blueplanet
a division of ciena

Unified Assurance and Analytics

Upgrade MOP from legacy to (23.08.xx) software

Release 23.08.64
Issue 1.0 | November 21, 2023 | 450-3704-325-2308

Table of Contents

Overview

BP Unified Assurance and Analytics is an amalgamation of BP Assurance (aka vSure) and BP Analytics (aka NHP). This is a new fully integrated microservices architecture which is packaged into multiple containers as opposed to the previously used format RPM's with 3rd party dependencies with BP Assurance.

This document lists the procedure for upgrade from legacy 20.02.xx software to 23.08.xx version.

Upgrade Instructions

Pre-requisites

NOTE

The upgrade from 20.02. xx to 23.08.xx is supported only through parallel installation in which a new installation of 23.08. xx will be done and thereafter, the data shall be migrated from 20.02. xx to the 23.08.xx installation.

Make sure to migrate all existing python2 scripts (such as Node Actions, Action Templates, Trouble Ticket scripts, Appmon Scripts etc.) to python3 before upgrade.

If there are customization to system created templates (such as node templates, endpoint templates etc.), all such customizations will be lost post upgrade. Blue Planet recommends the users to create custom templates (not modifying the system default) before upgrading with required customizations so that the changes are not lost.

The upgrade procedure includes -

1. Migration of below mentioned existing data from old database to new database.
    a. ConfigDB *(sa DB)* data migration from MariaDb to PostgreSQL
    b. Data from ConfigDB of 20.02 to 23.08 Neo4j
    c. Data from ConfigDB of 20.02 to 23.08 PM DB *(ClickHouse)*
    d. Data from FM DB *(MariaDB)* of 20.02 to 23.08 PostgreSQL
    e. Data from PM DB *(ShardDB)* of 20.02 to 23.08 PM DB *(ClickHouse)*
    f. User data from core-db will be migrated to 23.08 tron db
2. Transfer of workflow scripts, ticketing scripts, and generated reports
3. Deployment of RAs

For fresh installation of 23.08.xx follow the high-level installation steps -

1. Refer to the sizing guide to determine the server sizing for 23.08.xx installation.
2. Refer the 23.08.xx installation guide to set up the BP installer platform, license server (on-prem or cloud) and install UAA solutions.
3. Once the above is completed, follow the sections below to migrate reports and data, setup licenses, RA (Resource Adapters) deployment, and validation.

Before we begin,

- Make sure that the legacy 20.02.xx and 23.08.xx server are able to communicate without user intervention as the files are copied from 20.02.xx to 23.08.xx servers during migration.
- Update appropriate values in the commands were highlighted as yellow.
- Some commands in the document may be split into multiple lines. Before executing the command, ensure that the entire command is in a single line without any additional spaces in your terminal.
- Sudo prefix command is not required to be added when logged in as root.

NOTE

Make sure the email IDs of users are unique before upgrading. If the email Ids of more than one user is set as same, the system will add a random number (0-9) before the domain name in the email. Example: If the email ID is abc@xyz.com, the system will modify the email ID to abc2@xyz.com .

In such case, the admin will need to modify the email ID for the user to make it unique as per their requirements post upgrade. The list of updated email ID list will be present at the **/bp2/log/users_email_upgrade.log** file in uaa-core leader instance.

```
2021-12-20 11:54:47,285 INFO ########################-------------------------WARNINNG-------------------------########################
2021-12-20 11:54:47,286 INFO WE OBSERVED SAME EMAIL ADDRESS FOR MORE THAN ONE USERS SO WE HAVE CHANGED THE EMAIL ADDRESS FOR ONE OF THE USER'S
2021-12-20 11:54:47,286 INFO PLEASE CHANGE THE EMAIL ADDRESS FOR THE BELOW USER's AFTER LOGIN TO TRON
2021-12-20 11:54:47,286 INFO ##########################################################################################################################
2021-12-20 11:54:47,286 INFO changed_users_usersEmail_Dict
2021-12-20 11:54:47,286 INFO {'RohitSharma': 'mumbai_1@gm.com'}
2021-12-20 11:54:47,286 INFO ##########################################################################################################################
```

Post upgrade, the UAC Admin or UAC sysAdmin Role users will be able to update the password. Refer **User Access Control** section in Administrator's Guide on help for resetting password.

If the email ID is not configured for the user(s) before upgrade, make sure to update using either –

- the system will take default email ID (<db_username>@customer.com), where the db_username will be the old login ID.

If the username and password is same in 20.02.xx, the upgrade will migrate those users to new DB with password as 12345678. The sysadmin and admin users can login to the UI and change the respective user password from the App bar UI.

- Users with a unique email ID can also change their passwords from the App bar UI.

If the username and password is not same in 20.02.xx, those accounts will be successfully migrated to new DB in 23.08.xx and the users can login with their current login credentials **ONLY** if such user IDs are present in TRON DB.

Data Migration and Configuration

The data migration constitutes of migrating ConfigDB, Fault Management DB and Performance Management DB.

ConfigDB

Migration of ConfigDB is divided into 4 sections – Export, Import, File Migraton and Post SA Import. Follow the given procedure in each section to migrate ConfigDB to new installation of 23.08.xx
Config DB migration includes alarm customizations, metric customizations, PCRs, Workflows. It includes data related to all pages except the plugins as old plugins are not compatible with new 23.08.xx architecture.

The migration of ConfigDB may take ~45 minutes for approximately 300 nodes and 310K Endpoints.

The nodes will be imported in an unmanaged state and workflows will be disabled. Once, the import is successful, follow the post-import section to manage and enable the nodes and workflows, respectively.

NOTE

Before starting the upgrade process, need to install/deploy the **upgrade_from_20.02 solution on 23.08.xx server** using the following steps -

1. On 23.08.xx setup, login to any host as **bpadmin** user.
2. Enter solman using command '`sudo solman`'
3. For Single and multi-host upgrade, refer the example below steps.
   Single host –
   `solution_deploy artifactory.ciena.com.blueplanet.uaa_upgrade_from_20_02:23.08.xx`
   Multi host –
   `solution_deploy artifactory.ciena.com.blueplanet.uaa_upgrade_from_20_02:ha-23.08.xx`

For uaa_upgrade solution version, refer **lineup-uaa-<multi/single>-optional-solutions.yml** which will be presented in the BP_UAA-TAR file.

Export

NOTE

The following files are related to export process -

exportClientSaDb.py - Establishes the ssh connection and transfers the export script to 20.02.xx which runs and create zip file for table CSVs and then zip file is copied back to 23.08.xx server.

configDataExportToCSVFromMysql.py - Connects to mysql sa database and create zip file for CSVs which have table-wise data.

/bp2/scripts/migration.config - Contains configuration settings for export process.

