Containerized IBM Security Guardium Key Lifecycle Manager
Version 4.1

Documentation
(BETA 3)

IBM Confidential

Note

Before you use this information and the product it supports, read the information in "Notices" on page 63.

Product information

The Beta Program and this documentation is provided to you AS IS without any warranties express or implied, including the warranty of merchantability or fitness for a particular purpose. IBM may choose, in its own discretion, to change features and functions this Beta Program prior to being made generally available or choose not to make this Beta Program generally available.

Contents

Chapter 1. Deploying with containers

The containerized IBM® Security Guardium Key Lifecycle Manager application provides a simpler deployment experience that can be easily scaled and upgraded.

The containerized deployment comprises a set of preinstalled images for the database and the application. You must install the images in the following order:

1. Database
2. IBM Security Guardium Key Lifecycle Manager application

For more information, see "Overview of the containerized IBM Security Guardium Key Lifecycle Manager" on page 1.

You can deploy the containers on one of the following platforms:

Kubernetes

You can install the product in a containerized server cluster, managed by Kubernetes. IBM Security Guardium Key Lifecycle Manager works with Kubernetes to simplify application deployment and manage versions in the containers.

For more information, see "Deploying on a Kubernetes cluster" on page 2.

Docker

Docker is a state-of-the-art technology that creates, deploys, and runs applications by using containers.

For more information, see the topics:

- "Deploying with PostgreSQL using Docker" on page 3
- "Deploying with Db2 using Docker" on page 5
- "Deploying with HSM using Docker" on page 7

IBM zCX environment

IBM z/OS Container Extensions (zCX) provides a Docker runtime environment on z/OS.

If you have a zCX instance available, you can deploy an IBM Security Guardium Key Lifecycle Manager Docker container in it and have the product running in a z/OS address space.

For more information, see "Deploying on IBM zCX environment with Db2 for z/OS" on page 11.

Overview of the containerized IBM Security Guardium Key Lifecycle Manager

The containerized IBM Security Guardium Key Lifecycle Manager application provides a simpler deployment experience that can be easily scaled and upgraded.

A containerized deployment for IBM Security Guardium Key Lifecycle Manager requires a database container to be deployed prior to deploying the IBM Security Guardium Key Lifecycle Manager application container. The database can be Db2 or PostgreSQL.

The IBM Security Guardium Key Lifecycle Manager application container includes IBM WebSphere Liberty on which the IBM Security Guardium Key Lifecycle Manager application is installed. The IBM Security Guardium Key Lifecycle Manager application connects to the database that is deployed in the database container.

The database files, and the application files such as configuration information, keystore, truststore, user data, debug, and audit log files, are stored in Volumes (Persistent storage).

Deploying on a Kubernetes cluster

Use Helm charts to deploy IBM Security Guardium Key Lifecycle Manager containers on a Kubernetes cluster. This deployment is supported only with PostgreSQL database.

Before you begin

- Set up a Kubernetes cluster. You can use Version 1.17 or later. For more information, see https://kubernetes.io/docs/setup/.
- If you plan to use the Db2 database, ensure that you have an account on the Docker Hub.
- Install Helm Version 2.0 or later on the system from which you will access the Kubernetes cluster. For more information, see https://helm.sh/docs/intro/install/.
- To obtain the license activation file for IBM Security Key Lifecycle Manager, send us an email at: ibmsklm@in.ibm.com.

Procedure

Complete the following steps on the system on which you installed Helm:

1. Download the k8s-helm.zip file that contains the sample Helm charts for deploying IBM Security Guardium Key Lifecycle Manager.
2. Extract the `k8s-helm.zip` file.
3. In the directory where you extracted the files, navigate to **k8s-helm** > **sklm** directory.
4. Open the `values.yaml` file and modify the parameter values in the file as per your requirement.
5. Navigate to `k8s-helm` directory and run the following command:

   ```
   helm install sklm
   ```

   Note:

   If you are using Helm Version 3.0 or later, use the following command:

   ```
   helm install name sklm
   ```
6. Verify the installation by running the following commands:

   ```
   helm list
   kubectl get pods
   kubectl get pv
   kubectl get pvc
   ```
7. Launch the IBM Security Guardium Key Lifecycle Manager graphical user interface.

   ```
   https://ip-address:port/ibm/SKLM/login.jsp
   ```

   The value of *ip-address* is the IP address or DNS address of the IBM Security Guardium Key Lifecycle Manager server.

   The value of *port* is the port number that IBM Security Guardium Key Lifecycle Manager server listens on for requests.
8. On the Configuration page that appears, click the **License Agreements** link to review the license terms, and then select the **I accept the terms in the License Agreements** check box.
9. Click **Activate License**.
10. Upload the IBM Security Guardium Key Lifecycle Manager license activation file and activate the license.
11. Click **Login**.
12. Log in to the IBM Security Guardium Key Lifecycle Manager graphical user interface with the Administrator user credentials (`sklmadmin`).

13. Optional: Configure Kubernetes to call the Health Status REST ServiceHealth Status REST Service.

    Health checks are a simple way to determine whether a server-side application is working properly. Kubernetes requires two types of health checks: readiness probe and liveness probe. These probes are implemented by performing an HTTPS invocation by using the REST interface.

    For more information about configuring liveness and readiness probes, see the Kubernetes documentation.

What to do next

From the Welcome page, configure the drive types, keys, and certificates that your organization requires, or get started with using the product. See Working with IBM Security Key Lifecycle ManagerSee Administering.

Deploying with PostgreSQL using Docker

Use instructions in this topic to deploy IBM Security Guardium Key Lifecycle Manager with PostgreSQL by using Docker.

Before you begin

- Ensure that the host system meets these minimum system requirements:

Table 1. Minimum system requirements

Resource	Requirement
CPU	4 Cores
Memory	8 GB
Disk space	100 GB
Operating system and Supported architectures	Linux – x86_64 – s390x

- Install Docker engine on the host system. For instructions, see https://docs.docker.com/.
- Ensure that you have an account on the Docker Hub.
- To obtain the license activation file for IBM Security Key Lifecycle Manager, send us an email at: ibmsklm@in.ibm.com.
- Set up the PostgreSQL container. Sample command:

```
docker run -d -v sklmpostgresdbvolume:/var/lib/postgresql/data -e
 POSTGRES_PASSWORD=sklmpostgres -e POSTGRES_USER=sklmdb41 -e POSTGRES_DB=sklmdb41 -p 5432:5432
 postgres
```

  Note: The container might take a few minutes to start. You can monitor the progress by using the docker logs command.

Procedure

Complete the following steps on the host system:

1. Log in to Docker Hub.
2. Optional: Create an environment variable list file with the parameters for the IBM Security Guardium Key Lifecycle Manager container.

   For more information about the parameters, see "Parameters to deploy IBM Security Guardium Key Lifecycle Manager container using Docker" on page 13.
3. Ensure that the database container is running and ready to accept connections.

4. Run the IBM Security Guardium Key Lifecycle Manager application Docker container by using the parameter list file or by specifying the parameters in the command.

   Sample command with the parameter list file:

   ```
   docker run --name sklm_test -itd -h sklm.com -p 9443:9443 -p 3801:3801 -p 5696:5696 -p
    1441:1441  --env-file=sklmenvp.txt  -v sklmAppVolume_new:/opt/ibm/wlp/usr/products ibmcom/
   sklm
   ```

   Sample command with parameters:

   ```
   docker run --name sklm -itd -h sklm.com -p 9443:9443 -p 3801:3801 -p 5696:5696
    -p 1441:1441  -e LICENSE=accept -e SKLMADMIN_USERNAME=sklmadminuser -e
    LIBERTY_KEY_STORE_PASSWORD=kspassword -e SKLMADMIN_PASSWORD=sklmpswd -e DB_HOST=172.x.x.x
    -e DB_PORT=5432 -e SKLM_SEED=68d95f0081f1dbfc0b06de9b0916df1c -e DB_PASSWORD=dbpswd -e
    DB_TYPE=postgres -e DB_USER=sklmdb41 -e DB_NAME=sklmdb41 -v sklmAppVolume:/opt/ibm/wlp/usr/
   products ibmcom/sklm
   ```

   Note: The container might take a few minutes to start. You can monitor the progress by using the docker logs command.

5. Monitor the progress by using the docker logs command.

   ```
   docker logs -f sklm
   ```

   After you see the following message in the logs, proceed to the next step:

   ```
   IBM Security Guardium Key Lifecycle Manager server started.
   ```

6. Launch the IBM Security Guardium Key Lifecycle Manager graphical user interface.

   ```
   https://ip-address:port/ibm/SKLM/login.jsp
   ```

   The value of *ip-address* is the IP address or DNS address of the IBM Security Guardium Key Lifecycle Manager server.

   The value of *port* is the port number that IBM Security Guardium Key Lifecycle Manager server listens on for requests.

7. On the Configuration page that appears, click the **License Agreements** link to review the license terms, and then select the **I accept the terms in the License Agreements** check box.
8. Click **Activate License**.
9. Upload the IBM Security Guardium Key Lifecycle Manager license activation file and activate the license.
10. Click **Login**.
11. Log in to the IBM Security Guardium Key Lifecycle Manager graphical user interface with the Administrator user credentials (`sklmadmin`).

What to do next

From the Welcome page, configure the drive types, keys, and certificates that your organization requires, or get started with using the product. See Working with IBM Security Key Lifecycle ManagerSee Administering.

Deploying with Db2 using Docker

Use instructions in this topic to deploy IBM Security Guardium Key Lifecycle Manager with Db2 by using Docker.

Before you begin

- Ensure that the host system meets these minimum system requirements:

Table 2. Minimum system requirements

Resource	**Requirement**
CPU	4 Cores
Memory	8 GB
Disk space	100 GB
Operating system and supported architectures	Linux – x86_64 – s390x

- Install Docker engine on the host system. For instructions, see https://docs.docker.com/.
- Ensure that you have an account on the Docker Hub.
- To obtain the license activation file for IBM Security Key Lifecycle Manager, send us an email at: ibmsklm@in.ibm.com.
- Set up Db2 by using one of the following options:

Obtain the Db2 container image from Docker Hub and customize it

Note: You can customize Db2 for IBM Security Guardium Key Lifecycle Manager only with the **Standard** or **Advanced** edition of Db2. Ensure that you are using the required license key for one of these editions. The file type for the license is **.lic**. For example, `db2awse_c_np.lic`.

To obtain the Db2 image, go to the IBM Db2 container.

