td>
<td>* * * * %MM%DD%YYYY%hh%mm%ss_*_.gz</td>
<td>%DC_* _ _ %MM%DD%YYYY%hh%mm%ss*.gz</td>
</tr>
<tr>
<td>Example: Gatewayname_str1_str2_str3_flow_string_timestamp_str5_str6.gz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timestamp: MMDDYYYYhhmmmss</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Filename Pattern

<table>
<thead>
<tr>
<th>Gatewayname_(multiple strings separated by underscore or hyphen or both)_flow_string_timestamp_string_string.gz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: Gatewayname_str1_str2_str3_flow_str4_timestamp_str5_str6.gz</td>
</tr>
<tr>
<td>Timestamp: YYYYMMDDhhmmss</td>
</tr>
<tr>
<td><strong>Regex in Wrapper CLI</strong></td>
</tr>
<tr>
<td><em>.</em>.<em>_</em>%YYYY%MM%DD%hh%mm%ss_*_.gz</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gatewayname_(multiple strings separated by underscore or hyphen or both)_flow-string_timestamp_string_string_string.gz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: Gatewayname_str1_str2_str3_flow-str4_timestamp_str5_str6.gz</td>
</tr>
<tr>
<td>Timestamp: MMDDYYYYhhmmss</td>
</tr>
<tr>
<td><em>.</em><em>*-%MM%DD%YYYY%hh%mm%ss</em>*_.gz</td>
</tr>
</tbody>
</table>

### Notes:

- It is mandatory to send gateway name as the first substring followed by an underscore.
- You can replace 'http' with 'flow'.
- If the timestamp is in the MMDDYYYYhhmmss format, set the %MM%DD%YYYY%hh%mm%ss parameter in the Regex pattern.
- If the timestamp is in the YYYYMMDDhhmmss format, set the %YYYY%MM%DD%hh%mm%ss parameter in the Regex pattern.
- Rubix with the following values:
### Application Settings

<table>
<thead>
<tr>
<th>App Name</th>
<th>App Profile</th>
<th>Settings</th>
</tr>
</thead>
</table>
| Content Analytics (Atlas)        | rubix_custom_atlas_distributed_mural | **Application Instance = 1**  
                             | rubix_custom_rge_mural         | Cluster name = GCR-CLUS  
                             | rubix_custom_cacheless_mural   | application.atlas.rubixInstance.1.tomcatInstanceMaxSize — Set it as half of the total RAM on the GCR node  
                             |                               | application.atlas.rubixInstance.1.initialJavaHeapSize — Set it as half of the total RAM on the GCR node  |
| Bulkstats                       | rubix_custom_bulkstats_mural  | **Application Instance = 2**  
                             |                               | Cluster name = GCR-CLUS  |
| HTTP Error Tracking             | rubix_custom_launcher_mural   | **Application Instance = 1**  
                             | rubix_custom_httperror_mural  | Cluster name = GCR-CLUS  |
| Anomaly                         | rubix_custom_anomaly_mural    | **Application Instance = 2**  
                             |                               | Cluster name = GCR-CLUS  |

Verify the values of the following parameters, in all applied templates:

- **numOwners value:**
  - application.atlas.numOwners
  - application.bulkstats.numOwners
  Set 2 for single instance of Rubix in 1+1 HA cluster)
• URL address to access MURAL UI (Example ucsd.cisco.com):
  application.launcher.dpiURL
  application.launcher.httpURL
  application.atlas.rubixFQDN
  application.atlas.rgeSolutionURL
  application.atlas.anomalySolutionURL
  application.atlas.bulkStatsURL

• Domain (Example cisco.com)
  application.launcher.sessionCookieDomain
  application.atlas.sessionCookieDomain
  application.bulkstats.sessionCookieDomain
  application.rge.sessionCookieDomain
  application.reportAtlas.sessionCookieDomain
  application.anomaly.sessionCookieDomain
  application.httperror.sessionCookieDomain

  For example, if the URL to be used is
  https://abc.mural.com https://today.mural.com
  then the correct value is mural.com.

• The time zone of location of the ASR, for example,
  America/Argentina/Buenos_Aires.

  Note: Set the application.timeZone property to the same TimeZone
  value for all profiles (such as Atlas, RGE, and Bulkstats). Mural
  support timezones configured in IANA format. Refer to the table
  for the valid timezone strings.

• Time-zone properties
  application.atlas.timeZone
  application.bulkstats.timeZone
  application.reportAtlas.timeZone
  application.anomaly.timezone
  application.httperror.timeZone
Customer mail configurations, for example, mail server - 
mx1.cisco.com sender - support@cisco.com and SMTP port - 25
application.rge.mailHost
application.anomaly.mailHost
application.rge.mailSender
application.anomaly.mailSender
application.rge.mailPort
application.anomaly.mailPort

In case LUNs are not available for Disk-based caching, set value for 
below properties as false
application.atlas.cachePresistToDisk
application.bulkstats.cachePresistToDisk

Following screens show examples of various Rubix profiles after setup specific changes (Values for highlighted rows from different templates, need to be updated):