1. On 23.08.xx setup, login to any host as **bpadmin** user.
2. Enter solman using command '`sudo solman`'
3. Enter uaa-upgrade-from-20.02 container leader instance.
4. Execute following command to create sa directory –

   ```
   mkdir -p /bp2/data/migration/sa
   ```

5. Edit the **migration.config** file at **/bp2/scripts/** location -

```
vi /bp2/scripts/migration.config
```

Update the following properties in the file –

NOTE

homeDirOfRemoteUser should be the home directory of the given user under [export_ssh_connection]

Example:
If export ssh connection user is centina, homeDirOfRemoteUser should be **/home/centina**
If export ssh connection user is root homeDirOfRemoteUser should be **/root**

```
[export_build_version]

build_version=<build version of legacy UAA>
target_build_version=<Latest UAA 23.08 version>


[sa_export_directories]

migrationDirectory=<dir on the 20.02.xx server where the dump will be taken. Make sure to give a separate folder for sa migration and the folder is available>

exportScriptPath=<path where configDataExportToCSVFromMysql.py script is placed including the file name eg. /home/bpadmin/configDataExportToCSVFromMysql.py>

targetLocalPath=<Path to which the csv dump file zip is copied on the local server i.e., server (23.08.xx) from which we run exportClientSaDb.py. Make sure to give a separate folder for sa import alone and folder is available>

homeDirOfRemoteUser=<home directory of 20.02.xx server>

[export_ssh_connection]
username=<SSH username for 20.02.xx Master DB server>
password=<SSH password for 20.02.xx Master DB server>
host=<ip of the 20.02.xx Master DB server>
port=22
PemFilePath=<.pem file if needed to login to 20.02.xx Master DB server, else leave it blank>
exportDbPassword=centina
```

Example:

```
[export_build_version]
build_version=20.02-60
target_build_version=23.08-07

[sa_export_directories]
migrationDirectory=/home/centina/21.02_migration/sa
exportScriptPath=configDataExportToCSVFromMysql.py
targetLocalPath= /bp2/sa-migration/
homeDirOfRemoteUser=/root

[export_ssh_connection]
username=root
password= <if the servers have password less authentication for users and may or may not specify password>
host=10.182.95.241 (Make sure to use correct host IP as per your Master DB server)
port=22
PemFilePath=Empty
exportDbPassword=centina
```

6. Execute the following commands –

```
cd /bp2/scripts/sa-migration-scripts

python3 exportClientSaDb.py
```

NOTE

The log file for exporting configDB will be located at the following location on 20.02.xx server - **/centina/logs/sa_db_migration.log**

If there is a failure during migration, the script will display and error and stop execution. The CSV file at **/centina/logs/saexportFailOver.csv** on 20.02.xx server will have the table name and time at which it got failed.

Re-run the script, to read the table name from **/centina/log/saexportFailOver.csv** which got failed and start export from that table.

Import

Once the export process is completed, the DB tables are zipped into CSVs. The import process will now unzip and load these csv in containerised postgres.

NOTE

The following files are related to import process -

importClientSaDb.py - Establishes the ssh connection, copies and runs the configDataImportToPsql.py in the uaa-core-db leader server instance.

configDataImportToPsql.py - This script runs inside uaa-core-db leader server instance, connects to psql sa database, and import CSVs from zip file.

/bp2/scripts/migration.config - Contains configuration settings for import process.

1. On 23.08.xx setup, login to any host as **bpadmin** user.
2. Enter solman using command '`sudo solman`'
3. Enter uaa-upgrade-from-20.02 container leader instance.
4. Create/edit the **migration.config** file at **/bp2/scripts/** location -

```
vi /bp2/scripts/migration.config
```

Update the following properties in the file –

```
[sa_import_directories]
importScriptName=configDataImportToPsql.py <don't modify>

localPathToZip=/bp2/sa-migration/sa <make sure that this value is equal to
'targetLocalPath' value under [sa_export_directories] section and the folder is available>

restoreDir=/bp2/restore/sa <make sure to give a separate folder for sa import
alone and the folder is available>

unzip=yes

[uaa_core_db_credentials]
port=5432
```

```
user=bpadmin <Do not change username>
password=sbpadminpw <Update the password as given in /dev/shm/users.json inside uaa-core-db container>
db_name=bpuaa
super_user=sbpadmin <Do not change username>
super_user_password=eo8u8kbt4ux7 <Update the password as given in /dev/shm/users.json inside uaa-core-db container>
```

5. Execute the following commands –

```
cd /bp2/scripts/sa-migration-scripts

sed -i "s/application-monitors/applications/g" configDataImportToPsql.py

python3 importClientSaDb.py
```

NOTE

A log file named **sa_db_migration.log** will be present inside uaa-upgrade-from-20.02 container at **/bp2/log** location in 23.08.xx

If there is a failure during migration, the script will continue the execution and keep the failed CSVs. Failed csv files will be present at `restoreDir` location in the uaa-core-db container once the script execution is completed.

Resolve the error and re-run the script after updating unzip=no, parameter in the migration.config file.

If the user needs to re-import the SA, follow the steps to setup the system for re-import and then continue with the following steps.

6. Execute the following commands on all *(in case of multi-host setup)* uaa-core instances –

```
sudo solman

enter_container uaa-core_23.08.xx_x

rm /bp2/data/.task_cookie.txt
```

7. Execute the following commands to restart all *(in case of multi-host setup)* uaa-core apps –

```
sudo solman

solution_app_restart artifactory.ciena.com.blueplanet.uaa_core:ha-23.08.27 uaa-core
```

8. Execute the following command in the leader role –

```
sudo solman

enter_container uaa-core-db_xxxx.x.x

ctl list
```

*(To verify the leader role from the list and use the same for below command **only** in a multi host setup)*

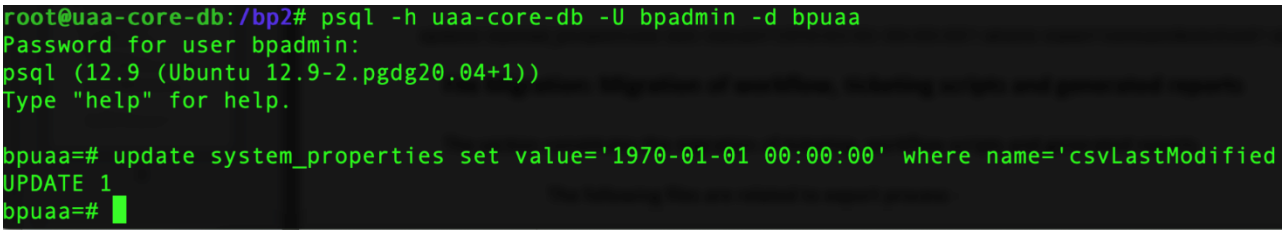
```
enter_container uaa-core-db_xxxx.x.x<leader Host>
```

9. Login to uaa-core-dbpostgresDB –

```
psql -h uaa-core-db -U bpadmin -d bpuaa
```

10. Execute the following command –

```
update system_properties set value='1970-01-01 00:00:00' where name='csvLastModified' and category='plugin';
```



File Migration: Migration of workflow, ticketing scripts and generated reports

The section constitutes the migration of ticketing, workflow scripts and generated reports.

NOTE

The following files are related to export process -

fileMigrationScript.py – Establishes the ssh connection to 20.02.xx server fetch the required (hbm/workflow scripts/ticketing scripts/generated reports) scripts/files and deploys into 23.08.xx server.

fileMigration.config - Contains configuration settings for export process.

The following steps are to be followed on all servers in multi-host environment.

1. On 23.08.xx setup, login to any host as **bpadmin** user.
2. Enter solman using command '`sudo solman`'
3. Enter uaa-upgrade-from-20.02 container leader instance.
4. Create/edit the **fileMigration.config** file at /bp2/scripts/file-migration-scripts location on host ***(leader instance)*** –

   ```
   vi /bp2/scripts/file-migration-scripts/fileMigration.config
   ```

   Update the following properties in the file -

   NOTE

   Only the following section should be modified.
   Make sure to change parameters under '**old-app-server-ssh-credentials**' head only.
   In case of AWS setup, **do NOT** comment the pem_file parameter in the fileMigration.config file.

