CipherTrust Cloud Key Manager Installation & Configuration Guide

Release 1.8

Document Version 2

October 07, 2020

CipherTrust Cloud Key Manager Installation & Configuration Guide
1.8
Document Version 2
October 07, 2020

All information herein is either public information or is the property of and owned solely by Thales DIS France S.A. and/or its subsidiaries or affiliates who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of Thales DIS France S.A. and any of its subsidiaries and affiliates (collectively referred to herein after as “Thales”) information.

This document can be used for informational, non-commercial, internal and personal use only provided that:

- The copyright notice below, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- This document shall not be posted on any network computer or broadcast in any media and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities.

The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, Thales makes no warranty as to the value or accuracy of information contained herein.

The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, Thales reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time.

Thales hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Thales be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Thales does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Thales be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Thales products. Thales disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service or loss of privacy.

Contents

Preface

This guide describes how to deploy the CipherTrust Cloud Key Manager (CCKM), and how to configure it to connect with various data sources and clouds. It also describes how to use the CCKM web interface.

Intended audience

This guide is for the system administrator who will install and configure the CCKM. This role is the “CCKM Admin.” This guide describes how to use the CCKM GUI for managing Azure, Azure China, Azure Germany, Azure Stack, Google Cloud, IBM, Salesforce, Salesforce Sandbox, and/or AWS Key Management Service (KMS) keys. These users, who must have access credentials to their organizations’ Azure, Google Cloud, IBM, Salesforce, or AWS accounts, are called “Cloud administrators.”

Service updates and support information

The license agreement that you have into to acquire the Thales products (“License Agreement”) defines software updates and upgrades, support and services, and governs the term under which they are provided. Any statements made in this guide or collateral documents that conflict with the definitions or terms in the License Agreement shall be superseded by the definitions and terms of the License Agreement. Any references made to “upgrades” in this guide or collateral documentation can either apply to a software update or upgrade.

For support and troubleshooting issues:

- https://supportportal.thalesgroup.com
- (800) 545-6608

For Thales Sales:

- https://enterprise-encryption.vormetric.com/contact-sales.html
- sales@thalesgroup.com
- (877) 267-3247

Chapter 1: Overview

The CipherTrust Cloud Key Manager (CCKM) centralizes the management of the life cycle of cloud keys in multi-cloud, hybrid and enterprise environments. CCKM addresses the needs of these environments for encrypting data in the cloud while retaining custodianship of encryption keys to comply with data security mandates in cloud storage environments. For the clouds (or cloud services) CCKM supports, CCKM enables enterprises who use these services to bring their own keys to those clouds, back up keys on premise, destroy cloud keys in specific situations, and manage the life cycle of cloud keys. CCKM provides a web GUI for all functions.

CCKM Components

The CCKM solution comprises of the following components:

- CCKM with Web UI pages including an administrator portal (for CCKM administrators) and a user portal (for cloud administrators or cloud key administrative users)
- At least one of the supported clouds (you can use all clouds)
- A supported trusted key source
- MongoDB or MongoDB Atlas as a database
- A supported Internet browser

The product is delivered as a virtual appliance that can be installed on any one of the supported deployment methods.

Supported Clouds

CCKM supports the following clouds:

- Amazon Web Services (AWS)
- Google Cloud
- IBM
- Microsoft Azure (Azure, Azure China, Azure Germany, and Azure Stack)
- Salesforce (Salesforce and Salesforce Sandbox)

Supported Key Sources

CCKM uses trusted key sources for the encryption keys employed within the supported clouds. Certain trusted key sources also store the clouds’ backup keys. Currently, the supported key sources include:

- Vormetric Data Security Manager (DSM): The DSM supports all of the clouds that CCKM supports. The DSM keys and the backup keys from the supported clouds are stored in DSM.
- CipherTrust Manager: CipherTrust Manager supports all clouds that CCKM supports. The CipherTrust Manager keys and the backup keys from the supported clouds are stored in CipherTrust Manager.

> **Notes**
>
> SafeNet Next Generation KeySecure has been rebranded as CipherTrust Manager. Throughout this guide, CipherTrust Manager is employed to reflect the new name of this product.

- nShield Connect hardware security module (HSM): nShield Connect supports the AWS and Azure clouds. The nShield Connect keys and the backup keys from the clouds are stored in MongoDB, the supported CCKM database.

> **Notes**
>
> Concurrent use of the CipherTrust Manager, DSM, and nShield as key sources is not supported. Configure only ***one*** of these supported key sources as a key source.

Supported Database

CCKM employs MongoDB or MongoDB Atlas, a global cloud database service, to store key metadata, and CCKM system and user configuration information. Encrypted communication using TLS between CCKM and MongoDB or MongoDB Atlas is available. Furthermore, CCKM supports MongoDB Atlas on Google Cloud, as well as on Microsoft Azure.

> **Notes**
>
> - For the supported versions of MongoDB or MongoDB Atlas, see Chapter 4: "How it works: Installation and Configuration Flow" on page 32.
> - The MongoDB installation file includes a community (free) edition of MongoDB that can be used for evaluation and proof-of-concept (PoC) purposes. For a production setup, an external MongoDB in replica set configuration or a MongoDB Atlas configuration is recommended.

Supported Deployment Methods

CCKM is delivered as a virtual appliance that can be installed in one of the following methods:

- On-premises by deploying an .ova file
- In Amazon Web Services by deploying an Amazon Machine Image (AMI)
- By instantiating CCKM from the Azure or Azure Stack Market places by deploying a Virtual Hard Disk (VHD) image
- In a Microsoft Hyper-V environment by deploying a VHD image

Supported Internet Browsers

The CCKM supports the following Internet browsers:

- Chrome 51.0.2704 (64-bit) or later
- Firefox 45.0 or later
- Internet Explorer 11 or later

CCKM Functionality

CCKM offers similar functionality for each of the supported clouds within the supported deployment scenarios:

- **Life cycle management** of keys, key versions, and attributes:
  - View Keys
  - Update Keys
  - Upload Keys
  - Rotate Keys
  - Delete Keys
- **Disaster recovery** of keys
  - Backup Keys
  - Restore Keys
- **Hybrid key management**
  - On-premise Keys storage
  - Management of both keys originating from trusted key sources and cloud-provider-sourced keys
  - Key synchronization
- **Compliance Management**
  - Key storage within on premise FIPS 140-2 Level 3 certification
  - Key storage in public or private clouds, inaccessible to clouds with FIPS 140-2 Level 1
- **Certificate handling (Salesforce)**
  - View certificates
  - Synchronize certificates
  - Use certificates to wrap keys for secure key upload
- **Key visibility reporting**
  - Combined Key Activity Reconciliation Report

    Track and reconcile key activities between CCKM and a cloud

    > **Note:** This report is not available for Azure Stack.

  - Key Activity Report

    Insight into who, what and when keys were accessed, created, deleted across PaaS

    > **Note:** This report is not available for Azure Stack.

  - Key Aging Report

    Insight into who owns the keys and when keys expire
  - Key Service Usage Report

    Insight into applications consuming the keys across IaaS/PaaS/SaaS

    > **Note:** This report is not available for Azure Stack, Salesforce, and Salesforce Sandbox.

  - Cloud Key Manager User Action Report

Insight into aggregated user activities, such as delete or create keys

- Connection to a Remote Syslog Server for forwarding of CCKM activity logging. Syslog messages may also be directed to a SIEM system that accepts Syslog message.

Chapter 2: VM Specifications for CCKM Components

This chapter provides the minimum requirements for the CCKM and MongoDB virtual machines (VMs) when CCKM is running bundled with a local MongoDB or with external MongoDB node(s) or MongoDB Atlas.

With Local MongoDB bundled with CCKM Appliance (Recommended for PoCs Only)

PoC – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 4 GB Memory
- 100 GB Hard disk (Thin)

Development/Staging – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 4 GB Memory
- 100 GB Hard disk (Thin)

Production – Hardware Recommendation

CCKM VM minimum requirements:

- 4 CPUs
- 8 GB Memory (Use 16 GB if API driven high frequency workflows are needed)
- 250 GB Hard disk (Thin)

With External MongoDB (Recommended for Production)

PoC – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 4 GB Memory (Use 8 GB if API driven high frequency workflows are needed to be tested)
- 100 GB Hard disk (Thin)

MongoDB VM minimum requirements:

- 2 CPUs
- 4 GB Memory (Use 8 GB if API driven high frequency workflows are needed to be tested)
- 250 GB Hard disk (Thin)

Development/Staging – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 4 GB Memory (Use 8 GB if API driven high frequency workflows are needed to be tested)
- 100 GB Hard disk (Thin)

MongoDB VM minimum requirements:

- 2 CPUs
- 4 GB Memory (Use 8 GB if API driven high frequency workflows are needed to be tested)
- 250 GB Hard disk (Thin)

Production – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 8 GB Memory (Use 16 GB, if API driven high frequency workflows are needed)
- 100 GB Hard disk (Thin)

External MongoDB VM minimum requirements for each node:

- 4 CPUs
- 8 GB Memory (Use 16 GB, if API driven high frequency workflows are needed)
- 250 GB Hard disk (Thin)

With MongoDB Atlas (Recommended for Production)

PoC – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 4 GB Memory
- 100 GB Hard disk (Thin)

Development/Staging – Hardware Recommendation

CCKM VM minimum requirements:

- 2 CPUs
- 4 GB Memory
- 100 GB Hard disk (Thin)

Production – Hardware Recommendation

CCKM VM minimum requirements:

- 4 CPUs
- 8 GB Memory (Use 16 GB if API driven high frequency workflows are needed)
- 250 GB Hard disk (Thin)

Chapter 3: About CCKM Admin and User Portals

CCKM provides Web UI pages for a CCKM Admin Portal for CCKM administrators and a CCKM User Portal for CCKM users (also known as cloud administrators). The chapter provides information about these portals, administration privileges, support for multi-accounts per cloud, and the handling of user (cloud administrator) login.

For information about the available menu options used to configure and manage the CCKM from the CCKM Admin Portal and, see Chapter 12: "About the Admin Portal" on page 66.

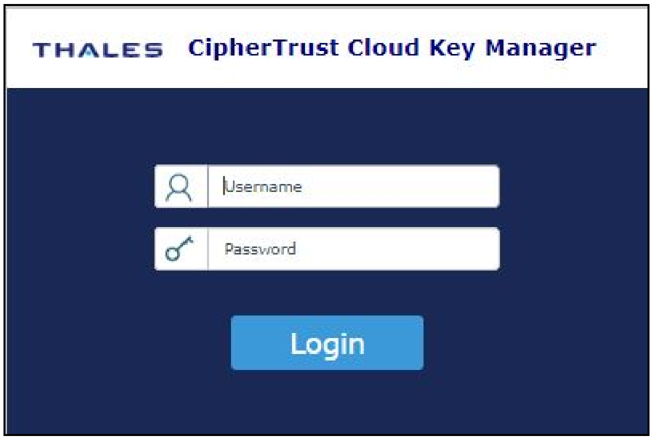
CCKM Administration Privileges

From the CCKM Admin Portal, a CCKM administrator can add and configure clouds, monitor CCKM logs to review user activities in CCKM, upload your CCKM license(s) and view all of the licenses for the configured clouds, and manage and configure CCKM settings. Unless the CCKM administrator also has user (cloud administrator) login credentials for Azure, AWS Key Management, Google Cloud, IBM or Salesforce, he or she will not be able to access the CCKM User Portal pages.

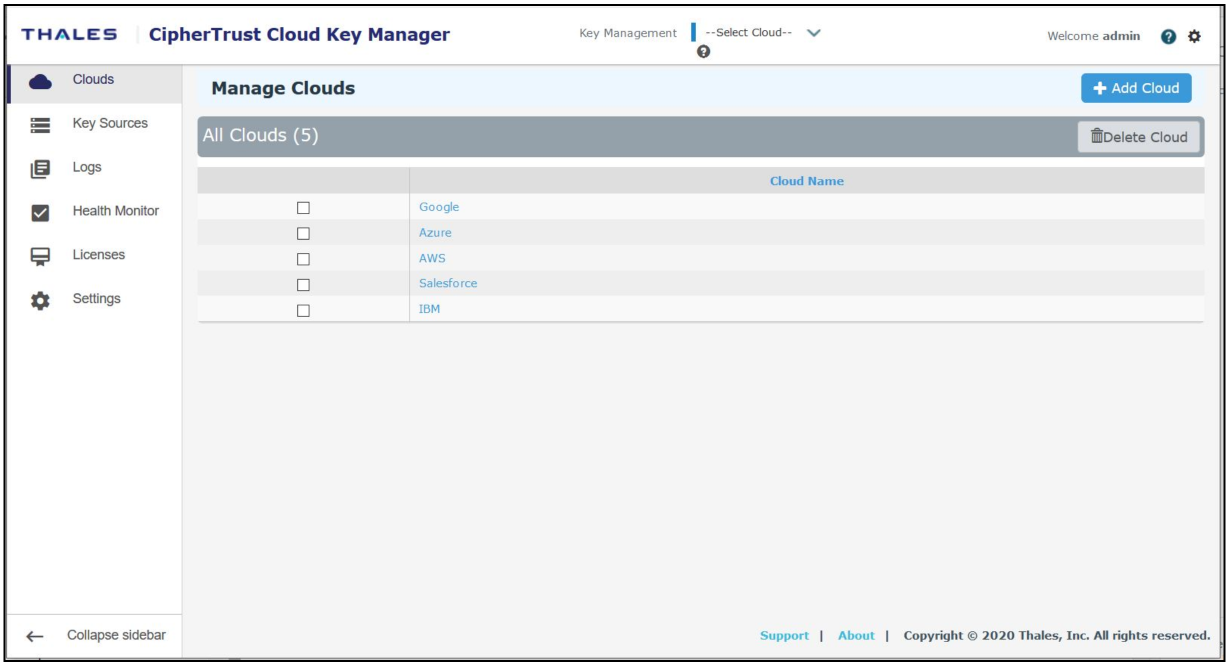
To access the CCKM User Portal, a CCKM user must have valid login credentials for the organization’s designated clouds. No specific “CCKM administrator” is created. Administrators have access from the CCKM User Portal to all clouds for which they have *valid* credentials.

Accessing CCKM Admin Portal (as CCKM Admin)

1. Enter “https://<IP address of CCKM instance>” in a browser. The CCKM Admin Login page is displayed:



2. Log in as a CCKM administrator using your administrator credentials. The CCKM Admin Portal is displayed:



Manage Clouds

All Clouds (5)

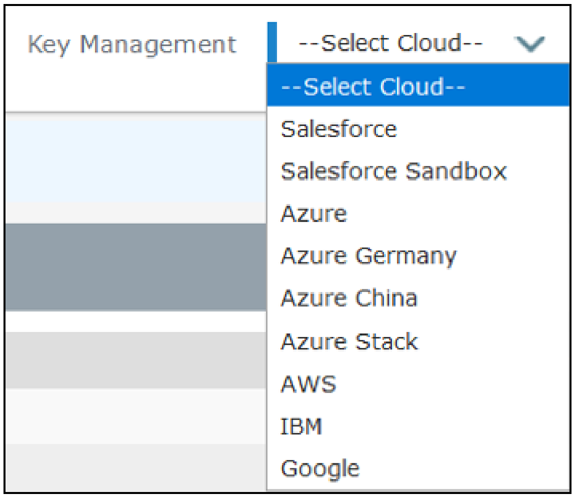
	Cloud Name
□	Google
□	Azure
□	AWS
□	Salesforce
□	IBM

Support | About | Copyright © 2020 Thales, Inc. All rights reserved.

Accessing CCKM User Portal from CCKM Admin Portal

From the CCKM Admin Portal, you can access the CCKM User Portal and CCKM Home page. However, you are required to enter the user login credentials for the given Key Management service to access the CCKM User Portal.

From the CCKM Admin Portal, select a configured cloud from the Key Management drop-down list at the top of the page.



The login page for the selected Key Management service is displayed.

Azure

The Microsoft login page opens at https://login.microsoftonline.com/

Enter your Microsoft user login credentials or click on the saved user name. The Microsoft Login page is displayed. Enter your Microsoft password and click **Log In**.

AWS

If you select **AWS** as the Key Management service, then the CCKM user login page is displayed. From this page, select the AWS key management service and the login page for AWS displays. Click **Log into AWS**. The AWS Password Login page is displayed. Enter your AWS key credentials in the **AWS Access Key ID** and **AWS Secret Access Key** boxes and click **Log In**.

Google

If you select **Google** as the Key Management service, then the CCKM user login page is displayed. From this page, select the Google key management service. You are redirected to https://accounts.google.com/. Enter your Google user login credentials (or click on the saved user name) and then enter your Google password. Click **Log In**. At the consent page, click **Allow** to proceed with the login.

IBM

If you select **IBM** as the Key Management service, then the CCKM user login page is displayed. From this page, select the IBM key management service and the login page for IBM displays. Click **Log into IBM**. The IBM Password Login page is displayed. Enter your IBM key credentials in the **IBM Username** and **IBM API Key** boxes and click **Log In**. You can also upload your IBM key credentials file under the **Load Credentials from IBM credentials file** by choosing the file and then clicking **Log In**.

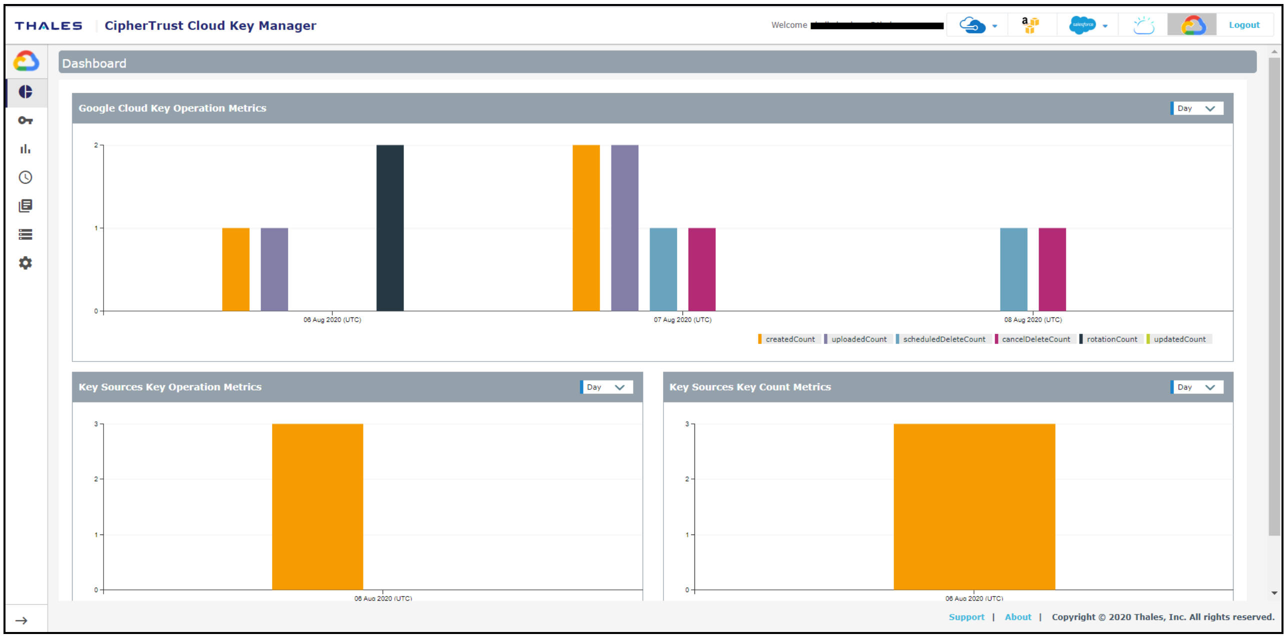
Salesforce

The Salesforce login page opens at https://login.salesforce.com/.

The Salesforce Sandbox login page opens at https://test.salesforce.com/

Enter your Salesforce credentials or click on the saved user name. The Salesforce Login page is displayed. Enter your Salesforce password and click **Log In**.

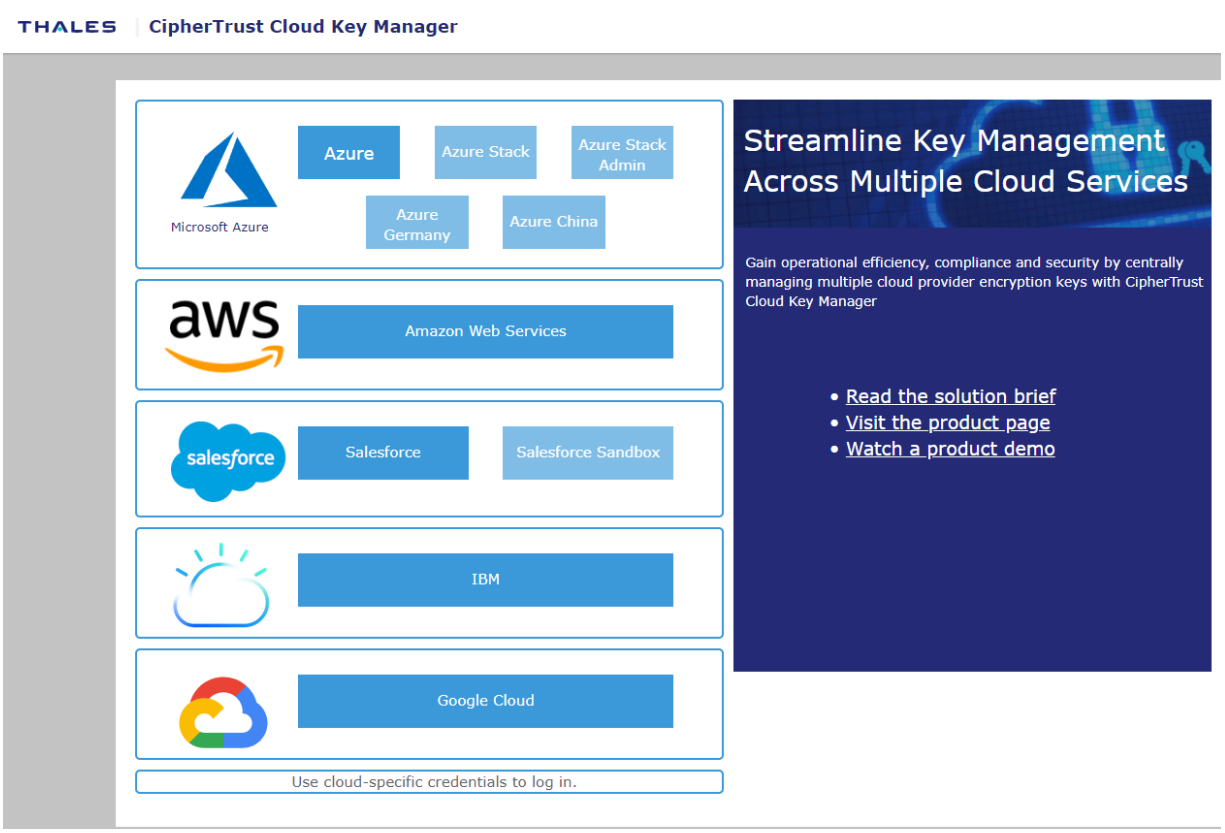
> **Note**
> The first time you access the CCKM, the keys are synchronized and a progress page is displayed. Once synchronization is complete, click **Continue**. The **CCKM Home** page is displayed within the CCKM User Portal.



Accessing CCKM User Portal (as CCKM User)

To access the CCKM User Portal and CCKM Home page as a CCKM user, do the following:

1. Enter “https://<CCKM IP address>:8443/kmaas” in a browser. The CCKM user login page is displayed.



Streamline Key Management Across Multiple Cloud Services

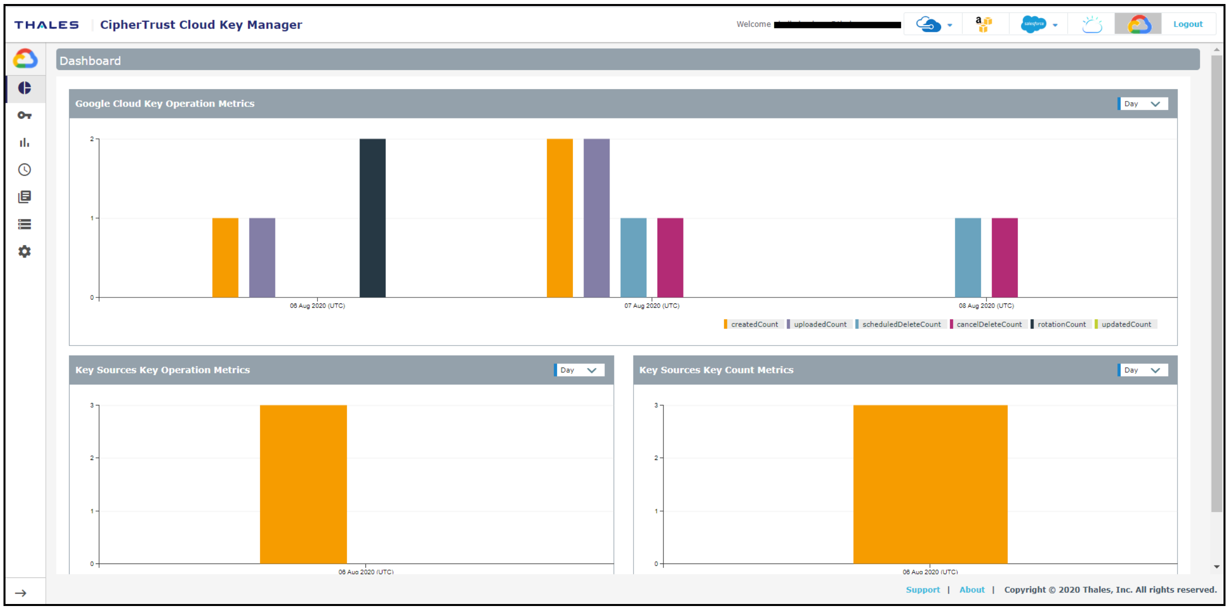
Gain operational efficiency, compliance and security by centrally managing multiple cloud provider encryption keys with CipherTrust Cloud Key Manager

- Read the solution brief
- Visit the product page
- Watch a product demo

Use cloud-specific credentials to log in.

2. Select the installed Key Management service you wish to use. The login page for the selected Key Management service is displayed.

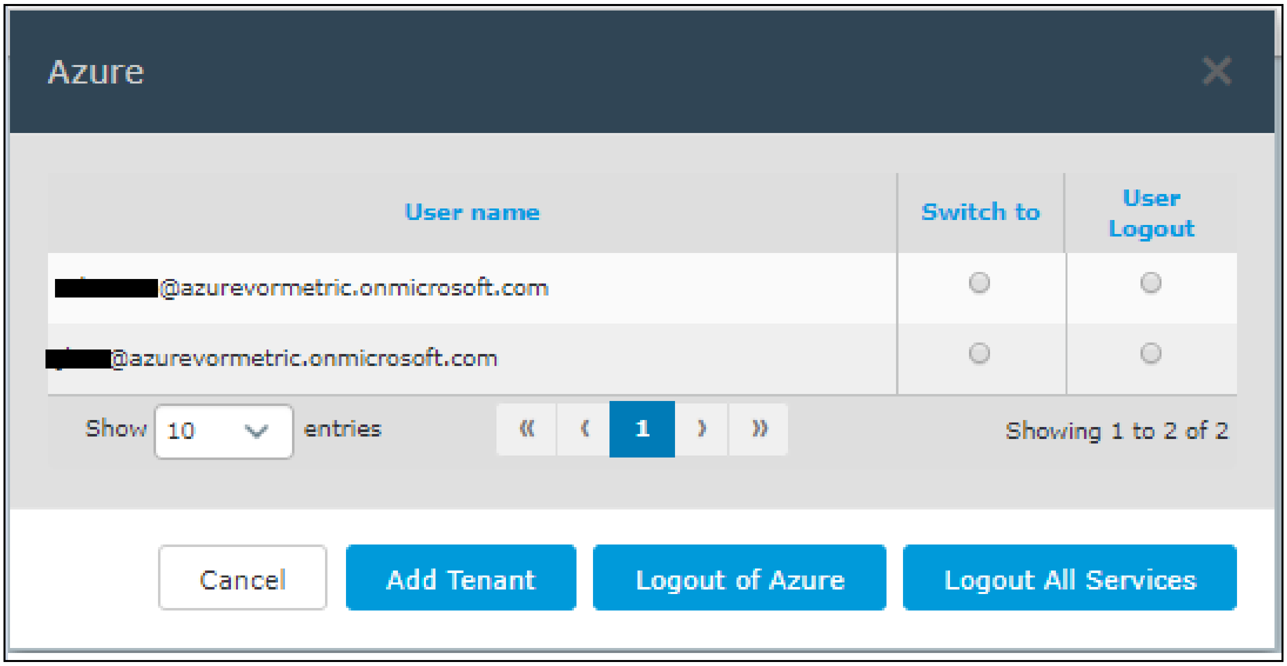
3. Enter your user login credentials for the given Key Management service to log into CCKM. The **CCKM Home** page is displayed within the CCKM User Portal.



Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI

CCKM supports login and key management for multiple accounts or multiple users per cloud through a single CCKM UI (Multi-account Login/Logout) and browser window.

The following **Multi-account Login/Logout** window reflects the Azure account (varies depending on the cloud account into which you are currently logged):



User name	Switch to	User Logout
[redacted]@azurevormetric.onmicrosoft.com	○	○
[redacted]@azurevormetric.onmicrosoft.com	○	○

Show 10 entries

Showing 1 to 2 of 2

You can perform the following using the **Multi-account Login/Logout** window.

- Logon to multiple accounts for a specific cloud (AWS/Azure/Google/IBM/Salesforce)
- Logout of specific or all accounts within one cloud or all clouds
- Perform key management for multiple accounts from your current CCKM browser window

Accessing the Multi-account Login/Logout Window

To access the **Multi-account Login/Logout** window for a specific cloud, you must have at least one user already logged on to an account. Once you are logged onto an account, from the service **Dashboard**, click on the cloud icon (Context Switcher) at the top of right of the page. The **Multi-account Login/Logout** window displays. The following sections provide instructions on how to use this window.

Switching to Another User Account (Already Logged On)

(Applicable for a given cloud on to which you are currently logged) To access another user account, which is already logged on, select the **Switch** to button next to the user name (account) for which to log on. You are then switched to that account with the user name displayed at the top of the **CCKM Home** page.

Logging on to Another User Account

(Applicable for a given cloud on to which you are currently logged) To log on to another user account, click one of following buttons depending on the service you are currently logged and then enter the user credentials in the login dialog boxes of the cloud:

- **Add Tenant** (applicable to Azure, Azure China, Azure Germany, and Azure Stack)
- **Add Organization** (applicable to Google Cloud)
- **Add OrgID** (applicable to Salesforce and Salesforce Sandbox)
- **Add Account** (applicable to AWS and IBM)

Once the user authentication is completed, you are logged on to the given user account and the username is displayed at the top of the CCKM Home page.

Logging Out of a User Account

(Applicable for a given cloud on to which you are currently logged) To logout of a specific user account, select the **User Logout** button next to the user name (account) for which to log out. The account is then logged out.

Logging Out of All User Accounts from a Specific Cloud Service or All Cloud Services

To logout of all of the user accounts for a specific cloud, click one of the following depending on the service to which you are currently logged:

- **Logout of Azure (Logout of Azure Germany, Logout of Azure China, or Logout of Azure Stack)**
- **Logout of AWS**
- **Logout of Google**
- **Logout of IBM**
- **Logout of Salesforce (Logout of Salesforce Sandbox)**

To logout of all clouds, click the **Logout All Services** button. All services are then logged out.

User Handling within CCKM User Portal (User Login Info)

This section describes how CCKM handles user authentication, user authorization, and user audits within the CCKM User Portal for each the clouds.

User Authentication

For AWS users, the user provides their AWS long-term credentials (for example, access key ID and secret access key) to CCKM. CCKM uses these credentials to obtain a one-time, temporary credential for the user session. *CCKM does not save a user's long-term credentials*.

For Azure, Google Cloud, and Salesforce users, the CCKM itself does not maintain user credentials. It uses OAuth2 to authenticate users.

During login, users are redirected to the Azure, Google Cloud, or Salesforce login page (depending on the cloud you accessing) to enter their respective username/password combinations. Once authenticated, the CCKM User Portal obtains an OAuth2 access token from the cloud and then maintains a user session between CCKM and the user’s browser. While the user session is active, CCKM uses the access token to call the given cloud's REST APIs, and will allow the logged-in user to access and manage keys within the given cloud per the granted permission.

For IBM users, the user provides their IBM long-term credentials (for example, IBM Username and API Key) to CCKM. CCKM uses these credentials to obtain a one-time, temporary credential for the user session. CCKM does not save a user's long-term credentials.

User Authorization

In AWS, user authorization defines what a user can do with AWS KMS. For a user to manage AWS CMKs, he or she must have proper KMS permissions. In addition, to use CCKM, the user should have IAM GetUser permission.

In Azure, user authorization defines what a user can do on Azure keys. For example, whether they can view, upload, update, delete, or restore keys in a key vault. This is managed within Azure Key Vault’s Access Control (IAM) and Access Policies. To allow a user to manage keys in a key vault, the key vault owner must give the user “Key Vault Contributor” role in Access Control, and appropriate permissions as needed on keys in **Access Policies**.

In Google Cloud, user authorization defines what a user can do on Google Cloud keys. For example, whether they can view, upload, update, rotate, or schedule destroy keys in Google Cloud KMS. This is managed within Google Cloud's Identity and Access Management (IAM). You will need to have at least the Viewer role on the Key Rings that you wish to manage and the appropriate permissions as needed on the keys and key versions. For more information, see Chapter 15: " CCKM for Google Key Management" on page 134.

In IBM, user authorization defines what a user can do within IBM Key Protect. For a user to manage an IBM Key Protect instance, he or she must have proper IBM Key Protect permissions. In addition, to use CCKM, the user should have IAM GetUser permission and keys level permission.

In Salesforce, user authorization defines whether the user has “Manage Encryption Keys” permission or not. Only users who have **Manage Encryption Keys** permission can manage Tenant Secrets.

User Audit

User operations (within the CCKM User Portal), including login and logout (successful or failed) are captured in CCKM audit logs. Currently, there are no access policies on CCKM audit logs and so any logged-in user for a specific organization and service can view all audit logs. These audit logs include operations made by other users in the past, within the scope of that organization.

Chapter 4: How it works: Installation and Configuration Flow

This chapter provides a summary of the steps to take to install and configure CCKM solution (CCKM and its components).

Installation and Configuration Flow

Use the following table to guide you in installing and configuring the CCKM solution. Included in the table is a column providing the sections to which to refer for more information about a given step.

Step #	Step description	Summary	Refer to these sections for more information
Step 1	**Install and configure MongoDB**		For more information, see Chapter 5: "Set Up MongoDB" on page 35.
		Choose one of the following and perform the pre-setup steps within MongoDB: • **Embedded (free) MongoDB**No pre-setup required • **External MongoDB (single or replica) Supported versions are 3.4.10, and later**. **Note:** This version of CCKM was tested with versions **3.4.10, 4.0.10**, **4.2.1**, and **4.2.3** of MongoDB. • **MongoDB Atlas Supported versions are 4.0.13, 4.2.1, 4.2.3, and 4.4.0.** **Note:** CCKM communicates with MongoDB through port 27017.	
Step 2	**Install and configure Key Source**		For more information, see Chapter 6: "Install and Configure Key Source" on page 39.

Step #	Step description	Summary	Refer to these sections for more information
	Install and configure CipherTrust Manager as your key source	If you plan to use the CipherTrust Manager as your key source, you must launch and configure it prior to configuring CCKM to integrate with the CipherTrust Manager. Use version **1.9.1.4265** or later.	For more information, see "Install and Configure CipherTrust Manager as Key Source " on page 39.
	Install and configure DSM as your key source	If you plan to use the DSM as your key source, then you must install and configure it prior to configuring the CCKM to integrate with the DSM. Install version **6.4.0.15031** or later. **Note:** CCKM communicates with the DSM through ports 443 and 8445.	For more for more information, see "Install and Configure DSM as Key Source" on page 40.
	Install and configure nShield as your key source	nShield Connect supports the AWS and Azure (Azure, Azure China, Azure Germany, and Azure Stack) clouds. Install version **12.40.2** or later of nShield Connect. With the use of nShield Connect, the keys are stored as blobs within the MongoDB. **Note:** CCKM communicates with nShield Connect through port 9004.	For more information, see "Install and Configure nShield Connect as Key Source " on page 40.
Step 3	**Setup to connect to cloud(s)**		

Step #	Step description	Summary	Refer to these sections for more information
	If using Azure, create an app as a doorway between CCKM and cloud(s)		For more information about Azure, Azure China and Azure Germany, see Chapter 7: "Register CCKM App in Azure Cloud" on page 42. For more information about creating an app for Azure Stack with AAD, see "Creating CCKM App in Azure Stack Cloud with Azure AD" on page 44. For more information about configuring Azure Stack using ADFS, see "Creating Service Principals for Azure Stack Cloud with ADFS " on page 47.
	If using AWS, there is no requirement to create an app in AWS to connect to CCKM.	AWS users should have: • Access key ID and secret access key • IAM GetUser permission for self (you as a user) • Proper KMS permissions	
	If using Google Cloud, perform all of the prerequisites to set up to connect to CCKM.		For more information about all of the prerequisites, see "Prerequisites on Google Cloud" on page 138.
	If using IBM, there is no requirement to create an app in IBM to connect to CCKM.	IBM cloud users should have: • IBM Username and API Key • IAM GetUser permission for self (you as a user) • Proper IBM Key Protect permissions • Proper IBM Cloud Activity Tracker and LogDNA permissions	
	If using Salesforce or Salesforce Sandbox, create a connected app in Salesforce		For more information about creating an app for Salesforce and Salesforce Sandbox, see Chapter 9: "Create a Connected App in Salesforce" on page 54.
Step 4	**Install CCKM in one of the supported deployment methods**		For more information, see Chapter 10: "Install CCKM" on page 55.
Step 5	**Perform required steps for the initial configuration of CCKM Admin Portal**		For more information, see Chapter 11: "Initial Configurations within CCKM Admin Portal " on page 63.

