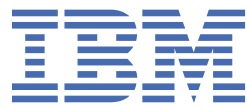


Containerized IBM Security Guardium Key  
Lifecycle Manager  
Version 4.1

*Documentation  
(BETA 4)*



IBM Confidential

**Note**

Before you use this information and the product it supports, read the information in [“Notices”](#) on page [111](#).

**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.

© **Copyright International Business Machines Corporation 2008, 2020.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Chapter 1. Installing on a container platform.....</b>	<b>1</b>
Overview of the containerized IBM Security Guardium Key Lifecycle Manager.....	1
Installing on OpenShift.....	2
Installing on a Kubernetes cluster.....	4
Installing with PostgreSQL using Docker.....	6
Installing with Db2 using Docker.....	7
Installing with HSM using Docker.....	9
Installing on IBM zCX environment with Db2 for z/OS.....	13
Parameters to deploy the container using Docker.....	15
Configuring CA-signed certificate based authentication.....	17
Configuring user authentication.....	21
User management.....	21
Adding a user.....	22
Modifying a user.....	22
Deleting a user.....	23
Managing roles.....	23
Managing groups.....	24
Known issues and limitations of using the containerized application.....	24
<b>Chapter 2. User Management REST Services.....</b>	<b>27</b>
User management REST services.....	27
Add Role REST Service.....	27
Add User Group REST Service.....	29
Assign Roles to User REST Service.....	31
Assign Roles to User Group REST Service.....	33
Assign User Groups to User REST Service.....	35
Delete Role REST Service.....	37
Delete User REST Service.....	39
Delete User Group REST Service.....	41
Get Users REST Service.....	43
Get User Group Details REST Service.....	44
Get User Groups REST Service.....	47
Get Role Details REST Service.....	50
Get Roles REST Service.....	52
Get User Details REST Service.....	56
Update User Group REST Service.....	59
<b>Chapter 3. User Authentication and Security Standards Configuration REST Services.....</b>	<b>63</b>
User authentication and security standards configuration REST services.....	63
Get Authentication Configuration Details REST Service.....	63
Get Security Configuration Details REST Service.....	66
Update File-Based Authentication Configuration REST Service.....	68
Update LDAP-Based Authentication Configuration REST Service.....	71
Update Security Configurations REST Service.....	74
<b>Chapter 4. Client and Client Group Management REST Services.....</b>	<b>79</b>
Client and client group management REST services.....	79
Assign Client Certificate to a Client REST Service.....	79
Assign Users to a Client REST Service.....	82

Create Client REST Service..... 84  
Delete Client REST Service.....86  
Get Client Details REST Service..... 88  
List All Clients REST Service..... 90  
Remove Users from a Client REST Service.....93  
Update Client Name REST Service..... 95  
Create Client Group REST Service..... 97  
Assign Clients to a Client Group REST Service..... 99  
Delete Client Group REST Service..... 100  
List All Client Group Details REST Service..... 102  
List Clients Not in Any Client Group REST Service.....104  
Remove Clients from Group REST Service..... 106  
Update Client Group Name REST Service..... 108

**Notices..... 111**  
Terms and conditions for product documentation..... 112  
Trademarks..... 113

---

## Chapter 1. Installing on a container platform

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

The containerized installation 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 install the containers on one of the following platforms:

### Openshift

You can install IBM Security Guardium Key Lifecycle Manager on OpenShift®.

For more information, see [“Installing on OpenShift” on page 2.](#)

### 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 [“Installing on a Kubernetes cluster” on page 4.](#)

### Docker

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

For more information, see the topics:

- [“Installing with PostgreSQL using Docker” on page 6](#)
- [“Installing with Db2 using Docker” on page 7](#)
- [“Installing with HSM using Docker” on page 9](#)

### 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 [“Installing on IBM zCX environment with Db2 for z/OS” on page 13.](#)

---

## 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 installation for IBM Security Guardium Key Lifecycle Manager requires a database container to be installed prior to installing 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 installed 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).

## Installing on OpenShift

---

Use Helm charts to install IBM Security Guardium Key Lifecycle Manager containers on OpenShift.

### Before you begin

#### Prepare the OpenShift Container Platform

- Obtain OpenShift Container Platform Version 4.2 or later.
- Set up an OCP cluster and ensure that it is up and running.

You can access the OpenShift documentation here: [https://access.redhat.com/documentation/en-us/openshift\\_container\\_platform/4.3/](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.3/)

#### Obtain the OpenShift Command line (CLI) tools

Obtain the following command line tools:

- oc
- helm

For instructions, see [https://docs.openshift.com/container-platform/4.3/cli\\_reference/openshift\\_cli/getting-started-cli.html](https://docs.openshift.com/container-platform/4.3/cli_reference/openshift_cli/getting-started-cli.html).

#### Install the database

You can use IBM Db2U Standard Edition Version 11.5 or PostgreSQL Version 10.

- To install IBM Db2U, see <https://github.com/IBM/Db2/tree/oc-4x/deployment>
- To install PostgreSQL, see <https://www.openshift.com/blog/deploy-postgresql-in-openshift-backed-by-openshift-container-storage>

**Tip:** To verify the database creation status, in the left navigation pane of the OpenShift dashboard, click **Workloads > Pods**.

#### Create Persistent Volume

Create a volume of minimum 15 GB. For more information, see [Creating persistent volumes](#).

#### Obtain License

To obtain the license activation file for IBM Security Guardium Key Lifecycle Manager, send us an email at: [ibmsklm@in.ibm.com](mailto:ibmsklm@in.ibm.com).

### Procedure

Complete the following steps on the system on which you installed the common tools:

1. Obtain the login token.
  - a) Log in to the OpenShift Container Platform.
  - b) Click **Display Token** link.
  - c) Copy the Login command that is displayed under **Log in with this token**.
  - d) Use the copied command to connect to the OCP server by using the command line tool (oc).
2. Download the [openshift-helm.zip](#) file that contains the sample Helm charts for installing IBM Security Guardium Key Lifecycle Manager.
3. Extract the `openshift-helm.zip` file.
4. In the directory where you extracted the files, navigate to **openshift-helm > sklmapp** directory.

5. Create the Security Context Constraint (SCC).

- a) Launch the command window on client system.
- b) Navigate to the location where the oc client is present and log in to OpenShift with the command given on the auth token link.

For example:

```
oc login --token=6rNowb9qoawao2VcSP0hzdd0wX0Y-GVbgxgiNMySOBw --server=servername:6443
```

- c) Run the following command: `oc apply -f liberty_scc.yaml`

The following output is displayed:

```
securitycontextconstraints.security.openshift.io/ibm-websphere-scc configured
```

6. Create the Websphere service account and bind the ibm-websphere-scc to the project.

```
#oc create serviceaccount websphere -n project name
#oc create serviceaccount websphere -n sklmb2
Output is like :
serviceaccount/websphere created

#oc adm policy add-scc-to-user ibm-websphere-scc -z websphere -n <project name>
#oc adm policy add-scc-to-user ibm-websphere-scc -z websphere -n sklmb2
```

The following output is displayed:

```
securitycontextconstraints.security.openshift.io/ibm-websphere-scc added to:
["system:serviceaccount:aaa:websphere"]
```

7. Open the `values.yaml` file and modify the parameter values in the file as per your requirement.

8. Navigate to `sklmapp` directory and run the following command:

```
helm install sklmapp-ocp sklmapp
```

9. Verify the installation.

- a) Log in to the OpenShift Container Platform.
- b) In the left pane, expand **Workloads > Pods**.

A new pod for the application is created with the status as **Running**.

10. To access the application, create a route.
  - a) In the left pane, expand **Networking** > **Routes**, and click **Create Route**.
  - b) Specify values for the properties on the page.