To customize the Db2 container:

1. Download the attached file and extract its content in a directory on the host system.
2. Edit the `Dockerfile.sample` file, as required, and save the file. You can use any text editor.
3. Run the following command from the directory where the `Dockerfile.sample` file is extracted:

```
docker build -t sklmdb -f Dockerfile.sample --no-cache .
```

4. Run the customized Db2 container. For example:

```
docker run --name sklmdb --restart=always --detach --ipc="" --cap-add=IPC_OWNER -p
 50000:50000 -e LICENSE=accept -e DB2INSTANCE=sklmdb41 -e DB2INST1_PASSWORD=SKLMdb2 -e
 DBNAME=sklmdb41 -v sklmDb2Volume:/database sklmdb
```

For more information, see https://hub.docker.com/r/ibmcom/db2.

Use an existing standalone (on-premise) version of Db2

You can use an existing version of Db2 and create an empty or blank database.

Note: Minimum supported version of the standalone Db2 is Version 11.1.4.4 interim fix 1.

The Db2 container might take a few minutes to start. You can monitor the progress by using the docker logs command.

Procedure

Complete the following steps on the host system:

1. Log in to Docker Hub.
2. Optional: Create an environment variable list file with the parameters for the IBM Security Guardium Key Lifecycle Manager container.

   For more information about the parameters, see “Parameters to deploy IBM Security Guardium Key Lifecycle Manager container using Docker” on page 13.
3. Ensure that the Db2 container is running and ready to accept connections.
4. Run the IBM Security Guardium Key Lifecycle Manager application Docker container by using the parameter list file or by specifying the parameters in the command.

   Sample command with the parameter list file:

   ```
   docker run --name sklmapp -itd -h sklm.com -p 9443:9443 -p 3801:3801 -p 5696:5696 -p
    1441:1441  --env-file=sklmenv.txt -v sklmAppVolume_db2:/opt/ibm/wlp/usr/products ibmcom/
   sklm
   ```

   Sample command with parameters:

   ```
   docker run --name sklm -itd -h sklm.com -p 9443:9443 -p 3801:3801 -p 5696:5696
    -p 1441:1441  -e LICENSE=accept -e LIBERTY_KEY_STORE_PASSWORD=keystorepswd -e
    SKLMADMIN_USERNAME=sklmadminuser -e SKLMADMIN_PASSWORD=sklmpswd -e DB_HOST=172.x.x.x -
   e DB_PORT=50000 -e SKLM_SEED=68d95f0081f1dbfc0b06de9b0916df1c -e DB_PASSWORD=sklmdb2 -e
    DB_TYPE=db2 -e DB_USER=sklmdb41 -e DBNAME=sklmdb41 -v sklmAppVolume_db2:/opt/ibm/wlp/usr/
   products ibmcom/sklm
   ```
5. Monitor the progress by using the docker logs command.

   ```
   docker logs -f sklm
   ```

   After you see the following message in the logs, proceed to the next step:

   ```
   IBM Security Guardium Key Lifecycle Manager server started.
   ```
6. Launch the IBM Security Guardium Key Lifecycle Manager graphical user interface.

   ```
   https://ip-address:port/ibm/SKLM/login.jsp
   ```

   The value of *ip-address* is the IP address or DNS address of the IBM Security Guardium Key Lifecycle Manager server.

   The value of *port* is the port number that IBM Security Guardium Key Lifecycle Manager server listens on for requests.
7. On the Configuration page that appears, click the **License Agreements** link to review the license terms, and then select the **I accept the terms in the License Agreements** check box.
8. Click **Activate License**.
9. Upload the IBM Security Guardium Key Lifecycle Manager license activation file and activate the license.
10. Click **Login**.
11. Log in to the IBM Security Guardium Key Lifecycle Manager graphical user interface with the Administrator user credentials (`sklmadmin`).

What to do next

From the Welcome page, configure the drive types, keys, and certificates that your organization requires, or get started with using the product. See Working with IBM Security Key Lifecycle ManagerSee Administering.

Deploying with HSM using Docker

You can configure the IBM Security Guardium Key Lifecycle Manager application container to use an Hardware Security Module (HSM) or a cryptographic card for storing the master encryption key, which protects the key materials that are stored in the database.

Before you begin

- Ensure that the host system meets these minimum system requirements:

Table 3. Minimum system requirements

Resource	Requirement
CPU	4 Cores
Memory	8 GB
Disk space	100 GB
Operating system and supported architectures	Linux – x86_64 – s390x

- Install Docker engine on the host system. For instructions, see https://docs.docker.com/.
- Ensure that you have an account on the Docker Hub.
- To obtain the license activation file for IBM Security Key Lifecycle Manager, send us an email at: ibmsklm@in.ibm.com.

- Ensure that the database container is running and ready to accept connections.
  - Set up Db2 by using one of the following options:

    Obtain the Db2 container image from Docker Hub and customize it

    Note: You can customize Db2 for IBM Security Guardium Key Lifecycle Manager only with the **Standard** or **Advanced** edition of Db2. Ensure that you are using the required license key for one of these editions. The file type for the license is **.lic**. For example, db2awse_c_np.lic.

    To obtain the Db2 image, go to the IBM Db2 container.

    To customize the Db2 container:

    1. Download the attached file and extract its content in a directory on the host system.
    2. Edit the Dockerfile.sample file, as required, and save the file. You can use any text editor.
    3. Run the following command from the directory where the Dockerfile.sample file is extracted:

       ```
       docker build -t sklmdb -f Dockerfile.sample --no-cache .
       ```

    4. Run the customized Db2 container. For example:

       ```
       docker run --name sklmdb --restart=always --detach --ipc="" --cap-add=IPC_OWNER -p
        50000:50000 -e LICENSE=accept -e DB2INSTANCE=sklmdb41 -e DB2INST1_PASSWORD=SKLM@db2
        -e DBNAME=sklmdb41 -v sklmDb2Volume:/database sklmdb
       ```

       For more information, see https://hub.docker.com/r/ibmcom/db2.

    Use an existing standalone (on-premise) version of Db2

    You can use an existing version of Db2 and create an empty or blank database.

    Note: Minimum supported version of the standalone Db2 is Version 11.1.4.4 interim fix 1.

    The Db2 container might take a few minutes to start. You can monitor the progress by using the docker logs command.
  - Set up the PostgreSQL container by running the following command:

    ```
    docker run -d -v sklmpostgresdbvolume:/var/lib/postgresql/data -e
     POSTGRES_PASSWORD=SKLM@postgres -e POSTGRES_USER=sklmdb41 -e POSTGRES_DB=sklmdb41 -p
     5432:5432 postgres
    ```

    Note: The container might take a few minutes to start. You can monitor the progress by using the docker logs command.
- Ensure that the HSM server is installed and connected with the host system on which you plan to deploy the application container.
- Create Setups directory on the host system and copy the HSM client installer into it.
- Create a file (Dockerfile) with the following content:

  ```
  # Extend from SKLM Application Repository
  ARG LATEST_IMAGE
  FROM ${LATEST_IMAGE}

  ARG CLIENT_CERT_NAME

  USER root

  #Copy HSM Client Installer Setup
  COPY ./Setups /Setups

  # Install HSM Client
  #  && Register Server certificate
  #  && Create Client Certificate
  #  && Copy Client Certificate to HSM Server
  #  && Display the Client Certificate content
  RUN yes|/Setups/linux/64/install.sh -p sa -c jsp jcprov \
          &&  /usr/safenet/lunaclient/bin/vtl addServer -n <IP address of HSM server> -c /
  Setups/server.pem \
          && /usr/safenet/lunaclient/bin/vtl  createcert -n ${CLIENT_CERT_NAME} \
  ```

```
        && cp /usr/safenet/lunaclient/cert/client/${CLIENT_CERT_NAME}.pem /opt/ibm/wlp/usr/
sklm/stagingFiles/products/sklm/ \
        && echo 'COPY BELOW CERTIFICATE AND SAVE AS '${CLIENT_CERT_NAME}.pem && cat /usr/
safenet/lunaclient/cert/client/${CLIENT_CERT_NAME}.pem

#switch to non root user
USER 1001
```

In the file, replace *<IP address of HSM server>* with the IP address of your HSM server.

Note: If required, modify the Dockerfile file.

About this task

IBM Security Guardium Key Lifecycle Manager uses the IBM PKCS11 Cryptographic Provider, and works with the HSMs that the provider supports.

For a complete list of supported cryptographic cards, see IBM Security Key Lifecycle Manager Support Matrix.

Procedure

1. On the host system on which you plan to deploy the application container, build the IBM Security Guardium Key Lifecycle Manager application container image using the docker file.

   The docker file installs the HSM client on the host system.

   ```
   docker build -t sklmhsm --build-arg LATEST_IMAGE=ibmcom/sklm --build-arg
    CLIENT_CERT_NAME=fqdn_name --no-cache .
   ```

   where, *fqdn_name* is the fully-qualified domain name of the host system.

   Copy the client certificate details that are displayed on the screen.
2. Copy the client certificate details to a file and name the file as *fqdn_name* (Value of `CLIENT_CERT_NAME`).
3. Complete these steps on the host system on which the HSM server is installed.

   a) Log in to the HSM server.

   b) Register the HSM client IP address.

   ```
   client register -client luna_client_sklmapp  -ip HSM client IP address
   ```

   where, *HSM client IP address* is the IP address of the HSM client.

   c) Assign a server partition to the client.

   ```
   client assignPartition -client luna_client_sklmapp  -partition partition name
   ```

   where, *partition name* is the name of the server partition.
4. Run the IBM Security Guardium Key Lifecycle Manager application container.

   Sample command with parameters:

   ```
   docker run --name sklm -itd -h sklm.com -p 9080:9080 -p 9443:9443 -p 9083:9083 -p 5696:5696
    -p 1441:1441 -e DB_HOST=172.x.x.x -e DB_PORT=5432 -v sklmAppVolume:/opt/ibm/wlp/usr/
   products -e SKLM_SEED=53910c676589ec555eccfb4f25b235fc -e DB_PASSWORD=dbpassword -e
    DB_TYPE=postgres -e DB_USER=sklmdb41 -e DBNAME=sklmdb41 sklmhsm
   ```

   For more information about the parameters, see "Parameters to deploy IBM Security Guardium Key Lifecycle Manager container using Docker" on page 13.

   Note: The container might take a few minutes to start. You can monitor the progress by using the docker logs command.

5. Monitor the progress by using the docker logs command.

   ```
   docker logs -f sklm
   ```

   After you see the following message in the logs, proceed to the next step:

   ```
   IBM Security Guardium Key Lifecycle Manager server started.
   ```

6. Log in to the IBM Security Guardium Key Lifecycle Manager application container .
7. Update the `/opt/ibm/java/jre/lib/security/java.security` file with security provider as `com.ibm.crypto.pkcs11.provider.IBMPKCS11`.
8. Add the **`pkcs11.pin`**, **`pkcs11.config`**, and **`useMasterKeyInHSM`** parameters to the IBM Security Guardium Key Lifecycle Manager server configuration file (`SKLMConfig.properties`).