**Rubix Launcher profile**

<table>
<thead>
<tr>
<th>Property, Name</th>
<th>Property Value</th>
<th>Property Type</th>
<th>Property_Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>launcher</td>
<td>jdbc:postgresql:///APPLICATION.PostgreSQL.Mgmt.VP1.5432/rubixdb</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.launcher.sessionCookieDomain</td>
<td>cisco.com</td>
<td>read-write</td>
<td>string</td>
</tr>
<tr>
<td>application.launcher.distributedConnectionUrl</td>
<td><a href="http://mx1.cisco.com:6443/index.jsp">http://mx1.cisco.com:6443/index.jsp</a></td>
<td>read-only</td>
<td>string</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property, Name</th>
<th>Property Value</th>
<th>Property Type</th>
<th>Property_Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>application.launcher.rubixInstance</td>
<td>Rubix-instance</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.launcher.rubixInstance.1.tomcatInstanceMaxSize</td>
<td>1g</td>
<td>read-write</td>
<td>string</td>
</tr>
<tr>
<td>application.launcher.rubixInstance.1.initialJavaHeapSize</td>
<td>1g</td>
<td>read-write</td>
<td>string</td>
</tr>
</tbody>
</table>

**Rubix Content Analytics (Atlas) profile**

set memory values as half of the total RAM installed on the GCR node.
## Proxies

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>atlas</td>
<td>Rack Servers</td>
<td>read-only</td>
</tr>
<tr>
<td>numOwners</td>
<td>2</td>
<td>read-write</td>
</tr>
<tr>
<td>pghost</td>
<td>5432</td>
<td>read-only</td>
</tr>
<tr>
<td>sessionCookieDomain</td>
<td>cisco.com</td>
<td>read-write</td>
</tr>
<tr>
<td>timeZone</td>
<td>UTC</td>
<td>read-write</td>
</tr>
<tr>
<td>databaseConnectionInfo</td>
<td>dbx-postgresql://5432/rubixdb</td>
<td>read-only</td>
</tr>
<tr>
<td>bulkstats</td>
<td>60s</td>
<td>read-write</td>
</tr>
<tr>
<td>rubix_instance</td>
<td>Rubix-Instance1</td>
<td>read-only</td>
</tr>
<tr>
<td>redirectPort</td>
<td>6009</td>
<td>read-write</td>
</tr>
<tr>
<td>dbapp</td>
<td>6005</td>
<td>read-write</td>
</tr>
<tr>
<td>rubix_instance</td>
<td>1</td>
<td>read-write</td>
</tr>
<tr>
<td>dbapp</td>
<td>6010</td>
<td>read-write</td>
</tr>
<tr>
<td>atlas_instance</td>
<td>1</td>
<td>read-write</td>
</tr>
<tr>
<td>dbapp</td>
<td>4059</td>
<td>read-write</td>
</tr>
<tr>
<td>dbapp</td>
<td>90g</td>
<td>read-write</td>
</tr>
</tbody>
</table>

### Rubix Bulkstats profile

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>atlas</td>
<td>Rack Servers</td>
<td>read-only</td>
</tr>
<tr>
<td>numOwners</td>
<td>2</td>
<td>read-write</td>
</tr>
<tr>
<td>pghost</td>
<td>5432</td>
<td>read-only</td>
</tr>
<tr>
<td>sessionCookieDomain</td>
<td>cisco.com</td>
<td>read-write</td>
</tr>
<tr>
<td>timeZone</td>
<td>UTC</td>
<td>read-write</td>
</tr>
<tr>
<td>databaseConnectionInfo</td>
<td>dbx-postgresql://5432/rubixdb</td>
<td>read-only</td>
</tr>
<tr>
<td>bulkstats</td>
<td>60s</td>
<td>read-write</td>
</tr>
<tr>
<td>rubix_instance</td>
<td>Rubix-Instance1</td>
<td>read-only</td>
</tr>
<tr>
<td>redirectPort</td>
<td>6009</td>
<td>read-write</td>
</tr>
<tr>
<td>dbapp</td>
<td>6005</td>
<td>read-write</td>
</tr>
<tr>
<td>rubix_instance</td>
<td>1</td>
<td>read-write</td>
</tr>
<tr>
<td>dbapp</td>
<td>4059</td>
<td>read-write</td>
</tr>
<tr>
<td>dbapp</td>
<td>90g</td>
<td>read-write</td>
</tr>
</tbody>
</table>

### Rubix reportAtlas profile

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>atlas</td>
<td>Rack Servers</td>
<td>read-only</td>
</tr>
<tr>
<td>numOwners</td>
<td>2</td>
<td>read-write</td>
</tr>
<tr>
<td>pghost</td>
<td>5432</td>
<td>read-only</td>
</tr>
<tr>
<td>sessionCookieDomain</td>
<td>cisco.com</td>
<td>read-write</td>
</tr>
<tr>
<td>timeZone</td>
<td>UTC</td>
<td>read-write</td>
</tr>
<tr>
<td>databaseConnectionInfo</td>
<td>dbx-postgresql://5432/rubixdb</td>
<td>read-only</td>
</tr>
<tr>
<td>bulkstats</td>
<td>60s</td>
<td>read-write</td>
</tr>
<tr>
<td>rubix_instance</td>
<td>Rubix-Instance1</td>
<td>read-only</td>
</tr>
<tr>
<td>redirectPort</td>
<td>6009</td>
<td>read-write</td>
</tr>
<tr>
<td>dbapp</td>
<td>6005</td>
<td>read-write</td>
</tr>
<tr>
<td>rubix_instance</td>
<td>1</td>
<td>read-write</td>
</tr>
<tr>
<td>dbapp</td>
<td>4059</td>
<td>read-write</td>
</tr>
<tr>
<td>dbapp</td>
<td>90g</td>
<td>read-write</td>
</tr>
</tbody>
</table>
# Rubix RGE profile for offline reports