Ensure that you specify the following property settings:

**Name**  
sklmapp-route

**Service**  
sklmapp

**Target port**  
9443 -> 9443(TCP)

**Security**  
Select the **Secure route** check box.

**TLS Termination**  
Passthrough

**Insecure Traffic**  
Redirect
  - c) Click **Create**.

The application URL is generated and displayed in the **Location** field.
  - d) Copy the application URL.
11. Launch the IBM Security Guardium Key Lifecycle Manager graphical user interface by using the copied application URL.
12. 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.
13. Click **Activate License**.
14. Upload the IBM Security Guardium Key Lifecycle Manager license activation file and activate the license.
15. Click **Activate License**.
16. Upload the IBM Security Guardium Key Lifecycle Manager license activation file and activate the license.
17. Click **Login**.
18. 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 Manager](#) See [Administering](#).

## Installing on a Kubernetes cluster

---

Use Helm charts to install IBM Security Guardium Key Lifecycle Manager containers on a Kubernetes cluster. This installation 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](mailto: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 installing 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 Service](#)[Health 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 Manager](#) See [Administering](#).

## Installing with PostgreSQL using Docker

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

### Before you begin

- Ensure that the host system meets these 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](mailto: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=sklmbdb41 -e POSTGRES_DB=sklmbdb41 -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 15.

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=sklmb41 -e DB_NAME=sklmb41 -v sklAppVolume:/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.

- Monitor the progress by using the `docker logs` command.

```
docker logs -f skl
```

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

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

- 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.

- 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.
- Click **Activate License**.
- Upload the IBM Security Guardium Key Lifecycle Manager license activation file and activate the license.
- Click **Login**.
- Log in to the IBM Security Guardium Key Lifecycle Manager graphical user interface with the Administrator user credentials (`skladmin`).

### 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 Manager](#) See [Administering](#).

## Installing with Db2 using Docker

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

### Before you begin

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

<i>Table 2. Minimum system requirements</i>	
Resource	Requirement
CPU	4 Cores
Memory	8 GB
Disk space	100 GB
Operating system and supported architectures	Linux <ul style="list-style-type: none"> <li>– x86_64</li> <li>– s390x</li> </ul>

- 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](mailto: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 sklmbd -f Dockerfile.sample --no-cache .
```

4. Run the customized Db2 container. For example:

```
docker run --name sklmbd --restart=always --detach --ipc="" --cap-add=IPC_OWNER -p 50000:50000 -e LICENSE=accept -e DB2INSTANCE=sklmbd41 -e DB2INST1_PASSWORD=SKLMdb2 -e DBNAME=sklmbd41 -v sklmbd2Volume:/database sklmbd
```

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 15.

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 sklmap -itd -h skl.com -p 9443:9443 -p 3801:3801 -p 5696:5696 -p 1441:1441 --env-file=sklmenv.txt -v sklAppVolume_db2:/opt/ibm/wlp/usr/products/ibmcom/sklm
```

Sample command with parameters:

```
docker run --name skl -itd -h skl.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=skladminuser -e SKLMADMIN_PASSWORD=sklmpswd -e DB_HOST=172.x.x.x -e DB_PORT=50000 -e SKLM_SEED=68d95f0081f1dbfc0b06de9b0916df1c -e DB_PASSWORD=sklmbd2 -e
```

```
DB_TYPE=db2 -e DB_USER=sklmb41 -e DBNAME=sklmb41 -v sklAppVolume_db2:/opt/ibm/wlp/usr/
products_ibmcom/sklm
```

- Monitor the progress by using the docker logs command.

```
docker logs -f skl
```

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

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

- 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.

- 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.
- Click **Activate License**.
- Upload the IBM Security Guardium Key Lifecycle Manager license activation file and activate the license.
- Click **Login**.
- Log in to the IBM Security Guardium Key Lifecycle Manager graphical user interface with the Administrator user credentials (skladmin).

### 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 Manager See Administering](#).

## Installing 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:

Resource	Requirement
CPU	4 Cores
Memory	8 GB
Disk space	100 GB
Operating system and supported architectures	Linux <ul style="list-style-type: none"> <li>– x86_64</li> <li>– s390x</li> </ul>

- 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](mailto: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 sklmbd -f Dockerfile.sample --no-cache .
```

4. Run the customized Db2 container. For example:

```
docker run --name sklmbd --restart=always --detach --ipc="" --cap-add=IPC_OWNER -p
50000:50000 -e LICENSE=accept -e DB2INSTANCE=sklmbd41 -e DB2INST1_PASSWORD=SKLM@db2
-e DBNAME=sklmbd41 -v sklmbd2Volume:/database sklmbd
```

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=sklmbd41 -e POSTGRES_DB=sklmbd41 -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 install 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 install 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=sklmbd41 -e DBNAME=sklmbd41 sklmhsm

```

For more information about the parameters, see “Parameters to deploy IBM Security Guardium Key Lifecycle Manager container using Docker” on page 15.

**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 Manager See Administering](#).



## Installing on IBM zCX environment with Db2 for z/OS

Use instructions in this topic to install 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=sk1mdb41
DB_NAME=sk1mdb41
```

- 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:

<i>Table 4. Minimum system requirements</i>	
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](mailto: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.

- 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 .
```

- Ensure that the database (sklmb41) is running and ready to accept connections.
- 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 15.

- 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.

- 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.
```

- 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.

- 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.
- Click **Activate License**.
- Upload the IBM Security Guardium Key Lifecycle Manager license activation file and activate the license.
- Click **Login**.
- 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 Manager](#) See [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.

<i>Table 5. Parameters and their descriptions</i>		
Parameter	Mandatory/ Optional	Description
<b>Container name</b>		
<b>name</b>	Mandatory	Name for the container.
<b>Environment variables</b>		
<b>DB_PASSWORD</b>	Mandatory	Password to connect to the database instance where the IBM Security Guardium Key Lifecycle Manager database is running
<b>DB_TYPE</b>	Optional	Type of the database. Depending on the database that you use, specify one of the following values: <b>db2 (Default value)</b> Db2 database <b>postgres</b> PostgreSQL database <b>zos_db</b> Native (non-container) Db2 for z/OS <b>Note:</b> This parameter is ignored in the subsequent docker run commands when the same value of the <b>sk1mAppVolume</b> parameter is used.
<b>DB_USER</b>	Optional	User name of the database. Default value: sk1mdb41
<b>DB_NAME</b>	Optional	Name of the database. When the value of <b>DB_TYPE</b> is <b>zos_db</b> , specify the location name of the database. Default value: sk1mdb41
<b>DB_PORT</b>	Mandatory	Port number of the database instance where the IBM Security Guardium Key Lifecycle Manager database is running
<b>DB_HOST</b>	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.

<i>Table 5. Parameters and their descriptions (continued)</i>		
<b>Parameter</b>	<b>Mandatory/ Optional</b>	<b>Description</b>
<b>LICENSE</b>	Mandatory	Variable to accept license terms. Specify value as accept.
<b>SKLM_SEED</b>	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.  <b>Note:</b> 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 <b>sklmAppVolume</b> parameter is used.
<b>LIBERTY_KEY_STORE_PASSWORD</b>	Optional	Password for the IBM Security Guardium Key Lifecycle Manager keystore. Default value: Ch@ngemypa55word  <b>Note:</b> 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 <b>sklmAppVolume</b> parameter is used.
<b>LIBERTY_KEY_STORE_PASSWORD_OLD</b>	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 <b>LIBERTY_KEY_STORE_PASSWORD</b> parameter. Default value: Ch@ngemypa55word  <b>Note:</b> 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 <b>sklmAppVolume</b> parameter is used.
<b>LIBERTY_AES_ENCRYPTION_KEY</b>	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 <b>SKLM_SEED</b> parameter for encryption.
<b>HEALTH_AUTHORIZATION_TOKEN</b>	Optional	Health token in your Kubernetes environment.  <b>Note:</b> This parameter is applicable only when you are deploying on a Kubernetes cluster by using Helm charts.
<b>Port numbers</b>		

<i>Table 5. Parameters and their descriptions (continued)</i>		
<b>Parameter</b>	<b>Mandatory/ Optional</b>	<b>Description</b>
9443	Mandatory	Port number for the graphical user interface.
5696	Mandatory	KMIP TLS port
1441	Mandatory	IPP TLS port
3801	Mandatory	IPP TCP port
<b>Persistent storage</b>		
<b>sklmAppVolume</b>	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=sklmb41
DBNAME=sklmb41
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=sklmb41
DBNAME=sklmb41
```