Chapter 5: Set Up MongoDB

The CCKM uses MongoDB to store key metadata, and CCKM system and user configuration information. The supported MongoDB configurations are described in this chapter.

> **Notes**
>
> - CCKM supports both MongoDB and MongoDB Atlas.
> - Be sure to back up your MongoDB database(s) on a regular basis. For more information on how to back up MongoDB, see "Backup and Restore for MongoDB" on page 210.
> - Although the CCKM includes a version of MongoDB, the software is supported by MongoDB. Contact the MongoDB Support team with any inquiries or issues.

Scenario 1: Use Built-in MongoDB for Evaluation and PoC Purposes

No pre-setup of MongoDB is required for the use of the built-in MongoDB within CCKM. If you are using the built-in MongoDB database server within CCKM (for evaluation and PoC purposes only), select to use the local database option from the **Database** tab within the **Settings** page of the CCKM Admin Portal.

Scenario 2: Configure an Existing (External) MongoDB (Single Node or Replica Set)

To configure an existing external MongoDB instance (single node or replica set), do the following:

- (Optional) Enable TLS for encrypted communication between MongoDB and CCKM
- (Required) Create one MongoDB database named *kmaas* and the user named kmaas_user along with the user passwords and database roles.

Each of these configurations is described in this section.

(Optional) Enable TLS

If you choose to enable TLS in the MongoDB server, add TLS-related parameters to the mongod.conf file. Here is a sample of the TLS-related parameters to add to the file:

```
tls:
   mode: requireTLS
   certificateKeyFile: /path to MongoDB .pem file
   FIPSMode: true
   disabledProtocols: TLS1_0,TLS1_1

security:
   keyFile:  /path to keyFile for internal membership authentication
   authorization: enabled
```

For the mongoDB replica nodes, add these TLS-related parameters to each of the associated mongod.conf files.

Create a Database (*kmaas) and User (kmaas_user)*

> **Note**
> If the MongoDB server is TLS-enabled, then ensure that the MongoDB client has the root certificate authority file of MongoDB server as this file is required to perform the TLS handshake between the MongoDB server and CCKM.

In this section, create a database named kmaas and the MongoDB user of kmaas_user (along with the associated password and roles). CCKM uses the username and password to access the database.

To create a database and user, do the following:

1. Access the MongoDB server using the MongoDB client.

2. Create the kmaas database and a new user (kmaas_user) for this database by running the following commands (replacing <mypassword> with the password you choose):

```
use kmaas
db.createUser( { user: "kmaas_user", pwd: "<mypassword>", roles: [ "dbOwner" ] } )
```

Scenario 3: Configure MongoDB Atlas

Before configuring MongoDB Atlas within CCKM, configure MongoDB Atlas. Perform these configurations as described in this section:

- Whitelist Devices Allowed to Connect to MongoDB Atlas
- *(Optional)* Add MongoDB Role (kmaas_role)
- Add MongoDB User (kmaas_user)
- Create a New Cluster

If using a MongoDB Atlas cluster, you will also be required to obtain a connection string from MongoDB Atlas that is used to connect CCKM to your MongoDB Atlas cluster using a MongoDB driver.

The configuration steps in this section highlight the *optional* and *required* configurations for MongoDB Atlas to communicate with CCKM. It is assumed you are familiar with configuring MongoDB Atlas. For detailed information on how to configure these steps in MongoDB Atlas, refer to the MongoDB Atlas documentation.

Whitelist Devices Allowed to Connect to MongoDB Atlas

To whitelist the IP addresses of the devices allowed to connect to MongoDB Atlas, do the following:

1. In **SECURITY** in the left navigation bar, select **Network Access**.

2. Within **Network Access**, click **ADD IP ADDRESS** to add an IP address to the whitelist. The **Add Whitelist Entry** dialog box displays. In the **Whitelist Entry** box, enter the IP address.

3. Click **Confirm**.

4. Repeat steps 2 through 3 for each device to the whitelist.

(Optional) Add MongoDB (Custom) Role (kmaas_role)

Adding the customized MongoDB user role of kmaas_role to MongoDB Atlas is optional. When adding your MongoDB users to MongoDB Atlas, you can associate your MongoDB user to these custom roles. However, you can also associate the users with the pre-defined role of Read and write to any database. This role allows a user associated with this role to read and write to any database including the CCKM databases. If CCKM is the only application you are using with MongoDB Atlas, then assigning the Read and write to any database role to your users is sufficient. However, if you are using more than one application with MongoDB Atlas, then associating customized roles to your users is recommended.

To add the MongoDB user role of kmaas_role to MongoDB Atlas and associate actions and roles with a database, do the following:

1. In **SECURITY** in the left navigation bar, select **Database Access**.
2. In **MongoDB Roles**, click **ADD NEW CUSTOM ROLE**. The **Add Custom Role** dialog box displays.
3. In the **Custom Role Name** box, enter kmaas_role.
4. In the **Action** or **Role** box, select **Database Actions and Roles**.
5. In the **Database** box, enter kmaas as the name of the database to which to associate this role.
6. Click **Add Custom Role**.

Add MongoDB User (kmaas_user)

Perform one or the other of these configurations depending on which role you are assigning to your users.

To add the MongoDB user of kmaas_user to MongoDB Atlas and associate this user with the pre-defined role of **Read and write to any database**, do the following:

1. In **SECURITY** in the left navigation bar, select **Database Access**.
2. In **MongoDB Users**, click **ADD NEW USER**. The **Add New User** dialog box displays.
3. In the **Username** box, enter kmaas_user.
4. In the **Password** box, enter the password for the kmaas_user user.
5. In **User Privileges**, click **Read and write to any database** to associate this user with this role.
6. Click **Add User**.

To add the MongoDB user of kmaas_user to MongoDB Atlas and associate this user with the customized role of kmaas_role, do the following:

1. In SECURITY in the left navigation bar, select **Database Access**.
2. In **MongoDB Users**, click **ADD NEW USER**. The **Add New User** dialog box displays.
3. In the **Username** box, enter kmaas_user.
4. In the **Password** box, enter the password for the kmaas_user user.
5. In **User Privileges**, click **Select Custom Role** and select kmaas_role to associate this user with this role.
6. Click **Add User**.

Create a New Cluster

In this step, you will create a new cluster in MongoDB Atlas for the CCKM application. Configuring a new cluster includes setting global cluster configuration, cloud provider and region, cluster tier, and cluster name.

After you create a new cluster for CCKM, one database set will be added to MongoDB Atlas. Thereafter, you are required to initialize the kmaas database by adding the (admin) user document to this database.

To create a new cluster in MongoDB Atlas for CCKM, do the following:

1. Under **ATLAS** in the left navigation bar, select **Clusters**.
2. In **Clusters**, click **Build a New Cluster**. The **Create a New Cluster** page displays.
3. In **Cloud Provider & Region**, select the provider and region you wish to use for the cluster.
4. In **Cluster Tier**, select the tier you wish to use for the cluster.
5. In **Additional Settings > for MongoDB 4.0, Backup > Select a Version**, select MongoDB 4.0.
6. (Optional. The display of this setting depends on your account settings.) In the **Turn on Backup (M2 and up)**, set this setting to **YES** and select **Continuous**.
7. In **More Configuration Options > Set Minimum TLS Protocol Version**, select TLS 1.2 and above.
8. Within **Cluster Name** box, enter the name of the new cluster.
9. Click **Create Cluster**.

Obtain Connection String for Connection Using a MongoDB Driver

In this step, you will obtain a connection string from MongoDB Atlas that is used to connect CCKM to your MongoDB Atlas cluster using a MongoDB driver. The connection string is made available to you based on the selected driver type and version. Copy this string and save it to a directory within your laptop or desktop. You will enter this connection string when configuring MongoDB Atlas settings in the Settings page > MongoDB tab of CCKM Admin Portal.

To obtain a collection string using a MongoDB driver, do the following:

1. In MongoDB Atlas, under **ATLAS** in the left navigation bar, select **Clusters**.
2. Find the cluster you created and click **Connect**.
3. In the **Choose a connection method**, select **Connect your application**.
4. In the **Choose your driver version**, select your driver and version from the dropdown menus.
   a. Under **DRIVER**, select **Java**.
   b. Under **VERSION**, select **3.7** or *later*.
5. In the **Add your connection string into your application code**, the **Connection String Only** tab displays the connection string to use to connect to your MongoDB Atlas cluster from CCKM. Click **Copy** to copy the connection string.
6. Save the connection string to a directory within your laptop or desktop.

Chapter 6: Install and Configure Key Source

This chapter provides information at a high-level about the installation and configuration of the key source you plan to use with the CCKM ***prior*** to installing and configuring CCKM. As part of these steps, you will *gather* information that is required to be entered in CCKM as part of the key source configuration. The following are the supported key sources:

- CipherTrust Manager
- DSM
- nShield Connect HSM

> **Note**
> Concurrent use of the CipherTrust Manager, DSM, and nShield as key sources is not supported. Configure only ***one*** of these supported key sources as a key source.
>
> SafeNet Next Generation KeySecure has been rebranded as CipherTrust Manager. Throughout this guide, CipherTrust Manager is employed to reflect the new name of this product.

Install and Configure CipherTrust Manager as Key Source

> **Note**
> This documentation assumes that you are familiar with using the NextGen CipherTrust Manager. For information on how to launch and configure CipherTrust Manager, refer to the CipherTrust Manager Deployment Guide and CipherTrust Manager Administrator Guide (version 1.9).

If you plan to use CipherTrust Manager as your key source, install and configure it *prior* to installing and configuring CCKM.

> **Note**
> You will enter the CipherTrust Manager account information you gather during the installation and configuration of CipherTrust Manager in CCKM (described in this section) when configuring this key source in the **CipherTrust Manager** tab of the **Key Sources** page of the CCKM Admin Portal as part of the initial CCKM configuration. (This initial configuration of CCKM takes place after you install CCKM and login into the CCKM Admin Portal for the first time.)

For any CipherTrust Manager you plan to use as key source for CCKM, the CipherTrust Manager administrator must perform the following:

1. Launch and configure CipherTrust Manager.
2. In CipherTrust Manager, create a CipherTrust Manager user or a LDAP user that has domain and system administrator privileges. Note the user name and password for this user.
3. Note the hostname or the IP address of the CipherTrust Manager during the configuration of CipherTrust Manager. This information is required when you configure the CipherTrust Manager settings in the **CipherTrust Manager** tab of the **Key Sources** page

4. Note the Connection String that will be used to connect CCKM to CipherTrust Manager. By default, the Connection String is “local_account”. If an LDAP user is selected, the connection string is the LDAP domain name.

5. Download the root CA certificate (CA) (in a Base64 format) of the CipherTrust Manager from the CipherTrust Manager server.

> **Note:** The common name of the CipherTrust Manager certificate and the hostname of the CipherTrust Manager must match. Otherwise, when you proceed to test the connection between the CCKM and CipherTrust Manager (when configuring CipherTrust Manager in the **CipherTrust Manager** tab within **Key Sources** page in the CCKM Admin Portal), the connection will fail, and an error message will display.

Install and Configure DSM as Key Source

If you plan to use DSM as your key source, install and configure it prior to installing and configuring CCKM.

> **Note**
> You will enter the DSM account information you gather during the installation and configuration of DSM in CCKM (described in this section) when configuring this key source in the **DSM** tab of the **Key Sources** page of the CCKM Admin Portal as part of the initial CCKM configuration. (This initial configuration of CCKM takes place after you install CCKM and login into the CCKM Admin Portal for the first time.)

For any DSM you plan to use as key source for CCKM, a ***DSM administrator*** must perform the following:

1. Install and configure DSM. Refer to the latest version of the DSM Administration Guide.

2. When configuring DSM, create a DSM administrator account of type “All Admin” for CCKM.

3. Note the following information (as you will enter this information when configuring the DSM key source in the CCKM Admin Portal:

   - DSM fully-qualified host name
   - DSM CCKM administrator name and password for the “All Admin” account.

4. Download the DSM server certificate. For information on how to do this, see Appendix B: "How to Export Your DSM Server Certificate from DSM" on page 218.

Install and Configure nShield Connect as Key Source

> **Note**
> This documentation assumes that you are familiar with using the nShield Connect HSM. This documentation also assumes that you have installed nShield Connect, and the associated RFS and Security World. For information on how to install an nShield Connect, and the associated RFS and Security World, refer to the nShield® Connect Installation Guide.

If you plan to use nShield Connect as your key source, install and configure it *prior* to installing and configuring CCKM.

> **Note**
> You will enter the nShield Connect account information you gather during the installation and configuration of nShield Connect in CCKM (described in this section) when configuring this key source key source in the **nShield** tab of the **Key Sources** page of the CCKM Admin Portal as part of the initial CCKM configuration. (This initial configuration of CCKM takes place after you install CCKM and login into the CCKM Admin Portal for the first time.)

For any nShield Connect you plan to use as key source for CCKM, the nShield Connect administrator must perform the following:

> **Note**
> To complete the configuration of nShield Connect, you are required to install CCKM and note its IP address or already know the IP address of CCKM. After obtaining the IP address of the CCKM, you can return to completing the nShield Connect configuration by adding the IP address of CCKM as an nShield client (last step).

1. Configure nShield Connect and note its IP address. You will add this IP address when you configure CCKM to connect to the nShield Connect.
2. Set up RFS and note its IP address. You will add this IP address when you configure CCKM to connect with nShield Connect.
3. Set up Security World.
4. In the nShield Connect, configure CCKM as an nShield client by specifying the IP address of CCKM. For information on how to configure the CCKM as an nShield client, refer to the nShield Connect User Guide for Unix.

Chapter 7: Register CCKM App in Azure Cloud

Register CCKM app in Azure ***before*** adding an Azure cloud in CCKM. CCKM provides support for Azure clouds (Azure, Azure China, Azure Germany, and Azure Stack). The steps for creating a cloud for Azure, Azure Germany, and Azure China differ from those steps to creating a cloud for Azure Stack.

Registering CCKM App in Azure, Azure Germany, or Azure China Cloud

Before adding an Azure cloud in CCKM, you must register the CCKM App and assign required permissions in the Azure portal. Then depending on the type of app credential you plan to employ, either create a key (client secret) in the Azure portal or generate a certificate from CCKM, download it, and then upload it to Azure. This entire process generates the connection data needed to configure the Azure cloud.

Also before configuring an Azure cloud in CCKM, refer to the following related sections to review information about how Azure allows external applications, such as CCKM, to access and manage resources within Azure and the two types of app credentials employed when registering apps in Azure:

- "Accessing and Managing Azure Resources" on page 107
- "Registering Apps in Azure" on page 107

The steps to create the connection between CCKM and an Azure Key Vault are the same for Azure, Azure Germany, and Azure China.

> **Note**
> The UI within the Azure portal changes periodically. In regards to the documentation relating to Azure App Registrations, this documentation reflects the legacy App Registrations, which displays as App registrations (Legacy) in the Azure portal. If you are using the new Azure AD app registration, refer to the Azure online documentation for information regarding the equivalent steps (documented in this section) that you would perform in the new Azure AD app registration.

> **Note**
> In a production environment, Thales recommends that you **DO NOT** change the cloud configuration for Azure, Azure Germany, or Azure China by changing the setting for the **Use CCKM as Service Principal** check box.

Create an Azure Active Directory Application (Multi-tenanted) in Azure Portal

To create an Azure Active Directory application in the Azure portal:

1. In the Azure Active Directory > App registrations > New application registration, provide the following parameters:
   - **Name:** choose a name for the app that CCKM will use to access Azure.
   - **Application type:** Web app/API
   - **Sign-on URL:**
     https://{CCKM Instance Host name}:8443/kmaas/login/azure (for Azure)
     https://{hostname}:8443/kmaas/login/azureChina (for Azure China)
     https:// {hostname}:8443/kmaas/login/azureGermany (for Azure Germany)
   - If you are using CCKM as a service principal, then in addition to the above URLs, also add the following URLs.

     > **Note:** In all cases, {hostname} must resolve to either the IP address of your CCKM Instance, or, for CCKM clusters, the IP address of the load balancer. For more information about CCKM cluster configuration, see "CCKM Cluster Configuration" on page 62

     For Azure:
     https://{CCKM Instance Host name}:8443/kmaas/auth2/azure
     https://{CCKM Instance Host name}:8443/kmaas/azureAdminConsent/azure

     For Azure China:
     https://{hostname}:8443/kmaas/auth2/azureChina
     https://{hostname}:8443/kmaas/azureAdminConsent /azureChina

     For Azure Germany:
     https:// {hostname}:8443/kmaas/auth2/azureGermany
     https:// {hostname}:8443/kmaas/azureAdminConsent /azureGermany
2. Click **Create** to create the app.
3. Access the new app under *App Registrations*.
4. From *App Registrations* > *{App Name}* > *Settings:* Note the **Application ID**. This is the “Client ID” used when configuring the cloud within CCKM.
5. From App Registrations > {App Name} > Settings > Properties: select **Yes** for **Multi-tenanted**.
6. Click **Save**.

Set Required Permissions on CCKM App

To set the required permissions on the CCKM app:

1. Access **App Registrations > {App Name} > Settings > Required Permissions** in Azure.

2. Click **+Add** and add the following APIs and their associated permissions:

   - **Azure Key Vault | DELEGATED PERMISSIONS |** √ Have full access to Azure Key Vault service
   - **Windows Azure Service Management API | DELEGATED PERMISSIONS |** √ Access Azure Service Management as organization users (preview)
   - **Windows Azure Active Directory (Microsoft.Azure.ActiveDirectory) | DELEGATED PERMISSIONS |** √ Sign in and read user profile
   - **Microsoft Graph | DELEGATED PERMISSIONS |** √ Read directory data

   > **Note:** The Microsoft Graph permission is an ***optional*** permission. The Microsoft Graph permission is required to query the names of the Azure applications or services using an Azure key from Azure Logs Analytics when you generate the **Azure Key Service Usage Report** from CCKM. The generated report includes the names of these applications or services and other related information. ***Without this permission set, the IDs (instead of the names) of the applications or services using an Azure key are included in the report after you run it.***

3. Select **Grant Permissions** and select **Yes**. (You must have Azure administrator permissions to Grant Permissions as required by the Microsoft Graph API.)

Create a Key in Azure Portal

If you will use a secret key or client secret in Azure as part of registering the apps in Azure, create this key in Azure. When setting access to an Azure cloud in the CCKM as part of adding a cloud to CCKM, you will provide this secret key.

To create a secret key (or client secret) in the Azure portal:

1. Access **App Registrations > {App Name} > Settings > Keys**.

2. Enter a key name in **Key description** box.

3. Select a duration period in the **Duration** box.

4. Click **Save**. The **Key Value** column displays the new client secret.

> **WARNING**
> **Copy this value and store securely; it cannot be retrieved after leaving this view. It will display as Hidden within the Value column. (This value is the “client secret” used when configuring the cloud in CCKM.)**

Creating CCKM App in Azure Stack Cloud with Azure AD

Before configuring Azure Stack cloud with the use of the directory type of AAD, you must create (or register) an App in Azure, assign required permissions, and create a key. This process generates the connection data needed to configure the cloud.

If you are using AAD as the identity provider, you need to register CCKM application twice (one for the administrator portal and one for User Portal) in AAD and provide CCKM with the associated client credentials. The process is similar to adding the Azure cloud .

> **Notes**
>
> - This documentation assumes that you are familiar with using Microsoft Azure Stack. This documentation also assumes that you have installed and configured Azure Stack in your data center. For information on Azure Stack, refer to the Microsoft Azure Stack documentation.
> - The features that allow for the use of CCKM as a service principal to access and manage resources within Azure and the use of a certificate (public key) as an app credential within CCKM and Azure are currently not supported in Azure Stack.

Create an Azure Active Directory Application in Azure Portal

This section provides the steps to create an administrator application and a Tenant application.

> **Note**
>
> If you are enabling the Azure Stack administrator cloud and Azure Stack cloud, then you will need to register CCKM application twice--the first time for the "azureStackAdmin" login endpoint and the second time for "azureStack" login endpoint.

In the **Azure Active Directory**> **App registrations**> **New application registration**, provide the following parameters:

- **Name:** choose a name for the app that CCKM will use to access Azure Stack.
- **Application type:** Web app/API
- **Sign-on URL:** https://{ hostname }:8443/kmaas/login/azureStackAdmin (for Azure Stack administrator) https://{ hostname}:8443/kmaas/login/azureStack (for Azure Stack Tenant)

  {hostname} must resolve to the IP address of your CCKM Instance.

After clicking **Create**, access the new app under *App Registrations*.

From *App Registrations > {App Name} > Settings:* Note the Application ID. This is the "Client ID" used when configuring the CCKM cloud.

Set Required Permissions on CCKM App

To set the required permissions on the CCKM app:

- Access App Registrations > {App Name} > Settings > Required Permissions in Azure.
- **Add** the following APIs and their associated Permissions:

For Azure Stack (tenant), add:

- **Azure Stack - KeyVault | DELEGATED PERMISSIONS |** √ Access Azure Stack - KeyVault

For Azure Stack (administrator), add:

- **AzureStack - KeyVault Internal | DELEGATED PERMISSIONS |** √ **Access AzureStack KeyVault Internal**
- **AzureStack - Administration | DELEGATED PERMISSIONS |** √ **Access Azure Stack – Administration**

For both Azure Stack (tenant) and Azure Stack (administrator), add:

- **Windows Azure Service Management API | DELEGATED PERMISSIONS |** √ Access Azure Service Management as organization users (preview)
- **Windows Azure Active Directory (Microsoft.Azure.ActiveDirectory) | DELEGATED PERMISSIONS |** √ Sign in and read user profile

- **Microsoft Graph | DELEGATED PERMISSIONS |** √ Read directory data

> **Note:** The Microsoft Graph permission is an optional permission. The Microsoft Graph permission is required to query the names of the Azure applications or services using an Azure key from Azure Logs Analytics when you generate the Azure Key Service Usage Report from CCKM. The generated report includes the names of these applications or services and other related information. Without this permission set, the IDs (instead of the names) of the applications or services using an Azure key are included in the report after you run it.

- Under *Required Permissions*, select **Grant Permissions**. (You must have Azure administrator permissions to Grant Permissions, as required by the Microsoft Graph API.)

For AAD administrator app, add:

- **Azure Stack - Administration | Access Azure Stack - Administration Windows Azure Service Management API | Access Azure Service Management** as organization users (preview) **AzureStack KeyVault Internal | Access AzureStack KeyVault Internal Microsoft Graph | Read directory data (Delegated Permissions) Windows Azure Active Directory | Sign in and read user profile (Delegated Permissions)**

For AAD Tenant app, add:

- **Azure Stack | Access Azure Stack**
- **Windows Azure Service Management API | Access Azure Service Management as organization users (preview) Azure Stack KeyVault | Access Azure Stack KeyVault Microsoft Graph | Read directory data (Delegated Permissions) Windows Azure Active Directory | Sign in and read user profile (Delegated Permissions)**

Obtaining URLs for Full Path To Endpoints

This section provides the steps for setting access to Azure Stack with AAD as well as the associated prerequisite steps.

Prerequisites to Setting Access to Azure Stack with AAD

Part of setting access to Azure Stack with ADD includes entering the required URLs for the full path to the endpoints. To obtain the URLs used in your cloud environment, run the following Azure commands using the Azure CLI in a terminal:

> **Note**
> You are required to run these commands for the "azureStackAdmin" cloud environment as an administrator and the "azureStack" cloud environment as a user. Essentially, you will run these commands twice—as an administrator and as a user.

1. In the role of administrator, set the Azure CLI environment for the "azureStackAdmin" cloud environment:

   ```
   az cloud set --name AzureStackAdmin
   ```

   In the role of an Azure user, set the Azure CLI environment for the "azureStack" cloud environment:

   ```
   az cloud set --name AzureStackUser
   ```

2. Login to Azure CLI using the user name and password of the administrator or user (depending on the role you are using for the login):

   ```
   az login
   ```

3. Display the configuration details of the cloud environment including the full path to the endpoints:

   ```
   az cloud show
   ```

   Take note of the displayed URLs. You will need to enter those URLs in the "Set Access to Azure Stack with AAD" procedure that follows.

For more information about the Azure cloud-management commands and login command listed in this section, see the Azure Command-Line Interface (CLI) documentation within the Microsoft Azure online documentation.

Set Access to Azure Stack with AAD

For information about this step, refer to "Set Access to Azure Stack with AAD" on page 69.

Creating Service Principals for Azure Stack Cloud with ADFS

This section describes the prerequisite steps to take before adding an Azure Stack cloud with Active Directory Federation Service (ADFS) in CCKM.

If you are using ADFS as the identity provider, you need to create two service principals for CCKM (one for the administrator portal and one for User Portal) in ADFS and provide CCKM with the associated client assertion credentials.

> **Note**
> This documentation assumes that you are familiar with using Microsoft Azure Stack. This documentation also assumes that you have installed and configured Azure Stack in your data center. For information on Azure Stack, refer to the Microsoft Azure Stack documentation.

Creating Two Service Principals (for Admin and Tenant)

Before adding an Azure Stack cloud with the use of the directory type of ADFS, perform the following steps:

- Create two service principals, one for administrator and one for tenant.
  See the following Azure Stack documentation covering how to create the service principals for ADFS:

  https://docs.microsoft.com/en-us/azure/azure-stack/azure-stack-create-service-principals#create-service-principal-for-ad-fs

  Creating the service principals requires you to generate a certificate for the service principals. The certificate will be used as the client assertion credentials for the client assertion authentication. When creating a service principal, ensure to provide the clientRedirectUris. The following are examples of clientRedirectUris for the CCKM administrator and tenant:

- New-GraphApplication -Name ‘CCKM_Admin’ -ClientRedirectUris "https://<hostname>:8443/kmaas/login/azureStackAdmin" -ClientCertificates $using: <certificate>}

- New-GraphApplication -Name ‘CCKM_tenant’ -ClientRedirectUris "https://<hostname>:8443/kmaas/login/azureStack" -ClientCertificates $using:cert}

> ⚠ **CAUTION**
> **Be sure to note the app ID of the service principal and store the certificate securely. This is the client certificate used when configuring the CCKM cloud.**

Chapter 8: Setting Up CCKM App for Access to Google Cloud

Prior to configuring and accessing the Google Cloud KMS in CCKM, ensure all of the prerequisites are met within the Google Cloud console. See "Prerequisites on Google Cloud" on page 138 for a list of all of the prerequisites. This chapter provides information on how to perform most of these prerequisites. For information on how to address the other prerequisites not covered in this document, refer to the Google Cloud online documentation.

> **Note**
> This documentation assumes that you are familiar with using Google Cloud Platform, Google Cloud KMS, and Google Identity and Access Management (IAM). For details about these Google technologies, refer to the Google Cloud documentation at https://cloud.google.com/docs.

Set Up OAuth Consent Screen and Request Verification

Before your CCKM users can authenticate to Google Cloud, a consent screen will be required to display. This screen will allow your users to choose whether they will allow access to their private data, as well as provide them a link to your terms of service and privacy policy. The consent screen is set up at the project level.

> **Note**
> As part of the OAuth Consent screen setup, you will be adding a scope that is sensitive. This type of scope requires verification from the Google Cloud security team, which can take approximately four to six weeks to complete.
>
> For development purposes, the consent screen does not need to be submitted for verification. However, OAuth is restricted to 100 sensitive scope logins until the consent screen is published.

After performing the steps in this section, the next step to take is to enable IAM API (see "Enable Identity and Access Management API" on the next page.

To set up your consent screen for a project and request verification:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. From the left-side menu, select **APIs & Services**. The **APIs & Services** dashboard displays.
4. From the left-side menu, click **OAuth consent screen**. The **OAuth consent screen** displays.

5. From the **Application Type**, select **Public** or **Internal** depending on whether you will allow users to access CCKM outside of your organization. A public application allows access to users *outside* of your organization as well as inside your organization while an internal application allows access to users from your organization.
6. In the **Application name** box, enter the name of the app asking for consent. This is the name of your CCKM app.
7. Under **Scopes for Google APIs**, click **Add scope** and add "**…/auth/cloudkms**" and "**.../auth/cloud-platform**" as required scopes.
8. Under **Authorized domains**, enter the domain name of your organization and press **Enter**.
9. Click **Save** to save the setting.
10. Click **Submit** for verification.

Enable Identity and Access Management API

> **Note**
> Enable IAM APIs after you set up your consent screen for a project and request verification (see "Create OAuth2 Client ID Credentials " on the facing pageand after you create a service account for a project (see "Create Service Account" on page 51).

To enable IAM APIs:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. In the search box, enter "Identity and Access Management".
4. Select **Identity and Access Management (IAM) API**. The **Identity and Access Management (IAM) API** page displays.
5. Select the **ENABLE** button.

Enable Cloud Resource Manager API

To enable Cloud Resource Manager APIs:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. In the search box, enter "Cloud Resource Manager API".
4. Select **Cloud Resource Manager API**. The **Cloud Resource Manager API** page displays.
5. Select the **ENABLE** button.

Enable Cloud Key Management Service (KMS) API

To enable Cloud Key Management Service (KMS) APIs:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. In the search box, enter "Cloud Key Management Service (KMS) API".

4. Select **Cloud Key Management Service (KMS) API**. The **Cloud Key Management Service (KMS) API** page displays.
5. Select the **ENABLE** button.

Create OAuth2 Client ID Credentials

CCKM must have authorization credentials that identify itself to Google's OAuth 2.0 server. The OAuth credentials will be known to both CCKM and Google Cloud. You are required to create credentials for your project. CCKM will use these credentials that you have enabled for your project to access Google Cloud APIs. It is these credentials (Client ID and Client secret) that you will enter in the CCKM Admin portal to add Google Cloud service to CCKM.

> **Note**
> After creating the OAuth2 client ID credentials, copy the Client ID and Client secret from the **Web application** page, paste this information into a file, and then save it. When adding a Google Cloud service to CCKM, refer to this file to obtain the Client ID and Client secret.

To create the OAuth 2 Client ID credentials:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. From the left-side menu, select **APIs & Services**. The **APIs & Services** dashboard displays.
4. From the left-side menu, click **Credentials**. The **Credentials** page displays.
5. Click **+ Create Credentials**, then select **OAuth client ID**.
6. In the **Application type** drop-down menu, select **Web application**.
7. In the **Name** box, enter the name of your OAuth client (CCKM app name).
8. Under **Authorized redirect URIs**, click + **ADD URI** and enter the CCKM redirect URI(s) in the **URIs** box. The OAuth 2.0 server will use the redirect URI(s) (CCKM endpoint(s)) to which to send its responses. For example, ***https://cckm.example.com:8443/kmaas/login/googleCloud*** where "cckm.example.com" is the fully qualified domain name of your CCKM.
9. Click **Create**. The **Client ID for Web application** page displays including the name of OAuth client, Client ID, Client secret, and creation date.

Set Session Duration for Google Cloud Console

As organization administrator, you can configure a fixed session duration for specific apps and services that use Google Cloud. With a fixed session duration set, users are required to re-enter their login credentials to continue to access the Google Cloud console after the session expires.

If your Google user account belongs to an organization, your refresh token will be affected by your organization's security policy for Google Cloud session control. For more information, see Control session length for Google Cloud Console and gcloud CLI.

If you are employing key rotation and key synchronization schedules in CCKM, you are required to use the session duration option of **Session never expires** when setting your security policy for Google Cloud session control. Otherwise, your scheduled job will fail when the given schedule begins. This is due to CCKM saving the refresh token (to MongoDB) that it obtains from Google Cloud when a user logs into CCKM for the first time. When CCKM runs the scheduled job, it uses the refresh token to obtain an access token from Google Cloud. Due to the security policy for session control being set, the saved refresh token expires after the session duration (1 to 24 hours depending on configuration), disallowing CCKM from obtaining the access token to begin the scheduled job. As a result, the scheduled job will fail. To ensure the scheduled job is successful, set the session duration to **Session never expires**.

> **Note**
> If you are employing key rotation and key synchronization schedules in CCKM, it is recommended that you first create an organization unit, and then add all the users who will use CCKM to that unit. Thereafter, change the security policy for that unit to **Session never expires**.

As an *alternative* to setting the session duration to **Session never expires** for your security policy for session control to avoid the failure of scheduled jobs, use service accounts. You can create a service account, use the service account to authenticate to the CCKM REST API, and then add schedules using CCKM REST API. These schedules added by the service account do not use refresh tokens and are not affected by the security policy for session control. For more information about creating a service account, see "Create Service Account" below.

To set up a session duration for an organization:

1. Login as admin of your organization at the Google Admin console (https://admin.google.com).
2. From the home page, go to **Security** and then Google Cloud session control (Beta).
3. Under **Organization Unit** (on the left), select your organization unit.
4. Under **Session duration**, configure the session duration using *one* of the following ways:
   a. If you will set up key rotation and key synchronization schedules in CCKM, select **Session never expires** so that the session never expires.
   b. Select **Set session duration** and from the drop-down list for **session duration**, select from 1 to 24 hours.
5. Click **Save**.

Create Service Account

To employ CCKM REST APIs to manage Google Cloud keys, you will need to use a service account from Google Cloud to authenticate to the CCKM REST APIs.

> **Note**
> After successfully creating your service account, take note of the email address that Google Cloud assigned to it. You are required to enter this email address when adding service account to CCKM.

After performing the steps in this section, proceed to enabling IAM API (see "Enable Identity and Access Management API" on page 49. Thereafter, grant IAM Roles to the service account (see "Grant IAM Roles to User and Service Accounts" on the facing page).

To create a service account for a project:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. From the left-side menu, select **IAM & Admin** > **Service Accounts**. The **Service Accounts** dashboard displays.
4. Click **+ CREATE SERVICE ACCOUNT**. The **Create service account** page displays.
5. In the **Service account name** box, enter the name of the service account.
6. (Optional) In the **Service account description** box, enter a description of the service account.
7. Click **Create**.

Grant IAM Roles to User and Service Accounts

For your user account or service account to use CCKM, you need to grant Project Browser and Cloud KMS Admin roles to the account. If your project belongs to an organization, then you also need to grant Organization Viewer role to your accounts.

The role of Project Browser allows read access to browse the hierarchy of a project, including the folder, organization, and IAM policy. The role of Cloud KMS Admin gives full access to Cloud KMS resources with the exception of encrypt and decrypt operations. Access to view an organization is given through the use of the Organization Viewer role. This role allows CCKM to find the organization to which the account belongs. For more information about Google Cloud roles, see Predefined roles within Google Cloud online documentation.

If you are using a service account, the next step to take after performing the steps in this section is to add a service account in CCKM. See "Adding Google Cloud Service Account to CCKM" on page 163 for more information. Thereafter, you can use the service account to authenticate CCKM REST API. After successfully authenticating to the CCKM REST APIs, you can proceed to making calls to CCKM REST APIs to manage Google Cloud keys. For more information, see Appendix D: "Sample Python Scripts for Authenticate and Call CCKM REST API " on page 221and Appendix E: "Authenticate your REST API Client to CCKM REST APIs for Google Cloud Keys" on page 228.

Grant Project Browser and Cloud KMS Admin Roles

To grant the roles of Project Browser and Cloud KMS Admin to your user account or service account:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. From the left-side menu, select **IAM & Admin** > **IAM**. The **IAM** dashboard displays.
4. Click **+ ADD**. The **Create service account** page displays.
5. In the **New members** box, enter the username of your user account you wish to grant roles (if this is for a user account). Enter the email address of the service account you wish to grant roles (if this is for a service account).
6. In the **Select a role** box, select the role of **Project** (on the left), and **Browser** (on the right).
7. Click **ADD ANOTHER ROLE**.
8. In the **Select a role** box, select the role of **Cloud KMS** (on the left), and **Admin** (on the right).
9. Click **Save**.

