Post Upgrade Tasks

Upgrade Cisco Prime Collaboration Provisioning from Small to Medium Deployment Model

After you manually upgrade the system requirements (vRAM, vCPU, vDISK, and so on), you must run the following scripts as a troubleshooting user:

1. Execute the **memorymodel.sh** file under `/opt/cupm`:

   ```
   ./memorymodel.sh medium "-Xms512m -Xmx1024m -XX:MaxPermSize=256m
   -server" "-Xms512m -Xmx1024m -XX:MaxPermSize=256m" simple all
   ```

2. Execute **cpcmdiskutil.sh** under `/opt/cupm`:

   ```
   . /cpcmdiskutil.sh /dev/sda
   ```

3. Restart the server(VMware instance).

> ✎
>
> **Note** If you want to support endpoint count as per the upgraded deployment model in Cisco Prime Collaboration Provisioning, you must add a scale license.

Upgrade Cisco Prime Collaboration Provisioning Server from Small or Medium to Large Deployment Model

1. Back up the database from the Cisco Prime Collaboration Provisioning application by following the procedures provided in Cisco Prime Collaboration Provisioning Guide.

2. Deploy a large OVA as the server. If it is a distributed setup, deploy both application and database servers. For more information on deploying and configuring the OVA, refer to Installation Tasks. Ensure that the globaladmin password is the same as the password that you provided during the deployment.

   a. Copy the licenses from the old server to the new server.

   b. If you use the MAC address of the existing Cisco Prime Collaboration Provisioning server, then update the MAC address using the VMware client for this VMware instance.

   c. If you use a new MAC address for server, then you must rehost the licenses in the `/opt/cupm/license` directory to match the new server VM.

3. Perform the following steps from the Troubleshooting Account. For information on creating a Troubleshooting Account, see Create a Troubleshooting Account.

4. Stop provisioning services in the server.

   a. Go to the *opt/cupm* folder.

      Execute **sudo ./cupm-app-service.sh stop**

   b. Ensure that Apache, JBoss, and NICE services are stopped.

      ps -aef | grep standalone

      ps -aef | grep nice

   c. Stop any running process.

      kill -9 *standalone process id*

      kill -9 *nice process id*

   d. Check whether the nice process is still holding on the postgres connection.

      ps -aef

   e. Look for the process:

      /opt/cupm/jvm/bin/java -server -classpath /opt/cupm/sep/lib/dom.jar

   f. Stop any running nice process.

      kill -9 *Process-Id found earlier*

   g. Wait a minute to make the resources, such as ports, to become free.

5. Restore the database in the server using the backed-up database file taken from Step 1. For details, see the section Restoring Database in the server section in the Cisco Prime Collaboration Provisioning Guide.

6. Stop and then start the provisioning services in the server.

   a. **cd /opt/cupm** folder.

      sudo ./cupm-db-service.sh stop

   b. Wait 30 seconds before starting the database services

   c. Start the database services.

      cd /opt/cupm

      sudo ./cupm-db-service.sh start

7. Copy the following files from the original Cisco Prime Collaboration Provisioning server to the newly deployed server

    a. `/opt/cupm/sep/dfc.properties`

    b. `/opt/cupm/sep/dfc.keystore`

    c. `/opt/cupm/jboss/standalone/configuration/standalone-full.xml`

8. Change the directory to `/opt/cupm/sep` and edit the `dfc.properties` file using the vi editor

    a. **cd /opt/cupm/sep**

    b. **vi dfc.properties**

    c. Change the property dfc.memory.model=medium to **`dfc.memory.model=large`**

    d. Change the property dfc.postgres.host=localhost to **`dfc.postgres.host=IP of server Database`**

    e. Save changes and exit the editor

9. Start application services in the server.

    a. Change directory to `/opt/cupm` to start the application services

    b. **cd /opt/cupm**

    sudo ./cupm-app-service.sh start

    The system is now ready to be used.