   ```
   [old-app-server-ssh-credentials]
   ssh_user: centos (Mention which user to be used)
   ssh_password: Ciena123! (Mention which password to be used)
   ssh_host: 10.107.3.39 (Change IP# to 20.02-xx Application Server IP #)
   ssh_port: 22
   ```

5. Execute the following command –

   ```
   cd /bp2/scripts/file-migration-scripts

   python3 fileMigrationScript.py
   ```

NOTE	The log file for migrating scripts will be located at the following location on 23.08.xx server - **/bp2/scripts/file-migration-scripts/file_migration_debug.log**

Post SA Import

Once the import is successful, execute the following steps –

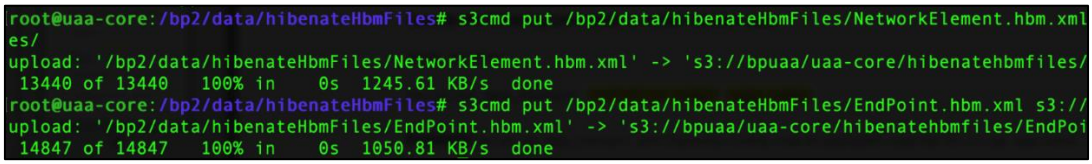
1. Copy **ONLY dynamic-component xml block** from the HBM files **(NetworkElement.hbm.xml, EndPoint.hbm.xml)** present in 20.02.xx server to the HBM files in **all the uaa-core containers** in 23.08.xx server –

   HBM files location in 20.02.xx server: **/centina/app/jboss/server/default/deployers/ejb3.deployer/**
   HBM files location in uaa-core container in 23.08.xx server: **/bp2/data/hibenateHbmFiles/**

```
<!-- STATIC-DEFINITION-ENDS -->

<!-- DYNAMIC-DEFINITION-STARTS -->
<!-- DYNAMIC-DEFINITION-ID:0: -->

<!-- NetworkElement hibernate mapping for dynamic fields start -->

      <!-- DYNAMIC COMPONENTS - FOR CUSTOM NODE FIELDS -->
      <dynamic-component insert="true" name="customFieldsMap" optimistic-lock="true" unique="false" update="true">
      </dynamic-component>
   </joined-subclass>

<!-- NetworkElement hibernate mapping for dynamic fields end -->

</hibernate-mapping>

<!-- DYNAMIC-DEFINITION-ENDS -->
```

2. Post updating HBM files under **/bp2/data/hibenateHbmFiles/,** copy the updated HBM files to S3 bucket by executing the following commands -

```
s3cmd put /bp2/data/hibenateHbmFiles/NetworkElement.hbm.xml s3://bpuaa/uaa-core/hibenatehbmfiles/

s3cmd put /bp2/data/hibenateHbmFiles/EndPoint.hbm.xml s3://bpuaa/uaa-core/hibenatehbmfiles/
```



```
root@uaa-core:/bp2/data/hibenateHbmFiles# s3cmd put /bp2/data/hibenateHbmFiles/NetworkElement.hbm.xml
es/
upload: '/bp2/data/hibenateHbmFiles/NetworkElement.hbm.xml' -> 's3://bpuaa/uaa-core/hibenatehbmfiles/
 13440 of 13440   100% in    0s  1245.61 KB/s  done
root@uaa-core:/bp2/data/hibenateHbmFiles# s3cmd put /bp2/data/hibenateHbmFiles/EndPoint.hbm.xml s3://
upload: '/bp2/data/hibenateHbmFiles/EndPoint.hbm.xml' -> 's3://bpuaa/uaa-core/hibenatehbmfiles/EndPoi
 14847 of 14847   100% in    0s  1050.81 KB/s  done
```

3. Execute the following commands in the leader host –

NOTE	Make sure to wait for uaa-core to come up before executing the following commands. Make sure that all the files *(if any)* under **/etc/bp2/bpuaa/import** directory are imported in uaa-core leader instance before proceeding.

```
enter_container uaa-core_23.08.xx_x

cd /etc/bp2/bpuaa/import

cp archived/profile-configs.csv ./

cp archived/*.* ./
```

Sound Files Import

1. Copy the sound files along with .soundfilenameidmap.properties file from 20.02 server to sound files present in uaa-ui and uaa-core containers in 23.08.xx server.

   Soundfiles location in 20.02.xx server: **/centina/soundFiles/**
   Soundfiles location in 23.08.xx server: **/bp2/data/soundFiles/**

   Create soundFiles-tmp folder in 23.08 server -

   ```
   cd /home/bpadmin/

   mkdir -p soundFiles-tmp
   ```

   Copy files from 20.02 to 23.08 server -

   ```
   scp /centina/soundFiles/* 23.08 server:/home/bpadmin/soundFiles-tmp

   scp /centina/soundFiles/.soundfilenameidmap.properties <bpadmin@23.08 server>:/home/bpadmin/soundFiles-tmp
   ```

   Example:

   ```
   scp /centina/soundFiles/* bpadmin@23.08 server:/home/bpadmin/soundFiles-tmp/

   scp /centina/soundFiles/.soundfilenameidmap.properties bpadmin@23.08 server:/home/bpadmin/soundFiles-tmp/

   sudo docker cp /home/bpadmin/soundFiles-tmp uaa-core_xx.xx.xx:/bp2/data
   ```

2. Copy sound files to s3 bucket by executing below commands -

   ```
   sudo solman

   enter_container uaa-core_xx.xx.xx
   ```
   (leader instance)

   ```
   s3cmd put /bp2/data/soundFiles-tmp/.soundfilenameidmap.properties s3://bpuaa/uaa-ui/soundFiles/

   s3cmd put /bp2/data/soundFiles-tmp/* s3://bpuaa/uaa-ui/soundFiles/

   s3cmd put /bp2/data/soundFiles-tmp/.soundfilenameidmap.properties s3://bpuaa/uaa-core/soundFiles/

   s3cmd put /bp2/data/soundFiles-tmp/* s3://bpuaa/uaa-core/soundFiles/
   ```

3. Verify if the copied soundfiles are present in uaa-core and uaa-ui containers at **/bp2/data/soundFiles** directory and execute the following commands *(if they are not present, it is recommended to wait for ~5 minutes before proceeding)* –

   ```
   enter_container uaa-core_23.08.xx_x

   cd /bp2/scripts

   python3 upgradeSoundFiles.py
   ```

NOTE The log file for script will be located at the following location on 23.08.xx server - **/bp2/log/upgradeSoundFiles.log**

Resource Adapter (RA) Deployment

An RA is a set of configuration information that tells UAA how to communicate with any given device. The commonly supported protocols and thus RA types are ASCII, SNMP, CORBA and TL1. RAs are created for each type of device that is supported.

RAs are stored as a series of XML documents bundled into a zip file with a .pro file extension. These files are stored in the **/etc/bp2/bpuaa/profiles** directory of the UAA application server main file system. The server continuously examines the contents of this directory looking for any new or changed RAs.

To facilitate download and installation of latest versions of RAs, an automatic and periodic "sync" between the central RA repository and the UAA deployment must be established.

This is achieved through a cronjob, which will perform periodic check on the RA repository where the latest versions of the RAs are available every 24 hours and update in the deployed UAA server.

Installing a Resource Adapter (RA) and Respective Licenses

Refer UAA user guide to install the required RAs and their respective licenses.

NOTE | For more details on installing/editing RA, user can refer the user guide/help.

IMPORTANT | All alarm customizations and the assigned plugin translators will be assigned by the script.

Login to UAA UI on 23.08.xx and re-enable the workflows which are disabled during the migration and change the operational state of all the nodes from unmanaged to managed.

Refer the user guide/online help available on how to change workflows and node’s operational state.

- Before managing a node, make sure that the nodes are reachable from 23.08.xx southbound collector (SBC).
- For receiving trap info, make sure the respective nodes are managed in the 23.08.xx setup and the nodes are configured to send the traps to the new destination i.e SBC.

Fault Management (FM) DB

Migration of FM DB is divided into 2 sections – Export and Import. Follow the given procedure in each section to migrate FM DB to new installation of 23.08.xx

The migration of FM DB may take ~60 minutes for 10M rows.

Export

NOTE

The following files are related to export process -

exportFmDbDriver.py - Establishes the ssh connection and transfers the export script to 20.02.xx which runs and create zip file for table CSVs and then zip file is copied back to 23.08.xx server.

fmDataExportToCSVFromMysql.py - Connects to mysql sa database and create zip file for CSVs which have table-wise data.