### 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=sklmb41
DB_USER=sklmb41
DB_PASSWORD=xxxxxx
DB_HOST=9.x.x.x
DB_PORT=446
LICENSE=accept
SKLM_SEED=68d95f0081f1dbfc0b06de9b0916df1c
SKLMADMIN_USERNAME=skladmin
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 than 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 user authentication

---

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

### About this task

Depending on the authentication mechanism, IBM Security Guardium Key Lifecycle Manager obtains the user information. If an LDAP server is configured, user information is obtained from the LDAP server. You can then authorize users to control their access to tasks and data in IBM Security Guardium Key Lifecycle Manager.

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.

You can configure file or LDAP based authentication by using the graphical user interface or the REST interface.

You can view the authentication configuration values. See [“Get Authentication Configuration Details REST Service” on page 63](#).

### Procedure

- Using graphical user interface
  - 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.
  - c) Click the **Configuration** tab.

The current user authentication properties are displayed.
  - d) To update the user authentication configuration, click **Update**.
  - e) Update the basic and advanced configuration properties as per your requirement and click **Save**.

- Using REST interface

To configure file-based authentication, use the [“Update File-Based Authentication Configuration REST Service” on page 68](#).

To configure LDAP-based authentication, use the [“Update LDAP-Based Authentication Configuration REST Service” on page 71](#).

## 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 and click **Search**.
4. From the search results, select the user.
5. 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.
6. 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.
7. 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**.
8. 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.
- IBM Java™ JCEPlusFIPS is not supported on the zCX platform.
- While configuring the security standards, ensure that you only use the Cipher Suites that are supported by the client.
- 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>

## 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

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

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

<i>Path parameter</i>	
JSON object with the following specification.	
JSON property name	Description
<b>roleName</b>	Specify a name for the user role that you want to add.

##### Response

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

<i>Success Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the code that is specified by the <b>status</b> property.
<b>status</b>	Returns the status to indicate if the operation was successful.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	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

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

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

<i>Path Parameter</i>	
JSON object with the following specification.	
JSON property name	Description
<b>groupName</b>	Specify a name for the user group that you want to add.

<i>Request Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>description</b>	Specify a description for the user group that you want to add.

### Response

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

<i>Success Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the code that is specified by the <b>status</b> property.
<b>status</b>	Returns the status to indicate if the operation was successful.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	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

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

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

<i>Query Parameter</i>	
JSON object with the following specification.	
JSON property name	Description
<b>userName</b>	Required. Specify the name of the user to which you want to assign the roles.

<i>Request Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>roleNames</b>	Required. Specify a JSON array of role names that you want to assign to the user.

### Response

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

<i>Success Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the code that is specified by the <b>status</b> property.
<b>status</b>	Returns the status to indicate if the operation was successful.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	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

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

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

<i>Query Parameter</i>	
JSON object with the following specification.	
JSON property name	Description
<b>groupName</b>	Required. Specify the name of the user group to which you want to assign the roles.

<i>Request Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>roleNames</b>	Required. Specify a JSON array of role names that you want to assign to the user.

### Response

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

<i>Success Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the code that is specified by the <b>status</b> property.
<b>status</b>	Returns the status to indicate if the operation was successful.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	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

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

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

<i>Query Parameter</i>	
JSON object with the following specification.	
JSON property name	Description
<b>userName</b>	Required. Specify the name of the user to which you want to assign the user groups.

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

### Response

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

<i>Success Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the code that is specified by the <b>status</b> property.
<b>status</b>	Returns the status to indicate if the operation was successful.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	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

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

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

<i>Path parameter</i>	
JSON object with the following specification.	
JSON property name	Description
<b>roleName</b>	Specify the name of the user role that you want to delete.

### Response

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

<i>Success Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the code that is specified by the <b>status</b> property.
<b>status</b>	Returns the status to indicate if the operation was successful.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	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

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

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

<i>Path Parameter</i>	
JSON object with the following specification.	
JSON property name	Description
<b>userName</b>	Specify the user name of the user that you want to delete.

### Response

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

<i>Success Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the code that is specified by the <b>status</b> property.
<b>status</b>	Returns the status to indicate if the operation was successful.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	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

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

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

<i>Path parameter</i>	
JSON object with the following specification.	
JSON property name	Description
<b>groupName</b>	Specify the name of the user group that you want to delete.

### Response

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

<i>Success Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the code that is specified by the <b>status</b> property.
<b>status</b>	Returns the status to indicate if the operation was successful.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	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

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

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

### Response

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

<i>Response Headers (continued)</i>	
Header name	Value and description
<b>Content-Type</b>	application/json
<b>Content-Language</b>	Locale for the response message.

<i>Success Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns 0 when the request is successful. Otherwise, returns 1.
<b>user</b>	Returns a comma-separated list of all the users.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	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": "skladmin",
    "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**

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

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

**Response**

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

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

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	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": [
    "skladmin"
  ]
}
```

#### 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

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

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

### Response

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

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

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	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": "sdfdsfdfsfd",
    "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**

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

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

**Response**

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

<i>Path parameter</i>	
JSON object with the following specification.	
JSON property name	Description
<b>roleName</b>	Specify the user role for which you want to retrieve the details.

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

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	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

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



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

**Response**

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

<i>Success Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns 0 when the request is successful. Otherwise, returns 1.
<i>For every role, the following details are returned:</i>	
<b>name</b>	Returns the name of the user role.
<b>description</b>	Returns the description of the user role.
<b>isDefault</b>	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
<b>code</b>	Returns the application error code.
<b>message</b>	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

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

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

<i>Path parameter</i>	
JSON object with the following specification.	
JSON property name	Description
<b>userName</b>	Specify the user name for which you want to retrieve the details.

## Response

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

*Success Response Body*

JSON object with the following specification.

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

*Error Response Body*

JSON object with the following specification.

JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	Returns a message that describes the error.

**Example****Get details of a user**

```
GET
https://host:port/SKLM/rest/v1/ckms/usermanagement/users/LT0user
```

**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
<b>host</b>	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.

<i>Request Parameters (continued)</i>	
Parameter	Description
<b>port</b>	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

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

<i>Path parameter</i>	
JSON object with the following specification.	
JSON property name	Description
<b>oldGroupName</b>	Specify the name of the user group that you want to modify.

<i>Request Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>newGroupName</b>	Specify the new name for the user group.
<b>description</b>	Specify a description for the user group that you want to modify.

## Response

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



*Success Response Body*

JSON object with the following specification.

JSON property name	Description
<b>code</b>	Returns the code that is specified by the <b>status</b> property.
<b>status</b>	Returns the status to indicate if the operation was successful.

*Error Response Body*

JSON object with the following specification.

JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	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."
}
```



## Chapter 3. User Authentication and Security Standards Configuration REST Services

### User authentication and security standards configuration REST services

You can use the user authentication REST services or APIs to configure and view the configuration parameters of containerized IBM Security Guardium Key Lifecycle Manager. Use the security standard configuration REST services to configure and view the security standards for the containerized product.