Grant Organization Viewer Role

If your user or service account belongs to a project that is a member of an organization, then grant the role of Organization Viewer to these accounts at the organization level, in addition to granting the roles of Project Browser and Cloud KMS Admin at the project level.

> **Note**
> To grant the role of Organization Viewer to your user or service account, the role of Organization Administrator is *required*. If you do not have this role, make the request of your Organization Administrator to grant this role to your user or service account.

To grant the Organization Viewer role to your user account or service account:

1. (*As an Organization Administrator*) Go to the Google Cloud Platform Console.
2. From the Organization list, select the organization in which the CCKM app belongs.
3. From the left-side menu, select **IAM & Admin** > **IAM**. The **IAM** dashboard displays.
4. Click **+ ADD**. The **Create service account** page displays.
5. In the **New members** box, enter the username of the user account you wish to grant the role (if this is for a user account). Enter the email address of the service account you wish to grant the role (if this is for a service account).
6. In the **Select a role** box, select the role of **Resource Manager** (on the left), and **Organization Viewer** (on the right).
7. Click **Save**.

Create Custom Roles

If you wish to grant specific permissions to users or groups to meet your specific requirements, then create custom roles. Group one or more of the supported permissions into a custom role and grant this role to users or groups. See "Create Custom Roles" above table for the permissions that are required for each of the supported CCKM actions.

To create custom roles to use for CCKM:

1. Go to the Google Cloud Platform Console.
2. From the projects list, select the project to which the CCKM app belongs.
3. From the left-side menu, select **IAM & Admin** > **Roles**. The **Create Role** dashboard displays.
4. In the **Title** box, enter the name of the custom role.
5. Click **+ ADD PERMISSIONS**. The **Add permissions** page displays.
6. From the list of permissions, select the permission(s) you wish to use for the custom role.
7. Click **ADD**.

Chapter 9: Create a Connected App in Salesforce

Before adding a Salesforce or Salesforce Sandbox cloud in CCKM, you must create the CCKM app and assign required permissions in the Salesforce or Salesforce Sandbox portal. Ensure you have the permissions to create the app.

> **Note**
> The information regarding Salesforce is applicable to both Salesforce and Salesforce Sandbox unless otherwise noted. The CCKM UI for Salesforce and Salesforce Sandbox is identical with the exception of the display of the names of the clouds. Your login credentials for Salesforce and Salesforce Sandbox are unique to each cloud. If you are configuring for Salesforce Sandbox, be sure to configure within the Salesforce Sandbox cloud.

Creating a Connected App in Salesforce

To create a connected CCKM app in Salesforce:

1. In Salesforce, ensure you have the permissions required to **Create** connected apps.
2. In the Salesforce app creation interface, supply basic information (app name, description, etc.).
3. Define how the app communicates with Salesforce using the following **API** parameters:
   - Enable OAuth Settings: check
   - Callback URL for Salesforce: `https://{hostname}:8443/kmaas/login/salesforce`

     Callback URL for Salesforce Sandbox: `https://{hostname}:8443/kmaas/login/salesforcesandbox`

     {hostname} must resolve to the IP address of your CCKM Instance.

     > **Note:** If you plan to deploy a CCKM cluster, then the {hostname} must resolve to the IP address of the load balancer. For more information about CCKM cluster configuration, see "CCKM Cluster Configuration" on page 62.

   - Selected OAuth Scopes: choose "*Access and manage your data (api)* " and "*Access your basic information (id, profile, email, address, phone)* ".
4. Click **Save**. The **Consumer Key** is created and displayed, and the **Consumer Secret** is created.

   Note the consumer key and secret and store them securely; you will use them when adding the Salesforce or Salesforce Sandbox cloud in CCKM.

Chapter 10: Install CCKM

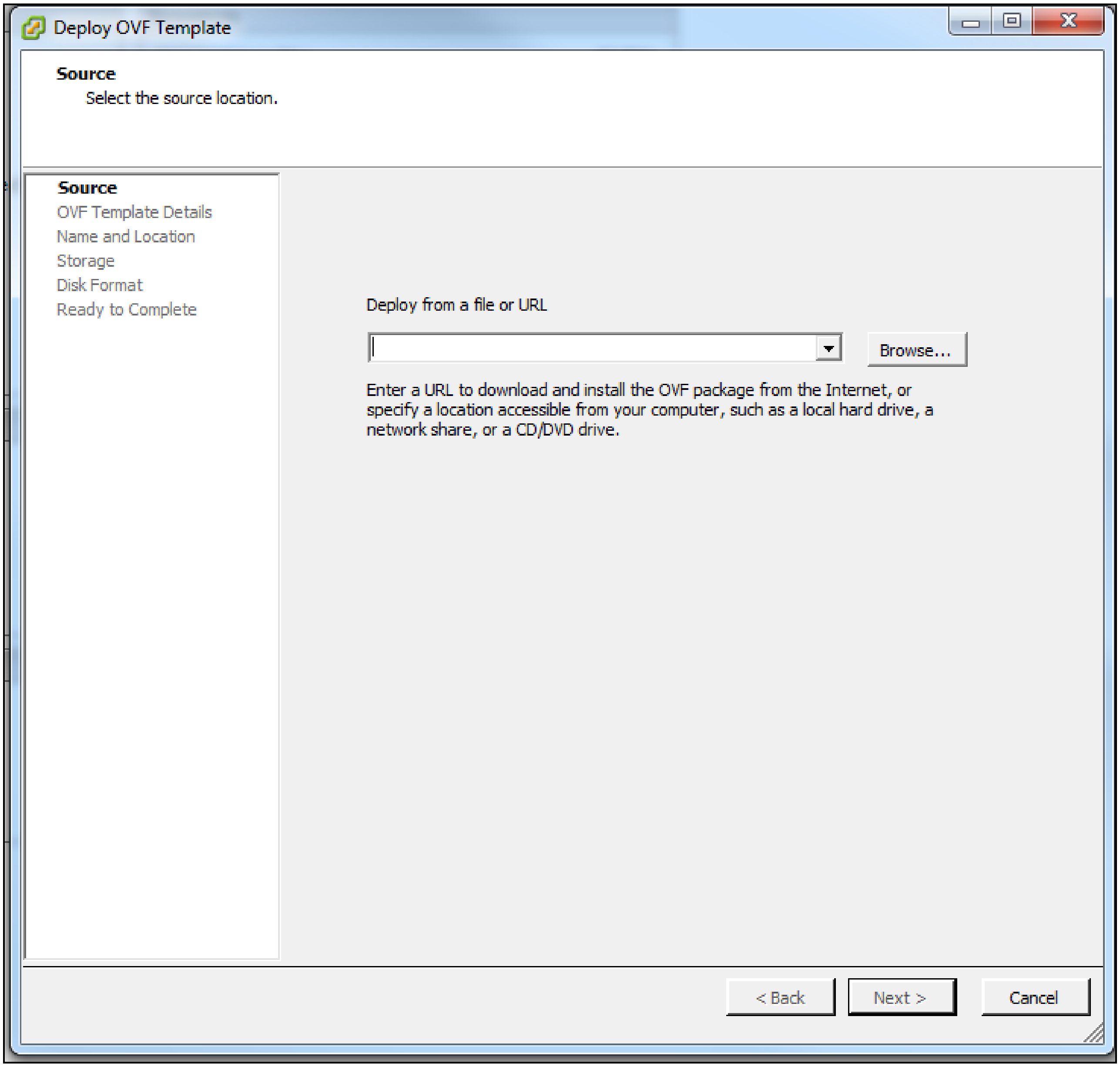
Install CCKM on VMware, AWS, or Microsoft Hyper-V environments. For the Azure Marketplace, launch and configure CCKM from the marketplace.

.

> **Note**
> CCKM is supported on VMware ESXi 6.0 and later versions.
>
> ***After installing CCKM*** and changing the default password, the ***first*** step to take is to synchronize the local time with network time using your preferred Network Time Protocol (NTP) server(s). Ensure at least one NTP server is configured. For more information, see "Synchronize CCKM Local time with Network Time using NTP Server" on page 59.
>
> *(Optional)* If you wish to use multiple NICs, configure the NIC(s) ***after*** adding an NTP server to CCKM and synchronizing with it. Then proceed to with accessing the CCKM GUI ***after*** completing the NIC configuration.

Install on VMware ESXi Server

To create the CCKM Server on a VMware ESXi server, load the CCKM server's .ova template file in the VMware ESXi server and create the VM. Give the CCKM VM a name and keep the default resource settings.



Deploy OVF Template

Source

Select the source location.

Deploy from a file or URL

Enter a URL to download and install the OVF package from the Internet, or specify a location accessible from your computer, such as a local hard drive, a network share, or a CD/DVD drive.

Take note of the IP address of the VM. This will be the IP Address of CCKM.

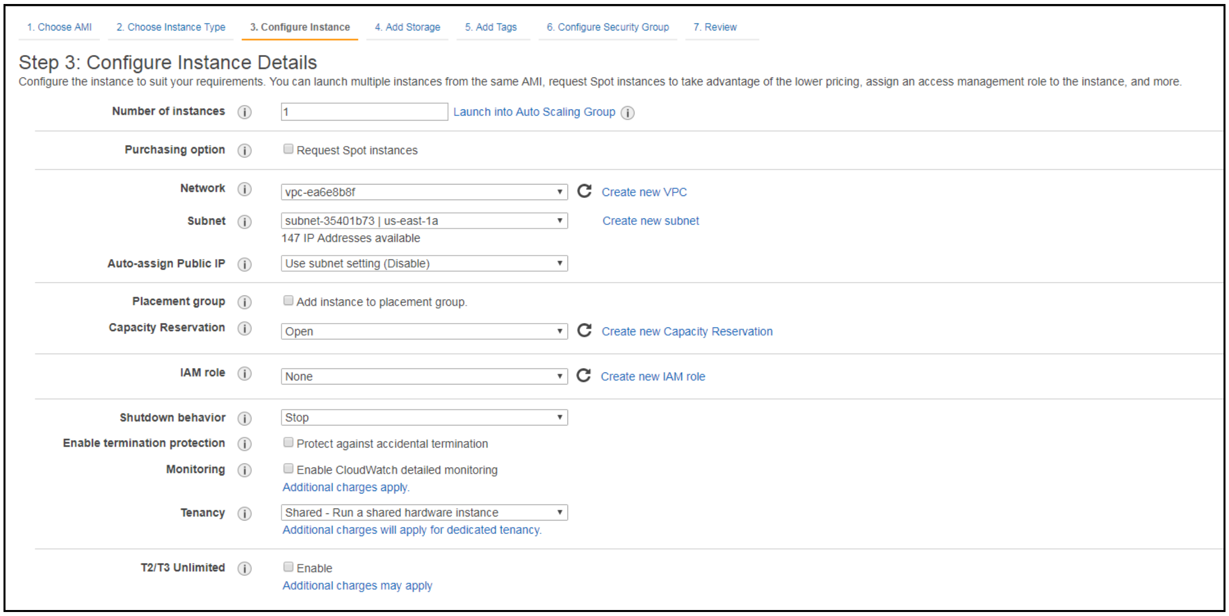
Install on AWS

The administrator should be familiar with AWS in general and AMI installations in particular.

Start the Installation

Contact Thales Support to obtain the CCKM Amazon Machine Image (**AMI**).

1. Log in to your AWS account.
2. Find the AMI in the AWS Dashboard at **EC2 > IMAGES > AMIs > Private images**.
3. Select the correct version of the CCKM AMI and click **Action > Launch**.
4. At **Choose an Instance Type**, select:
   - **Family**: General Purpose
   - **Type**: m4.xlarge
   - **vCPUs**: 4
   - **Memory**: 16
   - **Network Performance**: High
5. Click **Next**: Configure Instance Details.



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Network: vpc-ea6e8b8f — Create new VPC

Subnet: subnet-35401b73 | us-east-1a — Create new subnet
147 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

Placement group: □ Add instance to placement group.

Capacity Reservation: Open — Create new Capacity Reservation

IAM role: None — Create new IAM role

Shutdown behavior: Stop

Enable termination protection: □ Protect against accidental termination

Monitoring: □ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy: Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.

T2/T3 Unlimited: □ Enable
Additional charges may apply

Configure Instance Details

Fill out the *Configure Instance Details* page as follows:

1. In the **Network** box, select a VPC.
2. Select your subnet; it must be in the same subnet as the CipherTrust Manager or DSM depending on which key source you are using.
3. If you want the CCKM available on a public IP address, then set *Auto-assign Public IP* to **Enable**.
   This will become the CCKM GUI address.
4. Click **Next: Add Storage**.
5. Continue to "Complete the Installation" below.

Complete the Installation

On the *Add Storage* page:

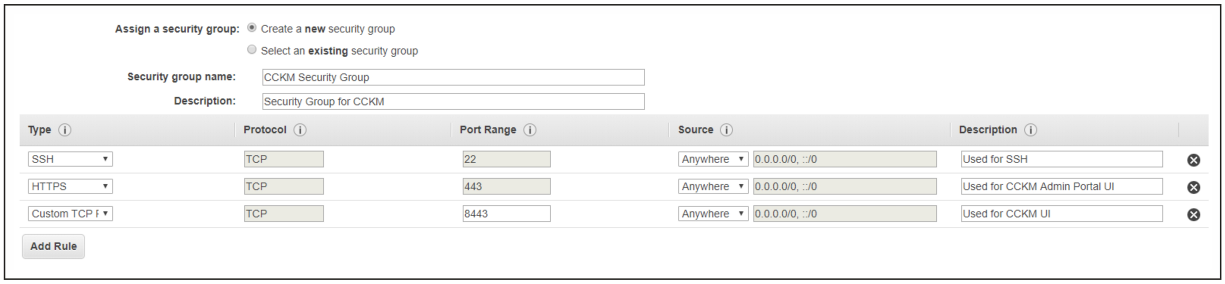
1. Use the default settings for adding storage. Click **Next: Tag Instance**.

2. In the *Tag Instance* page, create a name (key) and value to help organize your Amazon resources. In the *Name* column, type "**Name**". In the *Value* column, type the name you want to use for this instance.
   Note the following about tags (from Amazon's AWS documentation):

   Tags don't have any semantic meaning to Amazon EC2 and are interpreted strictly as a string of characters. Also, tags are not automatically assigned to your resources. You can edit tag keys and values, and you can remove tags from a resource at any time. You can set the value of a tag to an empty string, but you can't set the value of a tag to null. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value. If you delete a resource, any tags for the resource are also deleted.

3. Click **Next: Security Group**.

4. Select **Create a new security group**, enter a group name, and click **Add Rule** to configure the following sets of parameters:

   Assign a security group: ● Create a **new** security group ○ Select an **existing** security group

   Security group name: CCKM Security Group

   Description: Security Group for CCKM

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Anywhere	0.0.0.0/0, ::/0	Used for SSH
HTTPS	TCP	443	Anywhere	0.0.0.0/0, ::/0	Used for CCKM Admin Portal UI
Custom TCP F	TCP	8443	Anywhere	0.0.0.0/0, ::/0	Used for CCKM UI

   - **Type**: SSH, **Port**: 22, **Source**: specify the SSH source that can reach your instance
   - **Type**: HTTPS, **Port**: 443, **Source**: This port is used to access the CCKM Admin UI (CCKM Admin Portal)
   - **Type**: HTTPS, **Port**: 8443, **Source**: This port is used to access the CCKM GUI (CCKM user portal)

   > **Note:** Rules with source 0.0.0.0/0 allow all IP addresses to access your instance (***not recommended***). **Do not leave source IP addresses with the default 0.0.0.0/0.**

5. Click **Review** and Launch.

6. Review parameters and click **Launch**.

   A dialog box allows you to define a new public key and private key for EC2.

7. Create a new key/value pair and click **Launch Instances**.

8. You can now use the public or private DNS in your browser to access the CCKM GUI.

Install on Microsoft Hyper-V

To create a CCKM server on a Microsoft Hyper-V environment, load the CCKM VHD file and then create a virtual machine (VM) with the Hyper-V Manager. After you create the VM, you can begin to configure CCKM using the CCKM CLI.

Create CCKM VM

To create a CCKM VM with Hyper-V Manager:

1. In Hyper-V Manager, right-click and select **New Virtual Machine** to start the Hyper-V wizard.

2. Review the **Before You Begin** content (if it displays) and click **Next**.

3. At the **Specify Name and Location** page, specify the name for your VM and click **Next**.

4. At the **Specify Generation** page, leave the default of **Generation 1** and click **Next**.
5. At the **Assign Memory** page, enter the minimum of 4096 MB and click **Next**.
6. At the **Configure Networking** page, select the desired network connection and click **Next**.
7. At the **Connect Virtual Hard Disk** page, select **Use an existing virtual hard disk**, browse to select your CCKM VHD file, and click **Next**.
8. At the **Summary** page, click **Finish** to complete the wizard.

Power On CCKM VM and Connect to It

To power on the CCKM VM, connect to it, and then log into the CCKM CLI:

1. Right click on your new VM and select **Start** to power it on.
2. Right click on the VM again and select **Connect** to open a console for your VM.
3. Wait until the "cckm login" prompt displays (this may take a few minutes) and then log into CCKM CLI to configure your CCKM VM.

Launch on Azure Marketplace

Launch and configure CCKM from Azure Marketplace. Refer to the Azure online documentation for information.

Use SSH to Change the Default Password on the CCKM VM

After launching CCKM on a VM, it is important to change the default password as a security measure. To do so:

> **Note**
> ***After*** changing the default password, the ***next*** step is to add an NTP server to CCKM and synchronize the local time with the network time. For more information, see "Synchronize CCKM Local time with Network Time using NTP Server" below.

1. SSH to the IP address of the new VM (the IP address of CCKM).
2. Enter the username and password as prompted.
   Username: `cliadmin`
   Password: `cliadmin123`
3. Follow the prompt: Enter the old password, and then a new password twice. **Store securely**.

   ATTENTION: This username/password combination is used to access the VM via SSH. There is no way to recover a forgotten password— in this case, the VM would have to be discarded and a new CCKM launched.
4. The SSH session automatically ends. To validate, launch a new session and log in with the new password.

Synchronize CCKM Local time with Network Time using NTP Server

In this release of CCKM, the local time on CCKM server is no longer synchronized with the network time. ***After installing CCKM*** and changing the default password, the ***first*** step to take is to synchronize the local time with the network time using your preferred Network Time Protocol (NTP) server(s). For example purposes, the “pool.ntp.org” is used as the NTP server. However, you can use any NTP server you wish to use.

> **Note**
> At least one NTP server must be configured ***before*** you configure to add the NTP server to CCKM and synchronize with it.
>
> ***After*** adding an NTP server to CCKM and synchronizing the local time with the network time, the ***next*** step is to access the CCKM GUI. For more information, see "Access the CCKM GUI" on page 62. However, if you are using multiple NICs, then configure the NIC(s) ***after*** configuring to add the NTP server to CCKM and synchronize with it. Then proceed to access the CCKM GUI ***after*** completing the NIC configuration. For more information, see "Configure Multiple NICs on CCKM Instance" below.

To add an NTP server to CCKM and synchronize the CCKM local time to the network time, do the following:

1. Access your CCKM server through SSH (such as Putty):

   ```
   ssh cliadmin@<cckmhost>
   ```

2. Enter your user password when prompted:

   ```
   cliadmin@>cckmhost>'s password:
   CCKM CLI Main Menu
   cckm>
   ```

3. Access the CCKM Network command category:

   ```
   cckm> network
   ```

4. Access the NTP service settings and add the named NTP server to contact for time synchronization:

   ```
   network> ntpservice --add pool.ntp.org
   NTP Servers:
   pool.ntp.org
   ```

5. Perform immediate clock synchronization with the configured NTP server:

   ```
   network> ntpservice --sync
   Synchronizing time with NTP server(s):
   pool.ntp.org
   19 Mar 23:23:25 ntpdate[18641]: step time server 217.91.44.17 offset
   29286.555701 sec
   ```

6. Verify that the status of the NTP service is enabled:

   ```
   network>
   network> ntpservice --status
   enabled
   Active: active (running) since Thu 2020-03-19 14:58:26 PDT; 8h ago
   No association ID's returned
   network>
   ```

Configure Multiple NICs on CCKM Instance

CCKM supports multiple network interface controllers (NICs). This feature is available for ESXi and Azure cloud deployments. It is not available on AWS instances. Using the CLI network commands, you can assign a static IP to each NIC and add new routes to the routing table, ***Manually*** attach a NIC to a CCKM instance ***prior*** to performing the configurations in this section.

This section is only applicable to you if you are using multiple NICs.

Configure Network Interface

To configure the interface, do the following:

1. Log into the CCKM instance using cliadmin user.
2. View the network device configuration to verify the new NIC is attached to instance:
   ```
   network> show --list
   ```
3. Enter the network command category and configure a static IP address for the specified network interface:
   ```
   network> setup eth1
   <dhcp or static>? static
   ipv4 address? 10.3.200.201
   Subnet mask (e.g., 255.255.255.0)? 255.255.0.0
   ```

   OR

   Enable DHCP IP address leasing for the specified network interface:
   ```
   network> setup eth1
   <dhcp or static>? dhcp
   ```
4. Display the network interface setting to view the IP address configuration :
   ```
   network> ip --show
   1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
   default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
   valid_lft forever preferred_lft forever
   2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
   default qlen 1000
   link/ether 00:0d:3a:42:91:42 brd ff:ff:ff:ff:ff:ff
   inet 10.0.0.6/24 brd 10.0.0.255 scope global eth0
   valid_lft forever preferred_lft forever
   3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
   default qlen 1000
   link/ether 00:0d:3a:91:da:8c brd ff:ff:ff:ff:ff:ff
   inet 10.0.0.4/24 brd 10.0.0.255 scope global eth1 valid_lft forever preferred_
   lft forever
   ```

Add New Routes to Routing Table

To add a new route to the routing table, do the following:

> **Note**
> The default route is associated with eth0.

1. From the network commands, add a route to the routing table:
   ```
   route add -net 10.10.65.1/24 gw 10.10.65.254 dev eth1
   INFO: Execute 'route --make-persistent' command to make these changes reboot
   persistent
   ```

2. Set the routing table changes to be persistent after a reboot:

```
route –make-persistent
```

```
WARNING: This command will make the current routing table persistent after
reboot.
Any invalid configuration in routing table can make the system inaccessible over
the network. Do you wish to continue? <y/n> [n] y
SUCCESS: Routes are now reboot persistent
```

Access the CCKM GUI

Open a browser and point to the CCKM IP address with the https prefix (for example: *https://19.4.5.89*).

Once the CCKM is configured, the CCKM IP address will open to the CCKM GUI Home page.

CCKM Cluster Configuration

For high availability (HA) and improved performance of the CCKM solution, you can set up a CCKM cluster using at least two CCKM servers and one MongoDB database. To set up the CCKM cluster, launch two or more CCKM VMs using the same OVA, AMI, or VHD file for each VM you add to the cluster, configure the VMs to use the same MongoDB database.

When you configure the database information on the second or additional CCKMs (following the steps in section "Set Up Database" on page 64), enter the same configuration passphrase and database information that you entered (in the **Configuration Passphrase** box) for the first CCKM. This configuration allows the additional CCKM servers to join the first CCKM in the cluster and share the same CCKM configuration in the MongoDB database.

You can access the CCKM Web UI and REST API on any of the CCKM servers.

Optionally you can add a load balancer in front of all CCKM servers. In this case, you can access CCKM Web UI and REST API using your load balancer. When configuring the load balancer, configure the listeners to forward traffic for ports 443 and 8443. For port 8443, enable application generated cookie stickiness (or session stickiness) on the cookie "JSESSIONID". For health check, set the ping target to "https:8443/kmaas/health". Refer to the documentation for the load balancer you are employing in the CCKM cluster for information on how to configure this configuration.

Chapter 11: Initial Configurations within CCKM Admin Portal

This chapter provides the required steps for the initial configuration of the CCKM Admin Portal ***after*** you install CCKM and login into the CCKM Admin Portal for the ***first*** time.

After installing CCKM, access the CCKM Admin Portal at *https://<CCKM hostname or IP address>* to begin configuring CCKM.

Summary of Initial Configuration Steps

The following is a summary of the required steps to perform during the initial configuration of the CCKM within the CCKM Admin Portal to get you up and running to begin using CCKM. For steps 2 through 7, you are prompted (at the top of the page) with each step to take next. After you *initially* log into CCKM Admin Portal *after* the CCKM installation, you are prompted to change your CCKM Admin Portal password.

1. Change your default Admin Portal password. (See "Change Default Admin Portal Password (During First-time Login)" below.)
2. Login into Admin Portal. (See "Login into Admin Portal" on the next page.)
3. Configure your database settings. (See "Set Up Database" on the next page.)
4. Upload your license file. (See "Upload License" on the next page.)
5. Configure your key source (CipherTrust Manager, DSM, or nShield). (See "Configure Key Source" on the next page.)
6. Add cloud(s). (See "Add Cloud" on page 65.)

Change Default Admin Portal Password (During First-time Login)

You are required to change the default admin password for the CCKM Admin Portal ***after*** your ***initial*** or first-time login into this portal. As part of the initial login, enter the default admin username of "**admin**" and the default admin password of "**admin123**". When prompted to change the current password, do the following:

1. Enter the current admin password (default admin password).
2. Enter your new password.
3. Enter your new password again to confirm it.
4. Click **Update**.

Login into Admin Portal

After changing the admin password, you are prompted to log into CCKM Admin Portal again using your new password. Enter your admin username and password and click **Login**.

Set Up Database

You are prompted to set up your database within the **Settings** page. In the **Database** settings page, you have the following options to configure MongoDB:

- Local Database
- Remote Database
- ReplicaSet Database
- MongoDB Atlas

The CCKM server requires a MongoDB database to store user setup and cloud usage data.

For information on how to configure MongoDB, see "Configure Database" on page 78. **After setting up MongoDB settings successfully, CCKM restarts.**

Upload License

After the CCKM service is restarted, you are prompted to go to the **Licenses** page to upload your CCKM license(s) for the cloud(s). You must have a valid license file to upload to the **Licenses** page. Once the license file is uploaded to this page, information about this license displays on the page. To obtain your CCKM license(s) for one or more clouds, contact Thales Sales.

To upload a license file:

1. Click **Upload License** at the top of the page.
2. From the **Upload License** dialog box, select **Choose File** and choose the license.
3. Click **Upload**.The license is displayed on the **Licenses** page.

Configure Key Source

You are required to configure the key source as part of your initial configuration of CCKM. You have three options to configure your key source from the **Key Sources** page of the CCKM Admin Portal using the following tabs from this page:

- CipherTrust Manager
- DSM
- nShield

> **Note**
> Concurrent use of the CipherTrust Manager, DSM, and nShield as key sources is not supported. Configure ***only*** one of these supported key sources as a key source.

For information on how to configure a key source, see "Configure Key Source" on page 72

Add Cloud

As part of your ***initial*** configuration of CCKM, you are required to add at least one cloud to manage through CCKM from the **Clouds** page of the CCKM Admin Portal. You can also add more clouds after the initial configuration of CCKM. The following are your options:

- AWS
- Azure
- Azure China
- Azure Germany
- Azure Stack
- Google Cloud
- IBM Cloud
- Salesforce
- Salesforce Sandbox

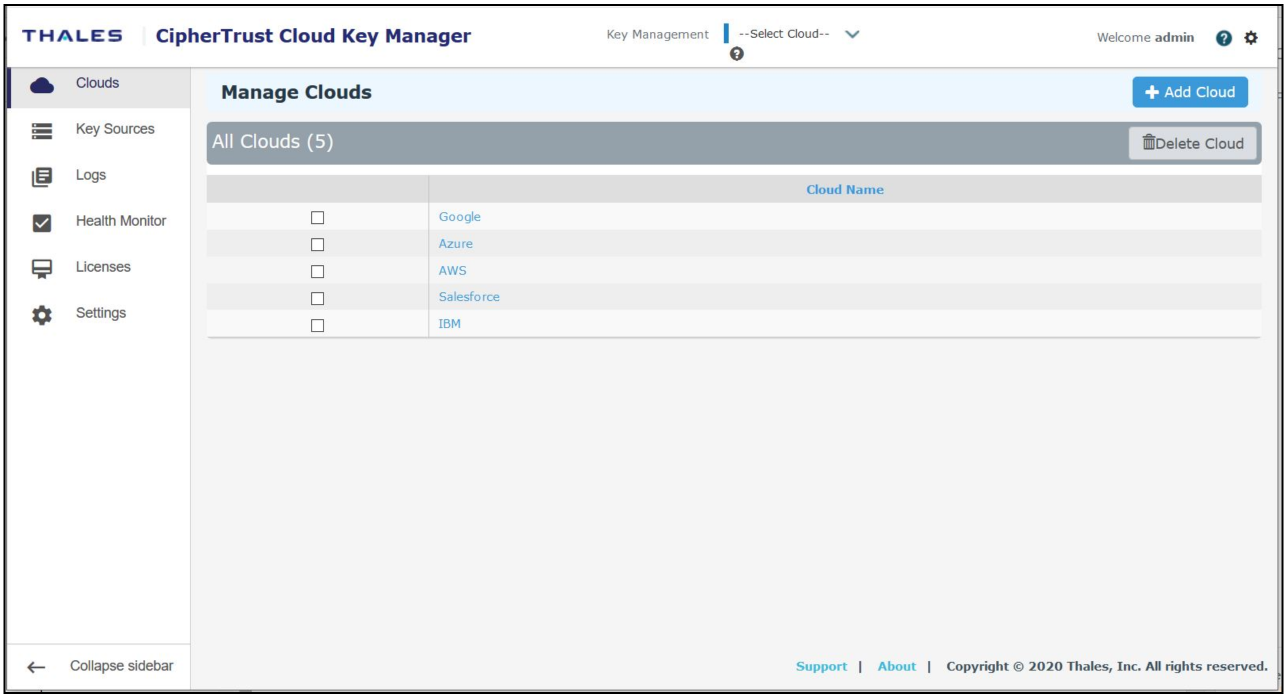
For information on how to add a cloud, see "Manage Clouds " on page 67

Chapter 12: About the Admin Portal

This chapter describes the Web UI pages of the CCKM Admin Portal and the settings and configurations you can perform using these pages.

CCKM Admin Portal

The CCKM Admin Portal contains the following options in the left-hand navigation bar:



	Cloud Name
□	Google
□	Azure
□	AWS
□	Salesforce
□	IBM

- **Cloud:** From the **Cloud** page, you can create a connection between CCKM and a given cloud to permit CCKM access to any of the cloud-service users with valid user credentials. From this page, you can also delete a cloud.
- **Key Sources:** From the **Key Sources** page, you can configure your key source.
- **Logs:** The **Logs** page lists audit logs related to the actions performed through the CCKM Admin Portal. You can search, sort, and view these audit logs.
- **Health Monitor:** The **Health Check Monitor** page provides a snapshot of the status of all major components in a CCKM solution. In addition, this page provides the option to remove a listing of a CCKM server that has a status of "Bad" from this page.

- **Licenses:** In **Licenses**, you can view all of your available licenses for your configured clouds.
- **Settings:** The **Settings** page allows you to configure database, proxy, SMTP, debug level, key deletion policy, and CCKM admin users.

Manage Clouds

From the **Clouds** page, you create a connection between CCKM and a given cloud to permit CCKM access to any of the cloud-users with valid user credentials. The following are the supported clouds:

- AWS
- Azure
- Azure China
- Azure Germany
- Azure Stack
- Google Cloud
- IBM Cloud
- Salesforce
- Salesforce Sandbox

From the **Clouds** page, you can also delete a cloud.

Select Cloud

Step 1: Select Cloud

1. Select **Clouds** from the left-hand navigation bar, select **Add Cloud**, and then select the cloud you wish to add from the **Cloud** drop-down list. The following are the options:
   - AWS
   - Azure
   - Azure China
   - Azure Germany
   - Azure Stack
   - Google Cloud
   - IBM Cloud
   - Salesforce
   - Salesforce Sandbox
2. Click **Next**.
3. Proceed to the next section to set access to the cloud.

Set Access to Cloud

Refer to the appropriate section below for information on how to set access to the cloud you are adding.

- If setting access to AWS cloud, see "Set Access to AWS Cloud" below
- If setting access to Azure cloud, see "Set Access to Azure (Azure, Azure Germany, Azure China) Cloud" below
- If setting access to Azure China cloud, see "Set Access to Azure (Azure, Azure Germany, Azure China) Cloud" below
- If setting access to Azure Germany cloud, see "Set Access to Azure (Azure, Azure Germany, Azure China) Cloud" below
- If setting access to Azure Stack cloud, see Set Access to Azure Stack
- If setting access to Google cloud, see "Set Access to Google Cloud" on page 71
- If setting access to IBM cloud, see "Set Access to IBM Cloud" on page 71
- If setting access to Salesforce cloud, see "Set Access to Salesforce Cloud" on page 71
- If setting access to Salesforce Sandbox cloud, see "Set Access to Salesforce Sandbox Cloud" on page 72

Set Access to AWS Cloud

Step 2: Set access to AWS cloud

In this step, set access to AWS cloud:

1. (Optional) Add a description for the cloud.
2. Click **Save**.

Set Access to Azure (Azure, Azure Germany, Azure China) Cloud

Step 2: Set Access to Azure (Azure, Azure Germany, Azure China) cloud

In this step, set access to Azure, Azure Germany, or Azure China cloud:

> **Note**
> If you will use CCKM as a service principal, then ensure to select the **Use CCKM as Service Principal** box.

1. Enter:

   Description: (Optional) the description of the cloud.

   Client ID: the 'application id' of the Azure app .

   If you using a client secret, select **Use client secret** and enter the client secret in the **Client Secret for the Azure Application** box.

   If you are using a certificate, select **Use certificate** and then click the **Generate and download** button. A message displays indicating a new certificate is generated and ready to download. Download the certificate and save it to a directory within your laptop or desktop. The certificate thumbprint automatically displays in the **Thumbprint** box after you download the certificate.

   > **Note:** Upload this certificate to the Azure Portal after completing the configuration of the cloud.

2. If you will use CCKM as a service principal, select the **Use CCKM as Service Principal** check box.

   > **Note:** The **Use CCKM as Service Principal** check box acts as a toggle to use CCKM on behalf of users or as a service principal. You can change this setting after completing the configuration of the cloud. To use CCKM on behalf of users, ensure this check box is deselected. In a production environment, Thales recommends that you **DO NOT** change the cloud configurationfor Azure, Azure Germany, or Azure China by changing the setting for the **Use CCKM as Service Principal** check box.

3. Click **Save**.

Set Access to Azure Stack with AAD

Step 2: Set Access to Azure Stack with AAD

In this step, set access to Azure Stack cloud with AAD:

1. From Azure Stack Type, select AAD.
2. Enter the Tenant Client Credentials:

   Tenant Client ID: The client ID of the Azure app (see "Create an Azure Active Directory Application (Multi-tenanted) in Azure Portal" on page 43.

   Tenant Secret Key: The client secret of Azure Stack tenant (see "Create an Azure Active Directory Application (Multi-tenanted) in Azure Portal" on page 43.

   Redirect URL: the tenant URL entered as a Sign-on URL in "Create an Azure Active Directory Application in Azure Portal" on page 45, or created in the 'Reply URLs' section of Azure Stack (see "Create an Azure Active Directory Application (Multi-tenanted) in Azure Portal" on page 43.

   Enter the **Admin Client Credentials**:

   Admin Client ID: The client ID of the Azure administrator (see "Create an Azure Active Directory Application (Multi-tenanted) in Azure Portal" on page 43.

   Admin Secret Key: The client secret of Azure Stack administrator.

   Redirect URL: the admin URL entered as a Sign-on URL in "Create an Azure Active Directory Application in Azure Portal" on page 45, or created in the 'Reply URLs' section of Azure Stack.