3 rubix_custom_rge_mural
4 rubix_custom_anomaly_mural
5 rubix_custom_alerts_mural

## Profiles

<table>
<thead>
<tr>
<th>Property_Name</th>
<th>Property_Value</th>
<th>Property_Type</th>
<th>Property_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>rubixApp</td>
<td>rge</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.nodeUrl</td>
<td>https://$APPLICATION.Rubix.RGE.Mgmt.VIP:80443</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.jobTrackerUrl</td>
<td>$APPLICATION.DPS.Data.VIP:5001</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.psqlUsername</td>
<td>admin</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.gmaster</td>
<td>$APPLICATION.Rubix.RGE.Mgmt.VIP</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.psqlServerUrl</td>
<td>http://$APPLICATION.DPS.Data.VIP:5000/psql2</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.alternateConnectionUrl</td>
<td>jdbc.postgresql://$APPLICATION.PostgreSQL.Mgmt.VIP:5432/rubixdb</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.distributedConnectionUrl</td>
<td>jdbc.postgresql://$APPLICATION.PostgreSQL.Mgmt.VIP:5432/rubixdb</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.sessionCookieDomain</td>
<td>$APPLICATION.DPS.Data.VIP:5001</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.nvl_cisco.com</td>
<td>nvl.cisco.com</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>Rubix Instance: &lt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Rubix Anomaly profile

3 rubix_custom_anomaly_mural
4 rubix_custom_alerts_mural
5 rubix_custom_alerts_mural

## Profiles

<table>
<thead>
<tr>
<th>Property_Name</th>
<th>Property_Value</th>
<th>Property_Type</th>
<th>Property_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>rubixApp</td>
<td>rge</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.nodeUrl</td>
<td>https://$APPLICATION.Rubix.RGE.Mgmt.VIP:80443</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.jobTrackerUrl</td>
<td>$APPLICATION.DPS.Data.VIP:5001</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.psqlUsername</td>
<td>admin</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.gmaster</td>
<td>$APPLICATION.Rubix.RGE.Mgmt.VIP</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.psqlServerUrl</td>
<td>http://$APPLICATION.DPS.Data.VIP:5000/psql2</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.alternateConnectionUrl</td>
<td>jdbc.postgresql://$APPLICATION.PostgreSQL.Mgmt.VIP:5432/rubixdb</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.distributedConnectionUrl</td>
<td>jdbc.postgresql://$APPLICATION.PostgreSQL.Mgmt.VIP:5432/rubixdb</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.sessionCookieDomain</td>
<td>$APPLICATION.DPS.Data.VIP:5001</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.nvl_cisco.com</td>
<td>nvl.cisco.com</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>Rubix Instance: &lt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property_Name</th>
<th>Property_Value</th>
<th>Property_Type</th>
<th>Property_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>rubixApp</td>
<td>rge</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.nodeUrl</td>
<td>https://$APPLICATION.Rubix.RGE.Mgmt.VIP:80443</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.jobTrackerUrl</td>
<td>$APPLICATION.DPS.Data.VIP:5001</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.psqlUsername</td>
<td>admin</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.gmaster</td>
<td>$APPLICATION.Rubix.RGE.Mgmt.VIP</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.psqlServerUrl</td>
<td>http://$APPLICATION.DPS.Data.VIP:5000/psql2</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.alternateConnectionUrl</td>
<td>jdbc.postgresql://$APPLICATION.PostgreSQL.Mgmt.VIP:5432/rubixdb</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.distributedConnectionUrl</td>
<td>jdbc.postgresql://$APPLICATION.PostgreSQL.Mgmt.VIP:5432/rubixdb</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.sessionCookieDomain</td>
<td>$APPLICATION.DPS.Data.VIP:5001</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>application.rge.nvl_cisco.com</td>
<td>nvl.cisco.com</td>
<td>read-only</td>
<td>string</td>
</tr>
<tr>
<td>Rubix Instance: &lt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Insta with the following values:

### Rubix HTTP Error Tracking profile

- **App Name**: Bulkstats
- **App Profile**: insta_custom_mural_bulkstats_flat
- **Cluster name**: INSTA-CLUS
### Applying Profile Settings

Apply the above profile settings by following these steps:

1. Select the **Applications** tab:

2. Start with the **Available Profiles** section:

   a. For **Application Name**, select the name of the system component from the drop-down menu.

   b. For **Application Profile**, select the profile you want to attach to the system component.

3. Scroll through the table in the **Application** section to find and select the Application Profile you just added. When you click on it, the UI opens two new sections below it: **Clusters** and **Profiles**.

4. In the **Cluster** section, click the **Add** button to add a new row to the empty table.

5. Attach the profiles as required.

After completing all configurations on the **Server Details**, **Networks**, **Global Settings**, **Nodes**, **Clusters**, and **Applications** tabs as described in the previous
sections, validate the configuration settings as described in the section "Validating the XML Configuration" below.

Validating the XML Configuration

To validate the GMS configuration, perform the following steps:

1. Click **Validate** on the bottom bar of the GMS interface.

   If any field or value entered is incorrect, a list of errors is displayed.