#### Get Authentication Configuration Details REST Service

Use **Get Authentication Configuration Details REST Service** to obtain the values of the authentication configurations in a containerized IBM Security Guardium Key Lifecycle Manager.

##### Operation

GET

##### URL

`https://host:port/SKLM/rest/v1/ckms/authnConfigProperties`

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

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

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

##### Response

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

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name	Description
<b>parameter</b>	Returns the following information for each parameter: <ul style="list-style-type: none"> <li>• Value of the parameter</li> <li>• Whether the parameter is editable</li> <li>• Whether the parameter is mandatory</li> </ul>
<b>code</b>	Returns the code that is specified by the status property.
<b>status</b>	Returns the status to indicate whether the node is added to the multi-master cluster.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	Returns a message that describes the error.

## Example

### Service request to obtain authentication configuration values

```
GET https://localhost:port/SKLM/rest/v1/ckms/SKLM/rest/v1/ckms/authnConfigProperties
```

### Success response

```
{
  "AUTH_TYPE_LDAP": {
    "value": "false",
    "editable": true,

```

```

    "mandatory": true
  },
  "LDAP_GR_OBJ_CLASS": {
    "value": "",
    "editable": true,
    "mandatory": false
  },
  "LDAP_PR_OBJ_CLASS": {
    "value": "",
    "editable": true,
    "mandatory": false
  },
  "LDAP_RECURSIVE_SEARCH": {
    "value": "",
    "editable": true,
    "mandatory": false
  },
  "LDAP_USER_FILTER": {
    "value": "",
    "editable": true,
    "mandatory": false
  },
  "LDAP_GROUP_FILTER": {
    "value": "",
    "editable": true,
    "mandatory": false
  },
  "LDAP_USER_ID_MAP": {
    "value": "",
    "editable": true,
    "mandatory": false
  },
  "LDAP_GROUP_ID_MAP": {
    "value": "",
    "editable": true,
    "mandatory": false
  },
  "LDAP_GROUP_MEMBER_ID_MAP": {
    "value": "",
    "editable": true,
    "mandatory": false
  },
  "LDAP_TYPE": {
    "value": "",
    "editable": true,
    "mandatory": true
  },
  "LDAP_HOST": {
    "value": "",
    "editable": true,
    "mandatory": true
  },
  "LDAP_PORT": {
    "value": "",
    "editable": true,
    "mandatory": true
  },
  "LDAP_BASE_ENTRY": {
    "value": "",
    "editable": true,
    "mandatory": true
  },
  "LDAP_BIND_DN": {
    "value": "",
    "editable": true,
    "mandatory": false
  },
  "LDAP_BIND_PASSWD": {
    "value": "",
    "editable": true,
    "mandatory": false
  },
  "LDAP_SSL_ENABLED": {
    "value": "",
    "editable": true,
    "mandatory": false
  },
  "LDAP_SKLMADMIN_USERNAME": {
    "value": "sklmadmin",
    "editable": true,
    "mandatory": false
  },
}

```

```

"LDAP_DISPLAY_NAME": {
  "value": "",
  "editable": true,
  "mandatory": false
},
"AUTH_TYPE_FILE": {
  "value": "true",
  "editable": true,
  "mandatory": true
},
"SKLMADMIN_USERNAME": {
  "value": "skladmin",
  "editable": true,
  "mandatory": true
},
"SKLMADMIN_PASSWORD": {
  "value": "",
  "editable": true,
  "mandatory": true
}
}

```

### Error response

```

Status Code : 400 Bad Request
Content-Language: en
{"code" : "CTGKM6002E", "message" : "CTGKM6002E Bad Request: Invalid user authentication ID or invalid request format."}

```

## Get Security Configuration Details REST Service

Use **Get Security Configuration Details REST Service** to obtain the values of the security configurations in a containerized IBM Security Guardium Key Lifecycle Manager.

### Operation

GET

### URL

`https://host:port/SKLM/rest/v1/ckms/securityConfigurations/{param}`

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

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

<i>Request Headers</i>	
Header name	Value
<b>Content-Type</b>	application/json
<b>Accept</b>	application/json
<b>Authorization</b>	SKLMAuth userAuthId=<authIdValue>

<i>Request Headers (continued)</i>	
Header name	Value
<b>Accept-Language</b>	Any valid locale that is supported by IBM Security Guardium Key Lifecycle Manager. For example: en or de

<i>Query parameters</i>	
JSON object with the following specification:	
Property name	Description
<b>param</b>	Specify one of the following parameters: <ul style="list-style-type: none"> <li>• FIPS</li> <li>• SUITE_B</li> <li>• SP800_131A</li> <li>• securityLevel</li> <li>• enabledCiphers</li> <li>• ALL</li> </ul> To obtain the values of all the parameters, specify ALL.

## Response

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

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name	Description
<b>parameter</b>	Returns the value for the requested parameter.
<b>code</b>	Returns the code that is specified by the status property.

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name (continued)	Description
<b>status</b>	Returns the status to indicate whether the node is added to the multi-master cluster.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	Returns a message that describes the error.

## Example

### Service request to obtain security configuration values

```
GET https://localhost:port/SKLM/rest/v1/ckms/SKLM/rest/v1/ckms/securityConfigurations/ALL
```

#### Success response

```
{
  "FIPS": "on",
  "Suite_B": "128",
  "SP800_131A": "on",
  "enabledCiphers":
  "SSL_ECDH_ECDSA_WITH_AES_128_GCM_SHA256,SSL_ECDH_RSA_WITH_AES_128_GCM_SHA256,SSL_DHE_RSA_WITH_AES_128_GCM_SHA256",
  "securityLevel": ""
}
```

### Service request to obtain security configuration values

```
GET https://localhost:port/SKLM/rest/v1/ckms/SKLM/rest/v1/ckms/securityConfigurations/FIPSPlus
```

#### Error response

```
{
  "code": "CTGKM3546E",
  "message": "CTGKM3546E Security Configurations with parameter value ALL, FIPS, Suite_B, SP800_131A, securityLevel and enabledCiphers are only allowed."
}
```

## Update File-Based Authentication Configuration REST Service

Use **Update File-Based Authentication Configuration REST Service** to configure the file-based authentication requirements in containerized IBM Security Guardium Key Lifecycle Manager.

### Operation

POST

### URL

`https://host:port/SKLM/rest/v1/ckms/authnConfigProperties/file`

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**

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

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

<i>Request body</i>	
JSON object with the following specification:	
<b>Property name</b>	<b>Description</b>
<b>AUTH_TYPE_FILE</b>	Specify true if you want to configure file-based user authentication. Other possible value is false.
<b>SKLMADMIN_USERNAME</b>	Specify the user name of the IBM Security Guardium Key Lifecycle Manager administrator. You can specify only alphanumeric characters. Default value: skladmin <b>Note:</b> This parameter is mandatory if you have specified true as a value in the <b>AUTH_TYPE_LDAP</b> parameter.
<b>SKLMADMIN_PASSWORD</b>	Specify the password for the IBM Security Guardium Key Lifecycle Manager administrator user that is specified in the <b>SKLMADMIN_USERNAME</b> parameter. <b>Note:</b> This parameter is mandatory if you have specified true as a value in the <b>AUTH_TYPE_LDAP</b> parameter.

**Response**

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

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name	Description
<b>code</b>	Returns the code that is specified by the status property.
<b>status</b>	Returns the status to indicate whether the node is added to the multi-master cluster.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	Returns a message that describes the error.