3. Enter the **Common Portal URIs and Endpoints** for **Azure Active Directory Authority:** the URL of the AD Authority. For example, https://login.windows.net
4. Enter **Tenant Portal URIs and Endpoints**:

   Refresh Token URL: the URL of the Refresh token. For example, https://vault.contoso.com/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

   Resource Manager URL: the URL of the Resource Manager. For example, URL https://management.local.azurestack.external

   Key Vault DNS Suffix: the URL of the Key Vault DNS Suffix. For example, vault.local.azurestack.external

   Graph DNS Suffix: the URL of the Graph DNS Suffix. For example, graph.local.azurestack.external

   Active Directory Service Endpoint Resource ID: the URL of the AD Service Endpoint Resource ID. For example, https://management.contoso.com/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

5. Enter **Admin Portal URIs and Endpoints**:

   Refresh Token URL: the URL of the Refresh token. For example, https://adminvault.contoso.com/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

   Resource Manager URL: the URL of the Resource Manager. For example, https://adminmanagement.local.azurestack.external

   Key Vault DNS Suffix: the URL of the Key Vault DNs Suffix. For example, adminvault.local.azurestack.external

   Graph DNS Suffix: the URL of the Graph DNS Suffix. For example, graph.local.azurestack.external

   Active Directory Service Endpoint Resource ID: the URL of the AD Service Endpoint Resource ID. For example, https://adminmanagement.contoso.com/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Set access to Azure Stack with ADFS Cloud

Step 2: Set access to Azure Stack with ADFS cloud

In this step, set access to Azure Stack cloud with ADFS:

1. From **Azure Stack Type**, select **ADFS**.

2. Enter the **Tenant Client Credentials**:

   Tenant Client ID: the App ID of the tenant service principal (see "Creating Two Service Principals (for Admin and Tenant)" on page 47).

   Tenant Client Certificate: Export the certificate part of the tenant service principal PFX certificate to a PEM file. Then click Choose File to select the exported PEM certificate of the Azure Stack tenant service principal.

   Tenant Client Certificate Private Key: Export the private key part of the tenant service principal PFX certificate to a PEM file. Then click Choose File to select the exported PEM private key of the Azure Stack tenant service principal.

   Redirect URL: the tenant URL entered as ClientRedirectURIs when creating the service principal.

3. Enter the **Admin Client Credentials**:

   Admin Client ID: the client ID of the Azure administrator app (see "Creating Two Service Principals (for Admin and Tenant)" on page 47).

   Admin Client Certificate: Export the certificate part of the administrator service principal PFX certificate to a PEM file. Then click Choose File to select the exported PEM certificate of the Azure Stack administrator service principal.

   Admin Client Certificate Private Key: Export the private key part of the administrator service principal PFX certificate to a PEM file. Then click Choose File to select the exported PEM private key of the Azure Stack administrator service principal.

   Redirect URL: the administrator URL entered as ClientRedirectURIs when creating the service principal.

4. Enter the **Common Portal URIs and Endpoints for Azure**:

   Authorization endpoint: the URL of the authorization endpoint. For example, https://adfs.local.azurestack.external/adfs/oauth2/authorize

   Token Endpoint URL: the URL of the token endpoint. For example, https://adfs.local.azurestack.external/adfs/oauth2/token

   Active Directory Authority: the URL of the AD Authority. For example, https://adfs.local.azurestack.external/adfs

5. Enter **Tenant Portal URIs and Endpoints**:

   Refresh Token URL: the URL of the Refresh token. For example, https://vault.adfs.azurestack.local/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

   Resource Manager URL: the URL of the Resource Manager. For example, URL https://management.adfs.azurestack.external

   Key Vault DNS Suffix: the URL of the Key Vault DNS Suffix. For example, vault.local.azurestack.external

   Graph DNS Suffix: the URL of the Graph DNs Suffix. For example, graph.local.azurestack.external

   Active Directory Service Endpoint Resource ID: the URL of the AD Service Endpoint Resource ID. For example, https://management.adfs.azurestack.local/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

6. Enter **Admin Portal URIs and Endpoints**:

   Refresh Token URL: the URL of the Refresh token. For example, https://adminvault.adfs.azurestack.local/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

   Resource Manager URL: the URL of the Resource Manager. For example, https://adminmanagement.adfs.azurestack.external

   Key Vault DNS Suffix: the URL of the Key Vault DNs Suffix. For example, vault.local.azurestack.external

   Graph DNS Suffix: the URL of the Graph DNS Suffix. For example, graph.local.azurestack.external

   Active Directory Service Endpoint Resource ID: the URL of the AD Service Endpoint Resource ID. For example, https://adminmanagement.adfs.azurestack.local/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Set Access to Google Cloud

Step 2: Set Access to Google Cloud

In this step, set access to Google Cloud:

1. (Optional) Add a description for the cloud.
2. Enter the Client credentials:
   - **Client ID:** The "application id" of the Google Cloud app.
   - **Client Secret:** This is the secret known only to the Google Cloud app and CCKM.
3. Click **Save**.

Set Access to IBM Cloud

Step 2: Set Access to IBM Cloud

In this step, set access to IBM cloud:

1. (Optional) Add a description for the cloud.
2. Click **Save**.

Set Access to Salesforce Cloud

Step 2: Set Access to Salesforce cloud

In this step, set access to Salesforce cloud:

1. (Optional) Add a description for the cloud.
2. Enter the Client credentials:
   - **Client ID:** The value of the 'Consumer Key' of the Salesforce app.
   - **Client Secret:** The value of the 'Consumer Secret' of the Salesforce app.
3. Click **Save**.

Set Access to Salesforce Sandbox Cloud

Step 2: Set Access to Salesforce Sandbox cloud

In this step, set access to Salesforce sandbox cloud:

1. (Optional) Add a description for the cloud.
2. Enter the Client credentials:
   - **Client ID:** The value of the 'Consumer Key' of the Salesforce app.
   - **Client Secret:** The value of the 'Consumer Secret' of the Salesforce app.
3. Click **Save**.

Configure Key Source

You are required to configure the key source as part of your initial configuration of CCKM. You have three options to configure your key source from the **Key Sources** page using the following tabs from this page:

- CipherTrust Manager
- DSM
- nShield

> **Note**
> Concurrent use of the CipherTrust Manager, DSM, and nShield as key sources is not supported. Configure only one of these supported key sources as a key source.

To configure CipherTrust Manager, select the **CipherTrust Manager** tab. For information on how to configure CipherTrust Manager, see "Configure CipherTrust Manager as Key Source" below.

To configure DSM, select the **DSM** tab. For information on how to configure DSM, see "Configure DSM as Key Source" on the next page.

To configure nShield, select the **nShield** tab. For information on how to configure nShield, see "Configure nShield Connect (for Azure and AWS Clouds)" on page 74.

Configure CipherTrust Manager as Key Source

CCKM provides support for the use of CipherTrust Manager as key source to generate, store, and retrieve the encryption keys that CCKM uses and manages. This section covers how to configure the connection between CCKM and CipherTrust Manager within the **Key Sources** page > **CipherTrust Manager** tab of the CCKM Admin Portal.

> **Note**
> Each time you add an additional node to your Key Secure instance that is connected to your (running) CCKM instance or CCKM cluster, a restart of the kmaas app service is required. To restart this service, in the CCKM command line prompt, enter the system command category and then enter the following command: `server --kmaas -restart`. For more information about CCKM CLI commands and the system command category and server command, see "System Commands" on page 216.

Prerequisites

Prior to configuring CipherTrust Manager within the **Key Sources** page, launch and configure CipherTrust Manager. For more information about these prerequisite steps, see "Install and Configure CipherTrust Manager as Key Source " on page 39.

Configuring CipherTrust Manager

To configure **CipherTrust Manager** in the **Key Sources** page > **CipherTrust Manager** tab within the CCKM Admin Portal, perform the following steps:

1. In the **Server** box, enter the hostname or the IP address of CipherTrust Manager.

   If the hostname of CipherTrust Manager is not reachable during this configuration, CCKM will display the **IP address** box. If the **IP address** box displays, enter the IP address of CipherTrust Manager in this box.

2. In the **Connection** box, enter the connection string. The default value is “local_account”. The value of "local" is supported. If CipherTrust Manager is integrated with an LDAP server, enter the domain name of the LDAP server.

3. In the **Username** box, enter the username for the CipherTrust Manager admin.

4. In the **Password** box, enter the password for the CipherTrust Manager admin.

5. From the **Certificate** box, select the certificate of CipherTrust Manager. Click **Browse** to select the file.

6. In the **Domain Prefix** box, enter a domain prefix for CipherTrust Manager. The domain prefix is the name of the CipherTrust Manager client. In this case, the client is the CCKM. This name must be unique within CipherTrust Manager.

7. Click **Test Connection** to test the connection between CCKM and CipherTrust Manager. If the test fails, verify that you entered the correct information in the previous steps. Reenter the correct information, if required, and then test the connection again. Otherwise, proceed to the next step.

8. Click **Save** to save the configurations.

Configure DSM as Key Source

CCKM provides support for the use of DSM as key source to generate, store, and retrieve the encryption keys that CCKM uses and manages. This section covers how to configure the connection between CCKM and DSM within the **Key Sources** page > **DSM** tab of the CCKM Admin Portal.

Prerequisites

Prior to configuring DSM within the **Key Sources** page, launch and configure DSM. For more information about these prerequisite steps, see "Install and Configure DSM as Key Source" on page 40.

Configuring DSM

To configure DSM in the **DSM** tab of **Key Sources** page of the CCKM Admin Portal, perform the following steps:

1. In the **DSM Hostname** box, the Fully Qualified Hostname of the DSM, which you can find on the dashboard of the DSM Management Console. Example: dsm-a43.vormetric.com.
2. In the **Domain Prefix** box, enter a unique DSM domain name prefix. Ensure this name conforms to DSM domain name restriction.
3. From the **Certificate** box, select the certificate of DSM . Click **Browse** to select the file.
4. In the **Username** box, enter the username for the DSM admin.
5. In the **Password** box, enter the password for the DSM admin.
6. Click **Save** to save the configurations.

Configure nShield Connect (for Azure and AWS Clouds)

If you are using CCKM to manage keys in Azure or AWS, support for the use of nShield Connect as a key source is available to generate source keys. This section covers how to configure the connection from CCKM to nShield Connect HSM(s), a remote file system (RFS), and the Security World for nShield. This configuration is available within the **Key Sources** page > **nShield** tab of CCKM Admin Portal. The prerequisites steps you are required to perform are also provided in this section.

After successfully configuring the nShield Connect and the associated RFS and Security World in the **Key Sources** page, CCKM runs a number of nShield Connect commands to enroll the CCKM as a new nShield client to the nShield Connect HSM(s), and updates the Security World configuration from the RFS computer. The output of these commands (provided in links) display on the right side of the **nShield Info** page. Be sure to click on each link within the **nShield Info** page to verify that each command ran successfully. If an error message displays, be sure to address the error. For common issues that may arise during the configuration of the connection from CCKM to nShield Connect and the associated RFS and Security World, see "Troubleshooting nShield Configurations " on page 212.

> **Note**
> This documentation assumes that you are familiar with using nShield Connect. This documentation also assumes that you have installed nShield Connect, and the associated RFS and Security World. For information on how to install an nShield Connect, and the associated RFS and Security World, refer to the *nShield® Connect Installation Guide*. For information on how to configure nShield Connect, and the associated RFS and Security World, refer to the *nShield Connect User Guide for Unix*.

Prerequisites

Prior to configuring the nShield Connect and the associated RFS and Security World within the Settings page, perform the following steps:

- Install and configure nShield Connect, RFS, and Security World.
- Configure CCKM as an nShield Connect client within the nShield Connect HSM by specifying the IP address of CCKM.

For more information regarding these prerequisite steps, see "Install and Configure nShield Connect as Key Source " on page 40

Configuring nShield

To configure nShield Connect and the associated RFS and Security World in the **Key Sources** page > **nShield** tab, perform the following steps:

1. From **nShield 1 IP Address**, enter the IP address of the nShield Connect.

2. Select **Add More nShield IP** to add and configure more nShield Connect HSMs.
3. From **nShield RFS IP Address**, enter the IP address of the remote file system (RFS).
4. Click **Add to Security World** to enroll CCKM to Security World. A set of links to the configuration command output display. Click on each link to verify that the commands ran successfully. If an error displays, address it before proceeding to the next step.
5. (Only perform this step, if all of the commands from the previous steps ran successfully.) Click **Save** to save the configurations. This step will restart the hardserver (Security World Software server) on which the CCKM is installed and configured.

   > **Note:** If an error occurred during any of the previous steps, and you click **Save**, you will not be able to generate an nShield Connect key within the CCKM Source Key page when you attempt to generate it. An error message displays. For common issues that may arise during the configuration of the connection from CCKM to nShield Connect HSM(s) and the associated RFS and Security World, see "Troubleshooting nShield Configurations " on page 212

Review Logs

Select **Logs** from the left-hand navigation bar within the CCKM Admin Portal to access the **Logs** page.

This page lists audit logs, describing user actions performed on the Admin Portal, set to Info, Warn, or Error alert levels.

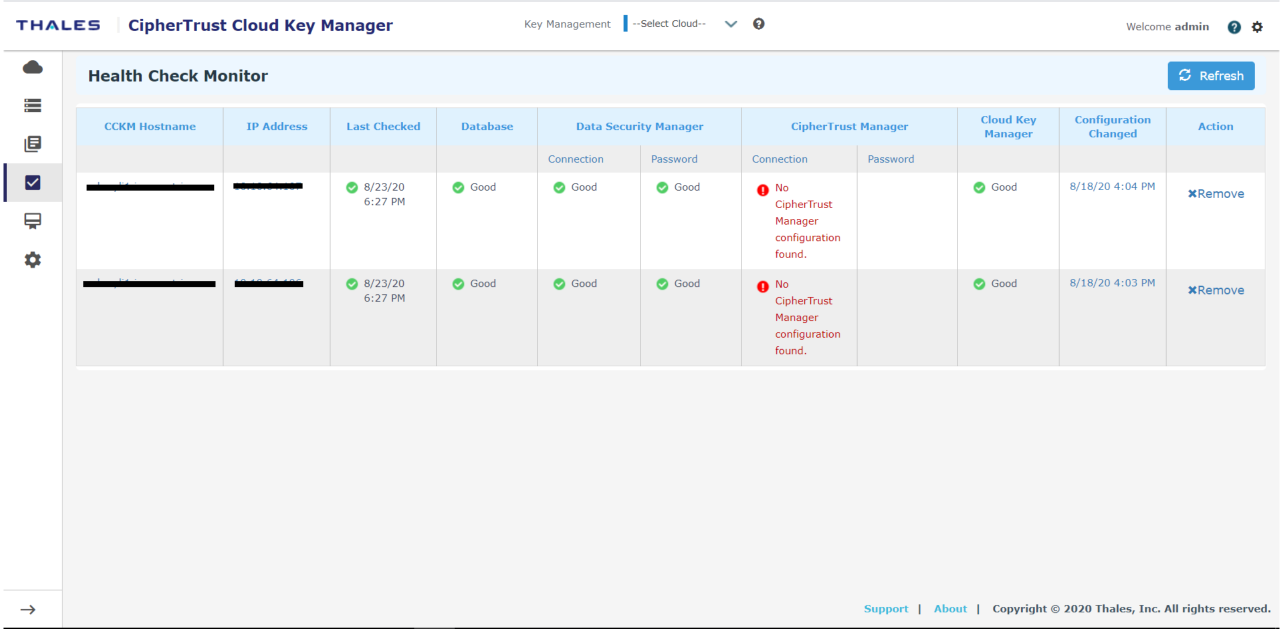
The CCKM administrator can use these logs to review admin activities in CCKM Admin Portal. Note that these logs differ from the logs used by Thales Support for deep troubleshooting (see "Troubleshooting Logs Files for Thales Support Team" on page 213), and CCKM user audit logs.

From the **Logs** page, you can:

- **Search** by log name or category filter
- **Search** for logs by **Date Range**
- Sort by **Date**, **Event Message**, **Severity**, **Category**, **Event name**, **Where** the activity occurred (which CCKM server), and **Severity** of the alert/Fineness of log capture level.
- Check **More** details to find, for example, uploaded file size, user ID, or timestamp.
- Select a log and **Refresh** it.
- Use **Download Debug Logs** to download debug log files.

Review Health Monitor

Select **Health Monitor** from the left-hand navigation bar within the CCKM Admin Portal to obtain a snapshot of the status of all major components in a CCKM solution.


THALES
CipherTrust Cloud Key Manager
Key Management
--Select Cloud--
Welcome admin
Refresh
Support | About | Copyright © 2020 Thales, Inc. All rights reserved.

Health Check Monitor

CCKM Hostname	IP Address	Last Checked	Database	Data Security Manager		CipherTrust Manager		Cloud Key Manager	Configuration Changed	Action
				Connection	Password	Connection	Password			
[redacted]	[redacted]	✓ 8/23/20 6:27 PM	✓ Good	✓ Good	✓ Good	! No CipherTrust Manager configuration found.		✓ Good	8/18/20 4:04 PM	✖Remove
[redacted]	[redacted]	✓ 8/23/20 6:27 PM	✓ Good	✓ Good	✓ Good	! No CipherTrust Manager configuration found.		✓ Good	8/18/20 4:03 PM	✖Remove

On the **Health Check Monitor** page, the following columns display including ones that provide the status (either "Good" or "Bad") of each of the major components of a CCKM solution:

- **CCKM Hostname**—The host name of the CCKM instance for which the information is displayed on the **Health Check Monitor** page.
- **IP Address**—The IP address of the CCKM server for which the information is displayed on the **Health Check Monitor** page.
- **Last Checked**—This column displays when the health status of this CCKM server was last checked . A status of "Bad" displayed in this column indicates that the last check took place more than 5 minutes ago, and the CCKM is down.
- **Database**—This column indicates the status of the connection between CCKM and the MongoDB server(s). A status of "Bad" indicates the connection between CCKM and the MongoDB server(s) is down or one or more of the MongoDB servers are down.
- **Data Security Manager, Connection**—This column indicates the status of the connection between the CCKM and DSM. A status of "Bad" indicates the network connection between the CCKM and DSM may be down or the DSM may be down. If a status of "Bad" is displayed for this column, check to see whether the DSM is running and reachable from CCKM.
- **Data Security Manager, Password**—This column indicates whether the configured DSM user password in the CCKM is valid. If the password status is "Bad", then change it in **DSM** tab of **Key Sources** page of the CCKM Admin Portal.

> **Note:** At a scheduled time (a configured setting in DSM), the DSM Administrator is required to change the DSM user password in the DSM based on security regulations. This change would require an update of this password within CCKM. This is a case when the password in both DSM and CCKM can get out of sync.

- **CipherTrust Manager, Connection**—This column indicates the status of the connection between the CCKM and CipherTrust Manager. A status of “Bad” indicates the network connection between the CCKM and CipherTrust Manager may be down or the CipherTrust Manager may be down. If a status of “Bad” is displayed for this column, check to see whether the CipherTrust Manager is running and reachable from CCKM.

  CipherTrust Manager, Password—This column indicates whether the configured CipherTrust Manager user password in the CCKM is valid. If the password status is “Bad”, then change it in **CipherTrust Manager** tab of **Key Sources** page of the CCKM Admin Portal.

- **Cloud Key Manager**—This column indicates the status of the CCKM (software) server running the CCKM User Portal. A status of “Bad” indicates the CCKM server is down, which means the CCKM User Portal is also down. To investigate this issue, check to see whether the kmaas (CCKM) process is running. If the process is not running, restart it by using CLI command **system->server --restart --kmaas**. Also download debug logs by clicking on **Download Debug Logs** on the **Logs** page within the CCKM Admin Portal. You can also download debug logs through the CCKM CLI command **applog->upload**.

- **Configuration Changed**—This column indicates the last time the CCKM configuration was changed on this CCKM.

- **Action**—This column provides the option to remove a listing of a CCKM server that has a status of "Bad" from the **Health Check Monitor** page. The status of a CCKM server displays under the **Cloud Key Manager** column. To remove a CCKM server with a status of "Bad", find the row that holds the CCKM you wish to remove and click **Remove** (from the **Action** column). The given CCKM server is removed from the **Health Check Monitor** page.

  > **Note**
  > If you remove a listing of a CCKM server that has a status of "Good" from the **Health Check Monitor** page, a background service running on CCKM will add the listing back to the page. This service periodically checks for active CCKM servers, which are connected to MongoDB(s). If an active CCKM server is not yet added to the page, this service adds it as an entry in the **Health Check Monitor** page.

Review Licenses

Select **Licenses** from the left-hand navigation bar within the CCKM Admin Portal to access the **Licenses** page where you can view all of the licenses for your configured clouds.

Cloud Accounts and Subscriptions

(Optional) From the **Cloud Accounts and Subscriptions** box, you can enter the ID numbers (one per line starting with #) for each of your cloud accounts, subscriptions, organizations, and projects that you wish to permit to use your CCKM. If you leave this box blank, users from other accounts can also use your CCKM to manage their cloud keys as long as they have valid cloud login credentials and can access your CCKM User Portal.

Settings

Select **Settings** from the left-hand navigation bar within the CCKM Admin Portal to access the **Settings** page. From the **Settings** page, you can:

- Set or modify the MongoDB settings
- Set or modify the proxy server settings
- Set or modify the SMTP settings
- Set or modify the debug log levels

- Configure key management by allowing CCKM administrators the capability to allow or disallow users to delete source keys
- Add and manage the administrator users and set or modify the password security

Configure Database

In the **Database** tab within the **Settings** page, you have the following options to configure MongoDB:

- **Local Database**
- **Remote Database**
- **ReplicaSet Database**
- **MongoDB Atlas**

The CCKM server requires a MongoDB database to store user setup and cloud usage data.

Prior to configuring any of the MongoDB options from the Database page, the MongoDB database server(s) must be deployed. The CCKM solution also includes a built-in version of MongoDB, which is to be used for evaluation and PoC purposes. For the prerequisites ***prior*** to performing the steps in this section, see Chapter 5: "Set Up MongoDB" on page 35.

> **Note**
> CCKM supports both MongoDB and MongoDB Atlas.

When configuring your database, you have the option to perform authentication of the MongoDB users through an AD server using LDAP. Perform the following steps *prior* to enabling this feature in this section: In the AD server, create a user "kmaas_user", and add it to AD group "dbOwner".

Also when configuring your database, it is required that you enter a passphrase. This passphrase is used to derive a master key. The master key, in turn, is used to encrypt all passwords, including the key source (CipherTrust Manager or DSM) and MongoDB passwords, and other sensitive user credentials that are entered into CCKM. The passphrase is shared among all CCKMs within a multi-CCKM environment and must be entered as part of the configuration of each additional CCKM server.

> **Note**
> After the passphrase is saved to CCKM, it cannot be changed or deleted. It is important that you note this passphrase for future use. If you configure additional CCKMs for a multi-CCKM environment, then you are required to enter the same passphrase when configuring these CCKMs.

After setting up MongoDB settings successfully, CCKM restarts.

Configure Local Database

If you are using the built-in MongoDB database server within CCKM (for evaluation and PoC purposes only), select to use the local database option from the **Database** page:

1. In **Database** page, select **Local Database**. The following boxes within the **Use Local Database** page are automatically filled with the default values:
   Database IP:localhost
   Port:27017
   Database Name: kmaas
   Database Login Credentials:

Username: kmaas_user

Password: ******

2. In the **Configuration Passphrase** box, enter a passphrase. The passphrase must be between 12 and 32 characters long (standard ASCII alphabet characters (a-z, A-Z), integers (0-9), and a limited set of special characters (!@#$%^&*(){}[])). **It is important that you note this passphrase and store it securely.**
3. Click **Test Connection** to verify that the connection between the MongoDB database server and the CCKM works. If the test is successful, 'the **Save** button is enabled and you are prompted to save the database setting.
4. Click **Save**. Thereafter, the CCKM Admin and User portals restart. **Be sure to wait for Admin Portal to restart before performing other operations.**

Configure Remote Database

If you are using an external or remote MongoDB database server (single instance), select to use the remote database option from the **Database** page:

1. In **Database** page, select **Remote Database**.
2. In the **Configuration Passphrase** box, enter a passphrase. The passphrase must be between 12 and 32 characters long (standard ASCII alphabet characters (a-z, A-Z), integers (0-9), and a limited set of special characters (!@#$%^&*(){}[])). **It is important that you note this passphrase and store it securely.**
3. In the **Database IP** box, enter the IP address of the MongoDB database server.
4. In the **Database Hostname** box, enter the host name of the MongoDB database server.
5. In the **Port** box, enter the port number of the MongoDB database server.
6. Optional. To use TLS or SSL for a secure connection to the MongoDB, select the **Enable TLS/SSL for secure connection to MongoDB** check box. The **Upload SSL Certificate box** displays. If you performed this step, proceed to the next step. Otherwise, skip to step 8.
7. From the **Upload SSL Certificate** box, click **Choose File** to select the MongoDB certificate authority (CA) file for TLS used to authenticate the MongoDB certificate.
8. In the **Database Name** box, enter the name of the MongoDB database.
9. In the **Database Login Credentials** boxes, enter your username and password for the MongoDB database.
10. Optional. Select the **Use LDAP for authentication** box to enable authentication of the MongoDB users through an AD server using LDAP.
11. Click **Check Database Connection** to verify that the connection between the MongoDB database server and the CCKM works. If the test is successful, 'the **Save** button is enabled.
12. Click **Save**.Thereafter, the CCKM Admin and User portals restart. **Be sure to wait for Admin Portal to restart before performing other operations.**

Configure ReplicaSet Database

If using external or remote MongoDB database servers (replica set), select to use the replica set database option from the **Database** page:

1. In **Database** page, select **ReplicaSet Database**.
2. In the **Configuration Passphrase** box, enter a passphrase. The passphrase must be between 12 and 32 characters long (standard ASCII alphabet characters (a-z, A-Z), integers (0-9), and a limited set of special characters (!@#$%^&*(){}[])). **It is important that you note this passphrase and store it securely.**

3. In the **Replica Set Name** box, enter the name of the MongoDB replica set.
4. For each instance of the MongoDB, enter the IP address, host name and port number of each database servers in the **Database IP**, **Database Hostname**, and **Port** boxes.
5. Optional. If you have more MongoDB instances to add, select the **Add More DB** box and enter the required information.
6. Optional. To use TLS or SSL for a secure connection to the MongoDB, select the **Enable TLS/SSL for secure connection to MongoDB** check box. The **Upload SSL Certificate box** displays. If you performed this step, proceed to the next step. Otherwise, skip to Step 8.
7. From the **Upload SSL Certificate** box, click **Choose File** to select the MongoDB root certificate file used to authenticate the MongoDB certificate.
8. In the **Database Name** box, enter the name of the MongoDB database.
9. In the **Database** Login Credentials boxes, enter your username and password for the MongoDB database.
10. Optional. Select the **Use LDAP for authentication** box to enable authentication of the MongoDB users through an AD server using LDAP.
11. Click **Test Connection** to verify that the connection between the MongoDB cluster and CCKM works. If the test is successful, 'the **Save** button is enabled.
12. Click **Save**.Thereafter, the CCKM Admin and User portals restart. **Be sure to wait for Admin Portal to restart before performing other operations.**

Configure MongoDB Atlas

If using MongoDB Atlas database servers, select to use the MongoDB Atlas option from the **Database** page:

1. In **Database** page, select **MongoDB Atlas**.
2. In the **Configuration Passphrase** box, enter a passphrase. The passphrase must be between 12 and 32 characters long (standard ASCII alphabet characters (a-z, A-Z), integers (0-9), and a limited set of special characters (!@#$%^&*(){}[])). **It is important that you note this passphrase and store it securely.**
3. In the **Atlas Connection String** box, enter the connection string of the MongoDB Atlas. For information on how to obtain this connection string, see "Obtain Connection String for Connection Using a MongoDB Driver" on page 38.
4. In the **Database Name** box, enter the name of the MongoDB Atlas database.
5. In the **Database Login Credentials** boxes, enter your username and password for the MongoDB database.
6. Click **Test Connection** to verify that the connection between the MongoDB Atlas and CCKM works. If the test is successful, 'the **Save** button is enabled.
7. Click **Save**.Thereafter, the CCKM Admin and User portals restart. **Be sure to wait for Admin Portal to restart before performing other operations.**

Set Proxy Settings

Select the **Proxy** tab from the **Settings** page to set (or modify) the proxy server settings.

> **Note**
> In the case of a multi-instance CCKM, proxy configuration is not propagated to other instances in the CCKM cluster. You are required to manually configure the proxy settings on each instance in the **Proxy** tab within **Settings** page of the Admin Portal. **If you edit the proxy settings, it is recommended that you logout of the CCKM Admin Portal and then login again to ensure the new settings become effective.**

1. In the **Settings** page, select the **Proxy** tab. The **Proxy Settings** page displays.
2. In the **Hostname** box, enter the hostname or IP address of the proxy server.
3. In the **Port** box, enter the port number of the proxy server.
4. In the **Skip Proxy List** box, enter the hostnames (separated by a pipe symbol “|”) that are to pass through the proxy server.
5. (Optional) In the **Username** box, enter the username of the proxy port.
6. (Optional) In the **Password** box, enter the password for the proxy server.
7. Click **Test Connection** to verify that the connection between the proxy server and the CCKM works. If the test is successful, the **Save** button is enabled.
8. Click **Save** to save the setting.

Set SMTP Settings

In CCKM, SMTP is used to send email alerts for user actions performed and scheduled for the supported clouds. Set optional notification information.

> **Note**
> A single configured SMTP host is common across all configured clouds. Therefore, the SMTP-related configurations that you enter in the SMTP page within the Settings page are applicable to all clouds that you configure.

Select the **SMTP** tab from the **Settings** page to set (or modify) SMTP settings.

1. In the **Settings** page, select the **SMTP** tab.
2. In the **Hostname** box, enter the hostname of the SMTP host.
3. In the **Port** box, enter the port number of the proxy server.
4. In the **From** box, enter the sender's email address.
5. (Optional) In the **Username** box, enter the username of the SMTP port.
6. (Optional) In the **Password** box, enter the password for the SMTP server.
7. Click **Save** to save the setting.

Set Debug Log Level

You can set or modify your the debug log levels (applied to logs that are stored in the file) from the **Debug Logs** tab within the **Settings** page. The following are the debug log levels you can set:

- Trace (Recommended for debugging issues in health check, DSM High Availability (HA), and propagation in cluster)
- Debug (Recommended for debugging any other issues)

- Info (Recommended for normal usage)
- Warn
- Error

To set the debug log levels, do the following:

1. Select the **Debug Logs** tab from the **Settings** page.
2. Select the Log level.
3. Click **Save**.

Configure Key Management

The **Key Management** tab from the **Settings** page provides CCKM administrators the capability to allow or disallow users to delete source keys by selecting or deselecting the **Allow Source Key Deletion** check box. When this check box is deselected, the **Delete** option in the **Actions** column of the **Key Sources** page is unavailable (not displayed as an option), and the calling of the REST API to delete a source key is denied. By default, the **Allow Source Key Deletion** check box is selected, which allows the deletion of source keys using the **Delete** option or the REST API call.

To manage the deletion of source keys:

1. In the **Settings** page, select the **Key Management** tab. The **Key Management Configuration** page displays. By default, the **Allow Source Key Deletion** check box is selected.
2. To disallow the deletion of source keys, deselect the **Allow Source Key Deletion** check box.
3. Click **Save Settings** to save the new setting.

Manage Users

Select the **User Management** tab from the **Settings** page to manage the users and set or modify the password security.

The CCKM user has the ability to create, enable, disable, and delete additional users from the **User Management** tab.

> **Note**
> A user with the username of 'admin' cannot be deleted.

Add User

1. Log in to CCKM, select the **Settings** icon, and choose the **User Management** tab.
2. Click **Add User**.

   The **User** dialog box is displayed.
3. Enter the required and optional fields:
   - **Username:** required
   - **First Name**, **Last Name**, **Email:** optional
   - **Password/Confirm password:** required. Must be 8 to 20 characters, with at least one capitalized letter, one lowercase, and one digit.
4. Select the **Active** check box (unless this user should be disabled).
5. Click **Submit**.

Disable User

There are two places from which to disable an administrator user:

1. Within the **Actions** column from the **User Management** page, click **Disable** next to the user's name.
2. Within the **Actions** column from the **User Management** page, click **Update** next the user's name.
3. From the **User** dialog box, you can change First Name, Last Name, Email, and clear the **Active** check box.
4. Click **Submit**.

Enable User

There are two places from which to enable an administrator user:

1. Within the **Actions** column from the **User Management** page, click **Enable** next to the user's name.
2. Within the **Actions** column from the **User Management** page, click **Update** next the user's name.
3. From the **User** dialog box, you can change First Name, Last Name, Email, and clear the **Active** check box.
4. Select the **Active** check box.
5. Click **Submit**.

Delete User

> **Note**
> The "admin" user cannot be deleted.

1. Within the **Actions** column from the **User Management** page, click **Delete** next to the username of the user you wish to delete. The **Delete User Confirmation** dialog box displays.
2. Click **Confirm** to confirm the deletion of the username.

Reset Password

The password of the "admin" user cannot be reset through the CCKM Admin Portal. You must use the CCKM CLI to reset password (**resetwebadmin**) of this user. For more information, see Appendix A: "CCKM CLI Commands" on page 214.

To reset the password of a user (besides the "admin" user), do the following:

> **Note**
> By default, the password you enter below will expire soon after entering it. You are prompted to change this password when you log into the Admin Portal *after* resetting the password.

1. Within the **Actions** column from the **User Management** page, click **Reset** next to the username of the user whose password you wish to reset. The **Enter New Password** dialog box displays.
2. Enter the new password and then enter the password again to confirm it.
3. Click **Confirm**. By default, this password will expire after entering it.
4. Log into the Admin Portal using the new password. You are prompted to change the password.

Update Admin Password Security

Any CCKM user can update the Admin password security information.

1. Log in to CCKM, select the **Settings** icon, and choose the **User Management** tab.
2. Click **Password Security**.

   The **Update password security** dialog box is displayed.
3. In the **Password duration (days)** box, enter the number of days in which the admin password expires. The password expiration interval is applied globally to each administrator account. If the administrator does not change the password prior to the expiration, the administrator must reset the password immediately the next time the administrator logs in. The expiration interval is an integer between 6 and 365. The default is 90. The password duration must be set to a value greater than 'Password Expiration Notification'.
4. In the **Password history** box, enter the number of times the current password is permitted to be reused. CCKM maintains a password history. You cannot use the same password more than once per the set limit. The default is 4, and the maximum value that can be set is 12. You can set this value to '0' to permit reuse of the current password.
5. In the **Minimum password length** box, set the minimum number of characters, including blank spaces that must be in a password. The minimum password length is an integer between 8 and 256. The default is 8
6. In the **Password expiration notification (days)** box, enter the number of days prior to the password expiration at which to begin telling the administrator that their password is about to expire. Administrators are notified of the impending expiration. The notification interval is an integer between 6 and 31. The default is 6.
7. Ensure the **Required upper case** check box is selected to enable the requirement of having at least one uppercase alphabet character in the administrator password. This check box is enabled by default. To disable this requirement, ensure this check box is deselected.
8. Ensure the **Required numbers** check box is selected to enable the requirement of having at least one integer in the administrator password. This is enabled by default. This check box is enabled by default. To disable this requirement, ensure this check box is deselected.
9. Ensure the **Require special characters** check box is selected to enable the requirement of having at least one special character (!@#$%^&*(){}[]) in the administrator password. This check box is enabled by default. To disable this requirement, ensure this check box is deselected.
10. In the **Maximum number of login tries** box, enter the maximum number of unsuccessful login attempts before disabling access for a set interval of time. The login page for the CCKM Admin Portal becomes inoperable and ignores further login attempts by an administrator for the specified interval. The range is between 1 and 10. The default number is 3.
11. In the **User lockout time (minutes)** box, enter the number of minutes to wait before allowing administrators to login. The default is 30 minutes
12. In the **Auth token validity duration (minutes)** box, enter the number of days in which the admin authentication token expires. The authentication token expiration interval is applied globally to each administrator account. If the administrator does not change the token prior to the expiration, the administrator must reset the password immediately the next time the administrator logs in. The expiration interval is an integer between 5 and 60. The default is 60.
13. Click **Submit**.

Chapter 13: CCKM for AWS Key Management

Overview

CCKM for AWS offers risk management and assessment of AWS customer-managed Customer Master Keys (CMKs) across AWS services. It provides operational visibility of AWS CMKs across AWS services, as well as tools to manage the complete life cycle of keys from on premise or from the cloud.

CCKM for AWS Key Management involves a few concepts specific to AWS:

- **Source key:** Source keys are generated from the supported CCKM key sources (CipherTrust Manager, DSM, or nShield Connect) and can be uploaded to AWS.
- **AWS regions:** Source keys can be uploaded to only one AWS region (chosen from a drop-down menu).
- **Customer Master Key (CMK):** There are two types of CMKs—AWS-managed or customer-managed. The CMK origin can be AWS_KMS (generated in KMS), or EXTERNAL (imported by user). CCKM can manage the life cycle of all keys, but owns the key material of only EXTERNAL keys imported by CCKM itself. CMKs are region-specific, meaning CMK operations and APIs work on CMKs in only one region at a time.
- CMKs can be disabled and enabled. CMKs cannot be deleted immediately, but can be scheduled for deletion 7 to 30 days in advance. CMKs can be identified by Key Id, Key ARN, or alias.
- **Alias:** Use a reader-friendly alias when uploading a source key to AWS to make searching for key names more intuitive. Multiple aliases can be assigned to the same CMK.

