



MURAL Software Upgrade Guide

Version: 5.0.2.p6

Last Updated: 28-07-2021

Copyright © 2021, 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 NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO 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.

All rights reserved.

Table Of Contents

1. Introduction	1
2. Prerequisites	2
2.1 Availability of Root Credentials for MoP	2
2.2 Check the PostgreSQL Health	2
2.3 Download RPMs	3
2.4 Compare md5sum	4
3. Pre-Upgrade Procedure	5
3.1 Take the Backup	5
3.2 Update the Repositories	6
3.3 Stop the Running Jobs	10
4. Upgrade Procedure	18
4.1 Running the installer	18
4.2 Verify Core Analytics Report Configurations	19
5. Post-Upgrade Procedure	22
5.1 Generate the encrypted password	22
5.2 Use encrypted password	22
5.3 Extracting the MURAL Job Scheduler Scripts	23
6. Start the Jobs	24
6.1 Clear the checkpoint	24
6.2 Run the Master Job	24
6.3 Run the Talend Job	25
6.4 Run the Aggregation Job	25
7. Cemus Report Verification	28

7.1 Verify the DailyCemusReport	32
7.2 Verify the MonthlyCemusReport	34
7.3 Verify the GRTCemusReport	35

1. Introduction

Mobility Unified Reporting and Analytics (MURAL) is a next-gen analytic solution, tailored for telecoms. This next-gen analytic solution, is multi-dimensional and has been enhanced to provide visibility into each subscriber's behavior and usage. The MURAL application requires nodes, such as

- Name node - To collect the data streams
- Compute nodes - To analyze and aggregate the data.
- Load Balancer and Management node - To manage the installation of all other nodes.

This document provides the step-by-step instructions on upgrading the MURAL from version 5.0.2.p5 to version 5.0.2.p6 .

Note: In this document, there are several files with long path names. When you copy-paste these paths on terminal, an extra space or hyphen is inserted in between. You must delete these extra characters manually. For more details on this issue, refer to the release notes.

2. Prerequisites

Ensure that the following prerequisites are met before you begin the upgrade procedure.

2.1 Availability of Root Credentials for MoP

Ensure that you have root credentials to execute the upgrade steps.

2.2 Check the PostgreSQL Health

Run the following command from management node to check the PostgreSQL health of your system:

```
crm_mon -Arfl
```

The following sample may resemble the output:

```
* Node mural-mgt-1.us.guavus.com:
+ master-pgsql : 10
+ pgsql-data-status : STREAMING|POTENTIAL
+ pgsql-master-baseline : 0000001285000090
+ pgsql-receiver-status : normal
+ pgsql-status : HS:alone
+ pgsql-xlog-loc : 000000128427F600
* Node mural-nn-1.us.guavus.com:
+ master-pgsql : 1000
+ pgsql-data-status : LATEST
+ pgsql-master-baseline : 0000001247000090
+ pgsql-receiver-status : normal (master)
+ pgsql-status : PRI
+ pgsql-xlog-loc : 000000128173E718
* Node mural-nn-2.us.guavus.com:
+ master-pgsql : 100
+ pgsql-data-status : STREAMING|SYNC
+ pgsql-receiver-status : normal
```



```
+ pgsql-status : HS:sync
+ pgsql-xlog-loc : 000000128173E888
```

Note: As illustrated above, all the nodes in PostgreSQL cluster must show status as **normal**.

2.2.1 Installing Ansible

Note: In the previous release of MURAL, Ansible and jinja2 were removed from the management node. In order to upgrade your system, you must re-install them.

Perform the following steps to install Ansible and jinja2

1. Run the following command to verify if Ansible and jinja packages are already installed or not:

```
pip list | grep -iE 'jinja2|ansible'
```

2. If Ansible 2.3.1.0 and Jinja 2.8.1 are not found on the management node, run the following command to install them:

```
cd /etc/reflex-provisioner/packages/pip
pip install ansible-2.3.1.0.tar.gz \
Jinja2-2.8.1-py2.py3-none-any.whl
```

3. Run the following command to verify if both the components are installed successfully:

```
pip list | grep -iE 'jinja2|ansible'
```

2.2.2 Verify all the services

To verify all the services are up and running, run the following command:

```
cd /etc/reflex-provisioner
```

```
ansible-playbook -i inventory/generated/prod/mural/hosts \
playbooks/platform/service_checks/all.yml -k
```

2.3 Download RPMs

Perform the following steps to download the RPMs:

1. Log into the management node.
2. Download the patch artifacts to `/opt/repos/mrx/5.6/5.6.2.rc1/` location from SFTP Server.

Note: Refer to the Release Notes of this patch for the complete list of artifacts. Contact technical support for SFTP access credentials.

2.4 Compare md5sum

Run the following command to verify the integrity of the copied packages and compare the values of `md5sum` of RPMs with the downloaded artifacts available in the Release Notes.

```
# md5sum *
```

The following sample may resemble the output:

```
f0a16490d8b9ac51051297a5feef98e0  mural5Jobs.tgz
be5d55e3a7eb26146a98ec8d071e600a  app_protocol_list.gz
c141d5fcec869a7a93b3c934948b903a  mrx-docker-release-5.6.2.rc1-
312.tar.gz
8b1488bf7c4b17de5b548d5c7901e9cb  reflex-aggregation-5.6.2.rc1-
312.el7.centos.x86_64.rpm
ccea72a7b33a930320ae55298346d0cc  reflex-datafactory-5.6.2.rc1-
312.el7.centos.x86_64.rpm
2522c99dc504ebadef1f120a78d5b668  reflex-solution-provisioner-
5.6.2.rc1-312.el7.centos.x86_64.rpm
```

3. Pre-Upgrade Procedure

You must complete all the tasks in the order as specified before you begin the upgrade procedure:

1. "Take the Backup" below
2. "Update the Repositories" on the next page
3. "Stop the Running Jobs" on page 10

3.1 Take the Backup

Before you install this patch and begin to upgrade MURAL to version 5.0.2.p6, ensure that you have backed up all the critical files from the last version installed that is 5.0.2.p5.