## Example

### Specify LDAP configuration

```
POST https://localhost:port/SKLM/rest/v1/ckms/SKLM/rest/v1/ckms/authnConfigProperties/file
{
  "AUTH_TYPE_FILE": "true",
  "SKLMADMIN_USERNAME": "sklmadmin",
  "SKLMADMIN_PASSWORD": "sklm@passwd"
}
```

### Success response

```
Status Code: 200 OK
[
  {
    "code": "0", "status": "Succeeded"
  }
]
```

**Error response**

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

**Update LDAP-Based Authentication Configuration REST Service**

Use **Update LDAP-Based Authentication Configuration REST Service** to configure the authentication requirements in containerized IBM Security Guardium Key Lifecycle Manager.

**Operation**

POST

**URL**

`https://host:port/SKLM/rest/v1/ckms/authnConfigProperties/ldap`

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**

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

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

<i>Request body</i>	
JSON object with the following specification:	
<b>Property name</b>	<b>Description</b>
<b>AUTH_TYPE_LDAP</b>	Specify true if you want to configure a supported LDAP provider for user authentication. Otherwise, by default, file-based authentication is available.  Other possible value is false.

<i>Request body</i>	
JSON object with the following specification:	
<b>Property name</b> <i>(continued)</i>	<b>Description</b>
<b>LDAP_TYPE</b>	Specify the type of LDAP server to be connected to.  Possible values: <b>IBM Tivoli® Directory Server</b> Configure the LDAP registry to use IBM Tivoli Directory Server. <b>Microsoft Active Directory</b> Configure the LDAP registry to use Microsoft Active Directory. <b>Note:</b> This parameter is mandatory if you have specified <code>true</code> as a value in the <b>AUTH_TYPE_LDAP</b> parameter.
<b>LDAP_HOST</b>	Specify the host name or IP address of the LDAP server.  <b>Note:</b> This parameter is mandatory if you have specified <code>true</code> as a value in the <b>AUTH_TYPE_LDAP</b> parameter.
<b>LDAP_PORT</b>	Specify the port number of the LDAP server.  <b>Note:</b> This parameter is mandatory if you have specified <code>true</code> as a value in the <b>AUTH_TYPE_LDAP</b> parameter.
<b>LDAP_BASE_ENTRY</b>	Specify the base node in the LDAP directory. For example:  <pre>"DC=k1m,DC=com"</pre> where, DC is Domain Component.  <b>Note:</b> This parameter is mandatory if you have specified <code>true</code> as a value in the <b>AUTH_TYPE_LDAP</b> parameter.
<b>LDAP_GR_OBJECT_CLASS</b>	Specify the object class that is defined for the group LDAP entity type in the LDAP server. For example, <code>GroupofUniqueNames</code> .
<b>LDAP_PR_OBJECT_CLASS</b>	Specify the object class that is defined for the person LDAP entity type in the LDAP server. For example, <code>person</code> .
<b>LDAP_RECURSIVE_SEARCH</b>	Specify <code>true</code> to indicate whether to perform a nested group search.  Other value: <code>false</code> (default)  Specify <code>true</code> only if the LDAP server does not support recursive server-side searches.
<b>LDAP_USER_FILTER</b>	Specify an LDAP filter clause for searching the user registry for users.
<b>LDAP_GROUP_FILTER</b>	Specify an LDAP filter clause for searching the user registry for groups.
<b>LDAP_USER_ID_MAP</b>	Specify an LDAP filter that maps the name of a user to an LDAP entry.
<b>LDAP_GROUP_ID_MAP</b>	Specify an LDAP filter that maps the name of a group to an LDAP entry.
<b>LDAP_GROUP_MEMBER_ID_MAP</b>	Specify an LDAP filter that identifies user to group memberships.
<b>LDAP_BIND_DN</b>	Specify the distinguished name (DN) for the application server, which is used to bind to the directory service.

<i>Request body</i>	
JSON object with the following specification:	
<b>Property name</b> <i>(continued)</i>	<b>Description</b>
<b>LDAP_BIND_PASSWD</b>	Specify the password for the DN that is specified in the <b>LDAP_BIND_DN</b> parameter. 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.
<b>LDAP_SSL_ENABLED</b>	Specify true to indicate that a TLS connection must be made to the LDAP server. Otherwise, specify false.
<b>LDAP_SKLMADMIN_USERNAME</b>	Specify the user name of the LDAP administrator. You can specify only alphanumeric characters.  <b>Note:</b> You must provide a value for this parameter or for the <b>SKLMADMIN_USERNAME</b> parameter if you have specified true as the value in the <b>AUTH_TYPE_LDAP</b> parameter.
<b>LDAP_DISPLAY_NAME</b>	Specify a name for the LDAP user to be displayed on the graphical user interface.

**Response**

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

<i>Success response body</i>	
JSON object with the following specification:	
<b>JSON property name</b>	<b>Description</b>
<b>code</b>	Returns the code that is specified by the status property.
<b>status</b>	Returns the status to indicate whether the node is added to the multi-master cluster.

*Error Response Body*

JSON object with the following specification.

JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	Returns a message that describes the error.

**Example****Specify LDAP configuration**

```
POST https://localhost:port/SKLM/rest/v1/ckms/SKLM/rest/v1/ckms/authnConfigProperties/ldap
{"AUTH_TYPE_LDAP": "true"
}
```

**Success response**

```
Status Code: 200 OK
[
  {
    "code": "0", "status": "Succeeded"
  }
]
```

**Error response**

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

**Update Security Configurations REST Service**

Use **Update Security Configurations REST Service** to set the security configurations in a containerized IBM Security Guardium Key Lifecycle Manager.

**Operation**

POST

**URL**

`https://host:port/SKLM/rest/v1/ckms/securityConfigurations/update`

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**

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

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

<i>Request body</i>	
JSON object with the following specification:	
Property name	Description
<b>FIPS</b>	<p>Specify the value as <code>on</code> or <code>off</code> to enable or disable Federal Information Processing Standards (FIPS) publication 140-2 standard compliance in Guardium Key Lifecycle Manager.</p> <p>By default, this flag is set to <code>off</code>.</p> <p>When you set this property to <code>on</code>, Guardium Key Lifecycle Manager uses the IBMJCEFIPS provider instead of the IBMJCE provider for all cryptographic functions.</p>
<b>Suite_B</b>	<p>Specify the value as <code>on</code> or <code>off</code> to enable or disable US National Security Agency (NSA) Suite B standard compliance in Guardium Key Lifecycle Manager.</p> <p>By default, this flag is set to <code>off</code>.</p> <p>Set the property with one of the following values:</p> <ul style="list-style-type: none"> <li>• 128</li> <li>• 192</li> </ul> <p>When you set this property to <code>on</code>, Guardium Key Lifecycle Manager uses the IBMJSSE2 provider instead of the IBMJCE provider for all cryptographic functions.</p>
<b>SP800_131A</b>	<p>Specify the value as <code>on</code> or <code>off</code> to enable or disable 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.</p> <p>By default, this flag is set to <code>off</code>.</p> <p>Set the property to <code>on</code> to enable this standard.</p>
<b>securityLevel</b>	<p>Specify one of the following values to configure the cipher suite group to be used by the TLS handshake.</p> <ul style="list-style-type: none"> <li>• HIGH: For 128-bit ciphers and higher</li> <li>• MEDIUM: For 40-bit ciphers</li> <li>• WEAK: For all ciphers without encryption</li> <li>• CUSTOM: When the cipher suite group is customized</li> </ul> <p>This property is ignored if you set the <b>enabledCiphers</b> property with a specific list of ciphers.</p>

<i>Request body</i>	
JSON object with the following specification:	
Property name (continued)	Description
<b>enabledCiphers</b>	<p>Specify a unique list of cipher suites. You can specify multiple cipher suites by separating the cipher suites with a space. For example:</p> <pre>enabledCiphers="TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256"</pre> <p>If you set this property, the <b>securityLevel</b> property is ignored.</p> <p><b>Note:</b> Ensure that you use only the Cipher Suites that are supported by the client.</p>

## Response

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

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name	Description
<b>code</b>	Returns the code that is specified by the status property.
<b>status</b>	Returns the status to indicate whether the node is added to the multi-master cluster.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	Returns a message that describes the error.