2. For each error, click the error and then the **Go** button to display the entry in the GMS configuration that is causing the error.

3. Correct any errors by following the steps in "Verifying XML Settings" on page 55

4. Click **Validate** again. When all errors have been resolved, the message **Validation successful** displays on the screen.

5. Click **Save Server File** to save the configuration to the GMS server.

   **Note:** The **Save Server File** is not operational until all validation errors are corrected. To save the file without correcting all errors and completing the validation, click **Save Local File**.

6. To activate the configuration after you have made changes, you must run the **gms config muralRS.xml activate** command. Until you activate the new configuration file, the system runs the previously saved configuration.
Using GMS Lite to Automate Installation of Nodes

This section explains how to use GMS Lite to manufacture the nodes, instead of manually manufacturing them individually.

Applying Patches on MURAL nodes

Download the patches from the FTP server to the master GCR VM in the /data directory. Apply all patches applicable for respective MURAL nodes.

Note: See the release notes for software version 3.4 for a complete list of patches and installation instructions.

Activating the XML File

1. Log into GMS Lite (using SSH) as admin user.

   ```
   > en
   # _shell
   ```

2. Activate the xml file on this VM

   ```
   (config)# gms config muralRS.xml activate
   ```

When complete, the output may resemble,

<table>
<thead>
<tr>
<th>File successfully activated</th>
</tr>
</thead>
<tbody>
<tr>
<td>(config)# gms config-runtime show status all</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM1-GCR-1</td>
<td>UNINIT</td>
</tr>
<tr>
<td>VM1-GCR-2</td>
<td>UNINIT</td>
</tr>
<tr>
<td>VM2-INSTA-1</td>
<td>UNINIT</td>
</tr>
<tr>
<td>VM3-DN-1</td>
<td>UNINIT</td>
</tr>
<tr>
<td>VM4-DN-2</td>
<td>UNINIT</td>
</tr>
</tbody>
</table>

Installing GCR Nodes

Run the following commands to bring up various clusters.
1. Log into the master GCR node where GMS is running and run the install command.

```
(config)# install appliance cluster cluster-name GCR-CLUS node VM1-GCR-1 force-format
```

This installation takes 1.5-2 hours. Periodically check the installation status by running:

```
[CLUSTER: master] (config)# install appliance show installation-status cluster GCR-CLUS node VM1-GCR-1
```

The resulting output may resemble:

```
VM1-GCR-1 : Node successfully installed
```

2. Install the standby GCR node.

```
(config)# install appliance cluster cluster-name GCR-CLUS node VM1-GCR-2 force-format
(config)# install appliance show installation-status cluster GCR-CLUS node VM1-GCR-2 force-format
```

The resulting output may resemble:

```
VM1-GCR-2 : Node successfully installed
```

3. Install the DN cluster.

```
(config)# install appliance cluster cluster-name DN-CLUS force-format
```

Installation in progress, check VM3-DN-1_cmc.log file for more details
Installation in progress, check VM4-DN-2_cmc.log file for more details

This installation takes 20-30 minutes. Periodically check the installation status by running:
(config)# install appliance show installation-status cluster DN-CLUS force-format

The resulting output may resemble:

VM3-DN-1: Node successfully installed.
VM4-DN-2: Node successfully installed.

4. Install the Insta cluster.

(config)# install appliance cluster cluster-name INSTA-CLUS force-format

Installation in progress, check VM2-INSTA-1_cmc.log file for more details

This installation takes 45-60 minutes. Periodically check the installation status by running:

(config)# install appliance show installation-status cluster INSTA-CLUS

The resulting output may resemble:

VM2-INSTA-1: Node successfully installed

After installing the GC node, perform the steps in "Repairing HDFS" below.

**Repairing HDFS**

1. To repair HDFS, log in to the master GC node and execute the following commands:

```
[admin@host ~]# pmx
Welcome to pmx configuration environment.
pm extension> subshell hadoop
pm extension (hadoop)> repair hdfs
Safe mode is OFF
The filesystem under path ‘/’ is HEALTHY
```
2. To ensure that Hadoop is not in the safe mode, execute the following commands:

```
[admin@host ~]# hadoop dfsadmin -report
```

3. To make the performance related changes, execute the following commands on both the master and the standby GC nodes:

```
> en
# conf t
(config) #internal set modify -
tps/process/hadoop/attribute/mapred.reduce.tasks/value value string 8
(config) #wr mem
(config) #pm process tps restart
```

### Configuring Application with Site Specifics

Execute below commands in both the GCR nodes to copy solution configuration file that is dependent on apps enabled for deployment. Content Analytics is default solution for Cisco MURAL sites. Following applications can also be enabled/disabled using this script.

- Tethering
- HTTP Error Tracking
- Anomaly

**Note:** Hardware at each deployment is sized to run a set of apps. Apps beyond this set will not be able to run on same hardware.

1. Log in to the master Insta node and reload the server for the zram changes to take effect:

```
> en
# conf t
```
(config) # wri mem
(config) # reload

After the server comes up as a standby server, repeat this step on the other Insta node (the new master Insta node).