Note: In this section, there are several files with long path names. When you copy-paste these paths on terminal, an extra space or hyphen is inserted in between. You must delete these extra characters manually. For more details on this issue, refer to the MURAL v5.0.2.p6 Release Notes.

1. Run the following commands to take the backup of the repositories on the management node:

```
mkdir ~/mural_5.0.2.p5_backup  
cd ~/mural_5.0.2.p5_backup/
```

```
cp -r -p /etc/reflex-  
provisioner/inventory/generated/prod/mural/group_vars/all/mrx  
.
```

```
cp -r -p /etc/reflex-  
provisioner/inventory/generated/prod/mural/vars/customer/commo  
n .
```

2. Run the following commands to take the backup of the streaming directory from HDFS on management node:

```
hdfs dfs -get /data/streaming data_streaming
```

3. Run the following commands to take the backup of `/opt/etc/scripts` directory on both the master nodes:

```
mkdir ~/mural_5.0.2.p5_backup
cd ~/mural_5.0.2.p5_backup/
cp -r -p /opt/etc/scripts .
```

3.2 Update the Repositories

Before updating the repositories, ensure that all the packages are downloaded from the SFTP server. For more information, refer to ["Prerequisites" on page 2](#). And, perform the following steps to update the repository with updated RPM's:

3.2.1 Update the Docker Images

Perform the following steps to extract the docker images:

1. Navigate to the `/mrX/5.6/` repository

```
cd /opt/repos/mrx/5.6/
```

2. Run the following commands:

```
mv 5.6.2.rc1/mrx-docker-release-5.6.2.rc1-312.tar.gz .

tar -zxvf mrx-docker-release-5.6.2.rc1-312.tar.gz -C \
/opt/repos/mrx/5.6/mrx-docker-5.6.2.rc1/
```

3.2.2 Update the Repository

Perform the following steps to update a repository:

1. Run the following command:

```
createrepo /opt/repos/mrx/5.6/5.6.2.rc1/
```

The following sample may resemble the output:

```
Spawning worker 0 with 8 pkgs
Spawning worker 1 with 8 pkgs
Spawning worker 2 with 8 pkgs
```

```
Spawning worker 3 with 8 pkgs
Spawning worker 4 with 8 pkgs
Spawning worker 5 with 8 pkgs
Spawning worker 6 with 8 pkgs
Spawning worker 7 with 8 pkgs
Workers Finished
Saving Primary metadata
Saving file lists metadata
Saving other metadata
Generating sqlite DBs
Sqlite DBs complete
```

3.2.3 Install Solution Provisioner Package

Note: In this section, there are several files with long path names. When you copy-paste these paths on terminal, an extra space or hyphen is inserted in between. You must delete these extra characters manually. For more details on this issue, refer to the MURAL v5.0.2.p6 Release Notes.

To replace the default configurations with the backup configuration files, run the following command:

```
yum install -y /opt/repos/mrx/5.6/5.6.2.rc1/reflex-solution-
provisioner-5.6.2.rc1-312.el7.centos.x86_64.rpm
```

3.2.4 Update Variable

Perform the following steps to update the value of variables such as `install_type` and tag version number:

1. Change `install_type` variable:
 - a. Open the file `/etc/reflex-provisioner/work_dir/reflex-solution-provisioner/inventory/templates/group_vars/global/all/mrx/all.yml`
 - b. Change `install_type` value to `upgrade`.

2. Edit `/etc/reflex-provisioner/work_dir/reflex-solution-provisioner/inventory/templates/group_vars/-global/all/mrx/ingestion/main.yml` file to update the following properties:

```
source_file_mask: "*http*.gz"
talend_http_topic: talendhttp
talend_nonhttp_topic: talendnonhttp
hdfs_dir: /user/mrx/ingestion/
hdfs_dir2: /user/mrx/ingestion/
```

3. Edit `/etc/reflex-provisioner/work_dir/reflex-configuration-module/conf/generate_inventory/conf_inventory/prod/mural/extra_vars/solution.yml` file to update / add the following properties:

```
source_file_mask: "*http*.gz"
source_file_mask2: "*flow*.gz"
odsClassification: true
```

4. Edit `extract.conf.j2` file available at `/etc/reflex-provisioner/work_dir/reflex-solution-provisioner/roles/mrx/extractJobs/deploy/templates/opt/mrx/ingestion/etc2` location to update the following properties:

```
source_file_mask = {{ source_file_mask2 }}
```

Note: Update the property values of `dme`, `5minAgg`, `hourlyAgg`, `dailyAgg`, `monthlyAgg` table names as per your setup.