## Chapter 4. Client and Client Group Management REST Services

### Client and client group management REST services

You can use the client and client group management REST services to manage and work with IBM Security Guardium Key Lifecycle Manager clients and groups. Cloud applications or clients that need to use keys and other cryptographic objects from IBM Security Guardium Key Lifecycle Manager use these REST APIs to communicate with the IBM Security Guardium Key Lifecycle Manager server.

For more information about IBM Security Guardium Key Lifecycle Manager client management, see [Managing client groups, clients, and their cryptographic objects](#).

#### Assign Client Certificate to a Client REST Service

Use **Assign Client Certificate to a Client REST Service** to associate a certificate with a client.

##### Operation

PUT

##### URL

`https://host:port/SKLM/rest/v1/clients/{clientName}/assignCertificate`

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

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

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

*Path parameters*

JSON object with the following specification:

Property name	Description
<b>clientName</b>	Required. Specify the name of the client.

*Request body*

JSON object with the following specification:

Property name	Description
<b>certUseOption</b>	<p>Required. Specify the option that you want to use to assign the certificate to the client.</p> <p>Specify one of the following values:</p> <ul style="list-style-type: none"> <li>• <b>ACCEPT_PENDING_CERT</b>: To accept the client communication certificate that was pushed to the IBM Security Guardium Key Lifecycle Manager server from a client device for secure communication</li> <li>• <b>IMPORT_CERT</b>: To import a certificate and assign it to the client</li> <li>• <b>USE_EXISTING</b>: To assign an existing certificate to the client</li> </ul>
<b>certAlias</b>	Required. Specify the alias of the certificate that you want to assign to the client.
<b>importPath</b>	<p>Required only if you specify <b>IMPORT_CERT</b> as the option for the <b>certUseOption</b> property.</p> <p>Specify the complete path of the certificate file to be imported. For example, C:/Program Files/IBM/WebSphere/AppServer/products/sklm/data/clientCert.cer</p>
<b>certUUID</b>	<p>Required only if you specify <b>ACCEPT_PENDING_CERT</b> as the option for the <b>certUseOption</b> property.</p> <p>Specify the universal unique identifier of the certificate that you want to accept and import.</p>

**Response**

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

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>message</b>	Returns the message to indicate the operation is successful.

<i>Error response body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>error</b>	Returns a message that describes the error.

## Example

### Assign a certificate to a client

```
PUT https://localhost:port/SKLM/rest/v1/clients/client_rest/assignCertificate
{
  "certUseOption": "IMPORT_CERT",
  "certAlias": "testcert",
  "importPath": "/opt/IBM/WebSphere/AppServer/products/sklm/data/clientsslcert.cer"
}
```

### Success response

```
{
  "message": "CTGKM3409I Successfully assigned certificate to client.",
  "messageId": "CTGKM3409I"
}
```

### Error response

```
{
```

```

"messageId": "CTGKM0543E",
"error": "CTGKM0543E An error occurred importing certificate: /opt/IBM/WebSphere/
AppServer/products/sklm/data/clientsslcer.cer (No such file or directory)"
}

```

## Assign Users to a Client REST Service

Use **Assign Users to a Client REST Service** to assign users to a client who can access the cryptographic objects of the client.

### Operation

PUT

### URL

`https://host:port/SKLM/rest/v1/clients/{clientName}/assignUsers`

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

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

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

<i>Path parameters</i>	
JSON object with the following specification:	
Property name	Description
<b>clientName</b>	Required. Specify the name of the client to which you want to assign the users.

<i>Request body</i>	
JSON object with the following specification:	
Property name	Description
<b>users</b>	Required. Specify names of the users that you want to assign to the client. Use a comma-separated list for multiple users. For example, "TestUser1, TestUser2"

**Response**

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

*Table 6. Success and Error response body*

JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>message</b>	Returns the message to indicate the success and error part of the operation.

**Example****Assign users to a client**

```
PUT https://localhost:port/SKLM/rest/v1/client_rest/assignUsers
{"users" : "KLMUserRole_Member_2"}
```

**Success and Error response**

```
{
  "message": "CTGKM3445I Assigned users [KLMUserRole_Member_2], Error processing users
[]",
  "messageId": "CTGKM3445I"
}
```

## Create Client REST Service

Use **Create Client REST Service** to create a client that is registered with Guardium Key Lifecycle Manager server.

### Operation

POST

### URL

`https://host:port/SKLM/rest/v1/clients`

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

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

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

<i>Request body</i>	
JSON object with the following specification:	
Property name	Description
<b>clientName</b>	Required. Specify the name of the client that you want to create.
<b>clientType</b>	Optional. Specify the way of client communication. Possible values: <ul style="list-style-type: none"> <li>• KMIP: This is the default value.</li> <li>• REST</li> </ul>

### Response



<i>Response Headers</i>	
Header name	Value and description
<b>Status Code</b>	<p><b>201 Created</b> The request was successful. The response body contains the requested representation.</p> <p><b>400 Bad Request</b> The authentication information was not provided in the correct format.</p> <p><b>401 Unauthorized</b> The authentication credentials were missing or incorrect.</p> <p><b>404 Not Found Error</b> The processing of the request fails.</p> <p><b>500 Internal Server Error</b> The processing of the request fails because of an unexpected condition on the server.</p>
<b>Content-Type</b>	application/json
<b>Content-Language</b>	Locale for the response message.

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>message</b>	Returns the message to indicate the operation is successful.

<i>Error response body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>error</b>	Returns a message that describes the error.

## Example

### Create a client

```
POST https://localhost:port/SKLM/rest/v1/clients
{
  "clientName": "client_rest1",
  "clientType": "REST"
}
```

### Success response

```
{
  "message": "CTGKM3411I Successfully created client CLIENT_REST1 .",
  "messageId": "CTGKM3411I"
}
```

### Error response

```
{
```

```

"error": "CTGKM3430E Failed to create client. Client with name CLIENT_REST1 already
exists.",
"messageId": "CTGKM3430E"
}

```

## Delete Client REST Service

Use **Delete Client REST Service** to delete a client and the cryptographic objects that are associated with it.

### Operation

DELETE

### URL

`https://host:port/SKLM/rest/v1/clients/{clientName}`

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

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

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

<i>Path parameters</i>	
JSON object with the following specification:	
Property name	Description
<b>clientName</b>	Required. Specify the name of the client that you want to delete.

### Response

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

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>message</b>	Returns the message to indicate the operation is successful.

<i>Error response body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>error</b>	Returns a message that describes the error.

## Example

### Delete a client

```
DELETE https://localhost:port/SKLM/rest/v1/clients/client_rest1
```

### Success response

-

### Error response

```
{
  "message": "CTGKM3408E Client with client_rest1 name not found.",
  "messageId": "CTGKM3408E"
}
```

## Get Client Details REST Service

Use **Get Client Details REST Service** to retrieve details of a specific client from the database.

### Operation

GET

### URL

`https://host:port/SKLM/rest/v1/clients/{clientName}`

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

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

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

<i>Path parameters</i>	
JSON object with the following specification:	
Property name	Description
<b>clientName</b>	Required. Specify the name of the client of which you want to retrieve the details.

### Response

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

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name	Description
<b>client</b>	Returns all the cryptographic objects that are associated with the client.

<i>Error response body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>error</b>	Returns a message that describes the error.