/bp2/scripts/migration.config - Contains configuration settings for export process.

1. On 23.08.xx setup, login to any host as **bpadmin** user.
2. Enter solman using command '`sudo solman`'.
3. Enter uaa-upgrade-from-20.02 container leader instance.
4. Execute the following command to create fm directory –

   ```
   mkdir -p /bp2/data/migration/fm
   ```

5. Edit the **migration.config** file at **/bp2/scripts/** location on host -

   ```
   vi /bp2/scripts/migration.config
   ```

   Update the following properties in the file –

   ```
   [fm_export_directories]
   migrationDirectory=<dir on the 20.02.xx server where the dump will be taken. Make sure to give separate directory for fm migration alone and the folder is available >
   targetLocalPath= /bp2/fm/ <Path to which the csv dump file zip is copied on the local server i.e, to the server from which ExportFmDbDriver.py is executed. Make sure to give separate directory for fm migration alone and the folder is available>

   [export_ssh_connection]
   username=<SSH username for 20.02.xx Master DB server>
   password=<SSH password for 20.02.xx Master DB server>
   host=<ip of the 20.02.xx Master Db server>
   port=<SSH port for 20.02.xx Master DB server>
   pemFilePath=<.pem file if needed to login to 20.02.xx Master DB server, else leave it blank>
   exportDbPassword=<20.02.xx Master DB login password>

   [fm_export_properties]
   ```

```
ResumeFromFailOver=<yes/no Incase. This value depends on if user wants to resume from failover state or not incase if export is failed>
StartDate=<yyyy-MM-DD HH:MM - start date from which dump will be taken, else leave it blank>
EndDate=<yyyy-MM-DD HH:MM - endDate up to which dump will be taken, else leave it blank>
```

NOTE

Create `targetLocalPath` directories in the uaa-upgrade-from-20.02 container as given in the above config file.
In case, startDate and EndDate are not specified, the entire data of FM DB will be migrated into the new 23.08.xx server.

6. Execute the following command –

```
cd /bp2/scripts/fm-migration-scripts

python3 ExportFmDbDriver.py
```

NOTE

The log files for exporting FM DB will be located at the following locations on 20.02.xx and 23.08.xx server respectively –
/home/centina/fm_db_migration_export.log
/bp2/scripts/fm-migration-scripts/fm_db_migration_export.log

If there is a failure during migration, the fm_dum.zip file will not be present under the target location or the log file will show error. Resolve all errors shown in the log file and execute the below commands –

```
cd /bp2/scripts/fm-migration-scripts
```

Edit **migration.config** script and update **ResumeFromFailOver=yes** and execute the export script

```
python3 ExportFmDbDriver.py
```

Import

Once the export process is completed, the DB tables are zipped into CSVs. The import process will now unzip and load these csv in containerised postgres.

NOTE

The following files are related to import process -

importFmDbDriver.py - Establishes the ssh connection, copies and runs the fmDataFromCSVtoPSQL.py script to transfer the zipped files to 23.08.xx server.

fmDataExportToCSVFromMysql.py - This script Connects to mysql sa database and import CSVs from zip file.

/bp2/scripts/migration.config - Contains configuration settings for import process.

1. On 23.08.xx setup, login to any host as **bpadmin** user.
2. Enter solman using command '`sudo solman`'
3. Enter uaa-upgrade-from-20.02 container leader instance.
4. Create/edit the **migration.config** file at **/bp2/scripts** location on host -

```
vi /bp2/scripts/migration.config
```

Update the following properties in the file –

```
[fm_import_directories]
localPathToZip=/bp2/fm/ <value of this should be equal to targetLocalPath under [fm_export_directories] section>
restoreDir=/bp2/restore/fm/ <make sure to give separate directory for fm alone and the directory is available>
unZip=yes

[uaa_core_db_credentials]
host=uaa-core-db
port=5432
user=bpadmin <Do not change username>
password=sbpadminpw <Update the password as given in /dev/shm/users.json inside uaa-core-db container>
db_name=bpuaa
super_user=sbpadmin <Do not change username>
super_user_password=eo8u8kbt4ux7 <Update the password as given in /dev/shm/users.json inside uaa-core-db container>
```

5. Execute the following command –

```
cd /bp2/scripts/fm-migration-scripts

python3 ImportFmDbDriver.py
```

NOTE

The following log files for importing FM DB will be located at following locations -
/bp2/scripts/fm-migration-scripts/fm_db_migration_import.log
/bp2/log/fm_db_migration_import_queries.log

If there is a failure during import, resolve all the errors shown in the log file and start over. User can choose to start the import from the failed state –

To start from failed state, update unZip=no in the /bp2/scripts/migration.config file and execute the following commands –

```
cd /bp2/scripts/fm-migration-scripts

python3 ImportFmDbDriver.py
```

If the user needs to re-import the FM, follow the steps to setup the system for re-import and then continue with the following steps.

6. Execute the following commands to restart all *(in case of multi-host setup)* **uaa-core** solution –

```
sudo solman

solution_app_restart artifactory.ciena.com.blueplanet.uaa_core:<23.08 version> uaa-core
```

Final Validation

To validate the script and reports migration –

1. On 23.08.xx setup, login to any host as **bpadmin** user.
2. Enter solman using command '`sudo solman`'
3. Enter uaa-upgrade-from-20.02 container leader instance and execute the following commands to check if workflow and ticketing scripts are migrated properly.

```
s3cmd ls s3://bpuaa/uaa-core/scripts/workflow/

s3cmd ls s3://bpuaa/uaa-fme/scripts/workflow/

s3cmd ls s3://bpuaa/uaa-core/scripts/ticketing/

s3cmd ls s3://bpuaa/uaa-fme/scripts/ticketing/
```

WARNING

Any configurations related to external system integrations (such as external DB configuration) should be re-configured on 23.08.xx server(s).

For newly created external DB connections in 23.08.xx, the user will need to edit the advanced reports and assign the new external DB connections and verify if the advanced reports are getting generated.

Since the DB architecture is changed from MySQL to Cickhouse/Postgress, make sure that the Advanced Report queries and Action Templates (with action template type as External Database Query) queries are working properly and does not have any syntactical errors after migrating them to 23.08.xx server(s). If found, please update the queries accordingly.

NOTE

Any node action scripts created in 20.02.xx are required to be copied manually from **/centina/med/<script location folder>** location on 20.02.xx server to **/opt/ciena/bpuaa/node-actions/** location on 23.08.xx host server(s).

Any custom application monitor scripts created in 20.02.xx are required to be copied manually from **/centina/med/<script location folder>** on 20.02.xx server to **/opt/ciena/bpuaa/application-monitors** location on 23.08.xx host server(s).

Performance Management (PM) DB

Migration of PM DB is divided into 2 sections – Export and Import. Follow the given procedure in each section to migrate PM DB to new installation of 23.08.xx

Files from the import folder will be deleted automatically once the import is successful or moved to replay location. The import thread will be running to scan for exported files and if found, the thread will import the data.

User can execute the export script first and the execute the import script in another terminal in parallel.

PM DB migration executes with 1 partition (1 day data) at a time. Only when 1 partition is migrated successfully, the next partition is taken for migration.

The required python modules like paramiko, pip, PyMySQL, and clickhouse_driver are installed automatically and clickhouse-client also installed automatically on uaa-upgrade-from-20.02 container.

NOTE

Export/Import scripts need to run in parallel. The export script generates the export CSV files into the specified export location in the config and the import scripts scan the export location per minute. If files exist import them into ClickHouse and check if any export threads are running.
If no threads are running, it will terminate the import script with message "*INFO [MainThread]: Terminating the import program since export is done*" in log file import_csvs_debug.log

In case of multi-node cluster, execute the export script in **all** the 23.08.xx uaa-upgrade-from-20.02 dockers *(if we must get data from multiple shards)* and specify which uaa-upgrade-from-20.02 container is responsible to get data from which shard databases and mediation servers through the configuration file.