- **Delete key:** Unlike some other key management cloud services, the AWS CMK APIs do not provide a direct mechanism for creating backup keys. Therefore, the process of key deletion in the CCKM has several phases.
  - A source key that has been generated in a key source (CipherTrust Manager, DSM, or nShield Connect) but not uploaded or used in AWS, can be deleted from the **Key Sources** page by clicking **Delete Key**.
  - Once a source key has been uploaded to AWS, deleting it will only remove it from the **Key Sources** page, but not delete it from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as the key source). CCKM will keep the key in CipherTrust Manager, DSM or MongoDB (depending on the key source you are using) as a backup. For the uploaded AWS key, you can do "**Delete > Delete Key Material Now**" to remove the key material from AWS, and then later do "Import" to re-import (restore) the key material in AWS.

    After a source key has been uploaded to AWS, to delete it from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as the key source), you must delete both the key uploaded to AWS and the key from the **Key Sources** page. First, schedule a deletion of the uploaded AWS key. After the AWS key is deleted on AWS, synchronize your keys on CCKM. Then delete the source key from the **Key Sources** page.
- **CMK key rotation:** The rotation of CMKs is encouraged as part of cryptographic best practices. Note that CCKM for AWS currently supports key rotation for CCKM keys only. In CCKM, key rotation involves creating and uploading new external CMK and moving the alias(es) from the old (previous key) to this newly uploaded key (current key). This key rotation does not affect the components of the current CMK, such as the key ID, key ARN, and region. The old key can be continued to be used to decrypt the data that was previously encrypted by this key.

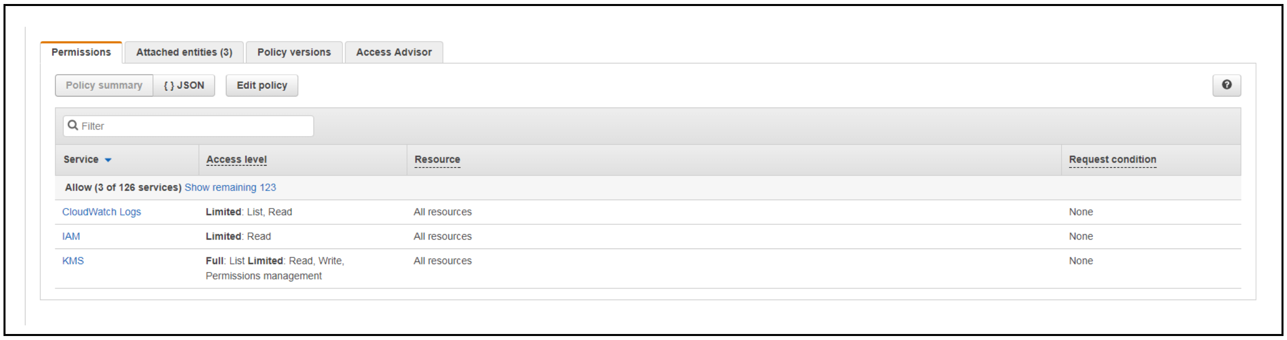
  CCKM allows for a manual or a scheduled rotation of a CMK. Scheduled rotation requires that the key rotation feature be configured in the **Settings** page within CCMK prior to enabling this feature in the **Rotate AWS Key** page.

  Note that by default all CKMs, which the CCKM creates, allow both encrypt and decrypt operations. However, in CCKM, when enabling key rotation (either manually or scheduled), there is an option (on by default) to prevent the current key from encrypting any new data going forward. During the configuration of key rotation, ensure the **Disable Encrypt permission on Current key** check box is cleared to keep the encryption permission on the current CMK key. **Note that for native AWS CMK rotation, the current key, once rotated out, can only be used to decrypt the previous data it encrypted. It cannot be used to encrypt any new data.**

Prerequisites on AWS

You must have:

- KMSFullAccess
- IAM GetUser permissions
- CloudWatch Logs permissions (Limited: List, Read All Resources)



Service	Access level	Resource	Request condition
Allow (3 of 126 services) Show remaining 123			
CloudWatch Logs	**Limited**: List, Read	All resources	None
IAM	**Limited**: Read	All resources	None
KMS	**Full**: List **Limited**: Read, Write, Permissions management	All resources	None

Access AWS Key Management

You can either enter your AWS secret key ID and access key when prompted, or upload an AWS credentials file on the login page. AWS Credentials file should be .json format (can be written in notepad and saved for upload).

For example:

```
{
"aws_access_key_id":"ABCDEFG1234hiJKLMN",
"aws_secret_access_key":"fgreg1234567tg\refg35"
}
```

Access CCKM for AWS

For details about accessing CCKM as a CCKM user for the CCKM service for AWS Key Management, see Chapter 3: "About CCKM Admin and User Portals" on page 24. This chapter includes information on the ways in which to access the CCKM service for Key Management.

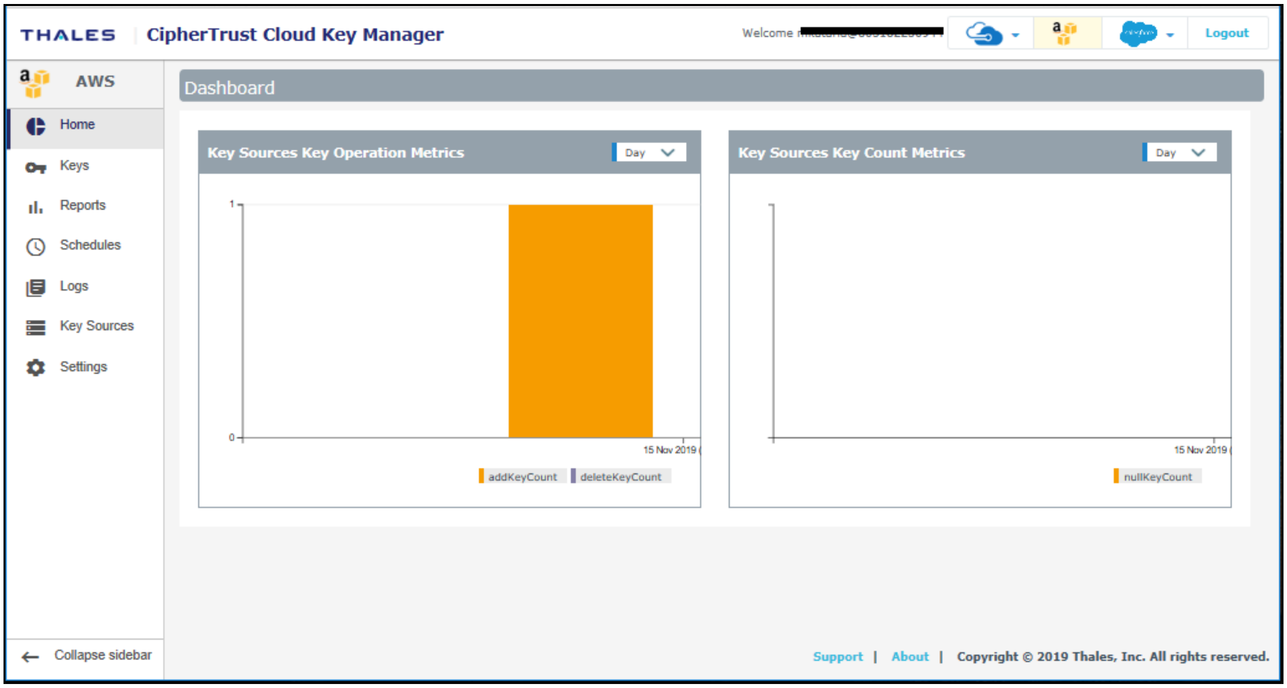
AWS CCKM User Portal

The CCKM User Portal for AWS contains the following menu options in the left-hand navigation bar:

- **Home:** Home page displays key operation and count metrics.
- **Keys:** Keys page displays key management overview and provides the Upload and Synchronize functionality, and the Delete, Restore, and Auto Rotate key operations available in CCKM.
- **Reports:** All Reports overview page links to detailed individual reports.
- **Schedule:** In Schedule, you can schedule automatic key rotation, key rotation based on key expiration, and key synchronization.
- **Logs:** Logs page lists individual logs that can be searched, sorted, and viewed.
- **Key Sources:** In **Key Sources**, you can generate keys in CCKM to upload to AWS. CCKM provides support for the use of CipherTrust Manager and DSM as key sources to generate, store, and retrieve the encryption keys that CCKM uses and manages. CCKM also provides support for the use of nShield Connect HSM as a key source to generate the encryption keys that CCKM uses and manages. These keys are then stored in MongoDB.

  > **Note:** CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

- **Settings:** The **Settings Management** page allows you to configure email alert settings, key alerts, remote syslog server, and proxy server settings. You can also view the release number of the CCKM running from this page.



To display the context of which Key Management service you are currently logged into within CCKM, the icon associated with the Key Management service displays at the top of the left-hand navigation bar.

The **arrow** button at the bottom of the navigation bar allows you to expand the bar to view the names of the menu options and collapse it to view only the icons associated with the menu options. Click the **arrow** button to either expand or collapse the navigation bar.

On the top-right side of the CCKM portal for AWS, the following options are available:

- **Azure Cloud Service Context Switcher:** This context switcher allows you to login into another Azure cloud service different from the current Azure cloud service into which you are currently logged. If you select the Azure cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Azure account(s) displays. For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.

- **AWS Login:** If you are not already logged into AWS, clicking on the AWS Login icon brings you to the AWS Password Login page where you can enter your AWS credentials to access the CCKM AWS cloud (from the current cloud in which you are currently logged). If you are already logged into AWS, clicking on the AWS Login icon brings you to the **Multi-account Login/Logout** window for the AWS account(s). For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.

- **Google Cloud Service Context Switcher:** This context switcher allows you to login into another Google Cloud organization different from the current Google Cloud organization into which you are currently logged. If you select the Google Cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Google account(s) displays. For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.

- **Salesforce Cloud Service Context Switcher:** This context switcher allows you to login into another Salesforce cloud service different from the current Salesforce cloud into which you are currently logged. This is either Salesforce or Salesforce Sandbox. If you select the Salesforce cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Salesforce account(s) displays. For more information about the **Multi-account Login/Logout** window,see "Support for Multiple accounts or Multiple Users per Cloud Service in

CCKM UI" on page 29.

- **Logout:** Logout from CCKM.

Keys

The **AWS Keys** page allows Cloud administrators to view all of the keys and manage them within the boundary of granted permissions. The keys or key vaults that do not have List permission for the respective Cloud administrator are not listed.

At the top right of the **Keys** page, the following buttons are available:

- **Export Keys:** Allows you to export the contents of the Keys list into a report in a CSV format.
- **Alias View:** Allows you to view the Keys list by the key alias names.
- **New Key:** Allows you to upload an external key (BYOK) or create a native key.
- **Synchronize:** Allows you to download any keys that were created in AWS, outside of the CCKM portal, into CCKM.

In this section, create, delete, schedule delete, import, update, rotate, synchronize, exports tenant secrets are described.

You can view the **AWS Keys** page by the **Keys** view or the **Alias** view. Here is the **Keys** view with **Key ID** as the first column.

To view the **AWS Keys** page by the **Alias** view, click **Alias View**. Any alias you assigned to the source keys when uploading these keys to AWS is shown in the Alias column, the first column listed within this view.

A **Search** box is available at the top the **AWS Keys** page allowing for a search of a specific key by entering any one of the following parameters:

- **Key ID:** The globally unique identifier for the CMK.
- **Source Key:** The name of the source key. This is the key generated in the key source you are using (CipherTrust Manager, DSM, or nShield Connect).
- **Region:** The AWS Region to which the key is uploaded.
- **ARN:** Amazon Resource Name of the CMK.
- **ARN description:** The description given to the CMK's ARN.
- **Origin:** The source of the key material. The valid values are:
  - AWS_KMS: Key material was created by AWS KMS.
  - EXTERNAL: Key material was imported from CCKM or another key management system.
- **Key Manager:** The manager of the CMK. The valid values are:
  - AWS: AWS-managed CMK.
  - CUSTOMER: Customer-managed CMKs that you create, use, and manage.
- **Key State:** The state of the CMK. The valid values are:
  - Enabled
  - Disabled
  - PendingDeletion
  - PendingImport

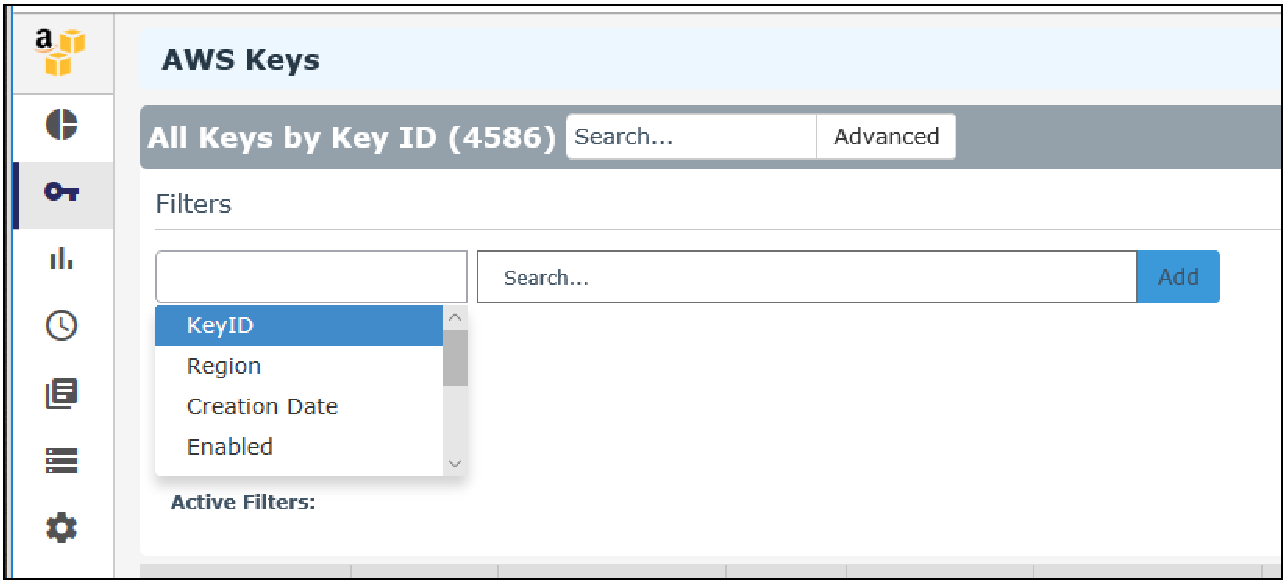
- **Key Usage:** The cryptographic operations for which the CMK is used. The only valid value is ENCRYPT_ DECRYPT.
- **Expiration Model:** Specifies whether the key material of the CMK expires. This value is present only when the source of the key material was imported from CCKM or another key management system. The valid values are:
  - KEY_MATERIAL_EXPIRES
  - KEY_MATERIAL_DOES_NOT_EXPIRE
- **Tags:** The label assigned to the key, which consists of a customer-defined key and an optional value. For example, department finance.
- **Key Aliases:** The alias or aliases assigned to the key.

The **Advanced** tab allows you to filter your search on the each of the columns displayed on the **AWS Keys** page with the exception of the **Action** columns:

- Key ID
- Region
- Creation Date—Enter START DATE and END DATE.
- Enabled— Available options from the drop-down menu are ENABLED and DISABLED.
- Key State— Available options from the drop-down menu are Enabled, Disabled, PendingImport, PendingDeletion
- Scheduled Deletion Date—Select a START DATE and END DATE.
- Key Material Expiration Date—Select a START DATE and END DATE.
- Key Material Origin
- Rotation Status— Available options from the drop-down menu are ACTIVE and ARCHIVED.

Each of these columns are described in more detail below in this section.

To select to filter based on a specific column from the **AWS Keys** page, click **Advanced**. The **Filters** section displays. From the column filter drop-down menu, select the name of the column on which to filter your search. For example, select **Region**. In the **Search** box, enter the value on which to search (for example, **US East (Ohio)** as the region name) and click **Add**. For the **Enabled**, **Key State**, and **Rotation Status** columns, a drop-down menu is available from which to select a value for the search. Click the **down** arrow, select a value from the available options, and then click **Add**. For the **Creation Date**, **Scheduled Deletion Date**, and **Key Material Expiration Date**, select a start and end date for the search and then click **Add**. The specified search filter is then added to **Active Filters**. In the example, **Region: US East (Ohio)** is added to **Active Filters**. The results of the search displays in the **AWS Keys** table. You can narrow your search by selecting more columns to add to the filter. To clear the filtered results, from **Active Filters**, click on the “X” (or the **Delete** button) for each of the named filters.



The following columns display on the AWS Keys page:

- **Key ID:** The globally unique identifier for the CMK.
- **Region:** The AWS Region to which the key is uploaded.
- **Creation Date:** Indicates the date on which the key was created.
- **Enabled:** Indicates whether the key is enabled for use.
- **Key State:** The state of the CMK. The valid values are:
  - Enabled
  - Disabled
  - PendingDeletion
  - PendingImport
- **Scheduled Deletion Date:** Indicates the date on which the key is scheduled to be deleted.
- **Key Material Expiration Date:** Indicates the date on which the key material is set to expire.
- **Key Material Origin:**
  - If the key was uploaded from CCKM, then the value is INTERNAL(<source_key_name>).
  - If the key was not uploaded from CCKM as a BYOK key, then the value is EXTERNAL.
  - If the key material was created by AWS KMS, then the value is KMS.
- **Rotation Status:** Indicates whether the key is in active rotation schedule or if it has been rotated in the past.
  - Auto—if the autorotation is currently enabled.
  - Active—if the key is the most latest in the rotation sequence.
  - Archived—if the key has been rotated out.

- **Actions:** Allows you to select any of the following actions on a given key:
  - Rotate
  - Delete
  - Schedule Delete
  - Cancel Schedule Delete
  - Import
  - Update

Create a Key

You have two options to create a new key:

- Upload an external key (BYOK)
- Create a native key

Upload your Own Key (BYOK)

Upload the AES256 keys created from the **Key Sources** page of CCKM to AWS. These are your "BYOK" keys.

To upload your BYOK key to CCKM:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, select the **New** button and then **Upload**. The **Upload a Key** dialog box displays.
3. From the **Region** box, select the AWS Region to which you will upload the key.
4. From the **Key Provider** box, select **CipherTrust Manager, DSM**, or **nShield** depending on the key source you are using for the key.
5. From the **Source Key** box, select one of the available source keys. The keys available in the drop-down menu are the keys generated from CipherTrust Manager, DSM, or nShield Connect depending on the key source you are using.
6. In the **Alias** box, assign an alias for intuitive key name searching and for use in key rotation. Required by AWS Management Console.

   > **Note:** In order to rotate keys, you must assign an alias, when either uploading or using the Update Key function.

7. (Optional) Enter the expiration date of the key material in the **Key Material Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
8. (Optional) In the **Description** box, enter a description for the key you are uploading.
9. (Optional) In the **Tags** boxes, enter the tag name and value of the key and then click the **+** button.
10. (Optional) In the **Allow Access for External Accounts** box, enter the account ID of an external AWS account with which the key is shared and then click the **+** button. This step is applicable only if you are to share this key with an external AWS account.
11. Click **Upload to AWS**.

    (Reset clears the page before it is uploaded, if necessary.)

    The uploaded key is displayed in the list.

Create a Native Key

Create a native key in AWS KMS using CCKM. After CCKM creates a native key, it displays the key as **KMS** under the **Key Material Origin** column within the **Keys > All Keys** list page in CCKM.

To create a native key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, select the **New Key** button and then **Upload**. The **Create Key** dialog box displays.
3. From the **Region** box, select the AWS Region to which you will add the new key.
4. In the **Alias** box, assign an alias for intuitive key name searching and for use in key rotation. Required by AWS Management Console.

   > **Note:** In order to rotate keys, you must assign an alias, when either uploading or using the Update Key function.

5. (Optional) In the **Description** box, enter a description of the key.
6. (Optional) In the **Tags** boxes, enter the tag name and value of the key and then click the **+** button.
7. (Optional) In the **Allow Access for External Accounts** box, enter the account ID of an external AWS account with which the key is shared and then click the **+** button. This step is applicable only if you are to share this key with an external AWS account.
8. Click **Create**.

   (Reset clears the page before it is created, if necessary.)

   The new key is displayed in the list.

Delete a Key

From the **Keys** page, deleting a key deletes it from AWS. AWS requires a minimum wait period of seven days before it fully deletes the entire key from AWS.

To delete a key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. Click **Delete** next to the target key.
3. In the **Delete Key** dialog, choose:
   - **Delete key material now**

     If you choose this option, then the **Action** column for the key will include a **Schedule Delete** link. While the key material will have been deleted from AWS, you still must enter a time of 7-30 days after which the remaining key metadata will be removed. Until that scheduled deletion, you can still Import the key again to AWS.

     The **Action** column will also show an **Import Key** link, whereby you could re-import the key to AWS before the scheduled deletion is completed.
   - **Schedule to delete key after <enter between 7 -30> number of days**

     If you choose this option, then nothing is deleted from AWS until the designated time frame, then the key material and metadata will be deleted simultaneously and cannot be reimported. During the “Pending” period, you can **Cancel Delete** from the **Action** column.
4. Type or copy/paste the confirmation phrase in the Warning box.

5. Click **Delete** to confirm.

   The full deletion will occur after the 7-30 period has passed. Until then, the **Action** list for the key will include **Import** key and **Schedule Delete** (if applicable).

Schedule Delete

See "Delete a Key" on the previous page. If you chose "Delete key material now" use the **Schedule Delete** link to a Pending period and finalize the deletion.

Import a Key

See "Delete a Key" on the previous page. If the source key originated in CCKM and the key material has been deleted from AWS but the full deletion has not completed, use **Import** to re-import the key from CCKM to AWS.

Update Key

Use **Update** to do the following:

- Create and/or assign an alias to a key.
- Adjust the Enabled status, Description, or Tags.
- Add the account ID of an external AWS account with which the key is shared.
- Add an AWS key policy used to share the key with an external AWS account.

> **Note**
> Keys cannot be updated when they are in a PendingImport or PendingDeletion state.

To update a key:

1. On the **Keys** list page (Keys View or Alias View), click **Update** in the **Action** column for a particular key.
2. From the **Key Properties** tab, select an available alias or enter a new alias name in the **Alias** dialog. Click **+**.
3. Adjust the **Enabled** check box, **Description**, or **Tags**.
4. Click **Update**.
5. (Optional) From the **Key Policy** tab, in **Default View**, in **Allow Access for External Accounts**, click **+** and enter the account ID of an external AWS account with which the key is shared. This step is applicable only if you are to share this key with an external AWS account and you have not yet customized the AWS key policy.

   You also have the option to enter an AWS key policy that allows you to share this key with an external AWS account. From the **Key Policy** tab, within **Key Policy** in **Policy View**, enter the AWS key policy copied from the associated JSON file.
6. Click **Save Accounts**.

Rotate Key

CCKM allows for a manual rotation and an autorotation of a key. You manually rotate a key by adding a new key (with the same alias name) using the **Rotate Now** tab in the **Rotate AWS Key** dialog box. Auto rotating a key is adding a new key (with the same alias name) at a scheduled time. Key autorotation requires that you first schedule the autorotation in the **Schedules** page prior to enabling this feature in the **Auto Rotate** tab in the **Rotate AWS Key** dialog box. Note that only source keys that have not yet been uploaded to AWS can be used as new key material in a key rotation.

> **Note**
> Any key that is shared among different IAM users within a single AWS account and is scheduled for autorotation by one IAM user cannot be scheduled for autorotation again by another IAM user. Any key that is scheduled for autorotation displays the status of “AUTO” within the **Rotation Status** column of the **Keys** page. If you attempt to schedule autorotation using **Rotate Key** within the **All Keys** list with a key already scheduled, a message displays indicating that the key is already scheduled for rotation with the scheduled interval, date, and time at which the rotation is to take place. In addition, the name of the user responsible for scheduling the rotation displays.

Rotate a Key Manually

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. From the **All Keys** list, choose the **Alias** or **Keys** view.
3. Find the row that holds the key you wish to rotate and select **Rotate Key** from the **Actions** column.

   The **Rotate AWS Key** dialog box is displayed.
4. Select the **Rotate Now** tab.
5. From **Key Provider** box, select **CipherTrust Manager, DSM**, or **nShield** depending on the key source you are using for the key.
6. From the **Source Key** list, select a new source key.

   Only source keys that have not yet been uploaded to AWS are listed. The **Aliases** box displays the alias associated with the selected key.
7. Deselect the **Disable Encrypt permission on Current key** check box, if you want to continue using the key that has been rotated out to encrypt new data. By default, this check box is selected. Best practice is not to use the old key that has been rotated out to encrypt new data. Strictly maintain the rotated-out key so that KMS can continue using it to decrypt the previously encrypted data by that key.
8. (Optional) Enter a description of the new CMK key in the **Description** box.
9. Click **Rotate**.

Enable Autorotation of Key

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. Choose **Alias** or **Keys** view.
3. Find the name of the key for which to enable autorotation and select **Rotate Key** from the Actions drop-down menu. The **Rotate AWS Key** dialog box is displayed.
4. Select the **Auto Rotate** tab.
5. From **Enable autorotation on this key?**, select **On** to enable the autorotation of this key. By default, this toggle is initially set to **Off**.

   > **Note:** If the toggle for **Enable autorotation on key expiration?** is displayed, skip it. This option displays only if the key has an expiration date associated with it. This option is not applicable to enabling autorotation of the key.
6. From the **Schedule** drop-down menu, select a rotation schedule to which to apply to this key. Any key-rotation schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.

7. From the **Key Provider** box, select **CipherTrust Manager, DSM**, or **nShield** depending on the key source you are using for the key.
8. Deselect the **Disable Encrypt permission on Current key** check box, if you want to continue using the key that has been rotated out to encrypt new data. By default, this check box is selected. Best practice is not to use the old key that has been rotated out to encrypt new data. Strictly maintain the rotated-out key so that KMS can continue using it to decrypt the previously encrypted data by that key.
9. Click **Submit**.

Synchronize Keys

The first login into the CCKM AWS automatically synchronizes keys from all AWS regions. If you make a key status change in AWS, you must synchronize to see the latest status in the CCKM GUI. If you schedule a key deletion, then after the designated time period has passed, CCKM will delete the key from AWS. You are then required to synchronize your keys from CCKM to remove this key permanently from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as your key source).

You have the option to schedule key synchronization. Perform this scheduling in the **Schedules** page.

To synchronize your keys:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. Click **Synchronize**. A warning box is displayed; click **Synchronize** again to proceed with the synchronization.

   New keys that are created in AWS are retrieved and displayed in the CCKM UI.

Enable Autorotation before Key Expiration

To enable autorotation of a key before it expires, you must first set up a key-expiration schedule in the **Schedules** page. Once the key is enabled for autorotation, the key will be rotated before it expires. After the key is autorotated, the key will have a new expiration date based on the key-expiration schedule.

To enable autorotation of a key before it expires:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. Choose **Alias** or **Keys** view.
3. Find the name of the key for which to enable autorotation before key expiration and select **Rotate Key** from the Actions drop-down menu. The **Rotate AWS Key** dialog box is displayed.
4. Select the **Auto Rotate** tab.
5. From **Enable autorotation on key expiration?**, select **On** to enable the autorotation of this key before it expires. By default, this toggle is initially set to **Off**.

   > **Note:** Skip the **Enable autorotation on this key?** as it is not applicable to enabling autorotation of this key before it expires.

6. From the **Schedule** drop-down menu, select a key-expiration schedule to which to apply to this key. Any key-expiration schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
7. From the **Key Provider** box, select **CipherTrust Manager, DSM**, or **nShield** depending on the key source you are using for the key.

8. Deselect the **Disable Encrypt permission on Current key** check box, if you want to continue using the key that has been rotated out to encrypt new data. By default, this check box is selected. Best practice is not to use the old key that has been rotated out to encrypt new data. Strictly maintain the rotated-out key so that KMS can continue using it to decrypt the previously encrypted data by that key.
9. Click **Submit**.

Export Keys

Export Keys allows you to export the contents of the **Keys** list into a report in a CSV format.

1. In the left navigation bar, click **Keys**.
2. Click **Export Keys**.

   This will download all of the keys in a report in a CSV format.

Reports

It is necessary to set up the CloudTrail and CloudWatch services in AWS before AWS keys will be represented in the CCKM reports. You also have the option to set these services up in CCKM using the **General** tab in the **Settings** page. For more information, see "Configure CloudTrail and CloudWatch in AWS to Use CCKM Reports" below.

Choose Reports in the left-hand navigation to access the following reports:

- **Combined Key Activity Reconciliation Report**

  Click **Run Report** to generate a new list of events connected to a key.
- **Key Activity Report**

  Illustrates the activity type performed by a specific user and at the stated time.
- **Key Service Usage Report**

  Indicates the:
  - Key name
  - Origin: AWS KMS or External
  - Requesting Service: Name or ID of the service using a key
  - Region
  - Request Timestamp: Time when the application used the key.
- **Key Aging Report**

  Provides information about key expiration using the following columns:
  - Key Name
  - Region
  - Scheduled key deletion date
  - Key expiration: Date and time at which this key expires.
- **Cloud Key Manager User Action Report**

  Lists the users and their actions in creating and deleting keys in the CCKM portal.

Configure CloudTrail and CloudWatch in AWS to Use CCKM Reports

CCKM uses AWS CloudTrail and CloudWatch services to generate reports.

Overview

AWS CloudTrail:

- Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail.
- Actions performed on or by AWS KMS keys are also recorded in CloudTrail.
- CCKM uses these audit records to generate ‘Key reconciliation report’, ‘Key service usage report’ and ‘Key activity report’.

AWS CloudWatch:

- CloudWatch allows the querying and filtering of CloudTrail logs through log groups.
- CCKM uses CloudWatch log group to fetch KMS audit logs and generate reports.

Configure CloudTrail

1. Log in the AWS Management Console.
2. **Important: Select any region. Note down this region.**

   This region will be provided in the CCKM in the last steps.
3. Go to the **CloudTrail** section in **Management Tools**, select **Trail**, and click **Create Trail**. Create trail in CloudTrail with following settings:
   a. Apply trail to all regions
   b. ‘Read/Write events’ in Management events
   c. Skip data events section.
   d. Select any Storage location as per your convenience for logs collected by CloudTrail.

https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-create-and-update-a-trail.html

Configure CloudWatch Log Group

Monitor the new trail using a CloudWatch log group.

1. Configure CloudWatch Logs to receive your logs from CloudTrail so that you can monitor for specific log events.
2. Edit trail created in the previous section.
3. Enter a new or existing CloudWatch log group name. **Note down this log group name**.

https://docs.aws.amazon.com/awscloudtrail/latest/userguide/send-cloudtrail-events-to-cloudwatch-logs.html

Schedules

The **Schedules** page allows Cloud administrators to view and schedule the following job types:

- Key Rotation
- Key Synchronization
- Key Expiration

CCKM provides support for multiple schedules for each of the job types.

At the top right of the **Schedules** page, the **Add Schedule** button is available from which to add a scheduled job type.

Schedule Key Rotation

CCKM provides support for multiple schedules of a rotation of an AWS key. A new version of the key is added to the AWS key during a scheduled key rotation. Scheduled rotation requires that this feature be configured in the **Schedules** page prior to enabling the feature in the **Auto Rotate Key** dialog box (from the **Actions** column of the **All Keys** list). Your AWS key credentials (AWS Access Key ID and AWS Secret Access Key) are required as part of the configuration and are saved to the key source you are using (CipherTrust Manager or DSM) or to MongoDB (if you are using nShield Connect as your key source). If you have both DSM and nShield Connect configured as your key sources, then your AWS key credentials are saved only to DSM where it is stored.

> **Note**
> CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

During a scheduled rotation of a key, CCKM uses your associated AWS key credentials to rotate the key. If the AWS Secret Access Key expired or changed, then a new Secret Access Key must be reentered. Otherwise, the scheduled rotation will fail.

You can pause or delete a scheduled key rotation in the **Schedules** page. Note that when a key rotation schedule is deleted from CCKM, then the AWS key credentials are also deleted from the key source you are using (CipherTrust Manager or DSM) or from MongoDB (if you are using nShield Connect as your key source).

Set up a Key Rotation Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_ROTATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).

   If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. Set the repeat interval of the key rotation by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

   If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
8. Enter your AWS key credentials in the **AWS Access Key ID** and **AWS Secret Access Key** boxes or click **Choose file** from **Upload Credentials** to upload an AWS credentials file that contains both the AWS Access Key ID and AWS Secret Access Key.
9. Click **Save** to save setting.

Pause a Key Rotation Schedule

To pause a key rotation schedule, select **Yes** for **Pause** within the **Add Schedule** dialog box and then click **Save**. The setting is saved. To resume the rotation schedule after pausing it, select **No** for **Pause** and then click **Save.**The setting is saved.

Delete a Key Rotation Schedule

To delete a key rotation schedule, select **Delete** from the **Actions** drop-down menu from the **Schedules** page. Note that the AWS key credentials (AWS Access Key ID and AWS Secret Access Key) associated with the given key rotation schedule are also deleted from the key source you are using (CipherTrust Manager or DSM) or from MongoDB (if you are using nShield Connect as your key source).

Schedule Key Synchronization

CCKM allows for multiple schedules of the synchronization of AWS keys. Synchronizing downloads any keys that were created in AWS, outside of the CCKM portal, into your CCKM. Scheduled key synchronization requires that this feature be configured in the **Schedules** page. You can also pause or delete a scheduled key synchronization in the **Schedules** page. Note that when a key synchronization schedule is deleted from CCKM, then the AWS key credentials (AWS Access Key ID and AWS Secret Access Key) associated with the given key scheduled synchronization are also deleted from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as the key source). If you have both DSM and nShield Connect configured as your key sources, then the AWS key credentials are deleted from DSM where it is stored.

> **Note**
> CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

Set up a Key Synchronization Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_SYNCHRONIZATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).

   If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. Set the repeat interval of the key rotation by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

   If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
8. Enter your AWS key credentials in the **AWS Access Key ID** and **AWS Secret Access Key** boxes or click **Choose file** from **Upload Credentials** to upload an AWS credentials file that contains both the AWS Access Key ID and AWS Secret Access Key.
9. Click **Save** to save setting.

Schedule Key Expiration

CCKM provides support for multiple schedules of the expiration of AWS keys in the **Schedules** page. You can also pause or delete a scheduled key expiration in the **Schedules** page. Note that when a key expiration schedule is deleted from CCKM, then the AWS key credentials (AWS Access Key ID and AWS Secret Access Key) are also deleted from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as your key source). If you have both DSM and nShield Connect configured as your key sources, then the AWS key credentials are deleted from DSM where it is stored.

> **Note**
> CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

Set up a Key Expiration Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_EXPIRATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. From **Start Date**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.
8. From **Expire a new key after**, enter a number and select the interval unit of Hour, Day, Week, Month, or Year from the drop-down menu after which the new key expires. The supported ranges are 6 through 24 hours, 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.
9. Enter your AWS key credentials in the **AWS Access Key ID** and **AWS Secret Access Key** boxes or click **Choose file** from **Upload Credentials** to upload an AWS credentials file that contains both the AWS Access Key ID and AWS Secret Access Key.
10. Click **Save** to save setting.

Logs

CCKM maintains logs for events, such as login, key generation, synchronizing of keys, and deletion of keys, and displays these logs in the **All Logs** list of the **Logs** page. For each logged event listed on the **Logs** page, the following information is displayed:

- Event name
- Severity level
- Date and time of the event
- Event message
- User (name) associated with the event

You can search for a specific logged event from the **Logs** page using the **Search** box. Allowable filters on the search are event name, severity level, event message, and user (name).

Key Sources

The **Key Sources** page displays all of the keys that are generated from the key source you are using. Three tabs are available at the top of the page:

- **DSM Key:** Use this tab to use DSM as your key source. The keys are generated and stored within DSM.
- **CipherTrust Manager:** Use this tab to use CipherTrust Manager as your key source. The keys are generated and stored within CipherTrust Manager.
- **nShield Key:** Use this tab to use nShield Connect as your key source. The keys are generated within nShield Connect and then stored as blobs within MongoDB.

Keys are listed by name, key type (the cloud in which this key is used), the algorithm of the key, the creation date of the key, key description, and actions you can take regarding that key. Note that the key algorithm currently supported for an AWS key in CCKM is AES256.

A **Search** box is available at the top left of the **Keys Sources** page allowing for a search of a specific key by entering any one of the following parameters:

- Name (key name)
- Key Type
- Algorithm
- Description
- Created By (a key created by a specific user)

Add (Generate) a Key

From the **Key Sources** page, you can add (or generate) keys in CCKM using CipherTrust Manager, DSM, or nShield Connect as a key source. These keys are of the AES256 algorithm type.