## Example

### Listing client details

```
GET https://localhost:port/SKLM/rest/v1/clients/client_rest
```

### Success response

```
{
  "client": {
    "clientName": "CLIENT_REST",
    "userCount": 0,
    "users": "",
    "clientCert": "",
    "clientType": "REST",
    "managedObjectCount": 0,
    "object": ""
  }
}
```

**Error response**

```
{
  "error": "CTGKM3408E Client with client_rest name not found.",
  "messageId": "CTGKM3408E"
}
```

**List All Clients REST Service**

Use **List All Clients REST Service** to retrieve the list of clients from the database.

**Operation**

GET

**URL**

`https://host:port/SKLM/rest/v1/clients?clientType={clientType}`

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**

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

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

<i>Pagination headers</i>	
JSON object with the following specification:	
Property name	Description
<b>Range</b>	Optional. Specify the number of items (in the range 0 to 9) to be displayed on a page. Provide the value in the following format: <pre>items=0-9</pre> where, 0 is the start and 9 is the end of the range. If this property is not specified, all items are listed.

<i>Query parameters</i>	
JSON object with the following specification:	
Property name	Description
<b>clientType</b>	Optional. If you want to list clients that are using a specific way of client communication, then specify this parameter. You can provide one of the following values: <ul style="list-style-type: none"> <li>• KMIP</li> <li>• REST</li> </ul> If this property is not specified, all clients are listed.

**Response**

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

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name	Description
<b>client</b>	Returns an array of clients.

<i>Error response body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>error</b>	Returns a message that describes the error.

## Examples

### List all clients

```
GET https://localhost:port/SKLM/rest/v1/clients
```

#### Success response

```
{
  "client": [
    {
      "clientName": "TEST",
      "userCount": 2,
      "users": "KLMUserRole_Member_2, KLMUserRole_Member_3",
      "clientCert": "",
      "clientType": "REST",
      "managedObjectCount": 13,
      "object": "Symmetric&nbsp;Key&nbsp;(10)\tPublic&nbsp;Key&nbsp;(1)\tPrivate&nbsp;Key&nbsp;(1)\tCertificate&nbsp;(1)\t"
    },
    {
      "clientName": "CLIENT_TEST",
      "userCount": 0,
      "users": "",
      "clientCert": "",
      "clientType": "KMIP",
      "managedObjectCount": 0,
      "object": ""
    },
    {
      "clientName": "CLIENT_REST",
      "userCount": 0,
      "users": "",
      "clientCert": "",
      "clientType": "REST",
      "managedObjectCount": 0,
      "object": ""
    }
  ]
}
```

#### Error response

```
{
  "messageId": "CTGKM6004E",
  "error": "CTGKM6004E User is not authenticated or has already logged out."
}
```

### List all clients that use REST as the way of communication

```
GET https://localhost:port/SKLM/rest/v1/clients?clientType=REST
```

#### Success response

```
{
  "client": [
    {
      "clientName": "TEST",
      "userCount": 2,
      "users": "KLMUserRole_Member_2, KLMUserRole_Member_3",
      "clientCert": "",
      "clientType": "REST",
      "managedObjectCount": 13,
      "object": "Symmetric&nbsp;Key&nbsp;(10)\tPublic&nbsp;Key&nbsp;(1)\tPrivate&nbsp;Key&nbsp;(1)\tCertificate&nbsp;(1)\t"
    },
    {
      "clientName": "CLIENT_REST",
      "userCount": 0,
      "users": "",
      "clientCert": "",
      "clientType": "REST",
      "managedObjectCount": 0,
      "object": ""
    }
  ]
}
```



**Error response**

```
{
  "messageId": "CTGKM6004E",
  "error": "CTGKM6004E User is not authenticated or has already logged out."
}
```

**Remove Users from a Client REST Service**

Use **Remove Users from a Client REST Service** to remove users that are assigned to a client.

**Operation**

PUT

**URL**

`https://host:port/SKLM/rest/v1/clients/{clientName}/removeUsers`

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**

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

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

<i>Path parameters</i>	
JSON object with the following specification:	
<b>Property name</b>	<b>Description</b>
<b>clientName</b>	Required. Specify the name of the client from which you want to remove the users.

*Request body*

JSON object with the following specification:

Property name	Description
<b>users</b>	Required. Specify names of the users that you want to remove from the client. Use a comma-separated list for multiple users. For example, "TestUser1, TestUser2".

**Response***Response Headers*

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

*Table 7. Success and Error response body*

JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>message</b>	Returns the message to indicate the success and error part of the operation.

**Example****Remove users from a client**

```
PUT https://localhost:port/SKLM/rest/v1/clients/client_rest/removeUsers
{"users" : "KLMUserRole_Member_2,KLMUserRole_Member_3"}
```

**Success and Error response**

```
{
  "message": "CTGKM3446I Removed users [KLMUserRole_Member_2], Error processing users
[KLMUserRole_Member_3].",
  "messageId": "CTGKM3446I"
}
```

## Update Client Name REST Service

Use **Update Client Name REST Service** to update the name of a client.

### Operation

PUT

### URL

`https://host:port/SKLM/rest/v1/clients/updateClientName`

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

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

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

<i>Request body</i>	
JSON object with the following specification:	
Property name	Description
<b>clientName</b>	Required. Specify the name of the client that you want to update.
<b>newClientName</b>	Required. Specify the new name of the client.

### Response

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

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>message</b>	Returns the message to indicate the operation is successful.

<i>Error response body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>error</b>	Returns a message that describes the error.

## Example

### Update the name of a client

```
PUT https://localhost:port/SKLM/rest/v1/clients/updateClientName
```

### Success response

```
{
  "message": "CTGKM3420I Client name updated successfully.",
  "messageId": "CTGKM3420I"
}
```

### Error response

```
{
  "messageId": "CTGKM3408E",
  "error": "CTGKM3408E Client with CLIENT_KMIP name not found."
}
```

## Create Client Group REST Service

Use **Create Client REST Service** to create a client group that is registered with IBM Security Guardium Key Lifecycle Manager.

### Operation

POST

### URL

`https://host:port/SKLM/rest/v1/clients`

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

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

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

<i>Request body</i>	
JSON object with the following specification:	
Property name	Description
<b>groupName</b>	Required. Specify the name for the client group that you want to create.

### Response

<i>Response Headers</i>	
Header name	Value and description
<b>Status Code</b>	<p><b>201 Created</b> The request was successful. The response body contains the requested representation.</p> <p><b>400 Bad Request</b> The authentication information was not provided in the correct format.</p> <p><b>401 Unauthorized</b> The authentication credentials were missing or incorrect.</p> <p><b>404 Not Found Error</b> The processing of the request fails.</p> <p><b>500 Internal Server Error</b> The processing of the request fails because of an unexpected condition on the server.</p>
<b>Content-Type</b>	application/json
<b>Content-Language</b>	Locale for the response message.

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>message</b>	Returns the message to indicate the operation is successful.

<i>Error response body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>error</b>	Returns a message that describes the error.

## Example

### Create a client group

```
POST https://localhost:port/SKLM/rest/v1/clients/group
{"groupName": "TEST_CLIENT_GROUP"}
```

### Success response

```
{
  "message": "CTGKM3533I Created group with name TEST_CLIENT_GROUP.",
  "messageId": "CTGKM3533I"
}
```

### Error response

```
{
  "error": "CTGKM3542E Unable to create group because a group or a client with the same name TEST_CLIENT_GROUP already exists.",
  "messageId": "CTGKM3542E"
}
```

}

## Assign Clients to a Client Group REST Service

Use **Assign Clients to a Client Group REST Service** to assign users to a client who can access the cryptographic objects of the client.