   ```
   "pkcs11.config": "/Setups/luna.cfg",
   "pkcs11.pin": "HSM_PIN",
   "useMasterKeyInHSM" : "true"
   ```

   where, *HSM_PIN* is the PIN for HSM. You can use the Update Config Property REST Service to update the configuration file.
9. Copy the `luna.cfg` and `server.pem` to the `/Setups` directory in the container.
10. Stop the IBM Security Guardium Key Lifecycle Manager application Docker container .

    ```
    docker stop container-id
    ```

    where, *container-id* is the ID that uniquely identifies the application Docker container.
11. Start the application container.

    ```
    docker start container-id
    ```

12. Launch the IBM Security Guardium Key Lifecycle Manager graphical user interface.

    ```
    https://ip-address:port/ibm/SKLM/login.jsp
    ```

    The value of *ip-address* is the IP address or DNS address of the IBM Security Guardium Key Lifecycle Manager server.

    The value of *port* is the port number that IBM Security Guardium Key Lifecycle Manager server listens on for requests.
13. On the Configuration page that appears, click the **License Agreements** link to review the license terms, and then select the **I accept the terms in the License Agreements** check box.
14. Click **Activate License**.
15. Upload the IBM Security Guardium Key Lifecycle Manager license activation file and activate the license.
16. Click **Login**.
17. Log in to the IBM Security Guardium Key Lifecycle Manager graphical user interface with the Administrator user credentials (`sklmadmin`).

What to do next

From the Welcome page, configure the drive types, keys, and certificates that your organization requires, or get started with using the product. See Working with IBM Security Key Lifecycle ManagerSee Administering.

Deploying on IBM zCX environment with Db2 for z/OS

Use instructions in this topic to deploy IBM Security Guardium Key Lifecycle Manager on IBM zCX environment with Db2 for z/OS.

Before you begin

- Prepare the database system:
  - Install Db2 for z/OS. For more information, see Installing and migrating Db2.
  - Create a database. You can use these parameter values:

    ```
    DB_USER=sklmdb41
    DB_NAME=sklmdb41
    ```

  - Ensure that you have the license file for Db2 for z/OS, db2jcc_license_cisuz.jar. This file is used by the Guardium Key Lifecycle Manager application container to connect to the Db2 for z/OS database.
- Prepare the host system with the IBM zCX environment.
  - Ensure that your host system meets these minimum system requirements for the Guardium Key Lifecycle Manager application container:

    Table 4. Minimum system requirements

Resource	**Requirement**
CPU	4 zIIP
Memory	8 GB
User data disk space	60 GB

  - Provision an IBM z/OS Container Extension (zCX) instance on the host system. For more information, see What is z/OS Container Extension? .
- To obtain the license activation file for IBM Security Key Lifecycle Manager, send us an email at: ibmsklm@in.ibm.com.
- Save the license file (sklm.license.zip) to the host system.
- Create a file `Dockerfile` with the following content and save the file in the same directory where you saved the license file (sklm.license.zip) on the host system.

  ```
  # Extend from SKLM Application Repository
  ARG LATEST_IMAGE
  FROM ${LATEST_IMAGE}
  ARG DB2_LICENSE_FILE=${DB2_LICENSE_FILE}


  #Copy license file to SKLM
  COPY $DB2_LICENSE_FILE /opt/ibm/wlp/usr/sklm/custom

  # Set Environment variable
  ENV DB2_LICENSE_FILE=$DB2_LICENSE_FILE
  ```

Procedure

Complete the following steps on the host system with the IBM zCX environment:

1. Log in to the host system and navigate to the directory where you saved the Guardium Key Lifecycle Manager license and Docker files.
2. Save the license file for Db2 for z/OS (db2jcc_license_cisuz.jar) in this directory where the Guardium Key Lifecycle Manager application files are located.

3. Build the Docker image of the Guardium Key Lifecycle Manager application by using the Docker file.

   ```
   docker build -t sklmzos --build-arg LATEST_IMAGE=ibmcom/sklm --build-arg
    DB2_LICENSE_FILE=db2jcc_license_cisuz.jar --no-cache .
   ```

4. Ensure that the database (sklmdb41) is running and ready to accept connections.
5. Create an environment variable list file with the parameters for the IBM Security Guardium Key Lifecycle Manager container.

   For more information about the parameters, see “Parameters to deploy IBM Security Guardium Key Lifecycle Manager container using Docker” on page 13.
6. Run the IBM Security Guardium Key Lifecycle Manager application Docker container.

   Sample command with the parameter list file:

   ```

   docker run --name sklmapp -itd -h sklm.com -p 9443:9443 -p 3801:3801 -p 5696:5696 -p
    1441:1441 --env-file=sklmenvz.txt -v sklmAppVolume:/opt/ibm/wlp/usr/products sklmzos
   ```

   Note: The container might take a few minutes to start. You can monitor the progress by using the docker logs command.
7. Monitor the progress by using the docker logs command.

   ```
   docker logs -f sklmapp
   ```

   After you see the following message in the logs, proceed to the next step:

   ```
   IBM Security Guardium Key Lifecycle Manager server started.
   ```

8. Launch the IBM Security Guardium Key Lifecycle Manager graphical user interface.

   ```
   https://ip-address:port/ibm/SKLM/login.jsp
   ```

   The value of *ip-address* is the IP address or DNS address of the IBM Security Guardium Key Lifecycle Manager server.

   The value of *port* is the port number that IBM Security Guardium Key Lifecycle Manager server listens on for requests.
9. On the Configuration page that appears, click the **License Agreements** link to review the license terms, and then select the **I accept the terms in the License Agreements** check box.
10. Click **Activate License**.
11. Upload the IBM Security Guardium Key Lifecycle Manager license activation file and activate the license.
12. Click **Login**.
13. Log in to the IBM Security Guardium Key Lifecycle Manager graphical user interface with the Administrator user credentials (`sklmadmin`).

What to do next

From the Welcome page, configure the drive types, keys, and certificates that your organization requires, or get started with using the product. See Working with IBM Security Key Lifecycle ManagerSee Administering.

Parameters to deploy IBM Security Guardium Key Lifecycle Manager container using Docker

You can create a file with environment variables and parameters that the IBM Security Guardium Key Lifecycle Manager container can use for deployment.

Table 5. Parameters and their descriptions

Parameter	Mandatory/ Optional	Description
Container name		
name	Mandatory	Name for the container.
Environment variables		
DB_PASSWORD	Mandatory	Password to connect to the database instance where the IBM Security Guardium Key Lifecycle Manager database is running
DB_TYPE	Optional	Type of the database. Depending on the database that you use, specify one of the following values: **db2 (Default value)** Db2 database **postgres** PostgreSQL database **zos_db** Native (non-container) Db2 for z/OS **Note:** This parameter is ignored in the subsequent docker run commands when the same value of the **sklmAppVolume** parameter is used.
DB_USER	Optional	User name of the database. Default value: sklmdb41
DB_NAME	Optional	Name of the database. When the value of **DB_TYPE** is zos_db, specify the location name of the database. Default value: sklmdb41
DB_PORT	Mandatory	Port number of the database instance where the IBM Security Guardium Key Lifecycle Manager database is running
DB_HOST	Mandatory	IP address or fully qualified host name of the system that hosts the database instance where the IBM Security Guardium Key Lifecycle Manager database is running. You can use the same system to host the database instance and the application Docker container, or choose a different system for each of them.

Table 5. Parameters and their descriptions (continued)

Parameter	Mandatory/ Optional	Description
LICENSE	Mandatory	Variable to accept license terms. Specify value as `accept`.
SKLM_SEED	Mandatory	Secret passcode that is unique for a deployment, and must be stored securely. The value is a random string of 32 or 64 characters that you can generate using an external utility. **Note:** Ensure that the value of this parameter in the subsequent docker run commands is the same as that used in the first docker run command, when the same value of the **sklmAppVolume** parameter is used.
SKLMADMIN_USERNAME	Optional	User name of the IBM Security Guardium Key Lifecycle Manager administrator. You can specify only alphanumeric characters. Default value: `sklmadmin` **Note:** This parameter is ignored in the subsequent docker run commands when the same value of the **sklmAppVolume** parameter is used.
SKLMADMIN_PASSWORD	Mandatory	Password for the IBM Security Guardium Key Lifecycle Manager administrator user that is specified in the **SKLMADMIN_USERNAME** parameter. **Note:** This parameter is ignored in the subsequent docker run commands when the same value of the **sklmAppVolume** parameter is used.
LIBERTY_KEY_STORE_PASSWORD	Optional	Password for the IBM Security Guardium Key Lifecycle Manager keystore. Default value: `Ch@ngemypa55word` **Note:** Ensure that the value of this parameter in the subsequent docker run commands is the same as that used in the first docker run command, when the same value of the **sklmAppVolume** parameter is used.

Table 5. Parameters and their descriptions (continued)

Parameter	Mandatory/ Optional	Description
LIBERTY_KEY_STORE_PASSWORD_OLD	Optional	Old password for the IBM Security Guardium Key Lifecycle Manager keystore. If you want to change the keystore password, specify the current password as the value of this parameter, and the new password in the **LIBERTY_KEY_STORE_PASSWORD** parameter. Default value: `Ch@ngemypa55word` **Note:** Ensure that the value of this parameter in the subsequent docker run commands is the same as that used in the first docker run command, when the same value of the **sklmAppVolume** parameter is used.
LIBERTY_AES_ENCRYPTION_KEY	Optional	Key for encrypting the password for the IBM Security Guardium Key Lifecycle Manager administrator user with the AES algorithm. If you do not provide a value for this property, IBM Security Guardium Key Lifecycle Manager uses the value of the **SKLM_SEED** parameter for encryption.
FIPS	Optional	Flag to enable Federal Information Processing Standards (FIPS) publication 140-2 standard compliance in Guardium Key Lifecycle Manager. By default, this flag is set to `off`. When you set this property to on, Guardium Key Lifecycle Manager uses the `IBMJCEFIPS` provider instead of the `IBMJCE` provider for all cryptographic functions.
SP800_131A	Optional	Flag to enable Guardium Key Lifecycle Manager to communicate over secure sockets in compliance with the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-131A standard in strict mode. By default, this flag is set to `off`. Set the property to on to enable this standard.

Table 5. Parameters and their descriptions (continued)

Parameter	**Mandatory/ Optional**	**Description**
`SUITE_B`	Optional	Flag to enable US National Security Agency (NSA) Suite B standard compliance in Guardium Key Lifecycle Manager. By default, this flag is set to `off`. Set the property with one of the following values: • 128 • 192 When you set this property to on, Guardium Key Lifecycle Manager uses the `IBMJSSE2` provider instead of the `IBMJCE` provider for all cryptographic functions.
`SECURITY_LEVEL`	Optional	The cipher suite group to be used by the TLS handshake. You can specify one of the following values: • HIGH: For 128-bit ciphers and higher • MEDIUM: For 40-bit ciphers • WEAK: For all ciphers without encryption • CUSTOM: When the cipher suite group is customized This property is ignored if you set the **`ENABLED_CIPHERS`** property with a specific list of ciphers.