5. Edit `/etc/reflex-provisioner/work_dir/reflex-solution-provisioner/inventory/templates/group_vars/-global/all/mrx/dme/main.yml` file to update the following properties:

```
kafkaTopicHttpPDM: httpPDM_new
kafkaTopicNonHttpPDM: nonhttpPDM_new
```

6. Edit `/etc/reflex-provisioner/work_dir/reflex-solution-provisioner/inventory/templates/group_vars/-global/all/mrx/5minAgg/main.yml` file to update the following property:

```
pdm_agg_output_table: 5min_points_new
```

7. **Edit** `/etc/reflex-provisioner/work_dir/reflex-solution-provisioner/inventory/templates/group_vars/-global/all/mrx/hourlyAgg/main.yml` file to update the following property:

```
hourly_output_tableName: hourly_points_new
```

8. **Edit** `/etc/reflex-provisioner/work_dir/reflex-solution-provisioner/inventory/templates/group_vars/-global/all/mrx/dailyAgg/main.yml` file to update the following property:

```
daily_output_tableName: daily_points_new
```

9. **Edit** `/etc/reflex-provisioner/work_dir/reflex-solution-provisioner/inventory/templates/group_vars/-global/all/mrx/monthlyAgg/main.yml` file to update the following property:

```
monthly_output_tableName: monthly_points_new
```

Note: Perform Step 10 and 11, only if Grafana UI opens from secure web protocol (https).

10. **Edit** `/etc/reflex-provisioner/work_dir/reflex-platform-module/inventory/templates/group_vars/-global/all/platform/grafana.yml` file to update `platform_grafana_api_url` property value from **http** to **https** as follows:

```
platform_grafana_api_url: "https://{{ platform_grafana_access_ip }}:{{ platform_grafana_web_ui_port }}"
```

11. **Edit** `/etc/reflex-provisioner/work_dir/reflex-solution-provisioner/roles/mrx/grafana/deploy/tasks/main.yml` to add `validate_certs: no` parameter below `url: parameter` line as follows:

```
url: '{{ platform_grafana_api_url }}/api/dashboards/db'
validate_certs: no
method: POST
```

3.2.5 Refresh the inventory

Perform the following steps to refresh the inventory:

1. Navigate to the directory, `reflex-provisioner`

```
cd /etc/reflex-provisioner
```

2. Run the following command to refresh the inventory:

```
./scripts/composition/refresh.sh -i mural -s prod
```

Output:

```
-i mural was triggered!
-s prod was triggered!
Refreshing init inventory
Refreshing mural inventory
```

3.3 Stop the Running Jobs

1. Stop the input data flow.
2. Stop all the running jobs before executing solution installer:
 - a. Log in to the name node where jobs are running. For example, jobs in Mural 5 lab run from the active namenode, NN2.

```
# ssh <NN2 FQDN>
```

- b. Stop Talend jobs on the active name node. Perform the following steps to check if the Talend job is running and to stop them respectively:

- i. Check status of talend http process:

```
# ps -ef | grep talend-http | grep -v grep
```

The following sample may resemble the output:

```
root 51484 45694 0 06:59 pts/2 00:00:00 sh
/root/jobs/ingestion_jobs/run-talend-http-job.sh
```


- ii. Kill talend http process if it is running:

```
# kill -9 <PID's>
```

Example:

```
# kill -9 51484
```

- iii. Check status of talend non-http process:

```
# ps -ef | grep talend-nonhttp | grep -v grep
```

The following sample may resemble the output:

```
root 51483 45693 0 06:59 pts/2 00:00:00 sh
/root/jobs/ingestion_jobs/run-talend-nonhttp-job.sh
```

- iv. Kill talend non-http process if it is running:

```
# kill -9 <PID's>
```

Example:

```
# kill -9 51483
```

- c. Verify if the jobs are killed

```
# ps -ef | egrep 'talend-http|talend-nonhttp' | grep -v
grep
```

- d. Stop the master job.

```
# ps -ef | grep master_http |grep -v grep
```

The following sample may resemble the output

```
root 6420 60609 3 07:05 pts/2 00:09:22
/usr/java/latest/bin/java -cp /usr/lib/spark2/jars/netty-
all-
4.0.42.Final.jar:/usr/lib/spark2/jars/*:/opt/tms/java/DataM
ediationEngine/WEB-INF/classes:/opt/tms/java/dme-with-
dependencies.jar:/opt/tms/java/ddj-with-
```

```
dependencies.jar:/usr/lib/hive/lib/*:/usr/lib/spark2/conf/
:/usr/lib/spark2/jars/*:/etc/hadoop/conf:/etc/hadoop/conf
:/usr/lib/hadoop/lib/*:/usr/lib/hadoop/.//*:/usr/lib/hado
op-hdfs/./:/usr/lib/hadoop-hdfs/lib/*:/usr/lib/hadoop-
hdfs/.//*:/usr/lib/hadoop-yarn/lib/*:/usr/lib/hadoop-
yarn/.//*:/usr/lib/hadoop-mapreduce/lib/*:/usr/lib/hadoop-
mapreduce/.//* -Xmx2g -XX:-ResizePLAB
org.apache.spark.deploy.SparkSubmit --master yarn-client -
-conf
spark.scheduler.allocation.file=/opt/tms/java/DataMediatio
nEngine/WEB-INF/classes/poolConfig.xml --conf
spark.driver.extraJavaOptions=-XX:-ResizePLAB --
properties-file /opt/tms/java/DataMediationEngine/WEB-
INF/classes/spark.properties --class
com.guavus.reflex.marketing.dme.job.MRXMasterJob --name
master_http --queue jobs.dme --files
/opt/tms/java/DataMediationEngine/WEB-INF/classes/log4j-
executor.properties,/opt/tms/java/DataMediationEngine/WEB-
INF/classes/streaming.ini --jars /opt/tms/java/dme-with-
dependencies.jar /opt/tms/java/dme-with-dependencies.jar
root 60609 45694 0 07:02 pts/2 00:00:00 sh
/root/jobs/streaming_jobs/master_http_wrapper.sh
```

- Obtain the ID of the pod from the preceding output and run the following command:

```
# kill -9 <PID's>
```