### Operation

PUT

### URL

`https://host:port/SKLM/rest/v1/clients/groups/{groupName}/assignClients`

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

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

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

<i>Path parameters</i>	
JSON object with the following specification:	
Property name	Description
<b>clients</b>	Required. Specify the name of the client that you want to assign to the group. To assign multiple clients, provide the names in a comma-separated list.

<i>Request body</i>	
JSON object with the following specification:	
Property name	Description
<b>users</b>	Required. Specify names of the users that you want to assign to the client. Use a comma-separated list for multiple users. For example, "TestUser1, TestUser2"

**Response**

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

*Table 8. Success and Error response body*

<b>JSON property name</b>	<b>Description</b>
<b>messageId</b>	Returns the message identifier.
<b>message</b>	Returns the message to indicate the success and error part of the operation.

**Example****Assign clients to a client group**

```
PUT https://localhost:port/SKLM/rest/v1/clients/groups/TEST_CLIENT_GROUP/assignClients
{"clients": "TEST_CLIENT_1, TEST_CLIENT_2"}
```

**Success response**

```
{
  "message": "CTGKM3534I Assigned clients [TEST_CLIENT_1, TEST_CLIENT_2], Error processing
clients [].",
  "messageId": "CTGKM3534I"
}
```

**Error response**

```
{
  "error": "CTGKM0550E Input value cannot be an empty string for parameter clients.",
  "messageId": "CTGKM0550E"
}
```

**Delete Client Group REST Service**

Use **Delete Client Group REST Service** to delete a client group. The clients in the group or its objects are not deleted.

**Operation**

DELETE



**URL**

`https://< host >:< port &gt;/SKLM/rest/v1/clients/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**

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

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

**Response**

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

*Success Response Body*

JSON object with the following specification.

JSON property name	Description
<b>code</b>	Returns the code that is specified by the <b>status</b> property.
<b>status</b>	Returns the status to indicate if the operation was successful.

*Error Response Body*

JSON object with the following specification.

JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	Returns a message that describes the error.

**Examples****Delete a client group**

```
DELETE https://host:port/SKLM/rest/v1/clients/groups/TEST_CLIENT_GROUP
```

**Success response**

```
{
  "message": "CTGKM3534I Assigned clients [TEST_CLIENT_1, TEST_CLIENT_2], Error
processing clients [].",
  "messageId": "CTGKM3534I"
}
```

**Error response**

```
{
  "error": "CTGKM0550E Input value cannot be an empty string for parameter clients.",
  "messageId": "CTGKM0550E"
}
```

**List All Client Group Details REST Service**

Use **List All Client Group Details REST Service** to get details of all the client groups.

**Operation**

GET

**URL**

https://< host >:< port &gt;/SKLM/rest/v1/clients/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**

<i>Request Parameters</i>	
Parameter	Description
<b>host</b>	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.

<i>Request Parameters (continued)</i>	
Parameter	Description
<b>port</b>	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

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

**Response**

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

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>message</b>	Returns the message to indicate the operation is successful.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.

*Error Response Body*

JSON object with the following specification.

**JSON property name**  
(continued)**Description****message**

Returns a message that describes the error.

**Example****Get list of all client groups**

GET https://host:port/SKLM/rest/v1/clients/groups

**Success response**

```
{
  "groups": [
    {
      "groupName": "TEST_CLIENT_GROUP",
      "clientCount": 2,
      "clients": "TEST_CLIENT_2, TEST_CLIENT_1",
      "managedObjectCount": 0,
      "object": ""
    },
    {
      "groupName": "TEST_CLIENT_GROUP_2",
      "clientCount": 1,
      "clients": "TEST_CLIENT_3",
      "managedObjectCount": 0,
      "object": ""
    }
  ]
}
```

**Error response**

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

**List Clients Not in Any Client Group REST Service**

Use **List Clients Not in Any Client Group REST Service** to get a list of all the clients that are not in any client group.

**Operation**

GET

**URL**

https://&lt; host &gt;:&lt; port &amp;gt;/SKLM/rest/v1/clients/groups/clientList

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**

<i>Request Parameters</i>	
<b>Parameter</b>	<b>Description</b>
<b>host</b>	Specify the IP address or host name of the IBM Security Guardium Key Lifecycle Manager server.

<i>Request Parameters (continued)</i>	
Parameter	Description
<b>port</b>	Specify the port number on which the IBM Security Guardium Key Lifecycle Manager server listens for requests.

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

**Response**

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

<i>Success response body</i>	
JSON object with the following specification:	
JSON property name	Description
<b>messageId</b>	Returns the message identifier.
<b>message</b>	Returns the message to indicate the operation is successful.

<i>Error Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the application error code.

<i>Error Response Body</i>	
JSON object with the following specification.	
<b>JSON property name</b> <i>(continued)</i>	<b>Description</b>
<b>message</b>	Returns a message that describes the error.

## Examples

### Get list of clients not in a group

```
GET https://host:port/SKLM/rest/v1/clients/groups/clientList
```

#### Success response

```
{
  "client": [
    "TEST_CLIENT_4",
    "TEST_CLIENT_5"
  ]
}
```

#### Error response

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

## Remove Clients from Group REST Service

Use **Remove Clients from Group REST Service** to remove clients from a client group.

### Operation

PUT

### URL

`https://host:port/SKLM/rest/v1/clients/groups/{groupName}/removeClients`

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

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

<i>Request Headers</i>	
<b>Header name</b>	<b>Value</b>
<b>Content-Type</b>	application/json

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

<i>Request Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>groupName</b>	Required. Specify the name of the client group from which you want to remove the clients.
<b>clients</b>	Specify names of the clients that you want to remove from the group. Use a comma-separated list for multiple clients. For example, TestClient1, TestClient2.

## Response

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

<i>Success Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the code that is specified by the <b>status</b> property.
<b>status</b>	Returns the status to indicate if the operation was successful.

*Error Response Body*

JSON object with the following specification.

JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	Returns a message that describes the error.

**Example****Remove clients from a client group**

```
PUT https://host:port/SKLM/rest/v1/clients/groups/TEST_CLIENT_GROUP/removeClients
{"clients": "TEST_CLIENT_1"}
```

**Success response**

```
{
  "message": "CTGKM3536I Removed clients [TEST_CLIENT_1], Error processing clients [].",
  "messageId": "CTGKM3536I"
}
```

**Error response**

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

**Update Client Group Name REST Service**

Use **Update Client Group Name REST Service** to update the name of an existing client group.

**Operation**

PUT

**URL**

https://< host >:< port &gt;/SKLM/rest/v1/clients/groups/updateGroupName

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**

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

<i>Request Headers</i>	
Header name	Value
<b>Content-Type</b>	application/json



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

<i>Request Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>clientGroupName</b>	Specify the name of the client group that you want to update.
<b>newClientGroupName</b>	Specify the new name of the client group.

**Response**

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

<i>Success Response Body</i>	
JSON object with the following specification.	
JSON property name	Description
<b>code</b>	Returns the code that is specified by the <b>status</b> property.
<b>status</b>	Returns the status to indicate if the operation was successful.

**Error Response Body**

JSON object with the following specification.

JSON property name	Description
<b>code</b>	Returns the application error code.
<b>message</b>	Returns a message that describes the error.

**Examples****Update a client group name**

```
PUT https://host:port/SKLM/rest/v1/clients/groups/updateGroupName
{"clientGroupName":"TEST_CLIENT_GROUP","newClientGroupName":"TEST_CLIENT_GROUP_2"}
```

**Success response**

```
{
  "message": "CTGKM3547I Group name updated successfully.",
  "messageId": "CTGKM3547I"
}
```

**Error response**

```
{
  "messageId": "CTGKM3542E",
  "error": "CTGKM3542E Unable to create group because a group or a client with the same
name TEST_CLIENT_GROUP_2 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](http://www.ibm.com)<sup>®</sup> 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