`ENABLED_CIPHERS`	Optional	A unique list of cipher suites. You can specify multiple cipher suites by separating the cipher suites with a space. For example:
		`enabledCiphers=”TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384`
		` TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256”`
		If you set this property, the **`securityLevel`** property is ignored.

`AUTH_TYPE`	Optional	Type of authentication used. By default, file-based authentication is available. You can configure Guardium Key Lifecycle Manager to use a supported LDAP providers for user authentication. You can also use both these options as the authentication type. Possible values: • FILE • LDAP • FILE,LDAP

Table 5. Parameters and their descriptions (continued)

Parameter	Mandatory/ Optional	Description
LDAP_GR_OBJ_CLASS	Optional	The object class that is defined for the group LDAP entity type in the LDAP server. For example, `GroupofUniquenames`.
LDAP_PR_OBJ_CLASS	Optional	The object class that is defined for the person LDAP entity type in the LDAP server. For example, `person`.
LDAP_TYPE	Optional	Type of LDAP server to be connected to. Possible values: **IBM Tivoli Directory Server** Configure the LDAP registry to use IBM Tivoli Directory Server. **Microsoft Active Directory** Configure the LDAP registry to use Microsoft Active Directory. **Note:** This parameter is mandatory if you have specified LDAP as a value in the **AUTH_TYPE** parameter.
LDAP_HOST	Optional	Host name or IP address of the LDAP server. **Note:** This parameter is mandatory if you have specified LDAP as a value in the **AUTH_TYPE** parameter.
LDAP_PORT	Optional	Port number of the LDAP server. **Note:** This parameter is mandatory if you have specified LDAP as a value in the **AUTH_TYPE** parameter.
LDAP_BASE_ENTRY	Optional	Base node in the LDAP directory. For example: `"DC=klm,DC=com"` where, DC is Domain Component. **Note:** This parameter is mandatory if you have specified LDAP as a value in the **AUTH_TYPE** parameter.
LDAP_BIND_DN	Optional	Distinguished name (DN) for the application server, which is used to bind to the directory service.
LDAP_BIND_PASSWD	Optional	Password for the DN that is specified in **LDAP_BIND_DN**. The value can be stored in clear text or encoded form. It is recommended that you encode the password. To do so, use the securityUtility tool with the encode option.
LDAP_TLS_ENABLED	Optional	Indicates whether a TLS connection must be made to the LDAP server.

Table 5. Parameters and their descriptions (continued)

Parameter	Mandatory/ Optional	Description
LDAP_SKLMADMIN_USERNAME	Optional	User name of the LDAP administrator. You can specify only alphanumeric characters. **Note:** You must provide a value for this parameter or for the **SKLMADMIN_USERNAME** parameter if you have specified LDAP as a value in the **AUTH_TYPE** parameter.
LDAP filters		
RECURSIVE_SEARCH	Optional	Indicates whether to perform a nested group search. Possible values: `true`, `false` (default) Specify `true` only if the LDAP server does not support recursive server-side searches.
USER_FILTER	Optional	An LDAP filter clause for searching the user registry for users.
GROUP_FILTER	Optional	An LDAP filter clause for searching the user registry for groups.
USER_ID_MAP	Optional	An LDAP filter that maps the name of a user to an LDAP entry.
GROUP_ID_MAP	Optional	An LDAP filter that maps the name of a group to an LDAP entry.
GROUP_MEMBER_ID_MAP	Optional	An LDAP filter that identifies user to group memberships.
HEALTH_AUTHORIZATION_TOKEN	Optional	Health token in your Kubernetes environment. **Note:** This parameter is applicable only when you are deploying on a Kubernetes cluster by using Helm charts.
Port numbers		
9443	Mandatory	Port number for the graphical user interface.
5696	Mandatory	KMIP TLS port
1441	Mandatory	IPP TLS port
3801	Mandatory	IPP TCP port
Persistent storage		
sklmAppVolume	Mandatory	Persistent storage to store the application server configuration and metadata information. Sample value: `/opt/ibm/wlp/usr/products`

Sample environment file (with PostgreSQL database)

```
LICENSE=accept
SKLMADMIN_USERNAME=sklm
KEY_STORE_PWD=keystorepwd
```

```
SKLMADMIN_PASSWORD=sklm
DB_HOST=172.x.x.x
DB_PORT=5432
DB_PASSWORD=databasepwd
DB_TYPE=postgres
DB_USER=sklmdb41
DBNAME=sklmdb41
SKLM_SEED=68d95f0081f1dbfc0b06de9b0916df1c
```

Sample environment file (with Db2 database)

```
LICENSE=accept
SKLMADMIN_USERNAME=sklm
KEY_STORE_PWD=keystorepwd
SKLMADMIN_PASSWORD=sklm
DB_HOST=172.x.x.x
DB_PORT=50000
SKLM_SEED=68d95f0081f1dbfc0b06de9b0916df1c
DB_PASSWORD=databasepwd
DB_TYPE=db2
DB_USER=sklmdb41
DBNAME=sklmdb41
```

Sample environment variables file when deploying IBM Security Guardium Key Lifecycle Manager on IBM zCX environment with Db2 on z/OS

```
DB_TYPE=zos_db
DB_NAME=sklmdb41
DB_USER=sklmdb41
DB_PASSWORD=xxxxx
DB_HOST=9.x.x.x
DB_PORT=446
LICENSE=accept
SKLM_SEED=68d95f0081f1dbfc0b06de9b0916df1c
SKLMADMIN_USERNAME=sklmadmin
SKLMADMIN_PASSWORD=adminpassword
```

Configuring CA-signed certificate based authentication

You can enhance secure communication between a client (for example, browser, REST client) and the IBM Security Guardium Key Lifecycle Manager server by using a certificate that is signed by a certificate authority (CA) for authentication in WebSphere® Application Server Liberty base. By default, a self-signed certificate is provided in the keystore.

About this task

The IBM Security Guardium Key Lifecycle Manager keystore is of Public Key Cryptography Standards #12 (PKCS12) keystore type. PKCS12 is an industry standard keystore type, which makes it compatible with other products.

The keystore, `key.p12` file, is created in the `resources/security` directory when the IBM Security Guardium Key Lifecycle Manager server starts. The default password for the keystore is `Ch@ngemypa55word`. You can specify a different password when you initiate the IBM Security Guardium Key Lifecycle Manager application container.

The default keystore location in the **`sklmAppVolume`** volume is ${*PRODUCTS_DIR*}/serverConfig/key.p12.

Procedure

1. List the existing entries in the keystore.

   ```
   keytool -list -v -keystore /path/key.p12 -storepass KEY_STORE_PWD -storetype PKCS12
   ```

   Sample output of the command:

   ```
   Keystore type: PKCS12
   Keystore provider: IBMJCE
   Your keystore contains 1 entry
   Alias name: default
   Creation date: Apr 29, 2020
   Entry type: keyEntry
   Certificate chain length: 1
   Certificate[1]:
   Owner: CN=localhost, OU=defaultServer, O=ibm, C=us
   Issuer: CN=localhost, OU=defaultServer, O=ibm, C=us
   Serial number: 1a2abd4
   Valid from: 4/29/20 6:57 AM until: 4/29/21 6:57 AM
   Certificate fingerprints:
            MD5:  F1:0C:C7:DF:5B:72:4C:F7:60:34:06:30:F0:C1:08:56
            SHA1: F3:12:4F:8B:FC:0E:84:8F:21:90:77:13:20:0E:21:DC:00:80:15:70
            SHA256:
    68:65:69:BA:E0:D5:BF:9C:D9:2E:DA:CD:DE:6C:52:8F:DF:48:61:FA:E0:34:9B:94:85:9B:4F:38:16:E6:CE:B9
            Signature algorithm name: SHA256withRSA
            Version: 3
   Extensions:
   #1: ObjectId: 2.5.29.14 Criticality=false
   SubjectKeyIdentifier [
   KeyIdentifier [
   0000: 59 a7 12 19 9d d6 73 6f  ac 06 e3 bf 33 cd c0 d3  Y.....so....3...
   0010: b5 5e 0f 90                                        ....
   ]
   ]
   #2: ObjectId: 2.5.29.17 Criticality=false
   SubjectAlternativeName [
   [DNSName: localhost]]
   ```

2. Generate a CA-signed certificate request.

   ```
   keytool -certreq -file path/liberty.csr -alias default -keyalg RSA -keypass KEY_STORE_PWD
   -storetype PKCS12 -keystore path/key.p12 -storepass KEY_STORE_PWD
   ```

   Note: If the keystore has entries, use the alias that is used in these existing entries.

3. After you receive the CA-signed certificate, add the public key of the certificate to the keystore.

   ```
   keytool -importcert -storetype PKCS12 -storepass KEY_STORE_PWD
   -keystore  path/key.p12 -file path/ca.crt
   ```

4. Add the CA-signed certificate to the keystore.

   ```
   keytool -importcert -alias default -storetype PKCS12 -storepass KEY_STORE_PWD
   -keystore  path/key.p12 -file OUTPUT_CA_SIGNED_FILE.crt
   ```

5. List the existing entries in the keystore and verify the newly added certificate.

   ```
   keytool -list -v -keystore /path/key.p12 -storepass KEY_STORE_PWD -storetype PKCS12
   ```

   Sample output of the command:

   ```
   Keystore type: PKCS12
   Keystore provider: IBMJCE
   Your keystore contains 2 entries
   Alias name: default
   Creation date: Apr 29, 2020
   Entry type: keyEntry
   Certificate chain length: 2
   Certificate[1]:
   Owner: CN=localhost, OU=defaultServer, O=ibm, C=us
   Issuer: EMAILADDRESS=abc@abc.com, CN=SKLM, OU=ILL, O=IBM, L=Pune, ST=Maharashtra, C=IN
   Serial number: beb07bbc3c4d7ba23fd0d17e9dc4d16215e4d27
   Valid from: 4/29/20 7:17 AM until: 5/29/20 7:17 AM
   Certificate fingerprints:
   ```

```
         MD5:  0A:C0:C7:7D:70:7E:0E:E2:CD:3A:B6:06:C6:35:8D:00
         SHA1: A1:E8:3F:8A:7A:78:AB:0C:5C:58:FA:1D:14:30:08:47:47:36:E5:36
         SHA256:
 51:08:A7:9A:E6:B7:D8:19:20:14:3C:47:CB:E9:2A:F1:42:A6:C0:5F:2D:AD:5C:65:CF:4F:76:5A:A4:18:3E:BE
         Signature algorithm name: SHA256withRSA
         Version: 1
Certificate[2]:
Owner: EMAILADDRESS=abc@abc.com, CN=SKLM, OU=ILL, O=IBM, L=Pune, ST=Maharashtra, C=IN
Issuer: EMAILADDRESS=abc@abc.com, CN=SKLM, OU=ILL, O=IBM, L=Pune, ST=Maharashtra, C=IN
Serial number: 1678019d645872109c3a78118050a6b23b2efe6a
Valid from: 4/29/20 7:16 AM until: 5/29/20 7:16 AM
Certificate fingerprints:
         MD5:  B6:1B:07:2E:AD:9B:0C:95:04:AE:E8:BA:4A:E5:ED:D9
         SHA1: F9:D0:D2:8C:44:01:EA:B7:81:4A:3F:10:7F:60:AF:4B:64:71:BC:08
         SHA256:
 50:8C:DF:5E:7B:B7:09:C9:2A:C2:81:3E:A4:0F:14:4E:07:1F:58:DC:E4:E0:70:19:C4:84:28:42:A5:E2:9C:2E
         Signature algorithm name: SHA256withRSA
         Version: 3
Extensions:
#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 12 71 bd c8 66 ac dd b4  9d 2d a1 86 ac ed 06 88  .q..f...........
0010: c9 4b bf bf                                        .K..
]
]
#2: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 12 71 bd c8 66 ac dd b4  9d 2d a1 86 ac ed 06 88  .q..f...........
0010: c9 4b bf bf                                        .K..
]
]
#3: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
CA:true
PathLen:2147483647
]
*******************************************
*******************************************
Alias name: mykey
Creation date: Apr 29, 2020
Entry type: trustedCertEntry
Owner: EMAILADDRESS=abc@abc.com, CN=SKLM, OU=ILL, O=IBM, L=Pune, ST=Maharashtra, C=IN
Issuer: EMAILADDRESS=abc@abc.com, CN=SKLM, OU=ILL, O=IBM, L=Pune, ST=Maharashtra, C=IN
Serial number: 1678019d645872109c3a78118050a6b23b2efe6a
Valid from: 4/29/20 7:16 AM until: 5/29/20 7:16 AM
Certificate fingerprints:
         MD5:  B6:1B:07:2E:AD:9B:0C:95:04:AE:E8:BA:4A:E5:ED:D9
         SHA1: F9:D0:D2:8C:44:01:EA:B7:81:4A:3F:10:7F:60:AF:4B:64:71:BC:08
         SHA256:
 50:8C:DF:5E:7B:B7:09:C9:2A:C2:81:3E:A4:0F:14:4E:07:1F:58:DC:E4:E0:70:19:C4:84:28:42:A5:E2:9C:2E
         Signature algorithm name: SHA256withRSA
         Version: 3
Extensions:
#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 12 71 bd c8 66 ac dd b4  9d 2d a1 86 ac ed 06 88  .q..f...........
0010: c9 4b bf bf                                        .K..
]
]
#2: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 12 71 bd c8 66 ac dd b4  9d 2d a1 86 ac ed 06 88  .q..f...........
0010: c9 4b bf bf                                        .K..
]
]
#3: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
CA:true
PathLen:2147483647
]
```

6. Enable TLS communication in Liberty and configure httpEndpoint to use a TLS configuration other than the default keystore.

   a) In the `server.xml` file, add the **`serverKeyAlias`** attribute.

   The **`serverKeyAlias`** attribute specifies the alias of the certificate in the keystore used as the server's key. This attribute is only needed if the keystore has more then one key entry.

   For example:

   ```
   <ssl id="defaultTLSConfig" keyStoreRef="defaultKeyStore"
   serverKeyAlias="default" sslProtocol="TLSv1.2"/>
   ```

   b) Configure the **`keystore`** element in the `server.xml` file by adding the attributes **`pollingRate`** and **`updateTrigger`**.

   pollingRate

   Rate at which the server checks for updates to a keystore file. Specify a positive integer followed by a unit of time, which can be hours (h), minutes (m), seconds (s), or milliseconds (ms).

   For example, specify 100 milliseconds as `100ms`; specify 1.5 seconds as `1s500ms`.

   updateTrigger

   Method that is used to trigger the server to reload a keystore file.

   Possible values:

   - disabled: Disables all update monitoring. Changes to the keystore file will not be applied while the server is running.
   - mbean: Server will only update the keystore when prompted by the FileNotificationMbean. The FileNotificationMbean is typically called by an external program such as an integrated development environment or a management application.
   - polled: Server will scan for keystore file changes at the polling interval and update if the keystore file has detectable changes.

   Specify `polled` to enable the server for checking the keystore file for changes.

   The keystore in the following example is configured to be monitored every 5 seconds for updates:

   ```
   <keyStore id="defaultKeyStore" location="${PRODUCTS_DIR}/serverConfig/key.p12"
   password="{xor}DBQTEgg6PR4M" pollingRate="5s" updateTrigger="polled"></keyStore>
   ```

   For more information, see Configuration attributes.

7. Launch the IBM Security Guardium Key Lifecycle Manager graphical user interface to confirm that no certificate error is displayed and verify that the CA-signed certificate is shown in the browser.

Configuring LDAP authentication

You can configure IBM Security Guardium Key Lifecycle Manager to use the supported Lightweight Directory Access Protocol (LDAP) providers for user authentication. User authorization is managed in IBM Security Guardium Key Lifecycle Manager.

About this task

IBM Security Guardium Key Lifecycle Manager uses the user information from the LDAP server. You can then authorize users to control their access to tasks and data in IBM Security Guardium Key Lifecycle Manager.

Note: While adding a user in IBM Security Guardium Key Lifecycle Manager, ensure that you use the same user name as it exists in the configured LDAP server.

In geographically dispersed environments, performance might be negatively impacted if the LDAP server and the IBM Security Guardium Key Lifecycle Manager console are not geographically close to each other.

Multiple LDAP server configurations are not supported.

To configure LDAP authentication, use the LDAP specific parameters while deploying the container. For more information, see "Parameters to deploy IBM Security Guardium Key Lifecycle Manager container using Docker" on page 13.

User management

Users, user roles, and user groups control who has access to the product, which tasks they can perform, and which data they can access.

As part of the initial configuration, use the User Management feature to configure and authorize users that require access to the product. The users must be defined in the authentication system that is configured with the product. You can then add the users in IBM Security Guardium Key Lifecycle Manager, and assign them the relevant roles. By assigning roles, you can control who has access to the product, which tasks they can perform, and which data they can access.

You can also create a user group and assign roles to it. All users that need to have these roles are then associated with the group.

If you are using a file-based authentication system, only the default administrator user; for example, SKLMAdmin, is available. If you have configured an LDAP-based authentication system, then all the users can be associated with specific roles and groups. A predefined set of roles and groups are available after you install the product.

You can use the graphical user interface or the REST interface to manage users, roles, and groups in the IBM Security Guardium Key Lifecycle Manager application container.

For information about the REST services, see "User management REST services" on page 27.

For instructions to manage users, roles, and groups from the graphical user interface, see the following topics:

Adding a user

Use the User Management page to add a user in IBM Security Guardium Key Lifecycle Manager. Depending on the functions that you want the user to perform, assign it the associated roles or groups.

Procedure

1. Access the User Management page.
   a) Log in to the graphical user interface by using your credentials.
   b) Click **User Management**.
      The Users page opens that displays a list of the users and their assigned roles and groups.
2. Click **Add**.
   The Add User window is displayed.
3. In the Basic Information page, specify the user name.

   Note: You must specify the same user name that is used in the file-based authentication system or LDAP server, depending on the authentication type that is configured.
4. Click **Add**.
   The Assign Roles tab is selected. You must assign at least a role or a group to the user to add the user.
5. To assign a role to the user, from the **Roles** list, select the roles to be assigned to the user, and click **Assign Roles**.
   For more information about a role, you can review its tooltip.
6. To assign a group to the user, complete these steps:
   a) Click the **Assign Groups** tab.
   b) From the **Groups** list, select the groups to be assigned to the user, and click **Assign Groups**.

7. Click **Close**.

   The user is added in IBM Security Guardium Key Lifecycle Manager.

Modifying a user

Use the User Management page to add, modify, or remove the assigned roles or groups to a user.

Procedure

1. Access the User Management page.
   a) Log in to the graphical user interface by using your credentials.
   b) Click **User Management**.
      The Users page opens that displays a list of the users and their assigned roles and groups.
2. To modify the user role assignment, complete these steps:
   a) Select the user to be modified, and click **Modify**.
      The Modify User window is displayed.
   b) Ensure that the **Assign Roles** tab is selected.
   c) From the **Roles** lists, select or remove the roles.
   d) Click **Assign Roles**.
3. To modify the user group assignment, complete these steps.
   a) Select the user to be modified, and click **Modify**.
      The Modify User window is displayed.
   b) Click the **Assign Groups** tab.
   c) To assign or remove a group, from the **Groups** lists, select or remove the groups.
   d) Click **Assign Groups**.

   If you do not want to change any assignment, click **Close**.

Deleting a user

Delete a user from Guardium Key Lifecycle Manager if it is no longer required. The user continues to exist in the file-based authentication system or LDAP server, whichever is configured with Guardium Key Lifecycle Manager.

About this task

You cannot delete the logged-in user.

Procedure

1. Access the User Management page.
   a) Log in to the graphical user interface by using your credentials.
   b) Click **User Management**.
      The Users page opens that displays a list of the users and their assigned roles and groups.
2. Select the user to be deleted, and click **Delete**.
3. Confirm the delete operation.

Managing roles

Use the User Management page to add and delete a user role in IBM Security Guardium Key Lifecycle Manager.

About this task

A user role defines the functions that a user can access. You can add only one role per device group. Some roles are available by default when you install the product. You cannot delete a default role.

Procedure

1. Access the User Management page.
   a) Log in to the graphical user interface by using your credentials.
   b) Click **User Management**.
      The Users page opens that displays a list of the users and their assigned roles and groups.
2. Click the **Roles** tab.
   The Roles page is displayed.
3. To add a role, complete these steps:
   a) Click **Add**.
   b) Select the device group for which you want to add a role.
   c) Click **Add Role**.
4. To delete a role, select the role to be deleted, and click **Delete**. Then confirm the delete operation.
   The role is deleted and disassociated from all users and groups to which it was assigned.

Managing groups

Use the User Management page to add, modify, and delete a user group in IBM Security Guardium Key Lifecycle Manager.

About this task

A group is a collection of users with a given set of permissions, which are defined in roles. All users in a user group inherit the permissions based on the roles that are assigned to them. Some groups are available by default when you install the product. You cannot delete a default group.