Example:

```
# kill -9 6420 60609
```

- Wait for counters in the `/var/log/mural_logs/master_http.out` file. Once counters show as 0, then proceed to stop the `master_nonhttp` job:

```
# ps -ef | grep master_nonhttp |grep -v grep
```

The following sample may resemble the output:

```
root 61263 45694  0 07:03 pts/2    00:00:00 sh
/root/jobs/streaming_jobs/master_nonhttp_wrapper.sh

root 61349 61263  9 07:03 pts/2    00:24:55
/usr/java/latest/bin/java -cp /usr/lib/spark2/jars/netty-
all-
4.0.42.Final.jar:/usr/lib/spark2/jars/*:/opt/tms/java/DataM
ediationEngine2/WEB-INF/classes:/opt/tms/java/dme-with-
dependencies.jar:/opt/tms/java/ddj-with-
dependencies.jar:/usr/lib/hive/lib/*:/usr/lib/spark2/conf:/
/usr/lib/spark2/jars/*:/etc/hadoop/conf:/etc/hadoop/conf:/
/usr/lib/hadoop/lib/*:/usr/lib/hadoop/.//*:/usr/lib/hadoop-
hdfs/./:/usr/lib/hadoop-hdfs/lib/*:/usr/lib/hadoop-
hdfs/.//*:/usr/lib/hadoop-yarn/lib/*:/usr/lib/hadoop-
yarn/.//*:/usr/lib/hadoop-mapreduce/lib/*:/usr/lib/hadoop-
mapreduce/.//* -Xmx2g -XX:-ResizePLAB
org.apache.spark.deploy.SparkSubmit --master yarn-client --
conf
spark.scheduler.allocation.file=/opt/tms/java/DataMediation
Engine2/WEB-INF/classes/poolConfig.xml --conf
spark.driver.extraJavaOptions=-XX:-ResizePLAB --properties-
file /opt/tms/java/DataMediationEngine2/WEB-
INF/classes/spark.properties --class
com.guavus.reflex.marketing.dme.job.MRXMasterJob --name
master_nonhttp --queue jobs.dme --files
/opt/tms/java/DataMediationEngine2/WEB-INF/classes/log4j-
executor.properties,/opt/tms/java/DataMediationEngine2/WEB-
INF/classes/streaming.ini --jars /opt/tms/java/dme-with-
dependencies.jar /opt/tms/java/dme-with-dependencies.jar
```

- Obtain the ID of the pod from the preceding output and run the following command:

```
# kill -9 <PID's>
```

Example:

```
# kill -9 61263 61349
```

- Verify if the jobs are killed:

```
ps -ef | egrep 'master_http|master_nonhttp' | grep -v \
grep
```

The following sample may resemble the output:

```
# ps -ef | egrep 'master_http|master_nonhttp' | grep -v
grep
```

- Wait for counters in the /var/log/mural_logs/master_non-http.out file. Once counters show as 0, then proceed to stop Aggregation job.

e. Stop CONV and SDR jobs:

```
# ps -ef | grep conv_config |grep -v grep
```

The following sample may resemble the output:

```
# ps -ef | grep conv_config | grep -v grep

root 38891 1 0 Apr22 ? 00:00:00 sh /root/jobs/aggregation_
jobs/run-conv_config_file.sh
```

- Obtain the ID of the pod from the preceding output and run the following command:

```
# kill -9 <PID's>
```

Example:

```
# kill -9 38891
```

- Find sdr process:

```
ps -ef | grep sdr_config |grep -v grep
```

The following sample may resemble the output:

```
# ps -ef | grep sdr_config | grep -v grep
```

```
root 55161 1 0 Apr22 ? 00:00:00 /bin/bash
/root/jobs/aggregation_jobs/run-sdr_config_file.sh
```

- Obtain the ID of the pod from the preceding output and run the following command:

```
# kill -9 <PID's>
```

Example:

```
# kill -9 55161
```

f. Stop the 5 minutes Aggregation Job:

```
ps -ef | grep 5min-agg |grep -v grep
```

The following sample may resemble the output:

```
# ps -ef | grep 5min-agg |grep -v grep

root 5131 45694 0 07:05 pts/2 00:00:00 sh
/root/jobs/aggregation_jobs/run-5min-agg-mgr_sh.sh
```

- Obtain the ID of the pod from the preceding output and run the following command:

```
# kill -9 <PID's>
```

Example:

```
#kill -9 5131
```

g. Stop the hourly Aggregation Job:

```
ps -ef | grep hourly-agg |grep -v grep
```

The following sample may resemble the output:

```
# ps -ef | grep hourly-agg | grep -v grep

root 17868 45694 0 07:10 pts/2 00:00:00 sh
/root/jobs/aggregation_jobs/run-hourly-agg-mgr_sh.sh
```

- Obtain the ID of the pod from the preceding output and run the following command:

```
# kill -9 <PID's>
```

Example:

```
#kill -9 17868
```

h. Stop the daily Aggregation Job:

```
ps -ef | grep daily-agg |grep -v grep
```

The following sample may resemble the output:

```
# ps -ef | grep daily-agg | grep -v grep

root      19338 74594  0 07:10 pts/2    00:00:00 sh
/root/jobs/aggregation_jobs/run-daily-agg-mgr_sh.sh
```

- Obtain the ID of the pod from the preceding output and run the following command:

```
# kill -9 <PID's>
```

Example:

```
#kill -9 19338
```

i. Stop the monthly Aggregation Job:

```
ps -ef | grep monthly-agg |grep -v grep
```