Export

PM migration scripts migrate huge amount of PM data from shard dBs to clickhouse. We can find the size of PM data in the shard DB by using the below query -

```
select table_schema, sum((data_length+index_length)/1024/1024/1024) AS GB from information_schema.tables group by 1;
```

For **6.5GB** PM data migration, the migration scripts take **~4.5 hours** with default retention periods. It may vary based on retention period. it may take more time to complete the migration, thus we suggest running the import and export scripts in the background.

Users can optionally execute the following command to verify if the import and export scripts are running *(login to the PM DB server in 20.02.xx, all uaa-upgrade-from-20.02 containers in 23.08.xx)* –

```
ps –ef
```

NOTE

The following files are related to export process -

export_perf_data_csvs.py - Creates the export and import locations by reading the upgrade_config.cfg file. In a single cycle, it exports partition data from the 20.02.xx shard server and gets the exported partition CSV data into the 23.08.xx server at the specified location. Checks md5sum and delete the CSV file from the 20.02.xx server.

export_slo_raw.py - Exports the slo data from the 20.02.xx mediation server and get the exported CSV files into the 23.08.xx server at the specified location.

upgrade_config.cfg - Contains configuration settings for export process.

1. On 23.08.xx setup, login to host 0 *(all hosts in multi-host setup)* as **bpadmin** user.
2. Enter solman using command '`sudo solman`'
3. Enter uaa-upgrade-from-20.02 container *(all uaa-upgrade-from-20.02 containers if we must get data from multiple shards)*.

NOTE

The export-location should be the under the location **/bp2/data**. Make sure to keep the default location.

The export script creates PM export threads < number of threads = No. of shard databases > to export data and get the partition CSVs simultaneously.

Export script creates SLO export threads < number of threads = No. of mediation servers > to export all CSVs and then import into 23.08.xx server.

Example: If perf_raw_logsize is 2, it exports previous 2 days data and the current day data till the time when the export is executed.

4. Edit the **upgrade_config.cfg** file at **/bp2/scripts/pm-migration-scripts** location *(all uaa-upgrade-from-20.02 containers in multi host setup)* -

```
vi /bp2/scripts/pm-migration-scripts/upgrade_config.cfg
```

Update the following properties in the file –

```
[shard-name-ip]
db1:<shard database1 ip>
db2:<shard database2 ip>

[shard-ssh-credentials]
ssh_user:<SSH username for 20.02.xx shard database server>
ssh_password:<SSH password for 20.02.xx shard database server>
ssh_port:<SSH port for 20.02.xx shard server>
pem_file:<.pem file if needed to login to 20.02.xx shard database server else
leave blank/None>

[shard-db]
username:<shard database username>
password:<shard database password>

[med-name-ip]
med0:<mediation server1 ip>
med1:<mediation server2 ip>

[med-ssh-credentials]
ssh_user:<SSH username for 20.02.xx mediation server>
ssh_password:<SSH password for 20.02 mediation server>
ssh_port:<SSH port for 20.02 mediation server>
pem_file:<.pem file if needed to login to 20.02.xx mediation server else leave
blank/None>

[retention-days]
perf_raw_logsize:<perf raw retention days>
perf_1hour_logsize:<perf 1hour retention days>
perf_1day_logsize:<perf 1day retention days>
sla_raw_logsize:<sla raw retention days>
sla_1day_logsize:<sla 1hour retention days>
baseline_period:<baseline retention days>
service_perf_raw_logsize:<service per raw retention days>
```

```
service_perf_1hour_logsize:<service perf 1hour retention days>
service_perf_1day_logsize:<service perf 1day retention days>

[location]
export_location:<Location of exported files in 23.08.xx server>
temp_export_location:<Temporary location of exported files in 23.08.xx server>
```

The retention days configuration may not be exactly as the DB configuration. Here, the user can select the retention days based on the requirement/migration of data into 23.08.xx

NOTE

The log files for exporting PM DB will be located at the following locations –

On 23.08.xx uaa-upgrade-from-20.02 docker -
/bp2/scripts/pm-migration-scripts/export_csvs_debug.log
/bp2/scripts/pm-migration-scripts/export_csvs_error.log
/bp2/scripts/pm-migration-scripts/status.csv
/bp2/scripts/pm-migration-scripts/import_termination.csv

On 20.02.xx server -
/root/export_csvs-debug.log
/root/status.csv

Export scripts should be stopped in all instances if the import script fails. This could be due to exported files getting accumulated in the server and the memory filling up quickly.

Import script scans for export status, so it stops automatically while import script has problem or completed.

If the export script fails, there will not be any entry in the status.csv file for the partition. Re-execute the script which checks for the entry in the status.csv file. If the entry does not exist, the export script will try to export again.

If you need to export a particular CSV partition data, set '**isExported**' field to False in the status.csv for the CSV file.

5. Execute the following command –

```
cd /bp2/scripts/pm-migration-scripts

nohup python3 export_perf_data_csvs.py &
```

Import

NOTE

The following files are related to import process -

restore_perf_data_from_csvs.py - Imports the CSV data from the 'export_location' specified in upgrade_config.cfg, It checks the export_location per minute and imports the CSV data if it finds the files.

upgrade_config.cfg - Contains configuration settings for import process.

As stated in the export section, both export and import process should run in parallel.

1. On 23.08.xx setup, login to host 0 *(all hosts in multi-host setup)* as **bpadmin** user.
2. Enter solman using command '`sudo solman`'
3. Enter uaa-upgrade-from-20.02 container *(all uaa-upgrade-from-20.02 containers where export script is running)*.
   The import scripts are present at **/bp2/scripts/pm-migration-scripts** location on the container.
4. Edit the **upgrade_config.cfg** file at **/bp2/scripts/pm-migration-scripts** location within the container -

   ```
   vi /bp2/scripts/pm-migration-scripts/upgrade_config.cfg
   ```

   Update the following properties in the file –

   ```
   [click-house]
   host: <pm docker name>
   user: bpadmin
   password: XXXXXX
   ```

   Example:
   `host: uaa-pm-db.docker` *(uaa-pm-db0.docker, uaa-pm-db1.docker...so on multi host setup)*
   Password will be updated automatically. Recheck the password once. The password should be as given in **/dev/shm/pm_db_users.json** inside uaa-upgrade-from-20.02 container.

   ```
   {
     "clientUserPwd": "s7pyiz4zz68t",
     "clientUser": "bpadmin"
   }

   [location]
   Replay_location: <Location of the failed CSV files in 23.08.xx server>
   ```

5. Execute the following command –

   ```
   cd /bp2/scripts/pm-migration-scripts

   nohup python3 restore_perf_data_from_csvs.py &
   ```

The log files for importing PM DB will be located at the following locations on 23.08.xx server –
/bp2/scripts/pm-migration-scripts/import_csvs_debug.log
/bp2/scripts/pm-migration-scripts/import_csvs_error.log

NOTE If the import script fails, the exported CSV files will be moved to **replay_location**, to re-import the CSV files. User needs to move from **replay_location** to **export_location**.

If the user needs to re-import the PM, follow the steps to setup the system for re-import and then continue with the following steps.

NOTE Users can complete the Post upgrade procedure while the PM DB migration is running.

Post-upgrade Procedures (Common)

The following section describes the post upgrade procedures for UAA.