Procedure

1. Access the User Management page.
   a) Log in to the graphical user interface by using your credentials.
   b) Click **User Management**.
      The Users page opens that displays a list of the users and their assigned roles and groups.
2. Click the **Groups** tab.
   The Groups page is displayed. The table lists the user groups and their assigned roles.
3. To add a group, complete these steps:
   a) Click **Add**.
   b) In the Basic Information tab, specify a name for the group and a description.
   c) Click **Add**.
      The **Assign Roles** tab is selected.
   d) From the **Roles** list, select the roles that you want to assign to the group, and click **Assign Roles**.
   e) Click **Close**.
      The group is added, and is displayed in the table.
4. To modify a group, complete these steps:
   a) Select the group to be modified, and click **Modify**.
   b) To modify the basic information, edit the group name and the description, and click **Update**.
   c) To modify the assigned roles, click **Assign Roles** tab, and from the **Roles** list, select or remove the roles.
   d) Click **Assign Roles**.
   e) Click **Close**.
      The group is modified.

5. To delete a group, select the group to be deleted, and click **Delete**. Then confirm the delete operation.

   The group is deleted and disassociated from all users to which it was assigned.

Known issues and limitations of using containerized IBM Security Guardium Key Lifecycle Manager

This Beta version of the containerized IBM Security Guardium Key Lifecycle Manager application has the following issues, limitations, and requirements

- The following features are not supported:
  - CLI commands. Alternatively, use REST APIs. Swagger UI is now integrated with IBM Security Guardium Key Lifecycle Manager, and you can use it to call any REST API.
  - Multi-Master cluster
  - Database password change from the user interface
- To support the keys rollover feature when using the PostgreSQL database, modify the `postgres.conf` file that exists in the Persistent storage or volume. In the Resource usage section, replace **`max_prepared_transactions = 0`** by **`max_prepared_transactions = 100`** and then restart the PostgreSQL container.
- When deploying the product in a zCX environment, you might see the following error:

  ```
  AZIF0144E An unexpected error occurred in LPAR xxxxxx
  GLZM009I zCX instance xxxx stored failure data
  ```

  Solution: Apply the fix given here: https://www.ibm.com/support/pages/apar/OA59111
- In the About IBM Security Guardium Key Lifecycle Manager window, the value of the **Image Tag** field is incorrectly displayed as **`4.1.0.0-beta2`**. The correct value is **`4.1.0.0-beta3`**.

Troubleshooting

For information about the workarounds and resolutions, see Troubleshooting and support section*Troubleshooting and support* PDF..

Chapter 2. User Management REST Services

User management REST services

You can use the user management REST services or APIs to manage and work with IBM Security Guardium Key Lifecycle Manager users, user roles, and user groups. You must have administrative privileges to perform these operations.

Add Role REST Service

Use **Add Role REST Service** to add a user role in IBM Security Guardium Key Lifecycle Manager.

Operation
POST

URL
`https://<host>:<port >/SKLM/rest/v1/ckms/usermanagement/roles/{roleName}`

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	`application/json`
Accept	`application/json`
Authorization	`SKLMAuth userAuthId=<authIdValue>`
Accept-Language	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Path parameter

JSON object with the following specification.

JSON property name	Description
roleName	Specify a name for the user role that you want to add.

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success Response Body

JSON object with the following specification.

JSON property name	Description
`code`	Returns the code that is specified by the `status` property.
`status`	Returns the status to indicate if the operation was successful.

Error Response Body

JSON object with the following specification.

JSON property name	Description
`code`	Returns the application error code.
`message`	Returns a message that describes the error.

Example

Add a user role

```
POST
https://host:port/SKLM/rest/v1/ckms/usermanagement/roles/DS5000_Custom
```

Success response

```
{
  "code": "0",
  "status": "Succeeded"
}
```

Error response

```
{
  "code": "CTGKM5004E",
  "message": "CTGKM5004E Object with the same name DS5000_Custom already exists."
}
```

Add User Group REST Service

Use **Add User Group REST Service** to add a user group in IBM Security Guardium Key Lifecycle Manager.

Operation

POST

URL

`https://<host>:<port >/SKLM/rest/v1/ckms/usermanagement/groups/{groupName}`

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Path Parameter

JSON object with the following specification.

JSON property name	Description
groupName	Specify a name for the user group that you want to add.

Request Body

JSON object with the following specification.

JSON property name	Description
description	Specify a description for the user group that you want to add.

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success Response Body

JSON object with the following specification.

JSON property name	Description
`code`	Returns the code that is specified by the `status` property.
`status`	Returns the status to indicate if the operation was successful.

Error Response Body

JSON object with the following specification.

JSON property name	Description
`code`	Returns the application error code.
`message`	Returns a message that describes the error.

Example

Add a new user group

```
POST
https://host:port/SKLM/rest/v1/ckms/usermanagement/groups/DS5000_CUSTOM_GROUP
{"description":"This group is created for administering DS5000 device groups"}
```

Success response

```
{
  "code": "0",
  "status": "Succeeded"
}
```

Error response

```
{
  "code": "CTGKM6002E",
  "message": "CTGKM6002E Bad Request: Invalid user authentication ID or invalid request
 format."
}
```

Assign Roles to User REST Service

Use **Assign Roles to User REST Service** to assign multiple roles to a user.

Operation
`POST`

URL
`https://<host>:<port/SKLM/rest/v1/ckms/usermanagement/users/{userName}/roles`

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	`application/json`
Accept	`application/json`
Authorization	`SKLMAuth userAuthId=<authIdValue>`
Accept-Language	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Query Parameter

JSON object with the following specification.

JSON property name	Description
userName	Required. Specify the name of the user to which you want to assign the roles.

Request Body

JSON object with the following specification.

JSON property name	Description
roleNames	Required. Specify a JSON array of role names that you want to assign to the user.

Response

Response Headers

Header name	**Value and description**
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success Response Body

JSON object with the following specification.

JSON property name	**Description**
`code`	Returns the code that is specified by the **`status`** property.
`status`	Returns the status to indicate if the operation was successful.

Error Response Body

JSON object with the following specification.

JSON property name	**Description**
`code`	Returns the application error code.
`message`	Returns a message that describes the error.

Examples

Assign roles to a user

```
POST
https://host:port/SKLM/rest/v1/ckms/usermanagement/users/DS5000user/roles
{"roleNames":"DS5000_Custom","DS8000_Custom"}
```

Success response

```
{
  "code": "0",
  "status": "Succeeded"
}
```

Error response

```
{
  "code": "CTGKM3040E",
  "message": "CTGKM3040E Object with identifier DS5000user cannot be found."
}
```

Assign Roles to User Group REST Service

Use **`Assign Roles to User Group REST Service`** to assign multiple user roles to a user group.

Operation

`POST`

URL

`https://<host>:<port >/SKLM/rest/v1/ckms/usermanagement/groups/{groupName}/roles`

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters	
Parameter	**Description**
`host`	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
`port`	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers	
Header name	**Value**
`Content-Type`	`application/json`
`Accept`	`application/json`
`Authorization`	`SKLMAuth userAuthId=<authIdValue>`
`Accept-Language`	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Query Parameter JSON object with the following specification.	
JSON property name	**Description**
`groupName`	Required. Specify the name of the user group to which you want to assign the roles.

Request Body JSON object with the following specification.	
JSON property name	**Description**
`roleNames`	Required. Specify a JSON array of role names that you want to assign to the user.

Response

Response Headers

Header name	**Value and description**
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success Response Body

JSON object with the following specification.

JSON property name	**Description**
`code`	Returns the code that is specified by the **`status`** property.
`status`	Returns the status to indicate if the operation was successful.

Error Response Body

JSON object with the following specification.

JSON property name	**Description**
`code`	Returns the application error code.
`message`	Returns a message that describes the error.

Example

Assign roles to a user group

```
POST
https://host:port/SKLM/rest/v1/ckms/usermanagement/groups/DS5000_CUSTOM_GROUP/roles
{"roleName":"DS5000_Custom","DS8000_Custom"}
```

Success response

```
{
  "code": "0",
  "status": "Succeeded"
}
```

Error response

```
{
  "code": "CTGKM6002E",
  "message": "CTGKM6002E Bad Request: Invalid user authentication ID or invalid request
 format."
}
```

Assign User Groups to User REST Service

Use **`Assign User Groups to User REST Service`** to assign multiple user groups to a user.

Operation

`POST`

URL

`https://<host>:<port >/SKLM/rest/v1/ckms/usermanagement/users/{userName}/groups`

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters	
Parameter	**Description**
host	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers	
Header name	**Value**
Content-Type	`application/json`
Accept	`application/json`
Authorization	`SKLMAuth userAuthId=<authIdValue>`
Accept-Language	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Query Parameter JSON object with the following specification.	
JSON property name	**Description**
userName	Required. Specify the name of the user to which you want to assign the user groups.

Request Body JSON object with the following specification.	
JSON property name	**Description**
groupNames	Required. Specify a JSON array of user group names that you want to assign to the user.

Response

Response Headers

Header name	**Value and description**
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success Response Body

JSON object with the following specification.

JSON property name	**Description**
`code`	Returns the code that is specified by the `status` property.
`status`	Returns the status to indicate if the operation was successful.

Error Response Body

JSON object with the following specification.

JSON property name	**Description**
`code`	Returns the application error code.
`message`	Returns a message that describes the error.

Example

Add a user to a group

```
POST
https://host:port/SKLM/rest/v1/ckms/usermanagement/users/DS5000user/groups
{"groupName":"DS5000_CUSTOM_GROUP","DS8000_Custom"}
```

Success response

```
{
  "code": "0",
  "status": "Succeeded"
}
```

Error response

```
{
  "code": "CTGKM3040E",
  "message": "CTGKM3040E Object with identifier DS5000user cannot be found."
}
```

Delete Role REST Service

Use **`Delete Role REST Service`** to delete a user role.

Operation

DELETE

URL

`https://<host>:<port>/SKLM/rest/v1/ckms/usermanagement/roles/{roleName}`

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Path parameter

JSON object with the following specification.

JSON property name	Description
roleName	Specify the name of the user role that you want to delete.

Response

Response Headers

Header name	**Value and description**
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success Response Body

JSON object with the following specification.

JSON property name	**Description**
`code`	Returns the code that is specified by the **`status`** property.
`status`	Returns the status to indicate if the operation was successful.

Error Response Body

JSON object with the following specification.

JSON property name	**Description**
`code`	Returns the application error code.
`message`	Returns a message that describes the error.

Example

Delete a user role

```
DELETE
https://host:port/SKLM/rest/v1/ckms/usermanagement/roles/DS5000_new
```

Success response

```
{
  "code": "0",
  "status": "Succeeded"
}
```

Error response

```
{
  "code": "CTGKM6002E",
  "message": "CTGKM6002E Bad Request: Invalid user authentication ID or invalid request
 format."
}
```

Delete User REST Service

Use **`Delete User REST Service`** to delete a user and remove all its assigned user roles and user groups.