2. Log into the master Collector node as an administrator.

VM1-GCR-1 [GCR-CLUS: master] > en
VM1-GCR-1 [GCR-CLUS: master] # _shell

3. Change directory to the folder with scripts.

[admin@VM1-GCR-1 ~]# cd /opt/etc/scripts/

4. Run the following installation script and respond to the prompts with an N.

[admin@VM1-GCR-1 scripts]# ./install_solution_config.sh
is tethering app enabled, enter (Y/N) : N
is httperror app enabled, enter (Y/N) : N
is anomalyDetection app enabled, enter (Y/N) : N

The resulting output may resemble:

atlas:
solutionConfig.json generated successfully at path
/data/CoreJob/config
[admin@VM1-GCR-1 scripts]#

5. If the Anomaly detection feature is enabled, run the following commands in both the GCR nodes for CoreJob solution config.

   a. Enter the pmx oozie subshell environment.

      (config)# pmx
      pm extension> subshell oozie

   b. Run the **set job** command for **CoreJob** and **CleanupAnomalyMonthly** actions.
pm extension (oozie)> set job CoreJob action CheckIbsDone
attribute waitOnJobs AnomalyRule:0
setting attribute (waitOnJobs)

pm extension (oozie)> set job CleanupAnomalyMonthly
action CleanupAction attribute cleanupDatasets anomaly_Agg_daily

c. Exit the oozie subshell using the **quit** command.

pm extension (oozie)> quit
pm extension> quit

d. Apply one more setting and write changes to memory before restarting processes.

```bash
(config)# internal set delete -
/tps/process/oozie/jobs/CleanupAnomaly5weekly/actions/CleanupAction/attribute/cleanupDatasets/values/anomaly_Agg_daily
(config)# write memory
(config)# pm process tps restart
```

6. If Rule base report feature is enabled, edit the following file in both the GCR nodes:

```plaintext
Filename: /opt/etc/oozie/EDR/Feed/EDRFlowFeed.json
Property:
"com.guavus.mapred.atlas.app.dpiapp.feature.rulebase.RulebaseMapper" : "true",
```

After changes are made, Property should appear as follows.

```bash
# grep
"com.guavus.mapred.atlas.app.dpiapp.feature.rulebase.RulebaseMapper" /opt/etc/oozie/EDR/Feed/EDRFlowFeed.json
"com.guavus.mapred.atlas.app.dpiapp.feature.rulebase.RulebaseMapper" /opt/etc/oozie/EDR/Feed/EDRFlowFeed.json
```
Troubleshooting Node Installation

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

Logs can be collected on the GCR server from the location /data/gms/logs.

Tuning the System for Performance

To tune the performance of the system:

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

   ```
   > en
   # configure terminal
   (config)# internal set modify -
   /tps/process/hadoop/attribute/mapred.tasktracker.map.tasks.maximum/value value string 8
   (config)# internal set create -
   /tps/process/hadoop/attribute/mapred.compress.map.output/value value string true
   (config)# write memory
   (config)# pm process tps restart
   ```

2. Repeat Step 1 on the standby Collector node.
Verifying that Processes are Running

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 GCR node and verify that all Compute nodes have joined the HDFS cluster. The output shown in the following sample command sequence indicates correct configuration.

```
> en
(config)# _shell
# ps -ef | grep hadoop | awk '{print $NF}'
org.apache.hadoop.hdfs.server.datanode.DataNode
org.apache.hadoop.hdfs.server.namenode.NameNode
hadoop
org.apache.hadoop.hdfs.server.namenode.NameNode
org.apache.hadoop.hdfs.server.namenode.SecondaryNameNode
org.apache.hadoop.mapred.JobTracker
# hadoop 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:** These are internal IP addresses.

```
> en
# conf t
(config)# show pm process collector | grep status
Current status: running
```

2. Log into the standby GCR node and repeat the previous step.
3. Log in to the Insta node, and run the following commands to check on the processes.

```
> en
# conf t
(config)# show pm process insta
Current status: running
(config)# insta infinidb get-status-info
```

**Note:** Processes are running correctly if the output reports `ACTIVE` state for all modules, and `RUNNING` status for all instances and Adapter.

4. Run the following command to verify that Postgres is running on the master GCR node.

```
> en
# _shell
# ps -ef |grep psql
```

The resulting output may resemble:

```
postgres 2990 1 0 timestamp /usr/pgsql-9.2/bin/postmaster -p 5432 -D /data/pgsql/9.2/data
```

5. Log into the standby GCR node and repeat the two previous steps.
Generating and Pushing the Information Bases

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.

Configuring IBs for EDR

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

<table>
<thead>
<tr>
<th>Region</th>
<th>Area</th>
<th>GGSN</th>
<th>GGSNIP</th>
<th>SGSN</th>
<th>SGSNIP</th>
<th>CYBERCITY</th>
<th>APN</th>
<th>broadband</th>
<th>GROUP</th>
<th>broadband</th>
<th>RATID</th>
<th>RATTYPE</th>
<th>3G</th>
</tr>
</thead>
<tbody>
<tr>
<td>MW</td>
<td>USA</td>
<td>GGSN</td>
<td>27.23.157.1</td>
<td>SGSN</td>
<td>22.2.1</td>
<td>SGSN</td>
<td>CYBERCITY</td>
<td>APN</td>
<td>phone</td>
<td>GROUP</td>
<td>phone</td>
<td>RATID</td>
<td>RATTYPE</td>
</tr>
</tbody>
</table>

**Note:** Use the above table for example purposes only. You should use the data that matches your environment. For example, for GGSN, you might use GGSN, PGW, or HA. In this case, GGSNIP is the management IP address. For SGSN, you might use SGSN, SGW, HSGW, or PDSN. In this case, SSGNIP is the service IP address.

To configure the IBs for EDR:

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

   ```
   > en
   # conf t(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:

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

Where `ib-file-name` is:

• ipGgsn
• ipSgsn
• apnGroup

The resulting output may resemble:

```
1  [27.23.157.1][GGSN1]
1  [2.2.2.1][SGSN1]
1  [Sushfone-1][Sushfone-1]
2  [Sushfone-2][Sushfone-2]
3  [Sushfone-3][Sushfone-3]
```

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

5. To configure the IBs for BulkStats:

```plaintext
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

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

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/edr111 and send the Bulkstats file to /data/collector/California/bs111.

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

- 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 filename pattern provided here should be in sync with the Collector configurations. For guidelines on the regex value to be used for the filename format, see the "Verifying Applications Tab and Specifying Component Profiles" section in "Verifying XML Settings " on page 55. Refer to the values in the Wrapper CLI column for sample filename formats.

- 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 (-) (thus, combinations can also be used, such as "_flow-" or "–http_").

To configure gateways:

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

   \[
   \text{pm extension (aggregation_center)}> \text{add gateway name gateway-name}
   \]
   \[
   \quad \text{region gateway-region} \\
   \quad \text{location gateway-area} \\
   \quad \text{schema_version bulkstat-schema-version} \\
   \quad \text{ip gateway-IP} \\
   \quad \text{timezone gateway-timezone} \\
   \quad \text{edr-filename-pattern incoming-EDR-fileName-pattern} \\
   \quad \text{bulkstat-filename-pattern incoming-BS-fileName-pattern} \\
   \quad \text{type gateway-type} \\
   \quad \text{edr-file-path gateway-edr-file-path} \\
   \quad \text{bulkstat-file-path incoming-BS-files-path-on-collector}
   \]

   \textbf{Note:} See "Modifying Gateway Attributes" on page 1 for more information on gateway attributes and a sample of the output generated by this command.

   \textbf{Note:} For \texttt{bulkstat-schema-version}, specify an integer only, for example use 15, do not use 15.0. The supported Bulkstats schema versions are 15, 16, and 17.

   For example:

   \[
   \text{pm extension (aggregation center)}> \text{add gateway name GMPLAB1}
   \]
   \[
   \quad \text{region EAST location USA} \\
   \quad \text{schema_version 15 ip 10.10.10.255} \\
   \quad \text{timezone America/Recife} \\
   \quad \text{edr-filename-pattern *_MURAL-edr_*_%MM%DD%YYYY%hh%mm%ss.*} \\
   \quad \text{bulkstat-filename-pattern} \\
   \quad \text{bulkstats_%YYYY%MM%DD%hh%mm%ss type HA edr-file-path}
   \]
2. Verify the new gateway has been added:

   pm extension (aggregation center)> show gateways

3. Add IPs for all Collectors to push this information to all Collectors:

   pm extension (aggregation center)> set collector IPs comma-separated-ip-list

   **Example:**
   pm extension (aggregation center)> set collector IPs 192.168.1.1,192.168.2.2

   **Note:** These are internal IP addresses.

4. Set and verify the IP addresses for all Collectors:

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

5. 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"

6. Push the gateway configuration to all the Collectors:

   pm extension (aggregation center)> push gateway configuration

7. Generate and push all IBs:

   pm extension (aggregation center)> generate all ibs
   pm extension (aggregation center)> push all ibs
8. Generate and push IBs on Bulkstats:

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

9. Write changes to memory:

```plaintext
(config)# write memory
```

### Copying IBs to the Anomaly Feature

After the above `push` command completes, update all IBs if the Anomaly Detection feature is enabled.

```
[admin@VM1-GCR-1 ~]# pmx
Welcome to pmx configuration environment
pm extension> subshell anomaly
pm extension (anomaly)> update all ibs
```

### Synchronize the IBs on the Standby GCR Node

After the above `push` command completes, run the following command on the standby GCR node from the CLI configure terminal.

1. Go to the `bulkstats` subshell.

```
host [cluster : master|standby](config)# pmx subshell bulkstats
```

2. Fetch all IBs.

```
pm extension (bulk stats)> fetch all ibs from inbox
```

3. Go to the `aggregation_center` subshell.
**MURAL Software Standard Installation Guide for Rack Servers**

<table>
<thead>
<tr>
<th>4.</th>
<th>Fetch all IBs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>**host [cluster : master</td>
<td>standby](config)#**</td>
</tr>
<tr>
<td><strong>pm extension (aggregation_center)&gt;</strong></td>
<td><strong>fetch all ibs from inbox</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5.</th>
<th>If Anomaly Detection is enabled, go to the anomaly subshell.</th>
</tr>
</thead>
<tbody>
<tr>
<td>**host [cluster : master</td>
<td>standby](config)#**</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6.</th>
<th>Fetch all IBs.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pm extension (anomaly)&gt;</strong></td>
<td><strong>fetch all ibs from inbox</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7.</th>
<th>Run the <strong>quit</strong> command twice to exit and run the following command to write changes to memory.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(config)#</strong></td>
<td><strong>wr mem</strong></td>
</tr>
</tbody>
</table>
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. Edit `/etc/ssh/sshd_config` to set the following parameters, in case a PAM related authentication error is reported in the master Collector node `/var/log/messages` for the ASR:

   ```
   # mount -o remount,rw /
   # vi /etc/ssh/sshd_config
   UsePAM no
   PasswordAuthentication yes
   ```

3. Set the sshd_config file attribute as immutable.

   ```
   # chattr +i /var/opt/tms/output/sshd_config
   ```

   Verify the sshd_config file attribute as below.

   ```
   # lsattr /var/opt/tms/output/sshd_config
   -----i-------- /var/opt/tms/output/sshd_config
   ```
4. Run the `sshd restart` command:

```
# en
> conf t
(config) pm process sshd restart
```

5. Repeat steps 1 through 3 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.

**Note:** It is assumed that the timestamp on the data that is pushed to the platform is greater than or equal to the current time, and not an old timestamp.
Validating Data on Nodes

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

Validating Data on the Collector Nodes

1. Log in to the master Collector node and go to `bash`

   ```
   > 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 PROD-NAME-SHORT 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 GCRnode and make the `/` 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 "Validating Data on the Collector Nodes" on the previous page.

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."

```bash
#!/ 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 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 "Starting the Data Processing" on the previous page. This allows sufficient time for the jobs that process the collector data to start, and the done.txt files to be updated. Not waiting could result in the checks failing.

**Validating EDR Data**

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

   ```
   > en
   # _shell
   ```

2. Check the last timestamp for the Core job.

   ```
   > en
   # _shell
   # hadoop dfs -text /data/CoreJob/done.txt 2>/dev/null
   ```

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

   ```
   # hadoop dfs -text /data/EDR/done.txt 2>/dev/null
   ```

4. Check the last timestamp for CubeExporter data cubes.

   ```
   # hadoop dfs -text /data/CubeExporter/done.txt 2>/dev/null
   ```

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

   ```
   # hadoop dfs -text /data/BulkStat/done.txt 2>/dev/null
   # hadoop dfs -text /data/BSAgg15min/done.txt 2>/dev/null
   # hadoop dfs -text /data/BulkStatExporter_15min/done.txt 2>/dev/null
   ```

**Validating Insta Data**

1. Log in to 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 0 cubes-database" | awk -F ' ' '{print $5}'
    database_mural

2. Open the `idbmysql` user interface and select the database.

    # idbmysql
    Welcome to the MySQL monitor. Commands end with ; or \g.
    ...
    mysql> use database_mural;
    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).

    mysql> select * from bin_metatable;
    +-----------------+-----------------+---+---------+---+--------+
    | binclass        | aggregationinterval | mints | maxts   | bintype|
    +-----------------+-----------------+---+---------+---+--------+
    | 60min           | -1              | 1350126000 | 1350594000 | NULL | |
    | 60min           | 86400           | 1350086400 | 1350432000 | NULL | |
    | 60min           | 604800          | 0         | 0        | NULL | |
    | 60min           | 2419200         | 0         | 0        | NULL | |
    +-----------------+-----------------+---+---------+---+--------+
    4 rows in set (1.14 sec)
    Press Ctrl+D to exit
    mysql> Bye

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

    The following example indicates that data was processed between 11:00 on October 13 and 21:00 on October 18.

    # date -d @1350126000
    Sat Oct 13 11:00:00 UTC 2012
Validating Bulk Stats Data on the Insta Blade

1. Use SSH to log in to 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 '{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;
<table>
<thead>
<tr>
<th>binclass</th>
<th>aggregationinterval</th>
<th>mints</th>
<th>maxts</th>
<th>binType</th>
</tr>
</thead>
<tbody>
<tr>
<td>5min</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>NULL</td>
</tr>
<tr>
<td>5min</td>
<td>900</td>
<td>1364713200</td>
<td>1367293500</td>
<td>NULL</td>
</tr>
<tr>
<td>5min</td>
<td>3600</td>
<td>1364713200</td>
<td>1365004800</td>
<td>NULL</td>
</tr>
<tr>
<td>5min</td>
<td>86400</td>
<td>1364688000</td>
<td>1364860800</td>
<td>NULL</td>
</tr>
<tr>
<td>5min</td>
<td>604800</td>
<td>0</td>
<td>0</td>
<td>NULL</td>
</tr>
<tr>
<td>5min</td>
<td>2419200</td>
<td>0</td>
<td>0</td>
<td>NULL</td>
</tr>
</tbody>
</table>
6 rows in set (12.18 sec)
```

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 07:00 on March 31 and 03:45 on April 30.

<table>
<thead>
<tr>
<th>Command</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>date -d @1367293500</code></td>
<td>Tue Apr 30 03:45:00 UTC 2013</td>
</tr>
<tr>
<td><code>date -d @1364713200</code></td>
<td>Sun Mar 31 07:00:00 UTC 2013</td>
</tr>
</tbody>
</table>

**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 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 in to the master GCR node.