User Migration

1. Log in to host 0 *(for multi host setup)* as bpadmin. Do not use the site IP; use the IP address of host 0.
2. Enter uaa-core-db container –

   ```
   sudo solman

   enter_container uaa-core-db_xxxx.xx.x-xx-xx
   ```

   Example: enter_container uaa-core-db_2023.07.2-pg12-5_0

3. Execute the following command to log into uaa-core-dbpostgresDB -

   ```
   psql -h uaa-core-db -U bpadmin -d bpuaa
   ```

4. Execute the following command *(make sure that the entire command is executed in 1 line)* –

   ```
   SELECT mo.id,ed.dbType FROM sa.external_dbconfig AS ed,sa.managed_object mo WHERE mo.id = ed.id and mo.name='defaultPostgresConnection';
   ```

   Check above query has result which should display the default External Database Connection *(It is recommended to wait ~5-10 min post upgrade before checking)*. If not available, please contact Blue Planet Support.

User & Visualisation/Controls/Dashboards/Reports Migration

1. Enter uaa-upgrade-from-20-02 container -

   ```
   sudo solman

   enter_container uaa-upgrade-from-20-02
   ```

2. Navigate to **/bp2/scripts/coreui/** location to run **coreui_migration** script for User & Visualisation/Controls/Dashboards/Reports Data migration from 20.02.xx to 23.08.xx

   NOTE Before running the coreui migration script below, make sure to update the customer field in **credentials.json** at **/bp2/scripts/coreui/** location.
   Example: "customer":"Abc"

   ```
   cd /bp2/scripts/coreui/

   nohup python3 coreui-migration.py &
   ```

   NOTE The **coreui_migration.log** file will be located at the **/bp2/log/** location inside the uaa-upgrade-from-20-02_23.08.02_0 container.

Restarting the UAA Servers

1. Execute the following command in the leader role –

   ```
   sudo solman

   enter_container uaa-core-db_xxxx.x.x

   ctl list
   ```
   *(To verify the leader role from the list and use the same for below command **only** in a multi host setup)*

   ```
   enter_container uaa-core-db_xxxx.x.x<leader Host>
   ```

2. Login to uaa-core-dbpostgresDB –

   ```
   psql -h uaa-core-db -U bpadmin -d bpuaa
   ```

3. Execute the following command –

   ```
   SELECT setval('outage_control_seq', (SELECT MAX(id) FROM outage_control), TRUE);
   ```

4. Execute graph db sync script in uaa-core_xx.xx.xx_x container ***(on leader container)***.

5. Login to uaa-core leader instance and execute following command -

   ```
   python3 /bp2/scripts/db/pg2neo/SyncPostgresObjectsToNeo4j.py
   ```

6. Execute the following commands in **uaa-pm-db-scheduler** container ***(on leader container)*** –

   ```
   enter_container uaa-pm-db-scheduler_x.x.xx_x

   python3 /bp2/scripts/pg2ch.py
   ```

   Verify the **/bp2/log/pg2ch.log** file for any errors. Contact Blue Planet Support for assistance.

7. Refer **UAA 23.08 Administration Guide > UAA Maintenance Procedures**, to restart all the UAA solutions.

Installing default Resource Adapters

1. Log in to host 0 (for multi host setup, repeat the steps in all hosts) as bpadmin. Do not use the site IP; use the IP address of host 0.
2. Enter uaa-core container –

   ```
   sudo solman

   enter_container uaa-core_23.08.xx_x
   ```

   Example: `enter_container uaa-core_23.08.56_1`

3. Navigate to **/bp2/data/profiles/** location.

   ```
   cd /bp2/data/profiles/
   ```

4. Execute the touch command –

```
touch <ra_name>.pro
```

Example: `touch appmon-pm.pro`

Update User Preferences

1. Users (admin) are required to set default landing page for all users or individual user need to set landing page from the **App Bar UI > Preferences > User Preferences > Default Landing Page**.
2. Once the upgrade is complete, manually configure the LDAP and TACACS settings from the external authentication page if not migrated. Refer user guide for more info.
3. The **fault severity color** for UAA is upgraded from legacy (vSure) UI and will get reset to default post upgrade. If the user needs to use different color, they can create new theme and change color under **Settings > Alarm > Fault Color**. Refer user guide for more info.
4. Users will be required to update their Custom Logo in the UI post upgrade.
5. While upgrading to new architecture, user is required to add/setup the selfmon nodes with the custom script. Refer *Administration Guide* for help.
6. Any display preferences such as Alarm Schema created in the legacy (vSure) UI will not be available in the new UI post upgrade. Users are required to create new schemes/template *(Refer Manage Fields section from the User guide for help)*.
7. Blue Planet recommends **not** to remove/uninstall NetOmnia Action Template as it will also require deleting the associated workflows.
8. Once the upgrade is complete, users are required to create any custom tree(s) they had in the legacy software in 23.08.xx.

Enabling Kafka log messages for File Import (Optional)

1. Log in to host 0 (for multi host setup, repeat the steps in all hosts) as bpadmin. Do not use the site IP; use the IP address of host 0.
2. Enter uaa-core container -

   ```
   sudo solman

   enter_container uaa-core_23.08.xx_x

   Example: enter_container uaa-core_23.08.56_1
   ```

3. Edit the install.properties file –

   ```
   vi /bp2/conf/install.properties
   ```

4. Change the **file_import_staus_kafka_message** value from **false** to **true**
5. Restart the container -

   ```
   solution_app_restart artifactory.ciena.com.blueplanet.uaa_core:ha-23.08.xx

   Example: solution_app_restart artifactory.ciena.com.blueplanet.uaa_core:ha-23.08.20
   ```

Re-Import of Data

The following section describes the re-import of different DB data for UAA.

Re-Import of SA/Config DB Data

There is NO need to drop any tables in config db when running the Re-Import of SA.

1. On 23.08.xx setup, login to any host as **bpadmin** user.
2. Enter solman using command '`sudo solman`'.
3. Enter uaa-upgrade-from-20.02 container leader instance.
4. Execute the following commands script –

   ```
   python3 /bp2/scripts/sa-migration-scripts/importClientSaDb.py
   ```

   When the script executes, it will delete all the config db tables and then re-import the data from the sa.zip file present in the location **/bp2/data/migration/sa/**.
5. Make sure to check the logs for any further errors.
6. Continue with the import as per above SA import section above.

Re-Import of FM Data

Execute the following commands to re-import the FM data -

1. On 23.08.xx setup, login to any host as **bpadmin** user.
2. Enter uaa-core-db container –

   ```
   sudo solman

   enter_container uaa-core-db_xxxx.x.x

   ctl list
   ```
   *(To verify the leader role from the list and use the same for below command **only** in a multi host setup)*

   ```
   enter_container uaa-core-db_xxxx.x.x<leader Host>
   ```

3. Login to uaa-core-dbpostgresDB –

   ```
   psql -h uaa-core-db -U bpadmin -d bpuaa
   ```

4. Execute the sql commands -
   a. The following commands are required for dropping the fault related tables:

      ```
      set search_path to fm;
      DROP TABLE fault_active_alarms;
      DROP SEQUENCE fault_active_alarms_seq;
      DROP TABLE fault_event;
      DROP SEQUENCE fault_event_seq;
      DROP TABLE scm_last_computed_state;
      DROP TABLE alarms_per_minute;
      DROP TABLE purge_history_data_request;
      ```