Add (Generate) a DSM Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **DSM Key** tab.
3. Select **Add Key > AWS Key**.
4. In the **Add DSM Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to **AWS** and (key) **Algorithm** is fixed to **AES256**.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. Click **Save** to save the new key to CCKM. The new key is added to the key list.

   (Reset clears the screen before it has been saved.)

Add (Generate) a CipherTrust Manager Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **CipherTrust Manager Key** tab.
3. Select **Add Key > AWS Key**.

4. In the **Add CipherTrust Manager Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to **AWS** and (key) **Algorithm** is fixed to **AES256**.

5. In the **Name** box, enter the name of the new key.

6. Enter a description of the key in the **Description** box.

7. Click **Save** to save the new key to CCKM. The new key is added to the key list.

   (Reset clears the screen before it has been saved.)

Add (Generate) an nShield Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.

2. Select the **nShield Key** tab.

3. Select **Add Key > AWS Key**.

4. In the **Add nShield Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to **AWS** and (key) **Algorithm** is fixed to **AES256**.

5. In the **Name** box, enter the name of the new key.

6. Enter a description of the key in the **Description** box.

7. Click **Save** to save the new key to CCKM. The new key is added to the key list.

   (Reset clears the screen before it has been saved.)

Delete a Key

Deleting a key from the **Key Sources** page deletes the key from CCKM. If you delete a key from **Key Sources** that has already been uploaded to AWS, the key will not be deleted from AWS.

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.

2. Select the **DSM Key**, **CipherTrust Manager**, or **nShield Key** tab depending on the key source you used to generate this key.

3. Click **Delete** next to the target key.

4. The **Delete Key** dialog box with a Warning displays asking you to confirm the key deletion by entering the supplied phrase.

5. Enter or copy/paste the confirmation phrase.

6. Click **Delete** to confirm.

Key Sources

The **Key Sources** page displays all of the keys that are generated from the key source you are using. Three tabs are available at the top of the page:

- **DSM Key:** Use this tab to use DSM as your key source. The keys are generated and stored within DSM.

- **CipherTrust Manager:** Use this tab to use CipherTrust Manager as your key source. The keys are generated and stored within CipherTrust Manager.

- **nShield Key:** Use this tab to use nShield Connect as your key source. The keys are generated within nShield Connect and then stored as blobs within MongoDB.

Keys are listed by name, key type (the cloud in which this key is used), the algorithm of the key, the creation date of the key, key description, and actions you can take regarding that key. Note that the key algorithm currently supported for an AWS key in CCKM is AES256.

A **Search** box is available at the top left of the **Keys Sources** page allowing for a search of a specific key by entering any one of the following parameters:

- Name (key name)
- Key Type
- Algorithm
- Description
- Created By (a key created by a specific user)

Manage keys in Key Sources List Page

In addition to adding and deleting keys, the **Key Sources** page allows you to sort the existing keys by Name, Key Type, Algorithm and Created Date.

Settings

To access the **Settings** page in CCKM, select **Settings** from the left-hand navigation bar. From the **Settings** page, you can:

- Enter the emails of the recipients who are to receive alerts and reminders.
- Set up key alerts.
- Enable reporting for AWS keys using an existing AWS CloudTrail and CloudWatch log group or a new trail and log group that you create (from the **Settings** page).
- Configure a remote syslog server to which to send syslog messages.
- View information about the software release number of the CCKM you are running including the build and version number.

Adding Recipients' Emails (for Alerts and Reminders)

Use the **General** settings tab to enter the emails of the recipients who are to receive alerts and reminders from the system. Enter the recipients' email addresses separated by a comma in the **Alert Format** box and then click **Save** to save setting.

Enabling Alerts and Reminders

Use the **Keys Alert** tab from the **Settings** page to specify whether to send an alert to the specified user email(s) when an AWS key is deleted, restored, or uploaded in AWS. You can also use this tab to set up a reminder to manually rotate (or rekey) a key along with the frequency (in days) at which the reminder is sent to the specified user email(s). The user emails are set up in the **General** tab. By default, no alerts or reminders are sent.

To set up key alerts or a reminder to manually rotate (or rekey) a key:

1. In the **Settings** page, select the **Keys Alert** tab.
2. Under **Alerts**, click **On** for each of the alert types and reminder you wish to enable:
   - Rekey Alert (reminder)
   - Delete AWS Key
   - Import AWS Key
   - Upload AWS Key
3. If you set a reminder for **Rekey Alert**, then in the **Days** box, enter the number of days for the frequency an email reminder to manually rotate a key is sent.
4. Click **Save** to save the setting.

Enabling Reporting for AWS Keys

Use the **General** tab within the **Settings** page to enable CCKM reporting for AWS keys using an existing AWS CloudTrail applicable to an AWS region and a CloudWatch log group to which the trail belongs. You can also use the **General** settings tab to create a new CloudTrail and CloudWatch log group to which the trail belongs. You first select an AWS region in which to create it and then add the name of a new CloudWatch log group. Thereafter, you configure the other required configurations.

Enabling Reporting for AWS Keys Using an Existing CloudTrail and CloudWatch Log Group

1. In the **Settings** page, select the **General**tab. The **General** page displays.
2. From the **Select CloudTrail Region** drop-down menu, select the AWS region to which the trail applies.
3. From the **Select CloudWatch Log Group Name** drop-down menu, select the name of the log group to which trail belongs.
4. Click **Save** to save setting.

Enabling Reporting for AWS Keys Using a New CloudTrail and CloudWatch Log Group

1. In the **Settings** page, select the **General**tab. The **General** page displays.
2. From the **Select CloudTrail Region** drop-down menu, select the AWS region in which to create the new trail.
3. From the **Select CloudWatch Log Group Name** drop-down menu, enter the name of the new CloudWatch Log Group to create. This log group receives your logs from CloudTrail so that you can monitor for specific log events. Entering a name of the new log group triggers the display of other configuration boxes associated with creating a new trail and log group.
4. From the **Stack Name** box, enter the name of the new stack to create. The AWS CloudFormation stack automates the creation of all resources required for creating a new trail (bucket, trail, and log group). If a resource fails, the stack rolls back to the previous version. From AWS CloudFormation, you can view information about any failure during the process of the creating the stack using the name of the stack. You can also delete the stack from CloudFormation without any impact to your AWS resources.
5. From the **S3 Bucket Name** box, enter the name of an S3 bucket to designate as the storage of the trail log files. The name must be globally unique. If you do not enter a bucket name, the default bucket name of cckm-bucket is used.

6. From the **Cloud Trail Name** box, enter the name of the new trail. If you do not enter a trail name, the default trail name of **cckm-bucket** is used.

7. From the **CCKM IAM Role Name** box, enter the name of the AWS IAM Role used for CCKM. If you enter an existing IAM role name, then CCKM uses this name to apply to the configuration. If you enter a new IAM Role name, CCKM creates this role with the required permissions to create the resources for the trail. If you do not enter an IAM role name for CCKM, the default role name of **AWS-<*region-name*>-cckm-cloudtrail-to-cloudwatch** is used.

   > **Note:** If you do not have the permissions to create a new IAM role, contact your AWS administrator to create the IAM role with permissions to create and delete log groups, trails and buckets.

8. Enter your AWS key credentials in the **AWS Access Key ID** and **AWS Secret Access Key** boxes or click **Choose file** from **Upload Credentials** to upload an AWS credentials file that contains both the AWS Access Key ID and AWS Secret Access Key.

9. Click **Save** to save setting.

Configuring Syslog Server

Use the **Syslog** tab from the **Settings** page to configure a remote syslog server to which to send syslog messages. Note that the default port number of 514 and the facility name of “LOCAL1” for the syslog server display in the **Port** and **Facility** boxes, respectively. These boxes are not available for modification.

Configure Remote Syslog Server

To configure a remote syslog server:

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.

2. Enter the hostname or IP address of the syslog server in the **Hostname** box.

3. (Optional) Enter a description of the syslog server in the **Description** box.

4. Click **Save** to save the setting.

Delete Remote Syslog Server Configuration

To delete the configuration of a remote syslog server, click **Delete** from the **Syslog** dialog box. No syslog messages are sent to the remote syslog server after the deletion of this configuration.

About

Use the **About** tab to view the release number of the CCKM you are running.

Chapter 14: CCKM for Azure Key Management

Overview

CCKM provides support for Azure clouds (Azure, Azure China, Azure Germany, or Azure Stack). Note that in this chapter, the information regarding Azure is applicable to all Azure clouds unless otherwise noted. The CCKM UI for Azure, Azure China, Azure Germany, and Azure Stack is identical with the exception of the display of the names of the clouds. And for Azure Stack, there are two portals--one for the administrator and one for the tenant.

Accessing and Managing Azure Resources

> **Note**
> This section is only applicable to Azure, Azure China, and Azure Germany. It is **not** applicable to Azure Stack.

Azure allows external applications, such as CCKM, to access and manage resources within Azure in the following ways:

- On behalf of a user
- As a service principal

Both ways require that the application be first registered in Azure. For applications on behalf of a user, it is required that the Azure users be created and given the appropriate access controls and policies on the resources. The application obtains an access token through app+user authenticate. What an application can do with the Azure resources depends on what the user can do.

For as-service-principal applications, the application must be given the appropriate access controls and policies on the resources. The application obtains an access token through an *app-only authentication*. What an application can do with Azure resources depends on whether the service principal has proper permissions on the resources.

Registering Apps in Azure

When registering apps in Azure, the following are the two types of app credentials employed:

- Secret (password)—The user will generate a secret key in Azure when registering the app, and then copy the secret key and provide it to the app. When making OAuth authentication calls, the app will send the secret key to Azure.
- Certificate (public key)—the user will create a private key and public key pair locally, create a certificate for the public key, and then provide the certificate to Azure when registering the app. For the private key, the app will create a client assertion and send it to Azure when making OAuth authentication calls.

CCKM currently supports both types of app credentials for Azure, Azure China, and Azure Germany.

Prerequisites on Azure

The Azure administrator must ensure the following prerequisites are met:

- Each Azure key vault must belong to an Azure Resource Group.
- Key vault owner must give designated users or the CCKM app (if CCKM is used as a service principal) the “Key Vault Contributor” role in Azure Access Control, and appropriate permissions as needed on keys in Access Policies.

  > **Note:** Changing key vault Access Control and/or Access Policies will not affect the active CCKM user sessions. To reflect the change, a user must re-log in to CCKM to establish a new user session.

- CCKM Reports draw on the Azure Log Analytics service, which must be pre-configured if the CCKM Reports functionality is desired (see "Reports" on page 124). At minimum, you must have:
  - An Azure *Log Analytics* service created and associated with the relevant key vaults.
  - In the *Log Analytics* > *{Log service name}* > *Access Control (IAM)* section of the Azure portal, set “**Log Analytics Contributor**” permissions for any user who should be able to generate reports in CCKM. Other users can view, but not create reports.

  > **Note**
  > Azure Stack does not currently support the *Log Analytics* service.

For more information about how to configure Log Analytics in Azure key vaults, see the latest version of the following Microsoft Azure documentation:

https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-azure-key-vault

Access CCKM for Azure Key Management

This section describes the two ways in which to access the CCKM service for Azure Key Management.

Accessing CCKM Using User Login

For details about accessing CCKM as a CCKM user for the CCKM service for Azure Key Management, see Chapter 3: "About CCKM Admin and User Portals" on page 24. This chapter includes information on the ways in which to access the CCKM service for Key Management.

Accessing CCKM Using Service Principal

> **Note**
> This section only applies to Azure, Azure Germany, and Azure China using CCKM as a service principal.

To log into CCKM for Azure using CCKM as a service principal, you are redirected to a CCKM login dialog box for Azure. If this is the first time you are logging into CCKM as a service principal, you (as an administrator user) are required to provide the administrator consent to grant permissions to the CCKM app to access your Azure resources (your organization data). To do this, click the "admin consent" link (indicated by "here" in blue text at the top CCKM login dialog box) after which the CCKM Admin Consent dialog box is displayed. In this box, enter your Azure tenant name, tenant ID, and a new tenant password. After you click on the "Admin Consent" button, you are redirected to the Azure consent page where you can provide your consent (after reviewing the page) by clicking **Accept**. As an administrator user, you can confirm that CCKM has been granted the permissions as a service principal by logging into the Azure portal and verifying that the CCKM app is listed in the **Enterprise Applications > All Applications** page.

To access the CCKM service for Azure Key Management using CCKM as a service principal:

1. In the CCKM Admin Portal, select **Key Management > Azure | Azure Germany | Azure China**.

   You are redirected to the CCKM login dialog box for Azure to use CCKM as a service principal.

2. If this is the first time you are logging into CCKM using CCKM as a service principal, do the following to create a tenant:

   a. Click on the "admin consent" link (indicated by “here” in blue text at the top CCKM login dialog box) to provide your administrator consent to grant permissions to the CCKM app to access your Azure resources. (You must be logged in as an administrator user to provide your admin consent.) The CCKM Admin Consent dialog box is displayed.

   b. In the **Tenant Name** box, enter the tenant name. To locate your tenant name in the Azure portal (after logging in as an administrator user), click **Directory + subscription** icon and view **All Directories**. If more than one directory is listed, find the domain name that is associated with the administrator username within the list. This domain name is the tenant name.

   c. In the **Tenant ID** box, enter the tenant ID. To locate your tenant ID in the Azure portal (after logging in as an administrator user), click **Directory + subscription** icon and view **All Directories**. If more than one directory is listed, find the domain name that is associated with the administrator username within the list. The value that displays under the domain name is your tenant ID. This value is equivalent to the **Directory ID** in Azure. Alternatively, you can locate the tenant ID within in the **Directory ID** box in **Directory properties** of the **Azure Active Directory > Properties** when logged in as an administrator user.

   d. To locate your tenant ID in the Azure portal (after login as administrator user), go to **Azure Active Directory > Properties**. In **Directory properties**, locate the value displayed in the **Directory ID** box. This value is your tenant ID.

   e. In the **Tenant Password** box, enter the tenant password. The password must be at least 8 characters long and contain one upper case letter, one lower case letter, one digit, and one special character from a limited set of special characters "!@#$%^&*()".

   f. In the **Confirm Tenant Password** box, reenter the tenant password.

   g. Click **Admin Consent** to provide your consent to Microsoft. You are redirected to the Microsoft login page to select an account to use for the login.

   h. Select the account. The Microsoft consent prompt is displayed.

   i. Review the information and then click Accept to accept the request to grant permissions to the CCKM app to access your Azure resources as a service principal.

3. In the CCKM Azure login dialog box, in the **Tenant Name** box, enter the tenant name.

4. In the **Tenant Password** box, enter the tenant password.

5. Click **Login**. The Azure CCKM dashboard is displayed.

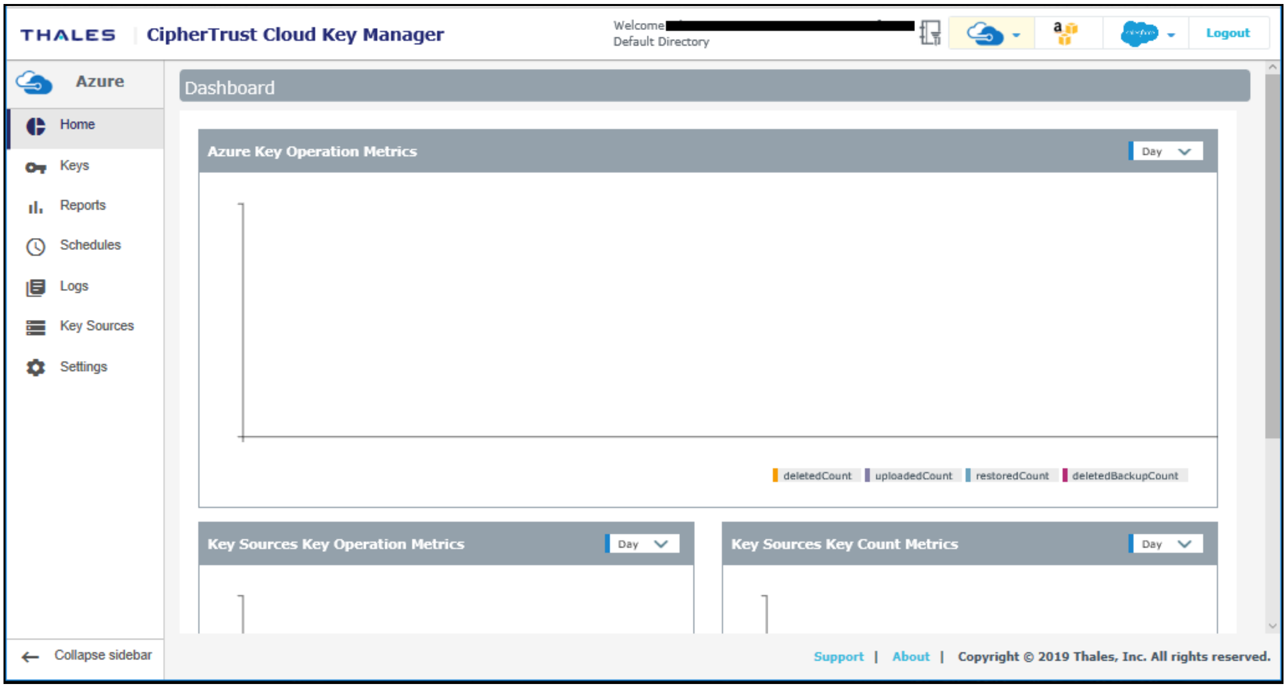
Azure CCKM User Portal

The CCKM User Portal for Azure contains the following menu options in the left-hand navigation bar:

- **Home:** Home page displays key operation and count metrics.
- **Keys: Keys** page displays key management overview and provides the Upload and Synchronize functionality, and the Delete, Restore, and Auto Rotate key operations available in CCKM.
- **Reports: All Reports** overview page links to detailed individual reports.
- **Schedule:** In **Schedule**, you can schedule automatic key rotation, key rotation based on key expiration, and key synchronization.
- **Logs: Logs** page lists individual logs that can be searched, sorted, and viewed.
- **Key Sources:** In **Key Sources**, you can create keys in CCKM to upload to Azure. CCKM provides support for the use of CipherTrust Manager and DSM as key sources to generate, store, and retrieve the encryption keys that CCKM uses and manages. CCKM also provides support for the use of nShield Connect HSM as a key source to generate the encryption keys that CCKM uses and manages. These keys are then stored in MongoDB. Also, supported in **Key Sources** is BYOK for all Azure clouds (including Azure Stack AAD), and PFX file upload (all Azure clouds including Azure Stack AAD and Azure Stack ADFS).

  > **Notes**
  > - CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.
  > - BYOK for Azure Stack ADFS is currently not supported.

- **Settings:** The **Settings Management** page allows you to configure the email settings, key alerts, remote syslog servers to which to send syslog messages, and proxy server settings. You can also view the release number of the CCKM you are running from this page. If you are using CCKM as a service principal, then you can change your tenant password or revoke your administrator consent from this page.



To display the context of which Key Management service you are currently logged into within CCKM, the icon associated with the Key Management service displays at the top of the left-hand navigation bar.

The **arrow** button at the bottom of the navigation bar allows you to expand the bar to view the names of the menu options and collapse it to view only the icons associated with the menu options. Click the **arrow** button to either expand or collapse the navigation bar.

On the top-right side of the Azure CCKM portal, the following options are available:

- **Azure Directory Context Switcher:** This context switcher allows you to switch to another Azure Active Directory to which you belong. For more information about this context switcher, see "Azure Directory Context Switcher (Support for Guest User for AAD B2B Collaboration)" below.

  > **Note:** This context switcher is not applicable to a CCKM that is used as a service principal.

- **Azure Cloud Service Context Switcher:** This context switcher allows you to login into another Azure cloud service different from the current Azure cloud service into which you are currently logged. If you select the Azure cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Azure account(s) displays. For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.
- **AWS Login:** If you are not already logged into AWS, clicking on the **AWS Login** icon brings you to the **AWS Password Login** page where you can enter your AWS credentials to access the CCKM AWS cloud service (from the current cloud service in which you are currently logged). If you are already logged into AWS, clicking on the **AWS Login** icon brings you to the **Multi-account Login/Logout** window for the AWS account(s). For more information about the Multi-account Login/Logout window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.
- **Google Cloud Service Context Switcher:** This context switcher allows you to login into another Google Cloud organization different from the current Google Cloud organization into which you are currently logged. If you select the Google Cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Google account(s) displays. For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.
- **Salesforce Cloud Service Context Switcher:** This context switcher allows you to login into another Salesforce cloud service different from the current Salesforce cloud service into which you are currently logged. This is either Salesforce or Salesforce Sandbox. If you select the Salesforce cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Salesforce account(s) displays. For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.
- **Logout:** Logout from CCKM.

Home

The **Home** page provides a dashboard to display key operation and count metrics.

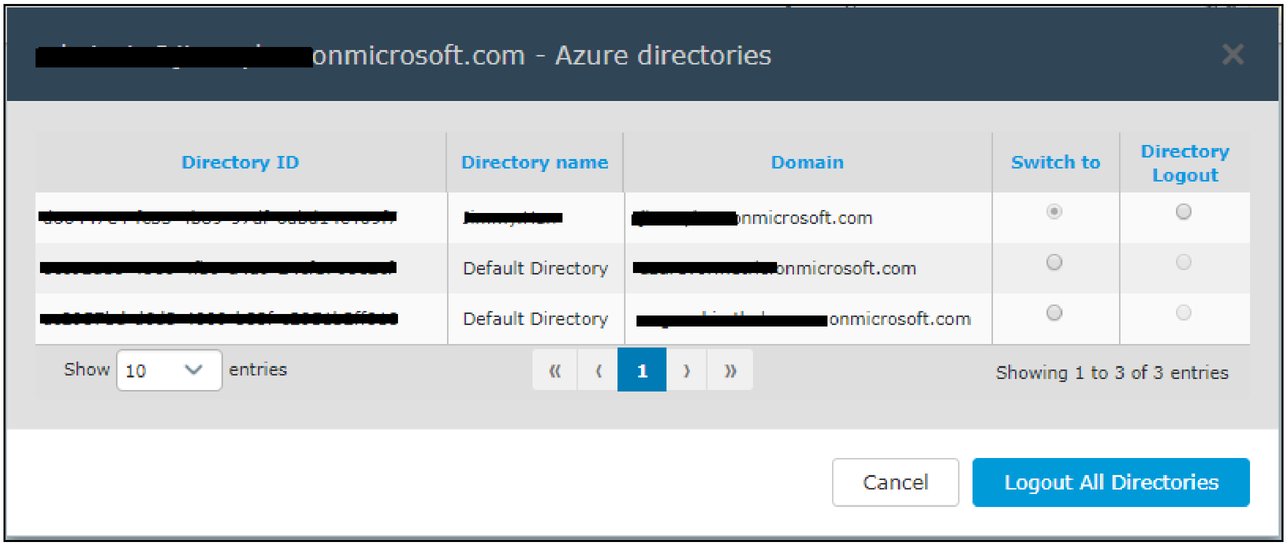
Azure Directory Context Switcher (Support for Guest User for AAD B2B Collaboration)

CCKM supports the ability to allow Azure guest users to take part in B2B collaboration for Azure tenants to share user authentication and applications with partners. Prior to using this feature in CCKM, the guest user account must already be set up in the Azure Active Directory as a guest user with the appropriate access policies applied and the account activated in Azure. Once you are logged into CCKM Azure, from the **CCKM Directory Context Switcher** window, you have the ability to view all of the Azure directories to which you belong, including your default directory and your business partner's directory to which you are set up as a guest user. Using the **CCKM Directory Context**

Switcher window, you can switch to any of the other directories to which you belong by logging into the directory from the same window. Once logged into a given directory, you can access and manage the resources and keys that reside in the given directory.

> **Note**
> The **Azure Directory Context Switcher** is not applicable to a CCKM that is used as a service principal.

The following **CCKM Directory Context Switcher** window displays the Azure directories in which a given user belongs:



[redacted]onmicrosoft.com - Azure directories

Directory ID	Directory name	Domain	Switch to	Directory Logout
[redacted]	[redacted]	[redacted]onmicrosoft.com	●	○
[redacted]	Default Directory	[redacted]onmicrosoft.com	○	○
[redacted]	Default Directory	[redacted]onmicrosoft.com	○	○

Show 10 entries

Showing 1 to 3 of 3 entries

You can perform the following actions using the CCKM Directory Context Switcher window:

- Logon to multiple directories for the Azure cloud
- Logout of a specific directory or all directories
- Perform key management for multiple directories from your current CCKM browser window

Accessing the CCKM Directory Context Switcher Window

To access the **CCKM Directory Context Switcher** window after you are logged into CCKM Azure, click on the **CCKM Directory Context Switcher** icon () at the top of right of the page. The CCKM Directory Context Switcher window displays. The following sections provide instructions on how to use this window.

Logging Out of User Directory

To logout of a specific directory, select the **Directory Logout** button next to the domain name associated with the directory for which to log out. The **Log out CipherTrust Cloud Key Manager** window displays. Click **Continue with Log Out** to log out. You are logged out of the directory and then the CCKM login page is displayed.

Switching to Another User Directory

To access another directory (you have not already logged into), select the **Switch to** button next to the domain name associated with the directory for which to log on. The Microsoft login page displays. Enter your user credentials for the given directory in the login dialog boxes. Once you successfully log into Microsoft, the Azure CCKM portal is displayed.

After you have logged into a directory, you are not prompted to log in again to switch to that directory.

Logging Out of All User Directories

To logout of all directories for Azure, click **Logout of All Directories**. The **Log out CipherTrust Cloud Key Manager** window displays. Click **Continue with Log Out** to log out. You are logged out of all the directories and then the CCKM login screen is displayed.

Keys

The **Keys** page allows Cloud administrators to view all the keys and manage them within the boundary of granted permissions. Those keys or key vaults that do not have List permission for the respective Cloud administrator will not be listed.

A **Search** box is available at the top left of the **Keys**page allowing for a search of a specific key by entering any one of the following parameters:

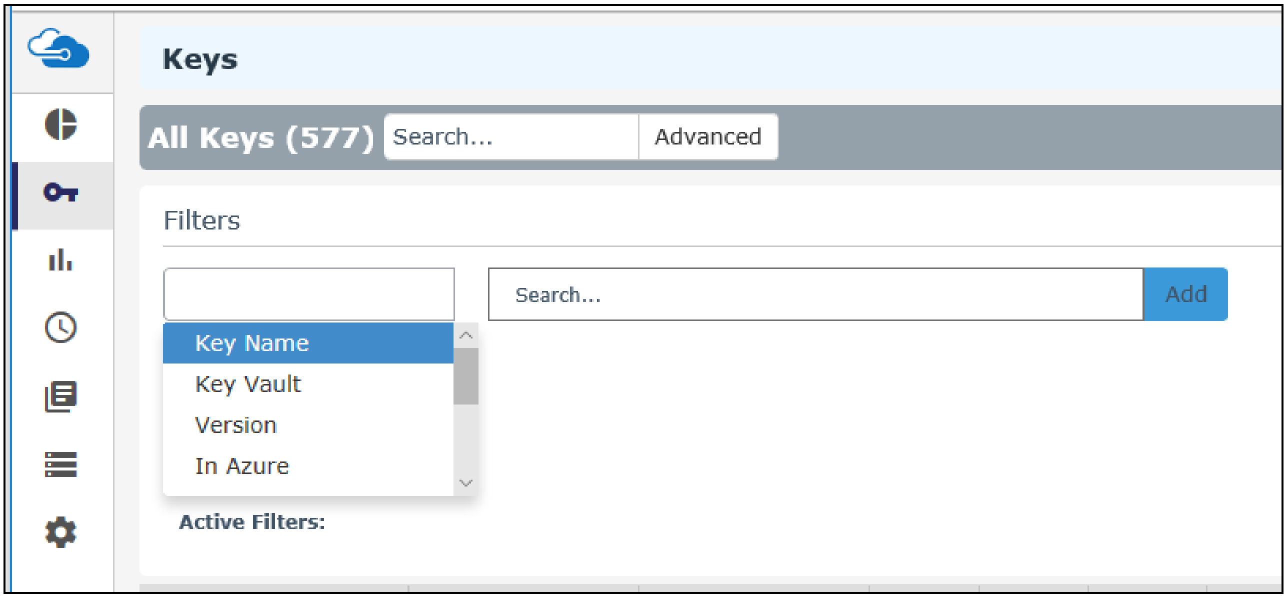
- Key ID
- Version Key ID
- Keyvault name
- Keyvault location
- Source key
- Subscription ID

The **Advanced** tab allows you to filter your search on any or a combination of these columns displayed on the **Keys** page with the exception of the **Count** and **Action** columns:

- Key Name
- Key Vault
- Version
- In Azure—Available options from the drop-down menu are In Azure, Soft Deleted, and Deleted.
- Backup—Available options from the drop-down menu are Backed Up and Not Backed Up.
- Enabled—Available options from the drop-down menu are Enabled and Disabled.
- Location
- Key Material Origin
- Auto Rotate—Available options from the drop-down menu are ON and OFF.

Each of these columns are described in more detail below in this section.

To select to filter based on a column or combination of these columns from the **Keys** page, click **Advanced**. The **Filters** section displays. From the column filter drop-down menu, select the name of the column on which to filter your search. For example, select **Key Name**. In the **Search** box, enter the value on which to search (for example, **MK-createkey** as the key name) and then click **Add**. For the **In Azure**, **Backup**, **Enabled** and **Auto Rotate** columns, a drop-down menu is available from which to select a value for the search. Click the down arrow, select a value from the available options, and then click **Add**. The specified search filter is then added to **Active Filters**. In the example, **Key Name: MK-createkey** is added to **Active Filters**. The results of the search displays in the **Keys** table. You can narrow your search by selecting more columns to add to the filter. To clear the filtered results, from **Active Filters**, click on the "X" (or the **Delete** button) for each of the named filters.



At the top right of the Keys page, the following buttons are available:

- **Export Keys:** Allows you to export the contents of the Keys list into a report in a CSV format.
- **New Key:** Allows you to upload an external BYOK key (CipherTrust Manager, DSM, nShield Connect, HSM .byok or PFX key) or create a native key. For information on how to upload an external BYOK key or create a native key, see "Create a Key" on the facing page.
- **Synchronize:** Allows you to download all the keys that were created in Azure (to which you have access) into CCKM. For information on how to upload a key, see "Synchronize Keys" on page 122.

The following columns display on the **Keys** page:

- Key Name: The name of the key.
- Key Vault: The name of the Azure Key Vault. If the key vault is a premium key vault, then (PREMIUM) will be appended to the key vault name.
- Version: The version number of the key.
- Count: The version count.
- In Azure: Indicates the status of the key in Azure, which can be one of the following:
  - A green check icon indicates the key exists in Azure.
  - An orange exclamation icon indicates the key vault or key was deleted from Azure with the Soft Delete feature enabled.
  - A red exclamation icon indicates the key vault or key was deleted from Azure without the Soft Delete feature enabled.
- Backup: Indicates whether a backup of the key exists in CCKM. The status can be one of the following:
  - A green check icon indicates the backup exists in CCKM.
  - A red exclamation icon indicates the backup key does not exist in CCKM.
- Enabled: Indicates whether the key is enabled for use.
- Location: Indicates the Azure region in which the key is stored. For example, West US or Central West US.

- Key Material Origin:
  - If the key was uploaded from CCKM, then the value is Internal(<source_key_name>).
  - If the key was not uploaded from CCKM as a BYOK key, then the value is EXTERNAL.
- Auto-Rotate: Indicates whether the key is set to be automatically rotated.
- Actions: Allows you to select any of the following actions on a given key:
  - Auto rotate
  - Delete
  - Add Version
  - Recover
  - Purge
  - Delete Backup
  - Restore

Create a Key

You have two options to create a new key:

- "Upload your Own Key (BYOK)" below
- "Create a Native Key" on the next page

Upload your Own Key (BYOK)

Upload the RSA keys created from the **Key Sources** page of CCKM to the Azure Key Vault. These are your "BYOK" keys.

As part of the upload process, Azure generates a corresponding backup key, which CCKM downloads and stores in the key source you are using (CipherTrust Manager or DSM) or in MongoDB (if you are using nShield Connect as your key source). For more information about the backup key, see "Backup Key" on page 1.

> **Note**
> You must have the import permissions for the Azure Key Vault to which you wish to upload the key.

To upload your BYOK key to CCKM:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, select the **New Key** button and then **Upload**. The **Upload a Key** dialog box displays.
3. From **Source Key Tier**, ensure the key source you wish to use is selected. The key sources that display within **Source Key Tier** depends on the key source you are using. The following are the options:
   - **DSM**—Generated a key from the DSM and imported it into a Standard or Premium Azure Key Vault.
   - **CipherTrust Manager**—Generated a key from the CipherTrust Manager and imported it into a Standard or Premium Azure Key Vault.
   - **HSM.byok**—Generated a .byok key from an external Thales HSM and imported it into a Premium Azure Key Vault from CCKM. Once imported from CCKM, this key is stored in the DSM securely as a blob.
   - **PFX**—Imported an external password-protected PFX key into a Standard or Premium Azure Key Vault. Once imported from CCKM, this PFX key will be stored in the DSM securely as a blob.
   - **nShield**—Generated a key from the nShield Connect and imported it into a Standard or Premium Azure Key Vault.

4. From **Source Key** box, select the source key. Enter the first three characters of the name of the key to bring it up for selection.
5. From **Key Vault Tier** box, select the key vault tier of **Standard** or **Premium** depending on the subscription service type you have purchased.
6. From the **Key Vault Region** box, select the region where your key vault resides and whether you want to use the Azure Standard or Premium subscription service. Based on the Standard or Premium Key Vault Tier selection, the regions are filtered accordingly displaying only the regions, which are either Standard or Premium.
7. From the **Key Vault** box, select the name of the Azure Key Vault to where you intend to upload the key.
8. Add the name of the key in the **Key Name** box. If you wish to use the key name you entered when you generated this key from the **Key Sources** page, select the **Use source key name** check box. The name will display in the **Key Name** box.
9. From the **Key Destination** box, select either **Software** or **HSM** (Premium subscription only) depending on which destination you want to use.
10. Click **Enabled**, if the key is enabled. Note that if the key is disabled, the key cannot be used for cryptographic operations.
11. Enter the activation date of the key in the **Activation Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
12. Enter the expiration date of the key in the **Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
13. From **Key Operations**, select the cryptographic operations the key will perform. The options are:
    - Encrypt
    - Decrypt
    - Sign
    - Verify
    - Wrap Key
    - UnwrapKey
14. (Optional) From **Tags**, enter the tag name and value of the key.
15. Click **Upload to Azure**.

    The uploaded key is displayed on the **All Keys** list page.

Create a Native Key

Create a native key in an Azure key vault and select the key types and sizes natively available in Azure. There is no source key tier as the Azure key vault will be the source of the key material. After CCKM creates a native key, it displays as **EXTERNAL** under the **Key Material Origin** column within the **All Keys** list page.

As part of creating a native key, Azure generates a corresponding backup key, which CCKM downloads and stores in the key source you are using (CipherTrust Manager or DSM), or in MongoDB (if you are using nShield Connect as your key source).

To create a native key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, select the **New Key** button and then **Create**. The **Create Key** dialog box displays.
3. From the **Key Vault Tier** box, select the key vault tier of **Standard** or **Premium** depending on the subscription service type you have purchased.
4. From the **Key Vault Region** box, select the region where your key vault resides. Based on the Standard or Premium Key Vault Tier selection, the regions are filtered accordingly displaying only the regions, which are either Standard or Premium.
5. From the **Key Vault** box, select the name of the Azure key vault to where you intend to create the key.
6. Add the name of the key in the **Key Name** box.
7. From the **Key Destination** box, select either **Software** or **HSM** (Premium subscription only) depending on which destination you want to use.
8. From **Key Type**, select **RSA** or **EC** for your key type.
9. If you selected **RSA** for key type, then select **2048**, **3072**, or **4096** from **Key Size** for the key size to apply to the new key. If you selected **EC** for key type, then select **P-256**, **P-384**, **P-521**, or **SECP256K1** from **Elliptical Curve Name** for the name of the elliptical curve to apply to the new version of the key.
10. Click **Enabled**, if the key will be enabled. Note that if the key is disabled, the key cannot be used for cryptographic operations.
11. Enter the activation date of the key in the **Activation Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
12. Enter the expiration date of the key in the **Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
13. From **Key Operations**, select the cryptographic operations the key will perform. The options are:
    - Encrypt
    - Decrypt
    - Sign
    - Verify
    - Wrap Key
    - UnwrapKey
14. (Optional) From **Tags**, enter the tag name and value of the key. Up to 15 tags are supported.
15. Click **Create**.

    The new key is displayed on the **All Keys** list page.

Delete a Key

From the **Keys** page, deleting a key deletes it from Azure Key Vault (but a backup key remains in the CCKM). After deletion, two new actions are displayed within the **Action** column: *Restore* and *Delete Backup*.