Operation

```
DELETE
```

URL

```
https://<host>:<port>/SKLM/rest/v1/ckms/usermanagement/users/{userName}
```

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters	
Parameter	**Description**
`host`	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
`port`	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers	
Header name	**Value**
`Content-Type`	`application/json`
`Accept`	`application/json`
`Authorization`	`SKLMAuth userAuthId=<authIdValue>`
`Accept-Language`	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Path Parameter JSON object with the following specification.	
JSON property name	**Description**
`userName`	Specify the user name of the user that you want to delete.

Response

Response Headers	
Header name	**Value and description**
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success Response Body JSON object with the following specification.	
JSON property name	**Description**
`code`	Returns the code that is specified by the **`status`** property.
`status`	Returns the status to indicate if the operation was successful.

Error Response Body JSON object with the following specification.	
JSON property name	**Description**
`code`	Returns the application error code.
`message`	Returns a message that describes the error.

Example

Delete a user

```
DELETE
https://host:port/SKLM/rest/v1/ckms/usermanagement/users/LTOAdmin
```

Success response

```
{
  "code": "0",
  "status": "Succeeded"
}
```

Error response

```
{
  "code": "CTGKM6002E",
  "message": "CTGKM6002E Bad Request: Invalid user authentication ID or invalid request
 format."
}
```

Delete User Group REST Service

Use **Delete User Group REST Service** to delete a user group. The assigned users and user roles are not deleted but are disassociated from the group.

Operation

`DELETE`

URL

`https://<host>:<port>/SKLM/rest/v1/ckms/usermanagement/groups/{groupName}`

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters	
Parameter	**Description**
host	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers	
Header name	**Value**
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Path parameter JSON object with the following specification.	
JSON property name	**Description**
groupName	Specify the name of the user group that you want to delete.

Response

Response Headers

Header name	**Value and description**
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success Response Body

JSON object with the following specification.

JSON property name	**Description**
`code`	Returns the code that is specified by the **`status`** property.
`status`	Returns the status to indicate if the operation was successful.

Error Response Body

JSON object with the following specification.

JSON property name	**Description**
`code`	Returns the application error code.
`message`	Returns a message that describes the error.

Example

Delete a user group

```
DELETE
https://<host>:<port>/SKLM/rest/v1/ckms/usermanagement/groups/LTO_NEW
```

Success response

```
{
  "code": "0",
  "status": "Succeeded"
}
```

Error response

```
{
  "code": "CTGKM6002E",
  "message": "CTGKM6002E Bad Request: Invalid user authentication ID or invalid request
 format."
}
```

Get Users REST Service

Use **`Get Users REST Service`** to retrieve a list of all the users in IBM Security Guardium Key Lifecycle Manager.

Operation
`GET`

URL
`https://<host>:<port>/SKLM/rest/v1/ckms/usermanagement/users`

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters

Parameter	Description
`host`	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
`port`	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
`Content-Type`	`application/json`
`Accept`	`application/json`
`Authorization`	`SKLMAuth userAuthId=<authIdValue>`
`Accept-Language`	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.

Response Headers (continued)

Header name	Value and description
Content-Type	application/json
Content-Language	Locale for the response message.

Success Response Body

JSON object with the following specification.

JSON property name	Description
code	Returns 0 when the request is successful. Otherwise, returns 1.
user	Returns a comma-separated list of all the users.

Error Response Body

JSON object with the following specification.

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Retrieve a list of all the users

```
GET
https://host:port/SKLM/rest/v1/ckms/usermanagement/users
```

Success response

```
[
  {
    "groups": [
      "klmGUICLIAccessGroup",
      "klmSecurityOfficerGroup"
    ],
    "isDefault": true,
    "name": "sklmadmin",
    "roles": [
      "klmSecurityOfficer",
      "suppressmonitor"
    ]
  }
]
```

Error response

```
{
  "code": "CTGKM6002E",
  "message": "CTGKM6002E Bad Request: Invalid user authentication ID or invalid request
 format."
}
```

Get User Group Details REST Service

Use **Get User Group Details REST Service** to retrieve details for a user group, such as assigned roles and assigned users.

Operation
GET

URL

`https://<host>:<port >/SKLM/rest/v1/ckms/usermanagement/groups/{groupName}`

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	`application/json`
Accept	`application/json`
Authorization	`SKLMAuth userAuthId=<authIdValue>`
Accept-Language	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Response

Response Headers

Header name	Value and description
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success Response Body JSON object with the following specification.	
JSON property name	**Description**
code	Returns 0 when the request is successful. Otherwise, returns 1.
name	Returns the name of the user group.
description	Returns the description of the user group.
isDefault	Indicates whether the user group and its associations are provided by default during product installation (return value = `true`) or added later by a user (return value = `false`). You cannot modify or delete a default user group.
roles	Returns details of the roles that are assigned to the user group
users	Returns details of the users that are assigned to the user group.

Error Response Body JSON object with the following specification.	
JSON property name	**Description**
code	Returns the application error code.
message	Returns a message that describes the error.

Example

Retrieve details of a user group

```
GET https://host:port/SKLM/rest/v1/ckms/usermanagement/groups/DS5000_CUSTOM_GROUP
```

Success response

```
{
  "description": "",
  "isDefault": true,
  "name": "klmSecurityOfficerGroup",
  "roles": [
    {
      "description": "Perform all IBM Security Guardium Key Lifecycle Manager
 administrative operations and has Super user access rights.",
      "isDefault": true,
      "name": "klmSecurityOfficer"
    },
    {
      "description": "Read or change properties, or act on certificates.",
      "isDefault": true,
      "name": "klmConfigure"
    },
    {
      "description": "Hide other tasks on the WebSphere Integrated Solutions Console.",
      "isDefault": true,
      "name": "suppressmonitor"
    }
  ],
  "users": [
    "sklmadmin"
  ]
}
```

Error response

```
{
  "code": "CTGKM6002E",
  "message": "CTGKM6002E Bad Request: Invalid user authentication ID or invalid request
 format."
}
```

Get User Groups REST Service

Use **Get User Groups REST Service** to retrieve all the user groups in IBM Security Guardium Key Lifecycle Manager.

Operation

GET

URL

https://*<host>*:*<port>*/SKLM/rest/v1/ckms/usermanagement/groups

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	application/json
Accept	application/json
Authorization	SKLMAuth userAuthId=<authIdValue>
Accept-Language	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Response

Response Headers	
Header name	**Value and description**
Status Code	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
Content-Type	`application/json`
Content-Language	Locale for the response message.

Success Response Body JSON object with the following specification.	
JSON property name	**Description**
code	Returns `0` when the request is successful. Otherwise, returns `1`.
For every user group, the following details are returned:	
name	Returns the name of the user group.
description	Returns the description of the user group.
isDefault	Indicates whether the user group and its associations are provided by default during product installation (return value = `true`) or added later by a user (return value = `false`). You cannot modify or delete a default user group.
roles	Returns details of the roles that are assigned to the user group.
users	Returns details of the users that are assigned to the user group.

Error Response Body JSON object with the following specification.	
JSON property name	**Description**
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Retrieve a list of all user groups

```
GET
https://host:port/SKLM/rest/v1/ckms/usermanagement/groups
```

Success response

```
[
  {
    "description": "",
    "isDefault": true,
    "name": "LTOAdmin",
    "roles": [
      "klmBackup",
      "klmConfigure",
      "klmAudit",
      "klmView",
      "klmCreate",
      "klmModify",
      "klmDelete",
      "klmGet",
      "LTO",
      "suppressmonitor"
    ]
  },
  {
    "description": "",
    "isDefault": true,
    "name": "LTOAuditor",
    "roles": [
      "klmAudit",
      "klmView",
      "LTO",
      "suppressmonitor"
    ]
  },
  {
    "description": "",
    "isDefault": true,
    "name": "LTOOperator",
    "roles": [
      "klmBackup",
      "klmView",
      "klmCreate",
      "klmModify",
      "LTO",
      "suppressmonitor"
    ]
  },
  {
    "description": "sdfdsfdsfsfd",
    "isDefault": false,
    "name": "group",
    "roles": [
      "DS8000"
    ]
  },
  {
    "description": "",
    "isDefault": true,
    "name": "klmBackupRestoreGroup",
    "roles": [
      "klmBackup",
      "klmRestore",
      "suppressmonitor"
    ]
  },
  {
    "description": "",
    "isDefault": true,
    "name": "klmGUICLIAccessGroup",
    "roles": [
      "suppressmonitor"
    ]
  },
  {
    "description": "",
    "isDefault": true,
    "name": "klmSecurityOfficerGroup",
    "roles": [
      "klmSecurityOfficer",
      "klmConfigure",
      "suppressmonitor"
    ]
  }
]
```

Error response

```
{
  "code": "CTGKM6002E",
  "message": "CTGKM6002E Bad Request: Invalid user authentication ID or invalid request
 format."
}
```

Get Role Details REST Service

Use **`Get Role Details`** to retrieve details for a user role, such as role description, assigned users, and assigned groups.

Operation
`GET`

URL
`https://<host>:<port>/SKLM/rest/v1/ckms/usermanagement/roles/{roleName}`

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters

Parameter	Description
host	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
port	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
Content-Type	`application/json`
Accept	`application/json`
Authorization	`SKLMAuth userAuthId=<authIdValue>`
Accept-Language	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Response

Response Headers	
Header name	**Value and description**
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Path parameter JSON object with the following specification.	
JSON property name	**Description**
`roleName`	Specify the user role for which you want to retrieve the details.

Success Response Body JSON object with the following specification.	
JSON property name	**Description**
`code`	Returns `0` when the request is successful. Otherwise, returns `1`.
`name`	Returns the name of the user role.
`description`	Returns the description of the user role.
`isDefault`	Indicates whether the user role and its associations are provided by default during product installation (return value = `true`) or added later by a user (return value = `false`). You cannot modify or delete a default user role.
`groups`	Returns details of the associated user groups.
`users`	Returns details of the associated users.

Error Response Body JSON object with the following specification.	
JSON property name	**Description**
`code`	Returns the application error code.
`message`	Returns a message that describes the error.

Examples

```
GET
https://host:port/SKLM/rest/v1/ckms/usermanagement/roles/LTO_CUSTOM
```

Success response

```
{
  "description": "Perform all IBM Security Guardium Key Lifecycle Manager administrative
 operations and has Super user access rights.",
  "groups": [
    {
      "description": "",
      "isDefault": false,
      "name": "klmSecurityOfficerGroup"
    }
  ],
  "isDefault": true,
  "name": "klmSecurityOfficer",
  "users": [
    {
      "description": "",
      "isDefault": true,
      "name": "sklmadmin"
    }
  ]
}
```

Error response

```
{
  "code": "CTGKM6002E",
  "message": "CTGKM6002E Bad Request: Invalid user authentication ID or invalid request
 format."
}
```

Get Roles REST Service

Use **`Get Roles REST Service`** to retrieve a list of all the user roles in IBM Security Guardium Key Lifecycle Manager.