b. The following commands are required to create the fault related tables:

```
set search_path to fm;
CREATE SEQUENCE fault_active_alarms_seq;

CREATE TABLE fm.fault_active_alarms
       (lastModifiedTime TIMESTAMP(0),
       sequenceNumber BIGINT CHECK (sequenceNumber > 0) NOT NULL DEFAULT NEXTVAL
('fault_active_alarms_seq'),
      raisedTime TIMESTAMP(0) NULL DEFAULT NULL,
      receivedTime TIMESTAMP(0) NULL DEFAULT NULL,
      subnetworkName VARCHAR(128) DEFAULT NULL,
      subnetworkId CHAR(36) DEFAULT NULL,
      networkElementName VARCHAR(128) DEFAULT NULL,
      networkElementId CHAR(36) DEFAULT NULL,
      eventType SMALLINT NOT NULL,
      networkElementSequenceNumber INT DEFAULT NULL,
      sourceName VARCHAR(256) DEFAULT NULL,
      sourceId VARCHAR(256) DEFAULT NULL,
      perceivedSeverity SMALLINT DEFAULT NULL,
      probableCause SMALLINT DEFAULT NULL,
      specificProblem VARCHAR(1024) DEFAULT NULL,
      additionalInformation VARCHAR(4096) DEFAULT NULL,
      acknowledged BOOLEAN DEFAULT NULL,
      owner VARCHAR(64) DEFAULT NULL,
      groupOwned boolean DEFAULT FALSE,
      acknowledgedTime TIMESTAMP(0) NULL DEFAULT NULL,
      timeToAcknowledge INT  DEFAULT NULL,
      commented BOOLEAN DEFAULT NULL,
      cleared BOOLEAN DEFAULT NULL,
      clearedBy VARCHAR(64) DEFAULT NULL,
      clearedTime TIMESTAMP(0) NULL DEFAULT NULL,
      clearSequenceNumber BIGINT DEFAULT NULL,
      priority SMALLINT  DEFAULT NULL,
      count INT DEFAULT 1,
      transitory BOOLEAN DEFAULT NULL,
      direction SMALLINT DEFAULT NULL,
      serviceAffecting BOOLEAN DEFAULT NULL,
      location SMALLINT DEFAULT NULL,
      monitorValue VARCHAR(32) DEFAULT NULL,
      thresholdLevel VARCHAR(32) DEFAULT NULL,
      timePeriod SMALLINT DEFAULT NULL,
      sourceDescription VARCHAR(256) DEFAULT NULL,
      duration INT  DEFAULT NULL,
      serviceAlarmSequenceNumber BIGINT  DEFAULT NULL,
      deferred boolean DEFAULT FALSE,
      deferredUntil TIMESTAMP(0) NULL DEFAULT NULL,
      serviceName VARCHAR(256) DEFAULT NULL,
      serviceId CHAR(36) DEFAULT NULL,
      serviceDescription VARCHAR(256) DEFAULT NULL,
      serviceType VARCHAR(64) DEFAULT NULL,
       profileId VARCHAR(64) DEFAULT NULL,
      resyncable BOOLEAN DEFAULT NULL,
      suppressed boolean DEFAULT FALSE,
       suppressedBySequenceNumber BIGINT DEFAULT NULL,
       troubleTicketId VARCHAR(64) DEFAULT NULL,
      lastRaisedTime TIMESTAMP(0) NULL DEFAULT NULL,
      lastReceivedTime TIMESTAMP(0) NULL DEFAULT NULL,
      customerName VARCHAR(128) DEFAULT NULL,
      customerId VARCHAR(64) DEFAULT NULL,
      underMaintenance boolean DEFAULT FALSE,
      scheduledMaintenanceName varchar(256) DEFAULT NULL,
```

```
      lastComment VARCHAR(512) DEFAULT NULL,
      id VARCHAR(1024) NOT NULL,
      idHash INT DEFAULT NULL,
      correlationPolicyName VARCHAR(255) DEFAULT NULL,
      correlationPolicyId VARCHAR(255) DEFAULT NULL,
      correlationGroupId VARCHAR(512) DEFAULT NULL,
      lastActiveCount INT DEFAULT 0,
      historySequenceNumber BIGINT DEFAULT NULL,
      reraisedTime TIMESTAMP(0) NULL DEFAULT NULL,
      domainName VARCHAR(255) DEFAULT NULL,
      classification int,
      isRootCause boolean,
      flapping BOOLEAN DEFAULT FALSE,
      important BOOLEAN DEFAULT NULL,
      probability DOUBLE PRECISION DEFAULT 0,
      aiPriority SMALLINT  DEFAULT NULL,
      aiPriorityScore INT  DEFAULT NULL,
      appliedRCARule SMALLINT DEFAULT 1000);

ALTER SEQUENCE fault_active_alarms_seq RESTART WITH 1;

CREATE INDEX fault_active_alarms_idx1 ON fault_active_alarms (sequenceNumber);
CREATE INDEX fault_active_alarms_idx2 on fault_active_alarms (receivedTime);
CREATE INDEX fault_active_alarms_idx3 on fault_active_alarms (clearedTime);
CREATE INDEX fault_active_alarms_idx4 on fault_active_alarms (transitory, cleared);
CREATE INDEX fault_active_alarms_idx5 on fault_active_alarms (eventType);
CREATE INDEX fault_active_alarms_idx6 on fault_active_alarms (serviceId);
CREATE INDEX fault_active_alarms_idx7 on fault_active_alarms (subnetworkId);
CREATE INDEX fault_active_alarms_idx8 on fault_active_alarms (networkElementId);
CREATE INDEX fault_active_alarms_idx9 on fault_active_alarms (idHash);
CREATE INDEX fault_active_alarms_idx10 on fault_active_alarms (deferred);
CREATE INDEX fault_active_alarms_idx11 on fault_active_alarms (id);
CREATE INDEX fault_active_alarms_idx12 on fault_active_alarms (domainName);
CREATE INDEX fault_active_alarms_idx13 on fault_active_alarms
(suppressedBySequenceNumber);

CREATE TABLE fm.alarms_per_minute
            (
               alarmCount BIGINT DEFAULT NULL,
               timeStamp timestamp(0)
            );


CREATE TABLE purge_history_data_request
            (
               objectId VARCHAR(128),
               objectType VARCHAR(128),
               objectDataType VARCHAR(128),
               ip VARCHAR(128),
               userName VARCHAR(128),
               startTime TIMESTAMP(0) DEFAULT NULL,
               endTime TIMESTAMP(0) DEFAULT NULL,
               status VARCHAR(128) DEFAULT 'REQUESTED'
            );
CREATE SEQUENCE fault_event_seq;




CREATE TABLE IF NOT EXISTS fm.fault_event
```

```
                (
                lastModifiedTime TIMESTAMP(0),
                sequenceNumber BIGINT CHECK (sequenceNumber > 0) NOT NULL DEFAULT
NEXTVAL ('fault_event_seq'),
                raisedTime TIMESTAMP(0) NULL DEFAULT NULL,
                receivedTime TIMESTAMP(0) NULL DEFAULT NULL,
                subnetworkName VARCHAR(128) DEFAULT NULL,
                subnetworkId CHAR(36) DEFAULT NULL,
                networkElementName VARCHAR(128) DEFAULT NULL,
                networkElementId CHAR(36) DEFAULT NULL,
                eventType SMALLINT NOT NULL,
                networkElementSequenceNumber INT  DEFAULT NULL,
                sourceName VARCHAR(256) DEFAULT NULL,
                sourceId VARCHAR(256) DEFAULT NULL,
                perceivedSeverity SMALLINT  DEFAULT NULL,
                probableCause SMALLINT  DEFAULT NULL,
                specificProblem VARCHAR(1024) DEFAULT NULL,
                additionalInformation VARCHAR(4096) DEFAULT NULL,
                acknowledged BOOLEAN DEFAULT NULL,
                owner VARCHAR(64) DEFAULT NULL,
                groupOwned boolean DEFAULT false,
                acknowledgedTime TIMESTAMP(0) NULL DEFAULT NULL,
                timeToAcknowledge INT  DEFAULT NULL,
                commented BOOLEAN DEFAULT NULL,
                cleared BOOLEAN DEFAULT NULL,
                clearedBy VARCHAR(64) DEFAULT NULL,
                clearedTime TIMESTAMP(0) NULL DEFAULT NULL,
                clearSequenceNumber BIGINT DEFAULT NULL,
                priority SMALLINT DEFAULT NULL,
                count INT DEFAULT 1,
                transitory BOOLEAN DEFAULT NULL,
                direction SMALLINT DEFAULT NULL,
                serviceAffecting BOOLEAN DEFAULT NULL,
                location SMALLINT DEFAULT NULL,
                monitorValue VARCHAR(32) DEFAULT NULL,
                thresholdLevel VARCHAR(32) DEFAULT NULL,
                timePeriod SMALLINT DEFAULT NULL,
                sourceDescription VARCHAR(256) DEFAULT NULL,
                duration INT DEFAULT NULL,
                serviceAlarmSequenceNumber BIGINT DEFAULT NULL,
                deferred boolean DEFAULT false,
                deferredUntil TIMESTAMP(0) NULL DEFAULT NULL,
                serviceName VARCHAR(256) DEFAULT NULL,
                serviceId CHAR(36) DEFAULT NULL,
                serviceDescription VARCHAR(256) DEFAULT NULL,
                serviceType VARCHAR(64) DEFAULT NULL,
                profileId VARCHAR(64) DEFAULT NULL,
                resyncable BOOLEAN DEFAULT NULL,
                suppressed boolean DEFAULT false,
                suppressedBySequenceNumber BIGINT DEFAULT NULL,
                troubleTicketId VARCHAR(64) DEFAULT NULL,
                lastRaisedTime TIMESTAMP(0) NULL DEFAULT NULL,
                lastReceivedTime TIMESTAMP(0) NULL DEFAULT NULL,
                customerName VARCHAR(128) DEFAULT NULL,
                customerId VARCHAR(64) DEFAULT NULL,
                underMaintenance boolean DEFAULT false,
                scheduledMaintenanceName varchar(256) DEFAULT NULL,
                lastComment VARCHAR(512) DEFAULT NULL,
                correlationPolicyName VARCHAR(255) DEFAULT NULL,
                correlationPolicyId VARCHAR(255) DEFAULT NULL,
                correlationGroupId VARCHAR(512) DEFAULT NULL,
                reraisedTime TIMESTAMP(0) NULL DEFAULT NULL,
                activeSequenceNumber BIGINT DEFAULT NULL,
```

```
                    domainName VARCHAR(255) DEFAULT NULL,
                    classification int,
                    isRootCause boolean,
                    important BOOLEAN DEFAULT NULL,
                    probability DOUBLE PRECISION DEFAULT 0,
                    aiPriority SMALLINT  DEFAULT NULL,
                    aiPriorityScore INT  DEFAULT NULL,
                    appliedRCARule SMALLINT DEFAULT 1000
                    )  PARTITION BY RANGE (receivedtime);

ALTER SEQUENCE fault_event_seq RESTART WITH 50000000000;

CREATE INDEX fault_event_idx1 ON fault_event (sequenceNumber);
CREATE INDEX fault_event_idx2 on fault_event (receivedTime);
CREATE INDEX fault_event_idx3 on fault_event (clearedTime);
CREATE INDEX fault_event_idx4 on fault_event (transitory, cleared);
CREATE INDEX fault_event_idx5 on fault_event (eventType);
CREATE INDEX fault_event_idx6 on fault_event (serviceAlarmSequenceNumber);
CREATE INDEX fault_event_idx7 on fault_event (subnetworkId);
CREATE INDEX fault_event_idx8 on fault_event (networkElementId);
CREATE INDEX fault_event_idx9 on fault_event (domainName);
CREATE INDEX fault_event_idx10 on fault_event (suppressedBySequenceNumber);
CREATE INDEX fault_event_idx_asn_cleared on fault_event (activeSequenceNumber,
cleared);
CREATE TABLE IF NOT EXISTS fm.scm_last_computed_state ( scmId VARCHAR(255) NOT NULL,
lastComputedStatus int DEFAULT NULL );
```