   `> en`  
   `# conf t`

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
- anomaly (if Anomaly 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 in to 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 via GMS for the UI nodes configuration details. For example:

**https://demo.sanmateo.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:
MURAL Software Standard Installation Guide for Rack Servers

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

For example:

https://demo.sanmateo.com:20443/
https://demo.sanmateo.com:30443/

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

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

**Note:** Once the installation is completed, be sure to 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:

```
# pmx subshell aggregation_center
pm extension (aggregation center)> generate whitelist
pm extension (aggregation center)> setting attribute (jobStart)
setting attribute (jobEnd)
INFO: GenericJobScheduler: starting job
job: 0000000-140809225350409-oozie-admi-C
INFO: GenericJobScheduler: job started successfully
Running Whitelist Creation Job
pm extension (aggregation center)> quit
pm extension> 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 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.

Tethering Reports

Create a file called serverFile_tethering with details of the ASR 5000 gateways, where the TAC, OS or UA databases, 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:

   
<table>
<thead>
<tr>
<th>Gateway-IP, gateway-username, gateway-password, location-to-copy-reports</th>
</tr>
</thead>
</table>

   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 `serverFile_tethering` file:

   ```
   # 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 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.
Rule-Based Reports

Create a file named `serverfile_Rulebase` with details of ASR IPs, access details and report destination paths.

1. The `serverFile_Rulebase` 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 `serverFile_tethering` file:

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

   **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.

**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
pm extension (aggregation center)> setting attribute (jobStart)
setting attribute (jobEnd)
INFO: GenericJobScheduler: starting job
job: 0000000-140809225350409-oozie-admi-C
INFO: GenericJobScheduler: job started successfully
Running Whitelist Creation Job
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.

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-start-time
- sn-end-time
- radius-calling-station-id
- radius-called-station-id
- sn-app-protocol
- sn-rulebase
- p2p-protocol
- sn-server-port
- sn-subscriber-port
- sn-direction
- traffic-type
- bearer-3gpp rat-type
- bearer-3gpp imei
- 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
- tethered

Sample:

```
```
Mural Software Standard Installation Guide for Rack Servers

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
- radius-called-station-id
- http-host
- http-url
- http-user-agent
- bearer-3gpp rat-type
- bearer-3gpp imei
- http reply code

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-
```

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

**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 1
   rule-variable ip subscriber-ip-address priority 4
   attribute sn-subscriber-port priority 6
   attribute sn-start-time format seconds priority 10

   attribute sn-direction priority 21
   rule-variable bearer ggsn-address priority 23
   rule-variable bearer 3gpp2 bsid priority 24
   attribute sn-flow-start-time format seconds priority 26
   attribute sn-flow-end-time format seconds priority 27

   attribute radius-calling-station-id priority 30
   rule-variable p2p protocol priority 31
   rule-variable bearer 3gpp imsi priority 35
   attribute radius-called-station-id priority 40
   rule-variable ip server-ip-address priority 60
   attribute sn-server-port priority 70
   attribute sn-app-protocol priority 80
   attribute sn-parent-protocol priority 81
   rule-variable ip protocol priority 82
   attribute sn-volume-amt ip bytes uplink priority 100
   attribute sn-volume-amt ip bytes downlink priority 110
   attribute sn-volume-amt ip pkts uplink priority 120
   attribute sn-volume-amt ip pkts downlink priority 130
```
MURAL Software Standard Installation Guide for Rack Servers

rule-variable bearer 3gpp imei priority 141
rule-variable bearer 3gpp rat-type priority 142
rule-variable bearer 3gpp user-location-information priority 143
rule-variable bearer 3gpp sgsn-address priority 144
rule-variable traffic-type priority 160

attribute sn-end-time format seconds priority 180
rule-variable tcp os-signature priority 190
rule-variable tcp tethered priority 191

#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 user-agent priority 55
rule-variable http url priority 60
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 141
rule-variable bearer 3gpp rat-type priority 142
rule-variable http reply code priority 160
rule-variable http referer priority 170

# exit
Appendix I: Setting Up Virtual Drives on Rack Server

Use the following steps to create and set up the virtual drives on the rack server.

1. Log into CIMC using admin username and password.

![CIMC Login Screen]

2. From the Server Status view, power on the server and select **Launch KVM Console** link.

![Server Status View]

3. Wait until the RAID controller configuration passes on the command screen.
4. From the CIMC interface, select **Storage** tab (left pane) > **Controller Info** (right pane).
5. Press **Create Virtual Drive from Unused Physical Drives**.
6. Select RAID Level as **RAID-5**, check the box for all disks under **Physical Drive** except one and then press the **>>** button.

7. Select **Device Group**, assign a **Virtual Drive Name**, ensuring to set
Write Policy as Write back and keep all other values as shown above.

Press Create Virtual Drive button.

8. Once the progress bar disappears, select Virtual Drive Info and verify a single virtual drive using multiple physical hard drives in RAID-5 configuration was created. Verify that Current Write Cache Policy is set to Write Back.

![Virtual Drive Info](image1)

9. Click Controller Info tab and ensure the Health status is Good.

![Controller Info](image2)

10. After successful Virtual Drive creation, select Physical Drive Info tab. All hard disks should be Online except one that is still in an Unconfigured status.

![Physical Drive Info](image3)
11. Select the unconfigured disk and under the **Actions** section, click on **Make Dedicated Hot Spare** link.

12. Press **Make Dedicated Hot Spare** button to create one disk as host spare disk.

13. Under the **Physical Drive Info** tab, the last disk should appear with Dedicated Hot Spare Status.

14. Click on **Virtual Drive Info** tab and select a virtual drive. Then click **Set**
as **Boot Drive** link.

15. Press **OK** to set this virtual drive as boot device.

16. Ensure the value under Boot Drive displays as true.

17. Now the virtual drive is ready to use for OS installation on the rack server.
## 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.
**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).
 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.