The following sample may resemble the output:

```
# ps -ef | grep monthly-agg | grep -v grep

root 16543 55644 0 07:10 pts/2    00:00:00 sh
/root/jobs/aggregation_jobs/run-monthly-agg-mgr_sh.sh
```

- Obtain the ID of the pod from the preceding output and run the following command:

```
# kill -9 <PID's>
```

Example:

```
#kill -9 16543
```

- j. Verify if the aggregation jobs are killed:

```
ps -ef | egrep '5min-agg|hourly-agg|daily-agg|monthly-agg'|
grep -v grep
```

The following sample may resemble the output:

```
# ps -ef | egrep '5min-agg|hourly-agg|daily-agg|monthly-
agg'| grep -v grep
```

- k. Stop the cleanup job

```
# ps -ef | grep cleanup |grep -v grep
```

The following sample may resemble the output:

```
# ps -ef | grep cleanup | grep -v grep

root      11249    1  0 Jun23 ?      00:00:00 sh /root/jobs/misc_
jobs/run_cleanup_job.sh
```

Obtain the ID of the pod from the preceding output and run the following command:

```
# kill -9 <PID's>
```

Example:

```
#kill -9 11249
```

- l. Run the following command to ensure that jobs are not running in Yarn:

```
# yarn application -list
```

The following sample may resemble the output:

```
# yarn application -list
```

```
Total number of applications (application-types: [] and
states: [SUBMITTED, ACCEPTED, RUNNING]):2 Application-Id
Application-Name Application-Type User Queue State Final-
State Progress Tracking-URL
```

```
#yarn application -kill <Application-Id>
```

4. Upgrade Procedure

This section describes the upgrade procedure when upgrading MURAL from version 5.0.2.p5 to version 5.0.2.p6.

4.1 Running the installer

Perform the following steps on management node to upgrade MURAL from version 5.0.2.p5 to version 5.0.2.p6:

1. Run the following command to navigate to the `reflex-provisioner` directory:

```
cd /etc/reflex-provisioner
```

2. Run the solution installer using following command:

```
ansible-playbook -i \
inventory/generated/prod/mural/hosts \
playbooks/mrx/deploy.yml -k --skip-tag azkacli
```

3. Run the following command to edit the value of `odsClassification` in *tomcat-mrxui* config map from `True` to **true**.

```
kubectl edit cm tomcat-mrxui
```

4. Run the following command to validate the value updated in Step-3:

```
kubectl describe cm tomcat-mrxui | grep odsClassification
```

The sample output may resemble as follows:

```
"odsClassification\":"true\n"}\n"
```

5. Run the following command to restart the pods:

```
kubectl get po | grep tomcat-mrxui | awk '{print $1}' | xargs
kubectl delete po
```

6. To verify that the preceding step is successfully executed and the UI is up and running, enter the URL <https://<lb-vip-fqdn>/login> in the browser.

7. Run the following command to update the `app_protocol_list` file on both the master nodes and on HDFS:

For master nodes:

```
gunzip /opt/repos/mrx/5.6/5.6.2.rc1/app_protocol_list.gz
scp /opt/repos/mrx/5.6/5.6.2.rc1/app_protocol_list <master-
nodes>:/opt/etc/scripts/
```

For HDFS:

```
hdfs dfs -put -f /opt/repos/mrx/5.6/5.6.2.rc1/app_protocol_list
/data/streaming/protocol-values.txt
```

8. Run the following command to copy job scheduler scripts to both the master nodes:

```
scp /opt/repos/mrx/5.6/5.6.2.rc1/mural5Jobs.tgz <master-nodes>:.
```

4.2 Verify Core Analytics Report Configurations

Perform the following steps on both the master nodes in the same order to verify and update the CAR files:

1. Run the following command to verify `gds_list` file:

```
cat /opt/etc/scripts/gds_list
```

The sample output may resemble the following:

```
VoIP
web applications
streaming applications
file transfer
peer to peer
IM
```

2. Run the following command to verify `ods_list` file:

```
cat /opt/etc/scripts/ods_list
```

The sample output may resemble as follows:

```
amazon-video
netFlix
NowTV
sky-sport-streaming
```

3. Run the following command to verify `app_protocol_list` file:

```
head /opt/etc/scripts/app_protocol_list
```

The sample output file may resemble as follows:

```
youtube=YouTube
https=HTTPS
google-play=Google Play
whatsapp-voice=WhatsApp Voice
whatsapp-transfer=WhatsApp Transfer
```

Note: The sample output generated is in form of key-value pair wherein key is the actual value generated in aggregation table and value is the name of protocol used for representation. If the key generated from aggregation job is different then update the `app_protocol_list` file based on key generated by your file.

4. Run the following command to verify `monthly_conf_vodafone` file:

```
cat /opt/etc/scripts/monthly_conf_vodafone
```

The sample output file may resemble as follows:

```
table_name=kafkaconnectdb.monthly_points
output_dir=/data/customer
run_duration=monthly
hive_server=192.168.193.15
hive_port=10000
smtp_server=192.168.104.25
server_port=25
sender_address=support-mrx@guavus.com
receiver_addresses=receiver-address1@gdomain.com, receiver-
```

```
address2@gdomain.com  
ods3_file=/opt/etc/scripts/ods_list
```

Note: You can change the config file based on your set up and email details.

5. Run the following command to verify `weekly_conf_vodafone` file:

```
cat /opt/etc/scripts/weekly_conf_vodafone
```

The sample output may resemble as follows:

```
table_name=kafkaconnectdb.hourly_points  
output_dir=/data/customer  
run_duration=weekly  
hive_server=192.168.193.15  
hive_port=10000  
smtp_server=192.168.104.25  
server_port=25  
sender_address=support-mrx@guavus.com  
receiver_addresses=receiver-address1@gdomain.com,receiver-  
address2@gdomain.com  
gds_file=/opt/etc/scripts/gds_list
```

Note: You can change the config file based on your set up and email details.