5. Restart uaa-fme leader instance to create partitions for the fault_event table:

   ```
   solution_app_restart artifactory.ciena.com.blueplanet.uaa_fme:23.xx.xx uaa-fme
   ```

6. It is required to clean up the data in the clickhouse database as it will be storing the fault comments table data. Login to leader instance of uaa-pm-db to delete the fault_comments table;

   ```
   enter_container uaa-pm-db_xx.xx-xx.xx

   clickhouse-client -u bpadmin --password <password>
   ```

   (password will be present under /dev/shm/users.json)

   ```
   use fm;

   TRUNCATE TABLE fm.fault_comments;

   use fm_local;

   TRUNCATE TABLE fm_local.fault_comments_local;
   ```

7. Continue with the import as per above FM import section.

Re-Import of PM Data

Execute the following commands to re-import the PM data –

1. To delete the cilckhouse data, execute the following command to re-deploy the uaa_pm_storage container by using the following commands on any host as **bpadmin**:

   ```
   sudo solman
   solution_redeploy artifactory.ciena.com.blueplanet.uaa_pm_storage:23.xx.xx --purge-host-vols
   ```

2. Restart the uaa-pm-db-scheduler to create the databases and its respective tables -

   ```
   sudo solman
   solution_app_restart artifactory.ciena.com.blueplanet.uaa_storage:23.xx.xx uaa-pm-db-scheduler
   ```

3. Post restarting the uaa-pm-db scheduler, login to clickhouse database and check if the databases are created.

   ```
   enter_container uaa-pm-db_xx.xx-xx.xx
   clickhouse-client -u bpadmin --password <password>
   ```

   (password will be present under /dev/shm/users.json)

   ```
   show databases;
   ```

   ```
   INFORMATION_SCHEMA
   default
   fm
   fm_local
   information_schema
   pm
   pm_local
   pmda
   sa
   sla
   spm
   system
   uaa_stats
   ```

8. Continue with the import as per above PM import section.
9. Make sure to delete the **status.csv** and **import_termination.csv** under the location **/bp2/scripts/pm-migration-scripts/** while running import again.

Once the PM Import is completed as per the above steps, continue below –

10. When PM data is re-imported, it is required to import the fm fault_comments data into the clickhouse. Execute the following steps to re-import fault_comments data into clickhouse.
    a. Navigate to the location

       ```
       cd /bp2/data/migration/fm/
       ```

    b. Unzip the fm_dump.zip folder

       ```
       unzip fm_dump.zip
       ```

c. Keep only csv's related to fault_comments and delete all other csv's from the fm _dump directory.
Example: Keep only csv file names which starts with fault comments 'fault_comments_2023-08-24_2023-08-31_0.csv'

d. Move the fm_dump.zip file to a temporary location of user's choice.

e. Zip all the fault_comment csv's present in the **bp2/data/migration/fm** directory.

```
cd /bp2/data/migration/fm/

zip -r fm_dump.zip *
```

f. Execute the following commands -

```
cd /bp2/scripts/fm-migration-scripts/

python3 ImportFmDbDriver.py
```

g. Check the logs, /bp2/scripts/fm-migration-scripts/fm_db_migration_import.log for any errors.

h. Verify if the fault comments data is migrated into clickhosue database.

```
enter_container uaa-pm-db_xx.xx-xx.xx

clickhouse-client -u bpadmin --password <password>

(password will be present under /dev/shm/users.json)

use fm;
```

sample query: `select * from fault_comments LIMIT 10;`

Contacting Blue Planet

Blue Planet Division Headquarters	7035 Ridge Road Hanover, MD 21076 +1 800-921-1144
Blue Planet Support	https://www.blueplanet.com/support
Sales and General Information	https://www.blueplanet.com/contact
Training	https://www.blueplanet.com/learning

For additional information, please visit https://www.blueplanet.com.

LEGAL NOTICES

Security

Ciena® cannot be responsible for unauthorized use of equipment and will not make allowance or credit for unauthorized use or access.