Operation
GET

URL
`https://<host>:<port>/SKLM/rest/v1/ckms/usermanagement/roles`

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters

Parameter	Description
`host`	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
`port`	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers	
Header name	**Value**
`Content-Type`	`application/json`
`Accept`	`application/json`
`Authorization`	`SKLMAuth userAuthId=<authIdValue>`
`Accept-Language`	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Response

Response Headers	
Header name	**Value and description**
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success Response Body JSON object with the following specification.	
JSON property name	**Description**
`code`	Returns `0` when the request is successful. Otherwise, returns `1`.
For every role, the following details are returned:	
`name`	Returns the name of the user role.
`description`	Returns the description of the user role.
`isDefault`	Indicates whether the user role and its associations are provided by default during product installation (return value = `true`) or added later by a user (return value = `false`). You cannot modify or delete a default user role.

Error Response Body

JSON object with the following specification.

JSON property name	**Description**
code	Returns the application error code.
message	Returns a message that describes the error.

Examples

Retrieve a list of all the user roles

```
GET
https://host:port/SKLM/rest/v1/ckms/usermanagement/roles
```

Success response

```
[
  {
    "name": "administrator",
    "description": "",
    "isDefault": true
  },
  {
    "name": "operator",
    "description": "",
    "isDefault": true
  },
  {
    "name": "configurator",
    "description": "",
    "isDefault": true
  },
  {
    "name": "monitor",
    "description": "",
    "isDefault": true
  },
  {
    "name": "deployer",
    "description": "",
    "isDefault": true
  },
  {
    "name": "adminsecuritymanager",
    "description": "",
    "isDefault": true
  },
  {
    "name": "nobody",
    "description": "",
    "isDefault": true
  },
  {
    "name": "iscadmins",
    "description": "",
    "isDefault": true
  },
  {
    "name": "klmSecurityOfficer",
    "description": "",
    "isDefault": true
  },
  {
    "name": "klmBackup",
    "description": "",
    "isDefault": true
  },
  {
    "name": "klmConfigure",
    "description": "",
    "isDefault": true
  },
  {
    "name": "klmRestore",
    "description": "",
    "isDefault": true
  },
  {
    "name": "klmAudit",
    "description": "",
    "isDefault": true
  },
  {
    "name": "klmView",
    "description": "",
    "isDefault": true
  },
  {
    "name": "klmCreate",
    "description": "",
    "isDefault": true
  },
  {
    "name": "klmModify",
    "description": "",
    "isDefault": true
  },
  {
    "name": "klmDelete",
    "description": "",
    "isDefault": true
  },
  {
    "name": "klmGet",
    "description": "",
    "isDefault": true
  },
  {
    "name": "klmAdminDeviceGroup",
    "description": "",
    "isDefault": true
  },
  {
    "name": "LTO",
    "description": "",
    "isDefault": true
  },
  {
    "name": "TS3592",
    "description": "",
    "isDefault": true
  },
  {
    "name": "DS5000",
    "description": "",
    "isDefault": true
  },
  {
    "name": "DS8000",
    "description": "",
    "isDefault": true
  },
  {
    "name": "GENERIC",
    "description": "",
    "isDefault": true
  },
  {
    "name": "BRCD_ENCRYPTOR",
    "description": "",
    "isDefault": true
  },
  {
    "name": "ONESECURE",
    "description": "",
    "isDefault": true
  },
  {
    "name": "suppressmonitor",
    "description": "",
    "isDefault": true
  },
  {
    "name": "ETERNUS_DX",
    "description": "",
    "isDefault": true
  },
  {
    "name": "IBM_SYSTEM_X_SED",
    "description": "",
    "isDefault": true
  },
  {
    "name": "XIV",
    "description": "",
    "isDefault": true
  },
  {
    "name": "GPFS",
    "description": "",
    "isDefault": true
  },
  {
    "name": "PEER_TO_PEER",
    "description": "",
    "isDefault": true
  },
  {
    "name": "DS8000_TCT",
    "description": "",
    "isDefault": true
  },
  {
    "name": "klmClientUser",
    "description": "",
    "isDefault": true
  },
  {
    "name": "klmFileTransfer",
    "description": "",
    "isDefault": true
  },
  {
    "name": "LTO_CUSTOM",
    "description": "This is a new role created for the custom LTO group",
    "isDefault": false
  },
  {
    "name": "DS5000_Custom",
    "description": "Role for DS500_custom device group",
    "isDefault": false
  }
]
```

Error response

```
{
  "code": "CTGKM6002E",
  "message": "CTGKM6002E Bad Request: Invalid user authentication ID or invalid request
 format."
}
```

Get User Details REST Service

Use **`Get User Details REST Service`** to retrieve details of a user such as the assigned roles and groups.

Operation

GET

URL

https://*<host>*:*<port>*/SKLM/rest/v1/ckms/usermanagement/users/{userName}

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters

Parameter	Description
`host`	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.
`port`	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers

Header name	Value
`Content-Type`	`application/json`
`Accept`	`application/json`
`Authorization`	`SKLMAuth userAuthId=<authIdValue>`
`Accept-Language`	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Path parameter

JSON object with the following specification.

JSON property name	Description
`userName`	Specify the user name for which you want to retrieve the details.

Response

Response Headers

Header name	Value and description
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success Response Body

JSON object with the following specification.

JSON property name	Description
code	Returns 0 when the request is successful. Otherwise, returns 1.
name	Returns the name of the user.
description	Returns the description of the user.
isDefault	Indicates whether the user and its associations are provided by default during product installation (return value = `true`) or added later by a user (return value = `false`). You cannot modify or delete a default user.
roles	Returns details of the user roles that are assigned to the user.
groups	Returns details of the user groups to which the user belongs.

Error Response Body

JSON object with the following specification.

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Example

Get details of a user

```
GET
https://host:port/SKLM/rest/v1/ckms/usermanagement/users/LTOuser
```

Success response

```
{
  "groups": [
    {
      "description": "",
      "name": "klmGUICLIAccessGroup",
      "roles": [
        "suppressmonitor"
      ]
    },
    {
      "description": "",
      "name": "klmSecurityOfficerGroup",
      "roles": [
        "klmSecurityOfficer",
        "suppressmonitor",
        "klmConfigure"
      ]
    }
  ],
  "isDefault": true,
  "name": "sklmadmin",
  "roles": [
    {
      "description": "Perform all IBM Security Guardium Key Lifecycle Manager
 administrative operations and has Super user access rights.",
      "groupName": "klmSecurityOfficerGroup",
      "isDefault": true,
      "name": "klmSecurityOfficer",
      "roleAssignment": "group"
    },
    {
      "description": "Hide other tasks on the WebSphere Integrated Solutions Console.",
      "groupName": "klmSecurityOfficerGroup",
      "isDefault": true,
      "name": "suppressmonitor",
      "roleAssignment": "group"
    },
    {
      "description": "Hide other tasks on the WebSphere Integrated Solutions Console.",
      "groupName": "klmGUICLIAccessGroup",
      "isDefault": true,
      "name": "suppressmonitor",
      "roleAssignment": "group"
    },
    {
      "description": "Read or change properties, or act on certificates.",
      "groupName": "klmSecurityOfficerGroup",
      "isDefault": true,
      "name": "klmConfigure",
      "roleAssignment": "group"
    },
    {
      "description": "Perform all IBM Security Guardium Key Lifecycle Manager
 administrative operations and has Super user access rights.",
      "groupName": "",
      "isDefault": true,
      "name": "klmSecurityOfficer",
      "roleAssignment": "user"
    },
    {
      "description": "Hide other tasks on the WebSphere Integrated Solutions Console.",
      "groupName": "",
      "isDefault": true,
      "name": "suppressmonitor",
      "roleAssignment": "user"
    }
  ]
}
```

Error response

```
{
  "code": "CTGKM6002E",
  "message": "CTGKM6002E Bad Request: Invalid user authentication ID or invalid request
 format."
}
```

Update User Group REST Service

Use **`Update User Group REST Service`** to modify an existing user group. You can modify the name and description of a user group.

Operation

PUT

URL

`https://<host>:<port>/SKLM/rest/v1/ckms/usermanagement/groups/{oldGroupName}`

By default, Guardium Key Lifecycle Manager server listens to non-secure port 9080 (HTTP) and secure port 9443 (HTTPS) for communication. During IBM Security Guardium Key Lifecycle Manager installation, you can modify these default ports.

Note: The non-secure port 9080 is not applicable when IBM Security Guardium Key Lifecycle Manager is deployed in a containerized environment.

Request

Request Parameters	
Parameter	**Description**
`host`	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.

Request Parameters (continued)	
Parameter	**Description**
`port`	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

Request Headers	
Header name	**Value**
`Content-Type`	`application/json`
`Accept`	`application/json`
`Authorization`	`SKLMAuth userAuthId=<authIdValue>`
`Accept-Language`	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

Path parameter JSON object with the following specification.	
JSON property name	**Description**
`oldGroupName`	Specify the name of the user group that you want to modify.

Request Body JSON object with the following specification.	
JSON property name	**Description**
`newGroupName`	Specify the new name for the user group.
`description`	Specify a description for the user group that you want to modify.

Response

Response Headers	
Header name	**Value and description**
`Status Code`	**200 OK** The request was successful. The response body contains the requested representation. **400 Bad Request** The authentication information was not provided in the correct format. **401 Unauthorized** The authentication credentials were missing or incorrect. **404 Not Found Error** The processing of the request fails. **500 Internal Server Error** The processing of the request fails because of an unexpected condition on the server.
`Content-Type`	`application/json`
`Content-Language`	Locale for the response message.

Success Response Body

JSON object with the following specification.

JSON property name	Description
code	Returns the code that is specified by the **status** property.
status	Returns the status to indicate if the operation was successful.

Error Response Body

JSON object with the following specification.

JSON property name	Description
code	Returns the application error code.
message	Returns a message that describes the error.

Example

Modify a user group

```
PUT https://host:port/SKLM/rest/v1/ckms/usermanagement/groups/LTO_CUSTOM_GROUP
{"newGroupName":"LTO_NEW","description":"changing group name"}
```

Success response

```
{
  "code": "0",
  "status": "Succeeded"
}
```

Error response

```
{
  "code": "CTGKM5004E",
  "message": "CTGKM5004E Object with the same name DS5000_Custom already exists."
}
```

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law :

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785

US

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at http://www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Product Number:

IBM Confidential