> **Note**
> This Delete action deletes ALL versions of the Azure key.

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. In the resulting **All Keys** list, find the name of the key and click **Delete** from the **Action** column. A warning message is displayed.
3. Enter or copy/paste the confirmation phrase.
4. Click **Delete**.

   The **Action** column for key should now show *Restore* and *Delete Backup* options.

Restore a Key (to Azure)

In the event that someone deletes an Azure key directly in the Azure portal, it is possible to restore it from the CCKM as follows:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. From the **All Keys** list page, select **Restore** from the **Action** column of a deleted key.

   The key and all its versions and metadata are restored to Azure Key Vault. The Success message is displayed, and the key is listed at the top of the **All Keys** page.

   > **Note**
   > If anyone deletes the backup key from CCKM, it will **not** be possible to restore the key to Azure.

Delete a “Soft-deleted” Key

From the **Keys** page, soft deleting a key deletes it from Azure Key Vault removing it from displaying in the CCKM UI. However, the key still exists in the Azure Key Vault and CCKM. After a soft delete, two new actions are displayed within the Action column: *Recover* and *Purge*.

To delete a key in a soft-delete state:

1. In the left navigation pane of the CCKM portal, select **Keys**.
2. From the **All Keys** list page, find the name of the key and click **Soft Delete** from the **Action** column.

   A warning displays indicating that this action will delete the key from your Azure key vault and that the operation may take some time to complete. The message also indicates to type the confirmation phrase (I wish to delete the key) in the text box if you wish to delete the key.
3. Enter or copy/paste the confirmation phrase in the text box.
4. Click **Delete**.

   The key and all its versions and metadata are deleted from the Azure Key Vault. The Success message is displayed, and the key is no longer listed on the **All Keys** page.

Recover a “Soft-delete” Key (to Azure)

From the **Keys** page, recovering a key that is in a “soft-delete” state allows you to restore the key to the Azure Key Vault. After the key is recovered, the key displays in the Azure Key Vault and is listed within the **All Keys** page.

To recover a key in a soft-delete state:

1. In the left navigation pane of the CCKM portal, select **Keys**.
2. From the **All Keys** list page, find the name of the key and select **Recover** from the **Action** column.

   A warning displays indicating that this action will recover your key from your Azure key vault and that the operation may take some time to complete.

3. Click **Recover**.

   The key and all its versions and metadata are restored to Azure Key Vault. The Success message is displayed, and the key is listed at the top of the **All Keys** page.

Purge a “Soft-deleted” Key

From the **Keys** page, purging a key that is in a “soft-delete” state allows you to delete it permanently from the Azure Key Vault. However, you still can restore this same key to the Azure Key Vault using the **Restore** action. For more information, see "Restore a “Soft-delete” Key (to Azure)" below.

To purge a key in a soft-delete state:

1. In the left navigation pane of the CCKM portal, select **Keys**.
2. From the **All Keys** list page, find the name of the key and click **Purge** from the **Action** column.

   A warning displays indicating that this action will purge the key from the Azure key vault. The message also indicates to type the confirmation phrase (I wish to purge the Azure key) in the text box if you wish to purge the key.
3. Enter or copy/paste the confirmation phrase in the text box.
4. Click **Purge**.

   The key and all its versions and metadata are deleted from the Azure Key Vault. The Success message is displayed, and the key is no longer listed on the **All Keys** page.

Restore a “Soft-delete” Key (to Azure)

From the **Keys** page, restoring a key that was previously set to a “soft-delete” state and then purged allows you to restore the key from the key source you are using (CipherTrust Manager or DSM) or from MongoDB (if you are using nShield Connect as your key source) and upload it to the Azure Key Vault. With a successful restore of a “soft-delete” key, the key and all its versions and metadata are restored to Azure Key Vault.

To restore a key that was previously set to a “soft-delete” state and then purged:

1. In the left navigation pane of the CCKM portal, select **Keys**.
2. From the **All Keys** list page, find the name of the key and select **Restore Key** from the **Action** column.

   A warning displays indicating that this action will restore your key to your Azure key vault and that the operation may take some time to complete.
3. Click **Restore**.

   If successful, the Success message is displayed, and the key is listed at the top of the **All Keys** page.

Rotate a Key

CCKM allows for a manual or autorotation of a key. You manually rotate a key by adding a new version from the **Add a version** dialog box. Auto rotating a key is adding a new version of the key at a scheduled time. Key autorotation requires that you first schedule the autorotation in the **Schedules** page prior to enabling this feature in the **Auto-rotate Key** dialog box. Note that the key you select to autorotate must be enabled for it to be used for cryptographic operations after the rotation.

> **Note**
> Any key that is shared among different users within the same Azure cloud and is scheduled for a key rotation by one user cannot be scheduled for autorotation again by another user. Any key that is scheduled for autorotation displays the status of “AUTO” with an orange check icon next to it within the **Auto Rotate** column of the **Keys** page. If you attempt to schedule autorotation using **Rotate Key** within the **Keys** page with a key already scheduled, a message displays indicating that the key is already scheduled for rotation with the scheduled interval, date, and time at which the rotation is to take place. In addition, the name of the user responsible for scheduling the rotation displays.

Manually Rotate a Key (Add a New Version of a Key)

To rotate a key manually (add a new version of a key):

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. In the **All Keys** list, find the name of the key to which to add a new version.
3. Select **Add Version** from the **Actions** drop-down menu. The **Add a Version** dialog box is displayed.
4. From the **Key Provider** box, select one of the following as the key source to which to add a new version:
   - **DSM**—Key was generated from DSM and imported into a Standard or Premium Azure Key Vault.
   - **CipherTrust Manager**—Key was generated from CipherTrust Manager and imported into a Standard or Premium Azure Key Vault.
   - **HSM.byok**—Your byok key was generated from an external Thales HSM and imported into a Premium Azure Key Vault from CCKM.
   - **PFX**—Key was imported as an external password-protected PFX key into a Standard or Premium Azure Key Vault.
   - **nShield**—Key was generated from nShield Connect and imported into a Standard or Premium Azure Key Vault.
   - **Native**—Key is generated natively in the designated Azure Key Vault using the CCKM UI. Note that this option is equivalent to generating a key in the Azure portal.
5. From the **PFX-File Password** box, enter your password for your PFX file. This box only displays, if you selected PFX as your source key tier.
6. From the **Source Key** box, select the source key to which to add a new version. Based on the selected source key tier and source key, the **Key Vault Tier**, **Key Vault Region**, and **Key Name** boxes are prepopulated with the values associated with your selections.
7. From the **Key Destination** box, select either **Software** or **HSM** (Premium subscription only) depending on which destination you want to use.
8. Click **Enabled**, if the key is enabled. Note that if the key is disabled, the key cannot be used for cryptographic operations. This option may already be selected depending on the selected source key.
9. From **Key Type**, select **RSA** or **EC** for your key type. **Key Type** only displays, if you selected **Native** as your source key tier. If you select **RSA**, then select **2048**, **3072**, or **4096** from **Key Size** for the key size to apply to the new version of the key. If you select **EC**, then select **P-256**, **P-384**, **P-521**, or **SECP256K1** from **Elliptical Curve Name** for the name of the elliptical curve to apply to the new version of the key.
10. Enter the activation date of the new version of the key in the **Activation Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
11. Enter the expiration date of the new version of the key in the **Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.

12. From **Key Operations**, click the cryptographic operations the new version of the key will perform. The options are:
    - Encrypt
    - Decrypt
    - Sign
    - Verify
    - Wrap Key
    - UnwrapKey
13. (Optional) From **Tags**, enter the tag name and value of the new version of the key.
14. Click **Upload to Azure**.

    The new version of the key is displayed in the list. As part of the process of adding a new key version of a key, Azure generates a corresponding backup key blob, which CCKM downloads and stores in DSM.

Enable Autorotation of Key

To enable autorotation of a key (add a new version of the key at a scheduled time):

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. In the **All Keys** list, find the name of the key for which to enable autorotation.
3. Select **Auto Rotate** from the **Actions** drop-down menu. The **Auto-rotate Key** dialog box is displayed. The **Key Algorithm** box is populated with the algorithm used in the selected key to autorotate.
4. From **Enable autorotation on this key?**, select **On** to enable the autorotation of this key. By default, this toggle is initially set to **Off**.

    > **Note:** If the toggle for **Enable autorotation on key expiration?** is displayed, skip it. This option displays only if the key has an expiration date associated with it. This option is not applicable to enabling autorotation of the key.

5. From the **Schedule** drop-down menu, select a rotation schedule to which to apply to this key. Any key-rotation schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
6. Select **Change key algorithm**, if you choose to use a different key algorithm than the one displayed in **Key Algorithm** for the new version of the key. If you choose to use a different key algorithm, **Key Provider**, **Key Destination**, and **Key Type** display for configuration.
7. From the **Key Provider** box, select one of the following as the key source to which to add a new version:
    - **DSM**—Key is to be generated from DSM and imported into a Standard or Premium Azure Key Vault.
    - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager and imported into a Standard or Premium Azure Key Vault.
    - **nShield**—Key is to be generated from nShield Connect and imported into a Standard or Premium Azure Key Vault.
    - **Native**—Key is to be generated natively in the designated Azure Key Vault using the CCKM UI. Note that this option is equivalent to generating a new key in the Azure portal.

8. From the **Key Destination** box, select either **Software** or **HSM** depending on whether you want to use the Azure Standard or Premium subscription service. If you are using the Standard subscription service for the selected key to be autorotated, then Software is selected. If you are using the Premium subscription service for the selected key to be autorotated, then HSM is selected. However, you have the option to select Software to use the Standard subscription service for the new version of the key.

9. From **Key Type**, select **RSA** or **EC** for the key type of the new version of the key.
   - If your source key tier is set to **DSM**, then the key type of **RSA** along with the key sizes of **2048** and **4096** for **Key Size** are your options.
   - If your source key tier is set to **nShield**, then the key type of **RSA** along with the key sizes of **2048** and **4096** for **Key Size** are your options.
   - If your source key tier is set to **Native**, then the key types of **RSA** and **EC** are your options. For **RSA**, the key sizes **2048**, **3072**, and **4096** for **Key Size** are your options. For **EC**, the name of the elliptical curves **P-256**, **P-384**, **P-521**, and **SECP256K1** for **Elliptical Curve Name** are your options.

10. Click **Submit**.

Synchronize Keys

Synchronizing allows you to download all the keys that were created in Azure (to which you have access) into CCKM. You have the option to schedule key synchronization. Perform this scheduling in the **Schedules** page.

The story below illustrates a use of synchronization and restoration:

> Sally is the Azure Cloud administrator with access to many vaults. She has synchronized keys in CCKM. Now, innocent, ignorant rogue Charlie logs into Azure portal and deletes all the keys in his vault, which also belonged to Sally. Now, if Sally synchronizes keys again from CCKM portal, she will see those keys as deleted in CCKM portal. However, she will also see options to restore the deleted keys back into the relevant Azure Key Vault. This is because CCKM also maintains Azure backup keys as part of basic key management.

To synchronize your keys:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. Click **Synchronize**. A warning box is displayed; click **Synchronize** again to proceed with the synchronization.

   New keys that are created in Azure are retrieved and displayed in the CCKM UI.

Enable Autorotation before Key Expiration

To enable autorotation of a key before it expires, you must first set up a key expiration schedule in the **Schedules** page. Once the key is enabled for autorotation, the key will be rotated before it expires. After the key is autorotated, the key will have a new expiration date based on the key-expiration schedule.

To enable autorotation of a key before it expires:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. In the **All Keys** list, find the name of the key for which schedule a key rotation.
3. Select **Auto Rotate** from the **Actions** drop-down menu. The **Auto-rotate Key** dialog box is displayed. The **Key Algorithm** box is populated with the algorithm used in the selected key to autorotate.
4. From **Enable autorotation on key expiration?**, select **On** to enable the autorotation of this key before it expires. By default, this toggle is initially set to **Off**.

> **Note:** Skip the **Enable autorotation on this key?** as it is not applicable to enabling autorotation of this key before it expires.

5. From the **Schedule** drop-down menu, select a key-expiration schedule to which to apply to this key. Any key-expiration schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.

6. Select **Change key algorithm**, if you choose to use a different key algorithm than the one displayed in **Key Algorithm** for the new version of the key. If you choose to use a different key algorithm, **Source Key Tier**, **Key Destination**, and **Key Type** display for configuration.

7. From **Source Key Tier**, select one of the following as the key source of the new version of the key:
   - **DSM**—Key is to be generated from DSM and imported into a Standard or Premium Azure Key Vault.
   - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager and imported into a Standard or Premium Azure Key Vault.
   - **nShield**—Key is to be generated from nShield Connect and imported into a Standard or Premium Azure Key Vault.
   - **Native**—Key is to be generated natively in the designated Azure Key Vault using the CCKM UI. Note that this option is equivalent to generating a new key in the Azure portal.

8. From the **Key Destination** box, select either **Software** or **HSM** depending on whether you want to use the Azure Standard or Premium subscription service. If you are using the Standard subscription service for the selected key to be autorotated, then Software is selected. If you are using the Premium subscription service for the selected key to be autorotated, then HSM is selected. However, you have the option to select Software to use the Standard subscription service for the new version of the key.

9. From **Key Type**, select **RSA** or **EC** for the key type of the new version of the key.
   - If your source key tier is set to **DSM**, then the key type of **RSA** along with the key sizes of **2048** and **4096** for **Key Size** are your options.
   - If your source key tier is set to **nShield**, then the key type of **RSA** along with the key sizes of **2048** and **4096** for **Key Size** are your options.
   - If your source key tier is set to **Native**, then the key types of **RSA** and **EC** are your options. For **RSA**, the key sizes **2048**, **3072**, and **4096** for **Key Size** are your options. For **EC**, the name of the elliptical curves **P-256**, **P-384**, **P-521**, and **SECP256K1** for **Elliptical Curve Name** are your options.

10. Click **Submit**.

Export Keys

Export Keys allows you to export the contents of the Keys list into a report in a CSV format.

1. In the left navigation bar, click **Keys**.

2. Click **Export Keys**.

   This will download all of the keys' versions in a report in a CSV format.

Backup Key

When a key is uploaded to Azure from CCKM, Azure creates a backup key (as an encrypted blob), which contains information about the Azure key and all its versions. The backup key can be used to restore a key that has been deleted. After Azure creates a backup key, CCKM automatically downloads and stores the key in the key source you are using (CipherTrust Manager or DSM), or in MongoDB (if you are using nShield Connect as your key source). In addition, when Azure keys are synchronized, the backup key of any key that was created directly in the Azure portal is also downloaded and stored in the key source you are using (CipherTrust Manager or DSM), or in MongoDB (if you are using nShield Connect as your key source).

> **Note**
> CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

Delete a Backup Key

Delete a backup key when you definitely do not want the key and have no corresponding data that will ever need decryption.

For a key that has been deleted:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. From the **All Keys** list page, select **Delete Backup** from the action column of a deleted key.
3. The **Delete Key** warning page is displayed. Type or Copy/paste the confirmation phrase (I wish to delete the Azure backup key).
4. Click **Delete**.

   The key is removed from the list.

Reports

CCKM includes the following reports:

- **Combined Key Activity Reconciliation Report**

  Provides the history of the key in a selected period. Click Run Report to generate a new list of events connected to a key.

- **Azure Key Activity Report**

  Illustrates the activity type performed by a specific user and at the stated time.

- **Azure Key Service Usage Report**

  Indicates the relationship between the application and the key, including:

  - Key name
  - Key Description
  - Requesting Service: Name or ID of the service using a key
  - Service Platform
  - Request Timestamp: Time when the application used the key.

- **Azure Key Aging Report**

  Provides information about key expiration using the following columns:

  - Key Name
  - Key description
  - Key destination: Indicates services, such as Azure
  - Key owner: Person who created the key. In Azure, there is no owner associated with the key.
  - Key expiration: Date and time at which this key expires.

- **Cloud Key Manager User Action Report**

  Lists the users and their actions in creating and deleting keys in the CCKM portal.

Schedules

The **Schedules** page allows Cloud administrators to view and schedule the following job types:

- Key Rotation
- Key Synchronization
- Key Expiration

CCKM provides support for multiple schedules for each of the job types.

At the top right of the **Schedules** page, the **Add Schedule** button is available from which to add a scheduled job type.

> **Note**
> If you are using CCKM as a service principal, you are not required to enter a username and password to schedule key rotation, key synchronization, or key expiration for Azure Key Management. The username and password boxes do not display in the dialog box for each of these job types.

Schedule Key Rotation

CCKM provides support for multiple schedules of the rotation of an Azure key. CCKM allows for a scheduled rotation (autorotation) of an Azure key. A new version of the key is added to the Azure key during a scheduled rotation. Scheduled rotation requires that this feature be configured in the **Schedules** page prior to enabling the feature in the **Auto Rotate Key** dialog box (from the **Actions** column of the **All Keys** list). Your Azure login password is required as part of the configuration and is saved in the key source you are using (CipherTrust Manager or DSM), or in MongoDB (if you are using nShield Connect as your key source). If you have both DSM and nShield Connect configured as your key sources, then the Azure login password is saved only to DSM where it is stored.

> **Note**
> CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

During a scheduled rotation of a key, CCKM uses the associated username and password to rotate the key. If the password expired or changed, then the new password must be reentered. Otherwise, the scheduled rotation will fail.

You can pause or delete a scheduled key rotation in the **Schedules** page. Note that when a key rotation schedule is deleted from CCKM, then the Azure user password is also deleted from the key source you are using (CipherTrust Manager or DSM) or from MongoDB (if you are using nShield Connect as your key source).

Set up a Key Rotation Schedule

To set up a key rotation schedule:

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.

   > **Note:** If you are using CCKM as a service principal, the **Username** box does not display as you are not required to enter a username when using CCKM as a service principal

3. From the **Job Type** drop-down menu, select **KEY_ROTATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.

6. For **Pause**, select **No**.

7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).

   If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. Set the repeat interval of the key rotation by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

   If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.

8. Enter your Azure user password in the **User Password** box.

   > **Note:** If you are using CCKM as a service principal, the **User Password** box does not display as you are not required to enter a password when using CCKM as a service principal.

9. Click **Save** to save the setting.

Pause a Key Rotation Schedule

To pause a key rotation schedule, select **Yes** for **Pause** within the **Add Schedule** dialog box and then click **Save**. The setting is saved. To resume the rotation schedule after pausing it, select **No** for **Pause** and then click **Save**. The setting is saved.

Delete a Key Rotation Schedule

To delete a key rotation schedule, select **Delete** from the **Actions** drop-down menu from the **Schedules** page. Note that the Azure user password associated with the given key rotation schedule is also deleted from the key source you are using (CipherTrust Manager or DSM) or from MongoDB (if you are using nShield Connect as your key source).

Schedule Key Synchronization

CCKM allows for multiple schedules of the synchronization of Azure keys. Synchronizing downloads any keys that were created in Azure, outside of the CCKM portal, into your CCKM. Scheduled key synchronization requires that this feature be configured in the **Schedules** page. You can also pause or delete a scheduled key synchronization in the **Schedules** page. Note that when a key synchronization schedule is deleted from CCKM, then the Azure user password is also deleted from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as the key source). If you have both DSM and nShield Connect configured as your key sources, then the Azure user password is deleted from DSM where it is stored.

> **Note**
> CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

Set up a Key Synchronization Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.

2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.

   > **Note:** If you are using CCKM as a service principal, the **Username** box does not display as you are not required to enter a username when using CCKM as a service principal.

3. From the **Job Type** drop-down menu, select **KEY_SYNCHRONIZATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the key synchronization by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).

   If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. Set the repeat interval of the key synchronization by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

   If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
8. Enter your Azure user password in the **User Password** box.

   If you are using CCKM as a service principal, the **User Password** box does not display as you are not required to enter a password when using CCKM as a service principal.
9. Click **Save** to save the setting.

Schedule Key Expiration

CCKM provides support for multiple schedules of the expiration of Azure keys in the **Schedules** page. You can also pause or delete a scheduled key expiration in the **Schedules** page. Note that when a key expiration schedule is deleted from CCKM, then the Azure user password is also deleted from the key source you are using (CipherTrust Manager or DSM), or from MongoDB (if you are using nShield Connect as your key source). If you have both DSM and nShield Connect configured as your key sources, then the Azure user password is deleted from DSM where it is stored.

> **Note**
> CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

Set up a Key Expiration Schedule

To set up a key expiration schedule:

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.

   > **Note:** If you are using CCKM as a service principal, the **Username** box does not display as you are not required to enter a username when using CCKM as a service principal.
3. From the **Job Type** drop-down menu, select **KEY_EXPIRATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.

7. From **Start Date**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.
8. From **Expire a new key after**, enter a number and select the interval unit of Hour, Day, Week, Month, or Year from the drop-down menu after which the new key expires. The supported ranges are 6 through 24 hours, 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.
9. Enter your Azure user password in the **User Password** box.

   If you are using CCKM as a service principal, the **User Password** box does not display as you are not required to enter a password when using CCKM as a service principal.
10. Click **Save** to save the setting.

Logs

CCKM maintains logs for events, such as login, key generation, synchronizing of keys, and deletion of keys, and displays these logs in the **All Logs** list of the **Logs** page. For each logged event listed on the **Logs** page, the following information is displayed:

CCKM maintains logs for events, such as login, key generation, synchronizing of keys, and deletion of keys, and displays these logs in the **All Logs** list of the **Logs** page. For each logged event listed on the **Logs** page, the following information is displayed:

- Event name
- Severity level
- Date and time of the event
- Event message
- User (name) associated with the event

You can search for a specific logged event from the **Logs** page using the **Search** box. Allowable filters on the search are event name, severity level, event message, and user (name).

Key Sources

The **Key Sources** page displays all of the keys that are generated from the key source you are using. Three tabs are available at the top of the page:

- **DSM Key:** Use this tab to use DSM as your key source. The keys are generated and stored within DSM.
- **CipherTrust Manager:** Use this tab to use CipherTrust Manager as your key source. The keys are generated and stored within CipherTrust Manager.
- **nShield Key:** Use this tab to use nShield Connect HSM as your key source. The keys are generated within nShield Connect and then stored as blobs within MongoDB.

  > **Note**
  > CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

Keys listed on the **Key Sources** page are listed by name, key type (the cloud in which this key is used), the algorithm of the key, the creation date of the key, key description and actions you can take regarding that key. Note that the key algorithm currently supported for an Azure key in CCKM is RSA.

A **Search** box is available at the top left of the **Keys Sources** page allowing for a search of a specific key by entering any one of the following parameters:

- Name (key name)
- Key Type
- Algorithm
- Description
- Created By (a key created by a specific user)

Add (Generate) a Key

From the **Key Sources** page, you can add (or generate) keys in CCKM using CipherTrust Manager, DSM, or nShield Connect HSM as a key source. These keys are of the RSA algorithm type. From the **Key Sources** page, the **DSM Key** tab also allows you to generate a .byok key from an external HSM (such as nShield Connect HSM) and import it into a Premium Azure Key Vault from CCKM. From the **DSM Key** tab, you can also import an external password-protected PFX key into a Standard or Premium Azure Key Vault. Once imported from CCKM, the .byok and PFX key is stored in DSM securely as blobs.

Add (Generate) a DSM Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **DSM Key** tab.
3. Select **Add Key > Azure Key**.
4. In the **Add DSM Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to Azure.
5. In the **Source Key Tier**, select one of the following as a key source of the key:
   - **DSM**—Generate a key from DSM and import it into a Standard or Premium Azure Key Vault.
   - **HSM.byok**—Generate a .byok key from an external Thales HSM and import it into a Premium Azure Key Vault from CCKM. Once imported from CCKM, this .byok key is stored in the DSM securely as a blob. The .byok key is user and region specific. Be sure to select the correct username and region when creating the .byok key.
   - **PFX**—Import an external password-protected PFX key into a Standard or Premium Azure Key Vault. Once imported from CCKM, this PFX key is stored in the DSM securely as a blob.
6. In the **Name** box, enter the name of the new key.
7. Enter a description of the key in the **Description** box.
8. From the **Algorithm** box, select the algorithm (**RSA 2048** or **RSA 4096**) to apply to the key.
9. Click **Save** to save the new key to CCKM. The new key is added to the keys list. (Reset clears the screen before it has been saved.)

Add (Generate) a CipherTrust Manager Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **CipherTrust Manager Key** tab.
3. Select **Add Key > Azure Key**.

4. In the **Add CipherTrust Manager Key** dialog box, select the name of the user of the key from the **User** drop-down menu. Service is fixed to **Azure**.
5. From the **Source Key Tier** box, select **CipherTrust Manager** from the drop-down menu.
6. In the **Name** box, enter the name of the new key.
7. Enter a description of the key in the **Description** box.
8. From the **Algorithm** box, select the algorithm (**RSA 2048** or **RSA 4096**) to apply to the key.
9. Click **Save** to save the new key to CCKM. The new key is added to the keys list. (Reset clears the screen before it has been saved.)

Add (Generate) an nShield Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **nShield Key** tab.
3. Select **Add Key > Azure Key**.
4. In the **Add nShield Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to Azure.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. From the **Algorithm** box, select the algorithm (**RSA 2048** or **RSA 4096**) to apply to the key.
8. Click **Save** to save the new key to CCKM. The new key is added to the keys list. (Reset clears the screen before it has been saved.)

Delete a Key

Deleting a key from the **Key Sources** page deletes the key from CCKM. When deleting a key from CCKM, the key is deleted from the key source you used to generate the key. However, if you generated the key from nShield Connect, the key is deleted from MongoDB where the key is stored.

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **DSM Key**, **CipherTrust Manager**, or **nShield Key** tab depending on the key source you used to generate this key initially.
3. Click **Delete** next to the target key.
4. The **Delete Key** dialog box with a Warning display asking you to confirm the key deletion by entering the supplied phrase.
5. Enter or copy/paste the confirmation phrase.
6. Click **Delete** to confirm.

Manage Keys in Key Sources List Page

In addition to adding and deleting keys, the **Key Sources** page allows you to sort the existing keys by Name, Key Type, Algorithm and Created Date.

Settings

To access the **Settings** page in CCKM, select **Settings** from the left-hand navigation bar. From the **Settings** page, you can:

- Enter the emails of the recipients who are to receive alerts and reminders.
- Set up key alerts.
- Configure remote syslog servers to which to send syslog messages.
- (Applicable to CCKM when used as a service principal) Change your tenant password or revoke your administrator consent.
- View information about the software release number of the CCKM you are running including the build and version number.

Note that for Azure, each of the supported clouds (Azure, Azure China, and Azure Germany) must be set up independently of each other, as these are separate and distinct key management services within Azure.

Adding Recipients’ Emails (for Alerts and Reminders)

Use the **General** settings tab to enter the designated recipients’ emails for email alerts and reminders to be sent from the system. Enter the recipients’ email addresses separated by a comma in the **Alert Format** box.

Setting Keys Alerts

Use the **Keys Alert** tab from the **Settings** page to enable whether to send key alerts to specified recipients. When a key is deleted, restored, or uploaded in Azure, a key alert can be sent. You can also to set up a reminder to manually rotate (or rekey) a key along with the frequency (in days) at which the reminder is sent. The recipients’ emails are set up in the **General** tab. By default, no alerts or reminders are sent.

To set up key alerts or a reminder to manually rotate (or rekey) a key:

1. In the **Settings** page, select the **Keys Alert** tab.
2. Under **Alerts**, click **On** for each of the alert types and reminder you wish to enable:
   - Rekey Alert (reminder)
   - Delete Azure Key
   - Restore Azure Key
   - Upload Azure Key
3. If you set a reminder for **Rekey Alert**, then in the **Days** box, enter the number of days for the frequency an email reminder to manually rotate a key is sent.
4. Click **Save** to save the setting.

Configuring Syslog Server

Use the **Syslog** tab from the **Settings** page to configure a remote syslog server to which to send syslog messages. Note that the default port number of 514 and the facility name of “LOCAL1” for the syslog server display in the **Port** and **Facility** boxes, respectively. These boxes are not available for modification.

Configure Remote Syslog Server

To configure a remote syslog server:

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.
2. Enter the hostname or IP address of the syslog server in the **Hostname** box.
3. (Optional) Enter a description of the syslog server in the **Description** box.
4. Click **Save** to save the setting.

Delete Remote Syslog Server Configuration

To delete the configuration of a remote syslog server, click **Delete** from the **Syslog** dialog box. No syslog messages are sent to the remote syslog server after the deletion of this configuration.

Configuring Tenant Password and Revoking Admin Consent (for Service Principal)

> **Note**
> The Tenant tab is available in the **Settings** page only if you have configured to use CCKM as a service principal.

The **Tenant** tab in the **Settings** page allows you to change your tenant password that is associated with the use of CCKM as a service principal.

> **Note**
> The **Tenant** tab is available in the **Settings** page only if you have configured to use CCKM as a service principal.

The **Tenant** tab also allows you to revoke the administrator consent, which you (as the administrator user) had provided during the initial login into CCKM for Azure Key Management (using CCKM as a service principal). See "Accessing CCKM Using Service Principal" on page 108, for more information about providing your administrator consent. If you revoke the administrator consent and then logout of CCKM, you will receive an error message when you attempt to log back into CCKM. The message indicates that the login failed due to having entered an incorrect tenant name or the given tenant not having granted permissions to CCKM (to be used as a service principal). You must provide the administrator consent again to login into CCKM. Click on the "admin consent" link (indicated by “here” in blue text at the top CCKM login dialog box) after which the CCKM Admin Consent dialog box is displayed.

Change Tenant Password

To change a tenant password that is associated with CCKM used as a service principal:

1. In the **Settings** page, select the **Tenant** tab. The **Tenant** dialog box displays.
2. Enter the new tenant password in the **New Tenant Password**box.
3. Reenter the new tenant password in the **Confirm Tenant Password**box.
4. Click **Save** to save the setting.

Revoke Admin Consent

To revoke your administrator consent that is associated with CCKM used as a service principal:

1. In the **Settings** page, select the **Tenant** tab. The **Tenant** dialog box displays.
2. Deselect the **Granted Permission to this CCKM** box to revoke your administrator consent. By default, this box is checked.

3. Click **Save** to save the setting.

About

Use the **About** tab to view the release number of the CCKM you are running.

Chapter 15: CCKM for Google Key Management

Overview

CCKM provides support for managing Google Cloud customer-managed encryption keys (CMEK). CMEK are used to encrypt data at rest within a Google Cloud project. In addition, several Google Cloud services offer the ability to protect data using encryption keys managed by customers within the Cloud Key Management System (KMS). CCKM can control the lifecycle of these keys by automatically or manually rotating CMEK. It also controls Google's ability to decrypt data at rest by disabling the key. Data is protected using a key that meets specific locality requirements using either one of the supported key sources (Data Security Manager (DSM) or CipherTrust Manager) with the CCKM solution. Additionally, CCKM can manage CMEK lifecycle in an automated way using Google Service account for REST API authentication.

> **Note**
> This documentation assumes that you are familiar with using Google Cloud Platform, Google Cloud KMS, and Google Identity and Access Management (IAM). For details about these Google technologies, refer to the Google Cloud documentation at https://cloud.google.com/docs.

Google Cloud IAM

Using Google Cloud's Identity and Access Management (IAM) allows you to manage access to Google Cloud resources by granting access to these resources. By defining the identity, role, and resource(s) in IAM to which to be given access, you are able to manage access to Google Cloud resources. Access permissions are grouped into roles and an authenticated Google member is granted at least one role to access to the resources.

IAM has three parts as described in the following sections.

Member

A member can be:

- Google Account (for end users)
- Service account (for apps, such as CCKM, and virtual machines)

- Google group (can contain multiple user and service accounts)
- G Suite domain
- Cloud Identity domain

An email address associated with a user, service account, or Google group is a member’s identity. The G Suite and Cloud Identity domains each use the associated domain name for its identity.

CCKM supports the use of Google user, group, and service accounts. You will need the user account to login to CCKM UI, and you will need service account to authenticate to CCKM REST API.

Roles

A role is a collection of permissions. There are three kinds of roles in Cloud IAM:

- Primitive roles: Owner, Editor, Viewer.
- Predefined roles: Predefined finer-grained access control roles. For your user account or service account to use CCKM, you need to grant Project Browser and Cloud KMS Admin roles to the account. Note that the Cloud KMS admin role grants all the permissions required to use CCKM. If your project belongs to an organization, then you also need to grant Organization Viewer role to your accounts.
- Custom roles: User defined roles (*recommended*). If you wish to grant specific permissions to users or groups to meet your specific requirements, then create custom roles. For example, one user or group should only be able to view keys but not perform any key operations. Another user or group should only be able to list keys and perform any operation with the exception of deleting key versions. See Table 15-1: "[CCKM Action Mapping to IAM KMS Permissions]" below for the permissions that are required for each of the supported CCKM actions. See "Create Custom Roles" on page 53 for information on how to create custom roles.

Table 15-1: [CCKM Action Mapping to IAM KMS Permissions]

CCKM Action	Cloud IAM KMS Permissions
Add key version import	• cloudkms.importJobs.create • cloudkms.importJobs.setIamPolicy • cloudkms.importJobs.get • cloudkms.importJobs.useToImport • cloudkms.cryptoKeyVersions.create • cloudkms.cryptoKeys.update
Add key version native	• cloudkms.cryptoKeyVersions.create • cloudkms.cryptoKeys.update
Authenticate	• resourcemanager.organizations.get • resourcemanager.projects.get
Cancel schedule delete key material	cloudkms.cryptoKeyVersions.restore
Create key	cloudkms.cryptoKeys.create
Rotate key import	• cloudkms.importJobs.create • cloudkms.importJobs.setIamPolicy • cloudkms.importJobs.get • cloudkms.importJobs.useToImport • cloudkms.cryptoKeyVersions.create • cloudkms.cryptoKeys.update
Rotate key native	• cloudkms.cryptoKeyVersions.create • cloudkms.cryptoKeys.update

Table 15-1: [CCKM Action Mapping to IAM KMS Permissions] (continued)

CCKM Action	Cloud IAM KMS Permissions
Schedule delete key material	cloudkms.cryptoKeyVersions.destroy
Synchronize	• cloudkms.keyRings.list • cloudkms.keyRings.getIamPolicy • cloudkms.cryptoKeys.getIamPolicy • cloudkms.cryptoKeys.list • cloudkms.cryptoKeyVersions.list
Update key	cloudkms.cryptoKeys.update
Update key version	• cloudkms.cryptoKeys.update • cloudkms.cryptoKeyVersions.update
Upload key	• cloudkms.cryptoKeys.create • cloudkms.importJobs.create • cloudkms.importJobs.setIamPolicy • cloudkms.importJobs.get • cloudkms.importJobs.useToImport • cloudkms.cryptoKeyVersions.create • cloudkms.cryptoKeys.update

For more information about Google Cloud roles, see Understanding Roles within Google's IAM.

Resource

Google Cloud resources are organized hierarchically:

- Organization: root node of the hierarchy
- Folders: children of the organization
- Projects: children of the organization or of a folder
- Resources: descendants of projects

A Cloud IAM policy grants a member permissions to a resource. Google Cloud KMS permissions can be granted on the following types of resources:

- **project** (*not recommended*): a member that is granted a role or one permission on a project will be given that privilege on all resources contained in the project. Resources contained in a project can be KMS resources or other service resources. KMS resources are key rings and all crypto keys contained in those key rings.
- **key ring**: a member that is granted a role or one permission on a key ring will be given that privilege on all crypto keys contained in the key ring.
- **crypto key**: a member that is granted a role or one permission on a crypto key will be given that privilege on all crypto key versions contained in the key.

About Google Cloud Keys

Google Cloud KMS stores cryptographic keys in a hierarchical structure. Here are levels starting from the top:

- Project
- Location
- Key ring

- Key
- Key version

When you create a key in Google Cloud KMS, you define the purpose and the algorithm of that key (see Table 15-2: "[KMS Keys]" below below for each of the available algorithms per key purpose). You have the option to change the algorithm when you create a new key version (depending on the scope of its purpose.). However, you cannot change the key purpose after a key is created.

Google Cloud KMS provides the following key purposes:

- ENCRYPT_DECRYPT (used in Symmetric encryption)
- ASYMMETRIC_SIGN (used in Asymmetric signing)
- ASYMMETRIC_DECRYPT (used in Asymmetric encryption)

The protection level, which you cannot change once the key is created, indicates how the cryptographic operations are performed. Google Cloud KMS provides the following protection levels:

- Software: software is used for cryptographic operations
- HSM: an HSM is used for cryptographic operations
- EXTERNAL *(Not currently supported in CCKM)*: An external key manager stores the key used in the cryptographic operations. Only applicable to symmetric encryption.

For each key type (and subtype), the following table provides the key purpose and supported algorithm(s) and protection levels.