5. Post-Upgrade Procedure

This section describes the post-upgrade tasks that must be performed after upgrade tasks in order to successfully complete the upgrade to version 5.0.2.p6.

5.1 Generate the encrypted password

Perform the following steps on active name-node to generate the encrypted password:

1. Go to the `scripts` directory:

```
cd /opt/etc/scripts
```

2. Run the following command:

```
/bin/bash EncryptPassword.sh
```

3. Enter the following details in the dialog box:

- `<password-to-encrypt>` : db password
- `<password-to-generate-key>` : Key used in password encryption

Note:

- a. The `db password` for `password-to-encrypt` property should be same as the value of `postgres_password` defined in `/etc/reflex-provisioner/inventory/templates/group_vars/-global/all/mrx/agg/main.yml` file.
- b. `password-to-generate-key` can be any string, it is used to generate encrypted password in combination with `postgres_password`

5.2 Use encrypted password

To use the password generated in the above section, perform the following steps:

1. Store `<password-to-generate-key>` in a HDFS file:

Note: The default path to store the generated key is : `/data/streaming/key.txt`

2. Perform the following steps on both the name nodes to update the local file, `ImpalaToPostgres_DB.properties` available at location `/opt/sample_jobs/dimensionImpalaToPostgres/`:
 - a. Use the encrypted password generated in the preceding steps for the property `db.pwd` property.
 - b. Use HDFS path as mentioned in step-1 for the property `key.filepath`.

Note: Default path is : `/data/streaming/key.txt`

3. Update HDFS file `postgres_fb.xml` available at path `/data/streaming`
 - a. Enter the encrypted password string for the field tag password.
 - b. Set the value of property `key_filepath` to HDFS path defined in step-1.

Note: Default Path from Installer is : `/data/streaming/key.txt`

5.3 Extracting the MURAL Job Scheduler Scripts

Run the following command on both the master nodes to extract the MURAL job scheduler scripts:

```
tar -zxvf mural5Jobs.tgz
```

6. Start the Jobs

This section provides the information on starting the jobs from active master node that were , stopped in section: "Stop the Running Jobs" on page 10.

To identify the active master node perform the following steps:

1. Log in to the master node using SSH.
2. Run the following command:

```
# hdfs haadmin -getServiceState `hostname -f`
```

The sample output may resemble as follows:

```
active
```

To start the jobs follow these steps in specified order on the active master node:

6.1 Clear the checkpoint

Run the following commands to clear the checkpoint value from the active name node:

```
hdfs dfs -rm -r -skipTrash /data/ddj/checkpoint
hdfs dfs -rm -r -skipTrash /data/ddj2/checkpoint
```

6.2 Run the Master Job

1. Run master jobs from the active name node:

```
nohup sh \
/root/jobs/streaming_jobs/master_http_wrapper.sh \
> /var/log/mural_logs/master-http.log &

nohup sh \
/root/jobs/streaming_jobs/master_nonhttp_wrapper.sh \
> /var/log/mural_logs/master-nonhttp.log &
```

2. Check the logs in the following files, and wait for zero counter to be displayed.

- /var/log/mural_logs/master-http.log
- /var/log/mural_logs/master-nonhttp.log

3. Start the input data flow, that was stopped in section: "Stop the Running Jobs" on page 10.

6.3 Run the Talend Job

```
nohup sh \  
/root/jobs/ingestion_jobs/run-talend-nonhttp-job.sh \  
> /var/log/mural_logs/talend-nonhttp.log &  
  
nohup sh \  
/root/jobs/ingestion_jobs/run-talend-http-job.sh \  
> /var/log/mural_logs/talend-http.log &
```

6.4 Run the Aggregation Job

6.4.1 Remove ts file

```
hdfs dfs -rm -r -skipTrash /data/streaming/*-ts
```

6.4.2 Run the 5-minute aggregation Job

```
nohup sh \  
/root/jobs/aggregation_jobs/run-5min-agg-mgr_sh.sh \  
> /var/log/mural_logs/5min-agg-mgr.log &
```

Run the following command to verify if the db password is updated correctly or not:

```
grep -A1 "postgres_fb.xml" /var/log/mural_logs/5min-agg-mgr.log
```

The sample logs generated may resemble as follows:

```
INFO FlexiBinLogger: FlexiBinMetaDataService initialising with config  
xml : /data/streaming/postgres_fb.xml  
INFO execution.SparkSqlParser: Parsing command: show tables in  
kafkaconnectdb
```

If not updated correctly, the sample error message may resemble as follows:

```
INFO FlexiBinLogger: FlexiBinMetaDataService initialising with config  
xml : /data/streaming/postgres_fb.xml  
Exception in thread "main" java.lang.IllegalArgumentException: Can  
not create a Path from an empty string
```

Note: To resolve this error, re-run the section "Generate the encrypted password" on page 22.

6.4.3 Run the Hourly Aggregation Job

```
nohup sh \  
/root/jobs/aggregation_jobs/run-hourly-agg-mgr_sh.sh \  
> /var/log/mural_logs/hourly-agg-mgr.log &
```

6.4.4 Run the Daily Aggregation Job

```
nohup sh \  
/root/jobs/aggregation_jobs/run-daily-agg-weekReport_sh.sh \  
> /var/log/mural_logs/run-daily-agg-weekReport.log &
```

6.4.5 Run the Monthly Aggregation Job

```
nohup sh \  
/root/jobs/aggregation_jobs/run-monthlyagg-monthlyReport.sh \  
> /var/log/mural_logs/monthlyagg-monthlyReport.log &
```

6.4.6 Run the Cleanup Job

1. Run the following command to verify `cleanup_config.xml` for correct table names and regex:

```
vi /opt/sample_jobs/cleanup_job/cleanup_config.xml
```

2. Run the following command to run the clean up job:


```
nohup sh \  
/root/jobs/misc_jobs/run_cleanup_job.sh \  
> /var/log/mural_logs/cleanup.log &
```

7. Cemus Report Verification

7.0.1 Verify the Location of Cemus Config Files

To verify the location of the Daily Cemus, Monthly Cemus, GRT Cemus reports log into the management node and run the following command:

```
hdfs dfs -ls /data/streaming/
```

- For *DailyCemusReport*, check the following files should be available at the location `/data/streaming`
 - `DailyCemusReport.properties`
 - `DailyQueriesCemus.txt`
- For *MonthlyCemusReport*, check the following files should be available at the location `/data/streaming`
 - `MonthlyCemusReport.properties`
 - `MonthlyQueriesCemus.txt`
- For *GRTCemusReport*, check the following files should be available at the location `/data/streaming`
 - `GRTCemusReport.properties`
 - `GRTQueriesCemus.txt`
 - `grt-protocol-values.csv`
- *protocol-values.txt*

This file consists of the list of protocols for which daily, monthly and GRT reports will be generated. This file remains common for daily, monthly and GRT aggregations and should also be available at the location `/data/streaming`. To verify its location, run the below command:

```
hdfs dfs -ls /data/streaming/protocol-values.txt
```

Output:

```
-rw-r--r--   3 root hadoop      75428 2020-03-31 16:37
/data/streaming/protocol-values.txt
```

7.0.2 Verify the Content of Cemus Config Files:

The content of the Daily Cemus, Monthly Cemus, GRT Cemus reports can be verified by the following steps:

1. For *DailyCemusReport*

- To verify the content of `DailyCemusReport.properties` file, run the following command:

```
hdfs dfs -cat /data/streaming/DailyCemusReport.properties
```

Output:

```
dbName=kafkaconnectdb
input.tableName=5min_points_new
reportType=daily
query.file=/data/streaming/DailyQueriesCemus.txt
download.threshold=1000
upload.threshold=1000
locality.code=UK
notAvailableString=NotAvailable
unknownString=Unknown
cemusReportRootDir=/data/mrx/customer/cemus-report
single.file.report=true
protocol.file=/data/streaming/protocol-values.txt
sftp.location=/root/sftp/cemus
```

Notes:

- Here, `download.threshold` and `upload.threshold` represents the threshold value in bytes for downloading and uploading of the files. Both the property values can be customized based on the client requirement.
- `sftp.location` represents the location where all the reports can be placed.
- The value for `single.file.report` should be set to **true**.

2. For *MonthlyCemusReport*

- To verify the content of `MonthlyCemusReport.properties`, run the following command:

```
hdfs dfs -cat /data/streaming/MonthlyCemusReport.properties
```

Output:

```
dbName=kafkaconnectdb
input.tableName=daily_points_new
input.tableName2=montly_points_new
reportType=monthly
query.file=/data/streaming/MonthlyQueriesCemus.txt
download.threshold=3000
upload.threshold=3000
locality.code=UK
notAvailableString=NotAvailable
unknownString=Unknown
cemusReportRootDir=/data/mrx/customer/cemus-report
single.file.report=true
protocol.file=/data/streaming/protocol-values.txt
sftp.location=/root/sftp/cemus
```

3. For *GRTCemusReport*

- To verify the content of `GRTCemusReport.properties`, run the following command:

```
hdfs dfs -cat /data/streaming/GRTCemusReport.properties
```

Output:

```
dbName=kafkaconnectdb
input.tableName=hourly_points_new
reportType=GRT
query.file=/data/streaming/GRTQueriesCemus.txt
download.threshold=3000
upload.threshold=3000
locality.code=UK
```

```
notAvailableString=NotAvailable
unknownString=Unknown
cemusReportRootDir=/data/mrx/customer/cemus-report
single.file.report=true
protocol.file=/data/streaming/protocol-values.txt
grt.protocol.file=/data/streaming/grt-protocol-values.txt
sftp.location=/root/sftp/cemus/GRT
smtp.server.ip=192.168.104.25
smtp.port=25
sender.address=support-mrx@guavus.com
receiver.addresses=sample1@host.com,sample2@host.com
```

Notes:

- i. Here, `download.threshold` and `upload.threshold` represents the threshold value in bytes for downloading and uploading of the files. Both the property values can be customized based on the client requirement.
- ii. `sftp.location` represents the location where all the reports can be placed.
- iii. The value for `single.file.report` should be set to true.

• grt-protocol-value.csv

- To verify the content of `grt-protocol-value.csv`, run the following command:

```
hdfs dfs -cat /data/streaming/grt-protocol-values.csv
```

Output:

```
protocol,audio_coded_kbps,video_coded_kbps
whatsapp,25,240
facebook,30,500
skype,50,390
viber,45,980
facetime,32,1000
googleduo,60,590
```

```
googlehangout,55,450
other voip,35,335
```

This file is used specifically for report Mobile-UK-2-PGW_04.16.19 08.00 AM.csv report generation. The property values like protocol, audio codec and video codec can be changed by the customer accordingly or customer can also add a new protocol or remove existing protocol as per the requirements.

protocol-values.txt

- To verify the content of `protocol-values.txt` file, run the following command.

```
hdfs dfs -cat /data/streaming/protocol-values.txt|head
```

The sample output may resemble as follows:

```
youtube=YouTube
https=HTTPS
google-play=Google Play
whatsapp-voice=WhatsApp Voice
whatsapp-transfer=WhatsApp Transfer
wechat-unclassified=Wechat
snapchat-unclassified=Snapchat
skype-streaming-video=Skype Streaming-Video
skype-unclassified=Skype
whatsapp-unclassified=Whatsapp
```

7.1 Verify the DailyCemusReport

This is the output file generated based on the configurations in `DailyCemusReport.properties` and `DailyQueriesCemus.txt` file.

To verify the location of the file, run the following command:

```
hdfs dfs -ls /data/mrx/customer/cemus-report/daily
```

For example,

```
hdfs dfs -ls /data/mrx/customer/cemus-report/daily/20210712
```

The sample output may resemble as follows:

```
Found 10 items:
rw-rr-   3 root hadoop      758682 2021-07-12 08:09
/data/mrx/customer/cemus-report/daily/20210712/Umbrella_
UK_DataServicesCorporate_ALL_ALL_D_2021-07-12-08-03-
56.csv
rw-rr-   3 root hadoop      1539476 2021-07-12 07:58
/data/mrx/customer/cemus-report/daily/20210712/Umbrella_
UK_DataServicesCorporate_ALL_TECH_D_2021-07-12-07-47-
47.csv
rw-rr-   3 root hadoop      3961523 2021-07-12 08:03
/data/mrx/customer/cemus-report/daily/20210712/Umbrella_
UK_DataServicesCorporate_DS_ALL_D_2021-07-12-07-58-08.csv
rw-rr-   3 root hadoop      6010909 2021-07-12 07:47
/data/mrx/customer/cemus-report/daily/20210712/Umbrella_
UK_DataServicesCorporate_DS_TECH_D_2021-07-12-07-37-
17.csv
rw-rr-   3 root hadoop      751497 2021-07-12 07:37
/data/mrx/customer/cemus-report/daily/20210712/Umbrella_
UK_DataServices_ALL_ALL_D_2021-07-12-07-31-55.csv
rw-rr-   3 root hadoop      1916603 2021-07-12 07:26
/data/mrx/customer/cemus-report/daily/20210712/Umbrella_
UK_DataServices_ALL_TECH_D_2021-07-12-07-15-55.csv
rw-rr-   3 root hadoop      3920690 2021-07-12 07:31
/data/mrx/customer/cemus-report/daily/20210712/Umbrella_
UK_DataServices_DS_ALL_D_2021-07-12-07-26-20.csv
rw-rr-   3 root hadoop      5947961 2021-07-12 07:15
/data/mrx/customer/cemus-report/daily/20210712/Umbrella_
UK_DataServices_DS_TECH_D_2021-07-12-06-56-35.csv
rw-rr-   3 root hadoop        644 2021-07-12 08:18
/data/mrx/customer/cemus-report/daily/20210712/Umbrella_
UK_DataServices_Time_BW_Distribution_ALL_D_2021-07-12-08-
15-39.csv
```

```
rw-rr-  3 root hadoop      1802 2021-07-12 08:15
/data/mrx/customer/cemus-report/daily/20210712/Umbrella_
UK_DataServices_Time_BW_Distribution_TECH_D_2021-07-12-
08-09-38.csv
```

7.2 Verify the MonthlyCemusReport

This is the output file generated based on the configurations in `MonthlyCemusReport.properties` and `MonthlyQueriesCemus.txt` file. To verify the location of the file, run the following command:

```
hdfs dfs -ls /data/mrx/customer/cemus-report/monthly
```

For example:

```
hdfs dfs -ls /data/mrx/customer/cemus-
report/monthly/20210710
```

The sample output may resemble as follows:

```
Found 4 items
rw-rr-  3 root hadoop   417 2021-07-10 03:52
/data/mrx/customer/cemus-
report/monthly/20210710/Umbrella_UK_DataServices_BW_
Distribution_ALL_M_2021-07-10-03-51-22.csv
rw-rr-  3 root hadoop   711 2021-07-10 03:51
/data/mrx/customer/cemus-
report/monthly/20210710/Umbrella_UK_DataServices_BW_
Distribution_TECH_M_2021-07-10-03-48-00.csv
rw-rr-  3 root hadoop   359 2021-07-10 03:53
/data/mrx/customer/cemus-
report/monthly/20210710/Umbrella_UK_DataServices_BW_
Percentile_ALL_M_2021-07-10-03-53-13.csv
rw-rr-  3 root hadoop  1004 2021-07-10 03:53
/data/mrx/customer/cemus-
report/monthly/20210710/Umbrella_UK_DataServices_BW_
```



```
Percentile_TECH_M_2021-07-10-03-52-48.csv
```

7.3 Verify the GRTCemusReport

This is the output file generated based on the configurations in `GRTCemusReport.properties` and `GRTQueriesCemus.txt` file. To verify the location of the file, run the following command:

```
hdfs dfs -ls /data/mrx/customer/cemus-report/GRT
```

For example:

```
hdfs dfs -ls /data/mrx/customer/cemus-report/GRT/20210711
```

Output:

```
Found 2 items
rw-r--r--  3 root hadoop          1528 2021-07-11 06:48
/data/mrx/customer/cemus-report/GRT/20210711/Mobile-[UK]-
1-PGW_07.11.21_06.48_AM.csv
rw-r--r--  3 root hadoop          8484 2021-07-11 06:49
/data/mrx/customer/cemus-report/GRT/20210711/Mobile-[UK]-
2-PGW_07.11.21_06.48_AM.csv
```