Table 15-2: [KMS Keys]

Key Type & Subtype	Key Purpose	Supported Algorithms	Protection Level
Symmetric *n/a*	ENCRYPT_DECRYPT	GOOGLE_SYMMETRIC_ENCRYPTION	SOFTWARE HSM
Asymmetric *signing*	ASYMMETRIC_SIGN	• EC_SIGN_P256_SHA256 • EC_SIGN_P384_SHA384 • RSA_SIGN_PSS_2048_SHA256 • RSA_SIGN_PSS_3072_SHA256 • RSA_SIGN_PSS_4096_SHA256 • RSA_SIGN_PSS_4096_SHA512 • RSA_SIGN_PKCS1_2048_SHA256 • RSA_SIGN_PKCS1_3072_SHA256 • RSA_SIGN_PKCS1_4096_SHA256 • RSA_SIGN_PKCS1_2048_SHA512	SOFTWARE HSM
Asymmetric *encryption*	ASYMMETRIC_ENCRYPT	• RSA_DECRYPT_OAEP_2048_SHA256 • RSA_DECRYPT_OAEP_3072_SHA256 • RSA_DECRYPT_OAEP_4096_SHA256 • RSA_DECRYPT_OAEP_4096_SHA512	SOFTWARE HSM

Prerequisites on Google Cloud

The Google Cloud administrator must ensure the following prerequisites are met prior to configuring and accessing the Google Cloud KMS in CCKM:

- Create at least one project. For information on how to create a project, see Creating and managing projects within the Google Cloud online documentation.
- Enable Cloud Resource Manager APIs. See "Enable Cloud Resource Manager API" on page 49 for more information.
- Enable IAM APIs. See "Enable Identity and Access Management API" on page 49 for more information.
- Enable Cloud KMS service and create a key ring in the project. See "Enable Cloud Key Management Service (KMS) API" on page 49 for information about enabling the Cloud KMS service.
- Create a user account and a service account. You will need the user account to login to CCKM UI, and the service account to authenticate to CCKM REST API. See "Create Service Account" on page 51 for more information.
- If your Google user account belongs to an organization, ensure your security policy for Google Cloud session control is set to **Session never expires**. See "Set Session Duration for Google Cloud Console" on page 50 for more information.
- Assign the roles of Project Browser and Cloud KMS Admin to the user account and service account. See "Grant IAM Roles to User and Service Accounts" on page 52 for more information.
- If you will use the user account to add a service account to CCKM, assign the Service Account Key Admin role to the user account. Refer to the Google Cloud online documentation for more information.
- If your project belongs to an organization, assign the Organization Viewer role to your user account and service account. See "Grant IAM Roles to User and Service Accounts" on page 52 for more information.
- (Optional) If you wish to grant specific permissions to users or groups, then create custom roles. See "Prerequisites on Google Cloud" above table for the permissions that are required for each of the supported CCKM actions. See "Create Custom Roles" on page 53 for information.
- If you need to generate CCKM reports, assign the following roles to your user account and service account:
  - Logging/Logs Viewer or Project/Viewer
  - Logging/Private Logs Viewer or Project/Owner

  CCKM accesses the Admin Activity audit logs and Data Access audit logs in Google Cloud to generate reports. These audit logs must be retained at log buckets in Google Cloud. *Ensure* logs are enabled in Google Cloud. For information about Google Cloud logs, refer to https://cloud.google.com/logging/docs/storage. For information about log retention periods, refer to https://cloud.google.com/logging/quotas#logs_retention_periods.
- Create OAuth2 Client ID credentials for CCKM app. Note the Client ID and secret. You will need these credentials when adding Google Cloud service to CCKM using the CCKM Admin Portal. See "Create OAuth2 Client ID Credentials " on page 50 for more information.

Access CCKM for Google Key Management

If you will use CCKM UI to manage your Google Cloud keys, then you will need to use your username and password for Google Cloud to login to the CCKM user portal. A Google Cloud service account cannot be used to login to the CCKM user portal. If you will use CCKM REST APIs to manage your Google Cloud keys, then you will need a Google Cloud service account. A Google Cloud user account cannot be used with CCKM REST APIs.

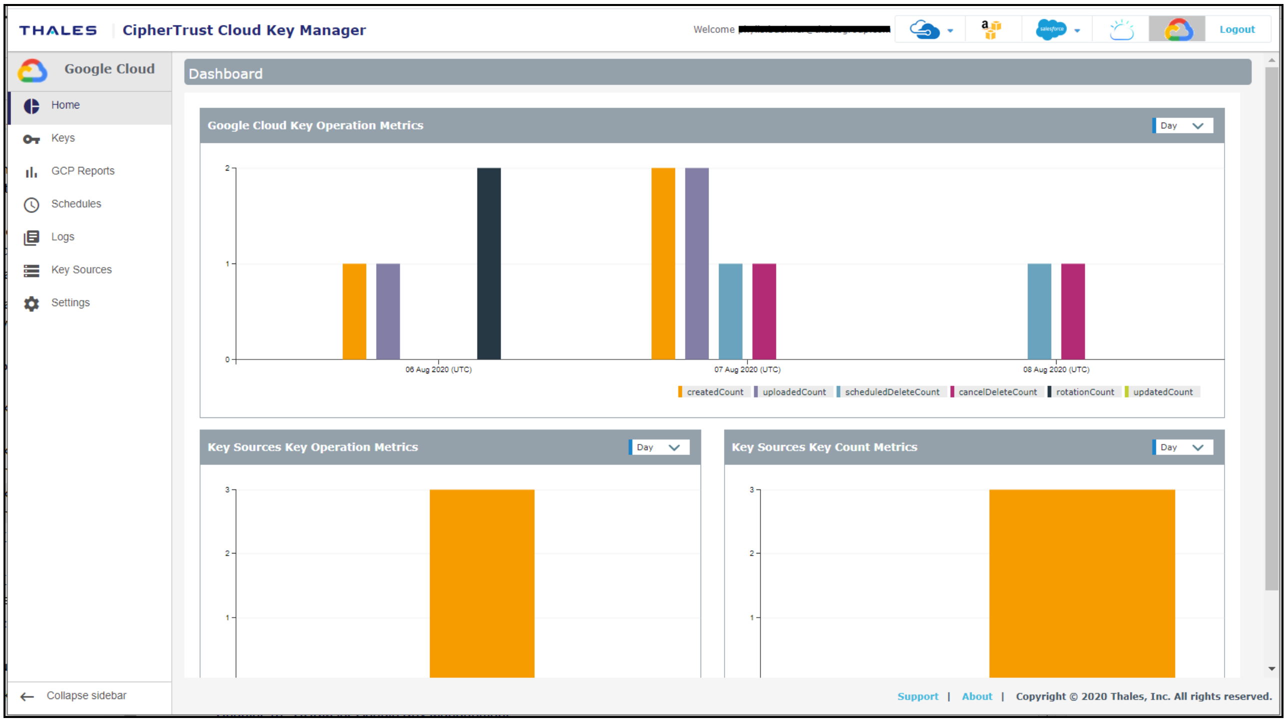
Google CCKM User Portal

The CCKM User Portal for Google Cloud contains the following menu options in the left-hand navigation bar:

- **Home:** Home page displays key operation and count metrics.
- **Keys: Google Cloud Keys** page displays key management overview and provides the functionalities to create a native key in Google Cloud KMS, upload your "BYOK" key, synchronize keys and key rings, and export the contents of the keys list into a report. Also available are the key operations you can perform for both Asymmetric and Symmetric keys and the key versions. For Asymmetric keys, the key operations of Update Key and Add Version are available. For the symmetric keys, the key operations of Update Key and Rotate Key are available. For the key versions of both types of keys, the actions of View all versions, Enable, Disable, Schedule Destroy, and Cancel Destroy are provided. Additionally, for the key versions of the Asymmetric keys, the action of Download (Public Key) is also provided.
- **Reports: All Reports** overview page links to detailed individual reports.
- **Schedule:** In **Schedule**, you can schedule automatic key rotation of Symmetric keys and schedule to add a new key version of Asymmetric keys. Additionally, you have the option to schedule key synchronization.
- **Logs: Logs** page lists individual logs that can be searched, sorted, and viewed.
- **Key Sources:** In **Key Sources**, you can create keys in CCKM to upload to Google Cloud. CCKM provides support for the use of CipherTrust Manager and DSM as key sources to generate, store, and retrieve the encryption keys that CCKM uses and manages.

  > **Notes**
  > - CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

- **Settings:** The **Settings Management** page allows you to configure the email settings, key alerts, and remote syslog servers to which to send syslog messages. You can also view the release number of the CCKM you are running from this page.



To display the context of which Key Management service you are currently logged into within CCKM, the icon associated with the Key Management service displays at the top of the left-hand navigation bar.

The **arrow** button at the bottom of the navigation bar allows you to expand the bar to view the names of the menu options and collapse it to view only the icons associated with the menu options. Click the **arrow** button to either expand or collapse the navigation bar.

On the top-right side of the Google CCKM portal, the following options are available:

- **AWS Login:** If you are not already logged into AWS, clicking on the **AWS Login** icon brings you to the **AWS Password Login** page where you can enter your AWS credentials to access the CCKM AWS cloud service (from the current cloud service in which you are currently logged). If you are already logged into AWS, clicking on the **AWS Login** icon brings you to the **Multi-account Login/Logout** window for the AWS account(s). For more information about the Multi-account Login/Logout window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.

- **Azure Cloud Service Context Switcher:** This context switcher allows you to login into another Azure cloud service different from the current Azure cloud service into which you are currently logged. If you select the Azure cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Azure account(s) displays. For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.

- **Google Cloud Service Context Switcher:** This context switcher allows you to login into another Google Cloud organization different from the current Google Cloud organization into which you are currently logged. If you select the Google Cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Google account(s) displays. For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.

- **Salesforce Cloud Service Context Switcher:** This context switcher allows you to login into another Salesforce cloud service different from the current Salesforce cloud service into which you are currently logged. This is either Salesforce or Salesforce Sandbox. If you select the Salesforce cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Salesforce account(s) displays. For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.

- **Logout:** Logout from CCKM.

Home

The **Home** page provides a dashboard to display key operation and count metrics.

Keys

The **Keys** page allows Cloud administrators to view all the keys and manage them within the boundary of granted permissions. Those keys that do not have List permission for the respective Cloud administrator will not be listed.

A **Search** box is available at the top left of the **Keys** page allowing for a search of a specific key by entering any one of the following parameters:

- Key Name
- Version Name
- Key Purpose
- Project Name
- Protection Level
- Template algorithm
- Label

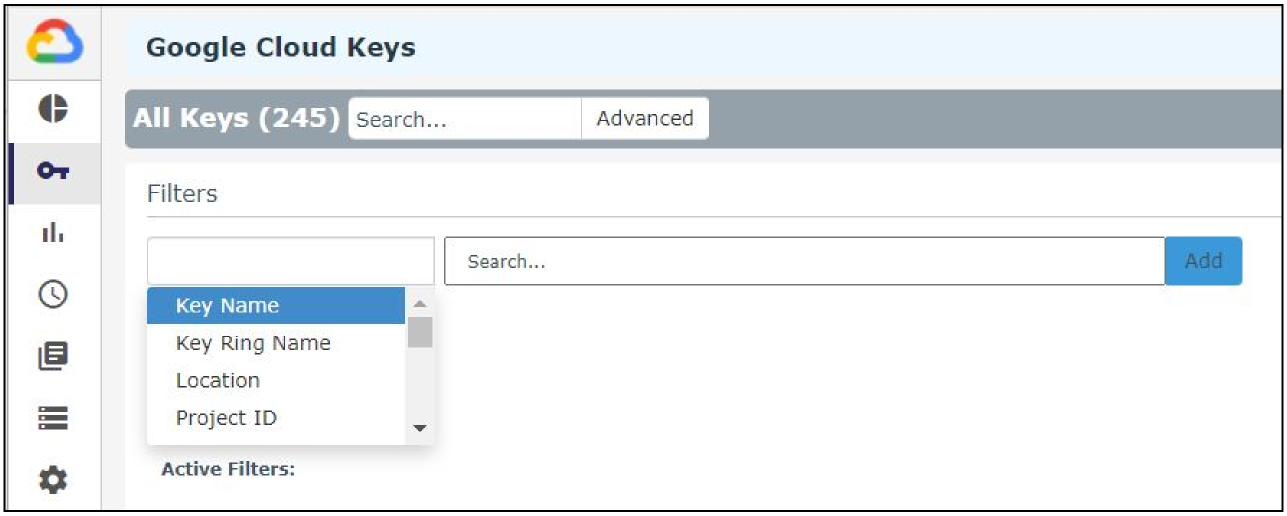
- Version state
- Version Algorithm
- Key Material Origin

The **Advanced** tab allows you to filter your search on any or a combination of these columns displayed on the **Keys** page with the exception of the **Count** and **Action** columns:

- Key Name
- Key Ring Name
- Location
- Project ID
- Project Name
- Key Purpose
- Protection Level
- Template algorithm
- Key Material Origin
- Version State
- Version Algorithm
- Rotation Status
- Primary Version
- Label

Each of these columns are described in more detail below in this section.

To select to filter based on a column or combination of these columns from the **Keys** page, click **Advanced**. The **Filters** section displays. From the column filter drop-down menu, select the name of the column on which to filter your search. For example, select **Key Name**. In the **Search** box, enter the value on which to search (for example, **MK-createkey** as the key name) and then click **Add**. For the **Key Purpose**, **Template Protection Level**, **Template algorithm**, **Version State**, **Version Protection Level**, **Version Algorithm**, **Auto Rotate**, and **Primary Version** columns, a drop-down menu is available from which to select a value for the search. Click the down arrow, select a value from the available options, and then click **Add**. The specified search filter is then added to **Active Filters**. In the example, **Key Name: MK-createkey** is added to **Active Filters**. The results of the search displays in the **Keys** table. You can narrow your search by selecting more columns to add to the filter. To clear the filtered results, from **Active Filters**, click on the “X” (or the **Delete** button) for each of the named filters.



At the top right of the Keys page, the following buttons are available:

- **Export Keys:** Allows you to export the contents of the Keys list into a report in a CSV format.
- **New Key:** Allows you to upload an external BYOK (CipherTrust Manager or DSM key) or create a native key. For information on how to upload an external BYOK or create a native key, see "Create a Key" on the next page.
- **Synchronize:** Allows you to download the key rings and key metadata of all the keys that were created in Google Cloud KMS (to which you have access) into CCKM. If you have External Key Manager (EKM) keys saved in Google Cloud KMS, the key rings and key metadata of EKM keys are also downloaded to CCKM during key synchronization . For information on how to synchronize keys, see "Synchronize Keys" on page 153.

The **Show EKM** check box is also located at the top right of the **Keys** page. Selecting this check box allows you to display information about EKM keys (stored in Google Cloud KMS and downloaded to CCKM during key synchronization) on the CCKM **Keys** page. By default, this box is deselected, and EKM key information is not displayed on the **Keys** page.

The following columns display on the **Keys** page:

Key ID: The ID of the key.

Key Ring: The ID of the key ring to which the key belongs.

Location: The name of the geographical region where your Cloud KMS key is stored and can be accessed.

Project ID: The ID of the project to which the key belongs.

Key Purpose: Defines a key's cryptographic capabilities and the algorithms that are supported for the key's version.

Protection Level: Indicates whether the key and key version are stored in Software or HSM.

Version: The version number of the key.

Version State: Indicates the status of the key version in Google Cloud KMS, which can be one of the following:

- **PENDING_GENERATION**: Key version is in the process of being generated. This state is only applicable to asymmetric keys. Once created and ready for use, the state of key version moves to the ENABLED state.
- **ENABLED**: Ready for use.
- **DISABLED**: Cannot be used. However, the key material is still available for use. You can set the state of the key version back to ENABLED to be able to use this key version again.
- **DESTROY_SCHEDULED**: Scheduled for destruction. This key version can be placed back into the DISABLED state.
- **DESTROYED**: Key version is destroyed and no longer available in the Google Cloud KMS. Included in this destruction is the key material. This key version cannot be restored.

Key Material Origin:

- If the key was uploaded from CCKM as a BYOK, then the value is INTERNAL(<source_key_name>).
- If the key was not uploaded from current CCKM as a BYOK, then the value is EXTERNAL.
- If the key material was created by Google Cloud KMS, then the value is KeyRing.

Rotation Status: Indicates whether the key is scheduled for autorotation in CCKM. AUTO is displayed in this column, if the key is scheduled for autorotation. This status *only* applies to scheduled autorotation in CCKM.

Actions: Allows you to select an action to apply to a given key or key version. To view the available actions for a key version, find the name of the key whose key version you wish to view and then expand the row to show the key versions underneath the key name. In the **Actions** column, click **Select**. The available actions are displayed. You may have to scroll through the list if you have more than two actions available for selection.
Actions available for Asymmetric keys:

- **Update Key**
- **Add Version**

Actions available for Asymmetric key versions

- **View all versions**
- **Enable**
- **Disable**
- **Schedule Destroy**
- **Cancel Destroy**
- **Download** (Public Key)

Actions available for Symmetric keys:

- **Update Key**
- **Rotate Key**

Actions available for Symmetric key versions

- **View all versions**
- **Enable**
- **Disable**
- **Schedule Destroy**
- **Cancel Destroy**

Create a Key

You have two options when you create a new key in Google Cloud KMS:

- "Create a Native Key" below
- "Upload your Own Key (BYOK)" on page 146

Part of the process of creating a new key with either option is adding it to a key ring in a given Google Cloud location and also selecting the key's purpose. The purpose of a key defines its cryptographic capabilities and the algorithms that are supported for the key's versions. The following are the supported key purposes in Google Cloud KMS:

- ENCRYPT_DECRYPT: used for Symmetric encryption. This key purpose is referred to as **Symmetric encrypt/decrypt** in CCKM.
- ASYMMETRIC_SIGN: used for Asymmetric signing. This key purpose is referred to as **Asymmetric sign** in CCKM.
- ASYMMETRIC_DECRYPT: used for Asymmetric encryption. This key purpose is referred to as **Asymmetric decrypt** in CCKM.

Create a Native Key

Create a native key in a Google Cloud KMS. In the process of creating a native key in Google Cloud KMS, key material is generated by Google Cloud KMS along with key metadata from CCKM are used. After CCKM creates a native key, it displays it as **KeyRing** under the **Key Material Origin** column within the **Google Cloud Keys** page.

For more information, see the following sections:

- "Create a Native Key (Asymmetric Key)" below
- "Create a Native Key (Symmetric Key)" on the next page

Create a Native Key (Asymmetric Key)

To create a native key that has a key purpose of **Asymmetric sign** (used for asymmetric signing) or **Asymmetric decrypt** (used for asymmetric encryption):

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, select the **New Key** button and then **Create**. The **Create Native Key** dialog box displays.
3. From the **Key Ring** box, select the key ring to which to add the key.
4. In the **Key Name** box, enter the name of the key.
5. From the **Key Purpose** box, select one of the following options depending on whether you will use the key for asymmetric signing or asymmetric encryption:
   - **Asymmetric sign**
   - **Asymmetric decrypt**
6. From the **Version Template Protection Level** area, select whether the key and key version will be stored in either **Software** or **HSM**. If no selection is made, then the default is **Software**.

7. From the **Version template algorithm** box, select the algorithm to be used in the key material to create the new version of the key. If you selected Asymmetric sign as the key purpose, then the following are your options:
   - **2048 bit RSA - PSS Padding - SHA256 Digest**
   - **3072 bit RSA - PSS Padding - SHA256 Digest**
   - **4096 bit RSA - PSS Padding - SHA256 Digest**
   - **4096 bit RSA - PSS Padding - SHA512 Digest**
   - **2048 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
   - **3072 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
   - **4096 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
   - **4096 bit RSA - PKCS#1 v1.5 padding - SHA512 Digest**
   - **Elliptic Curve P-256 - SHA256 Digest**
   - **Elliptic Curve P-384 - SHA384 Digest**

   If no selection is made, then the default is **2048 bit RSA - PSS Padding - SHA256 Digest**.
   If you selected Asymmetric decrypt as the key purpose, then the following are your options:
   - **2048 bit RSA - OAEP Padding - SHA256 Digest**
   - **3072 bit RSA - OAEP Padding - SHA256 Digest**
   - **4096 bit RSA - OAEP Padding - SHA256 Digest**
   - **4096 bit RSA - OAEP Padding - SHA512 Digest**

   If no selection is made, then the default is **2048 bit RSA - OAEP Padding - SHA256 Digest**.
8. (Optional) In the **Labels** boxes, enter the label name and value of the key and then click the **+** button. It is recommended you include “CCKM” in the name to reflect CCKM-associated information in the labels within Google Cloud list views and reports.
9. Click **Create**. The new key is displayed on the **Google Cloud Keys** page.

Create a Native Key (Symmetric Key)

To create a native key that has a key purpose of **Symmetric encrypt/decrypt** (used for symmetric encryption/decryption):

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, select the **New Key** button and then **Create**. The **Create Native Key** dialog box displays.
3. From the **Key Ring** box, select the key ring to which to add the key.
4. In the **Key Name** box, enter the name of the key.
5. From the **Key Purpose** box, select **Symmetric encrypt/decrypt**.
6. (Optional) From the **Version Template Protection Level** area, select whether the key and key version will be stored in either **Software** or **HSM**. If no selection is made, then the default is **Software**.
7. (Optional) In the **Rotation Period** box, enter the number of days for the key rotation period. Google Cloud performs and manages this key rotation in Google cloud. The minimum is 1, and the maximum is 36,500 days.
8. (Optional) In the **Rotation Start Date**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. This box displays only if you entered a rotation period in the previous step.

9. (Optional) In the **Labels** boxes, enter the label name and value of the key and then click the **+** button. It is recommended you include “CCKM” in the name to reflect CCKM-associated information in the labels within Google Cloud list views and reports.
10. Click **Create**. The new key is displayed on the **Google Cloud Keys** page.

Upload your Own Key (BYOK)

Upload the keys created from the **Key Sources** page of CCKM to the Google Cloud KMS. These are the keys (of algorithm type AES256, RSA2048, and RSA4096) you generated from the key source you are using (CipherTrust Manager or DSM). These are your "BYOK" keys.

For more information, see the following sections:

- "Upload your Own Key (BYOK) (Asymmetric Key)" below
- "Upload your Own Key (BYOK) (Symmetric Key)" on the next page

Upload your Own Key (BYOK) (Asymmetric Key)

> **Note**
> You must have the import permissions for the Google Cloud KMS to which you wish to upload the key.

To upload your BYOK to Google Cloud KMS that has a key purpose of **Asymmetric sign** (used for asymmetric signing) or **Asymmetric decrypt** (used for asymmetric encryption):

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, select the **New Key** button and then **Upload**. The **Upload a Key** dialog box displays.
3. From the **Key Provider** box, select CipherTrust Manager or DSM depending on the key source you are using for the key.
4. From the **Source Key** box, select one of the available source keys. The keys available in the drop-down menu are the keys generated from CipherTrust Manager or DSM depending on the key source you are using. Enter the first three characters of the name of the key to bring it up for selection.
5. From the **Key Ring** box, select the key ring to which the key is to be added.
6. In the **Key Name** box, enter the name of the key. If you wish to use the key name you entered when you generated this key from the **Key Sources** page, select the **Use source key name** check box. The name will display in the **Key Name** box.
7. From the **Key Purpose** box, select one of the following options depending on whether you will use the key for asymmetric signing or asymmetric encryption:
   - **Asymmetric sign**
   - **Asymmetric decrypt**
8. (Optional) From the **Version Template Protection Level** box, select whether the key and key version will be stored in either **Software** or **HSM**. If no selection is made, the default is **Software**.

9. (Optional) From the **Version template algorithm** box, select the algorithm to be used in the key material to create the new version of the key. If you selected Asymmetric sign as the key purpose, then the following are your options:
   - **2048 bit RSA - PSS Padding - SHA256 Digest**
   - **4096 bit RSA - PSS Padding - SHA256 Digest**
   - **4096 bit RSA - PSS Padding - SHA512 Digest**
   - **2048 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
   - **4096 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
   - **4096 bit RSA - PKCS#1 v1.5 padding - SHA512 Digest**

   If no selection is made, then the default is **2048 bit RSA - PSS Padding - SHA256 Digest**.
   If you selected Asymmetric decrypt as the key purpose, then the following are your options:
   - **2048 bit RSA - OAEP Padding - SHA256 Digest**
   - **4096 bit RSA - OAEP Padding - SHA256 Digest**
   - **4096 bit RSA - OAEP Padding - SHA512 Digest**

   If no selection is made, then the default is **2048 bit RSA - OAEP Padding - SHA256 Digest**.

10. (Optional) In the **Labels** boxes, enter the label name and value of the key and then click the **+** button. It is recommended you include “CCKM” in the name to reflect CCKM-associated information in the labels within Google Cloud list views and reports.

11. Click **Upload to Google Cloud**. The new key is uploaded to Google Cloud KMS and is displayed on the **Google Cloud Keys** page.

Upload your Own Key (BYOK) (Symmetric Key)

> **Note**
> You must have the import permissions for the Google Cloud KMS to which you wish to upload the key.

To upload your BYOK to Google Cloud KMS that has a key purpose of **Symmetric encrypt/decrypt** (used for symmetric encrypt/decrypt):

1. In the CCKM portal, in the left navigation bar, select **Keys**.

2. On the **Google Cloud Keys** page, select the **New Key** button and then **Upload**. The **Upload a Key** dialog box displays.

3. From the **Key Provider** box, select CipherTrust Manager or DSM depending on the key source you are using for the key.

4. From the **Source Key** box, select one of the available source keys. The keys available in the drop-down menu are the keys generated from CipherTrust Manager or DSM depending on the key source you are using. Enter the first three characters of the name of the key to bring it up for selection.

5. From the **Key Ring** box, select the key ring to which the key is to be added.

6. In the **Key Name** box, enter the name of the key. If you wish to use the key name you entered when you generated this key from the Key Sources page, select the **Use source key name** check box. The name will display in the **Key Name** box.

7. From the **Key Purpose** box, select **Symmetric encrypt/decrypt**.

8. (Optional) From the **Version Template Protection Level** box, select whether the key and key version will be stored in either **Software** or **HSM**. If no selection is made, the default is **Software**.

9. (Optional) In the **Rotation Period** box, enter the number of days for the key rotation period. Google Cloud performs and manages this key rotation in Google cloud. The minimum is 1, and the maximum is 36,500 days.

10. In the **Rotation Start Date**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. This box displays only if you entered a rotation period in the previous step.

11. (Optional) In the **Labels** boxes, enter the label name and value of the key and then click the **+** button. It is recommended you include “CCKM” in the name to reflect CCKM-associated information in the labels within Google Cloud list views and reports.

12. Click **Upload to Google Cloud**. The new key is uploaded to Google Cloud KMS and is displayed on the **Google Cloud Keys** page.

Update Key

Use **Update Key** dialog box to update the current key version of an asymmetric or symmetric key to create a *new* version of it to use. The following applies to **Update Key**:

- If the key purpose of the key you wish to update is Symmetric encrypt/decrypt, then you have the option to update the key rotation period and select a new rotation time. You also have the option to change the primary version of the key.
- If the key purpose is either Asymmetric sign or Asymmetric decrypt, then you have the option to update the Version Template Algorithm of the current key version. Changing the version template algorithm does **not** affect all of the other existing key versions of the key. This update applies to the new version of the key.
- For both asymmetric or symmetric keys, you have the option to update the Labels information.

For more information, see the following sections:

- "Update Key (Asymmetric Key)" below
- "Update Key (Symmetric Key)" on the next page

Update Key (Asymmetric Key)

> **Note**
> Keys cannot be updated when they are in a **DESTROYED** state.

To update a key that has a key purpose of **Asymmetric sign** (used for asymmetric signing) or **Asymmetric decrypt** (used for asymmetric encryption):

1. On the **Google Cloud Keys** page, click **Update Key** in the **Action** column for a particular key. The **Update key** dialog box displays. The **Project**, **Location**, **Key Ring**, and **Key Purpose** boxes and **Version Template Protection Level** area are prepopulated with the values associated with the key. These boxes and area cannot be updated.

2. From the **Version Template Algorithm** box, select the algorithm to be used in the key material of the version of the key. If you selected Asymmetric sign as the key purpose, then the following are your options:
   - **2048 bit RSA - PSS Padding - SHA256 Digest**
   - **3072 bit RSA - PSS Padding - SHA256 Digest**
   - **4096 bit RSA - PSS Padding - SHA256 Digest**
   - **4096 bit RSA - PSS Padding - SHA512 Digest**
   - **2048 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
   - **3072 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
   - **4096 bit RSA - PKCS#1 v1.5 padding - SHA256 Digest**
   - **4096 bit RSA - PKCS#1 v1.5 padding - SHA512 Digest**
   - **Elliptic Curve P-256 - SHA256 Digest**
   - **Elliptic Curve P-384 - SHA384 Digest**

   If you selected Asymmetric decrypt as the key purpose, then the following are your options:
   - **2048 bit RSA - OAEP Padding - SHA256 Digest**
   - **3072 bit RSA - OAEP Padding - SHA256 Digest**
   - **4096 bit RSA - OAEP Padding - SHA256 Digest**
   - **4096 bit RSA - OAEP Padding - SHA512 Digest**
3. (Optional) In the **Labels** boxes, enter the label name and value of the key and then click the **+** button.
4. Click **Update**.

Update Key (Symmetric Key)

> **Note**
> Keys cannot be updated when they are in a **DESTROYED** state.

To update a key with a key purpose of **Symmetric encrypt/decrypt** (used for symmetric encrypt/decrypt):

1. On the **Google Cloud Keys** page, click **Update Key** in the **Action** column for a particular key. The **Update key** dialog box displays. The **Project**, **Location**, **Key Ring**, **Key Purpose**, and **Version Template Algorithm** boxes and **Version Template Protection Level** area are prepopulated with the values associated with the key. These boxes and area cannot be updated.
2. (Optional) In the **Rotation Period** box, enter the number of days for the key rotation period. Google Cloud performs and manages this key rotation in Google Cloud. The minimum is 1, and the maximum is 36,500 days.
3. (Optional) In the **Rotation Start Date** box, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. This box displays only if you entered a rotation period in the previous step.
4. (Optional) To set a version of this key as the primary version, select the **Set as Primary Version** check box and then select the version number from the drop-down menu.
5. (Optional) In the **Labels** boxes, enter the label name and value of the key and then click the **+** button.
6. Click **Update**.

Rotate a Key

CCKM supports both scheduled rotation (or autorotation) and manual rotation of *symmetric* keys within Google cloud. When a key is rotated in CCKM for Google Cloud or in Google Cloud KMS, new key material is added to the key and the new version of the key is set as the primary key version. It is the primary key version of the key that Google Cloud KMS uses to encrypt data.

When you rotate a symmetric key, the following applies:

- Any data that was encrypted with previous versions of the key is not automatically re-encrypted.
- Existing key versions are not automatically disabled or destroyed.

Google Cloud KMS does ***not*** support autorotation and manual rotation of asymmetric keys. However, CCKM does provide this support partially. With either autorotation or manual rotation of asymmetric keys in CCKM, new key material is added to the key to create a new key version through the use of the **Add Version** action available on the **Google Cloud Keys** page. *However*, to begin using the new key version for either signing or encryption, you are required to take additional steps. Part of these additional steps includes distributing the public key of the new key version. CCKM provides support for downloading the public key of the new key version that you can distribute. For information about these additional steps, refer to Considerations for Asymmetric Keys within the online documentation for Google cloud. For more information about downloading a public key of a new key version of an asymmetric key, see "Download Public Key (Asymmetric Key Only)" on page 156.

Autorotation of a symmetric or an asymmetric key requires that this feature be configured in the **Schedules** page prior to enabling the feature from the **Google Cloud Keys** page. See "Schedule Key Rotation" on page 158 for more information about scheduling key rotation.

For more information, see the following sections:

- "Manually Rotate a Key (Symmetric Key)" below
- "Enable Autorotation of Key (Symmetric Key)" on the next page
- "Manually Rotate a Key (Add New Key Version) (Asymmetric Key) " on the next page
- "Enable Autorotation of Key (Add New Key Version) (Asymmetric Key)" on page 152

Manually Rotate a Key (Symmetric Key)

Manually rotating a symmetric key allows CCKM to add new key material to the selected key and set the new key version as the primary key version to be used for encrypting data.

After a new key version is generated, it will be added to the list of key versions as a primary version on the **Google Cloud Keys** page.

> **Note**
> During the rotation of a symmetric key, the template protection level that you initially configured for the key is used to generate the new key version.

To manually rotate a symmetric key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key to manually rotate.
3. Select **Rotate Key** from the **Actions** drop-down menu. The **Rotate Google Cloud Key** dialog box is displayed. The **Version Algorithm** box is prepopulated with the algorithm of Symmetric encrypt/decrypt.

4. From **Key Provider**, select the key provider (also known as key source) from which to generate the new key material for the key. Two options of the three displayed are available for selection (the current key source you are currently using and Native):
   - **DSM**—Key is to be generated from DSM.
   - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager.
   - **Native**—Key is to be generated natively in the designated Google Cloud KMS using the CCKM UI. Note that this option is equivalent to generating a key in the Google Cloud portal.
5. From the **Source Key** box, select the source key to use for the new key version. This box displays only if you selected either **DSM** or **CipherTrust Manager** in the previous step.
6. Click **Rotate**.

Enable Autorotation of Key (Symmetric Key)

Enabling autorotation of a symmetric key allows CCKM, at a scheduled time, to add new key material to the selected key and set the new key version as the primary key version to be used for encrypting data.

After a new key version is generated, it will be added to the list of key versions as a primary version on the **Google Cloud Keys** page.

> **Note**
> During the rotation of a symmetric key, the template protection level that you initially configured for the key is used to generate the new key version.

To enable autorotation of a symmetric key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key to autorotate.
3. Select **Rotate Key** from the **Actions** drop-down menu. The **Rotate Google Cloud Key** dialog box is displayed.
4. Select the **Auto Rotate** tab. The **Auto Rotate** dialog box is displayed. The **Version Algorithm** box is prepopulated with the algorithm of Symmetric encrypt/decrypt.
5. From **Enable autorotation on this key?**, select **On** to enable the autorotation of this key. By default, this toggle is initially set to **Off**.
6. From the **Schedule** drop-down menu, select a rotation schedule to which to apply to this key. Any key-rotation schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
7. From **Key Provider**, select the key provider (also known as key source) from which to generate the new key material for the key. Two options of the three displayed are available for selection (the current key source you are currently using and Native):
   - **DSM**—Key is to be generated from DSM.
   - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager.
   - **Native**—Key is to be generated natively in the designated Google Cloud KMS using the CCKM UI. Note that this option is equivalent to generating a key in the Google Cloud portal.
8. Click **Submit**.

Manually Rotate a Key (Add New Key Version) (Asymmetric Key)

Manually rotating an asymmetric key allows CCKM to add new key material to the selected key to create a new key version. You are required to take additional steps to begin using the new key version for either signing or encryption.

After a new key version is generated, it will be added to the list of key versions on the **Google Cloud Keys** page.

> **Note**
> During the rotation of an asymmetric key, the template protection level that you initially configured for the key is used to generate the new key version.

To add a new version of an asymmetric key to the key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the asymmetric key to which to add a new version.
3. Select **Add Version** from the **Actions** drop-down menu. The **Add Version for Google Cloud Key** dialog box is displayed.
4. From **Key Provider**, select the key provider (also known as key source) from which to generate the new key material for the new key version. Two options of the three displayed are available for selection (the current key source you are currently using and Native):
   - **DSM**—Key is to be generated from DSM.
   - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager.
   - **Native**—Key is to be generated natively in the designated Google Cloud KMS using the CCKM UI. Note that this option is equivalent to generating a key in the Google Cloud portal.
5. From the **Source Key** box, select the source key to use for this new key version. This box displays only if you selected either **DSM** or **CipherTrust Manager** in the previous step.
6. (Optional) From the **Version Algorithm** box, select the algorithm to use for the new key version. This box is prepopulated with the algorithm from the template version algorithm initially configured for the key. If you selected either **DSM** or **CipherTrust Manager** as the key provider, you have the option of changing the version algorithm. If you selected **Native**, you do not have the option to change the version algorithm.
7. Click **Add Version**.

Enable Autorotation of Key (Add New Key Version) (Asymmetric Key)

Enabling autorotation of an asymmetric key allows CCKM, at a scheduled time, to add new key material to the selected key to create a new key version. You are required to take additional steps to begin using the new key version for either signing or encryption.

After a new key version is generated, it will be added to the list of key versions on the **Google Cloud Keys** page.

> **Note**
> During the rotation of an asymmetric key, the template protection level that you initially configured for the key is used to generate the new key version.

To enable autorotation of adding a new version of an asymmetric key to the key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the antisymmetric key to which to add a new version.
3. Select **Add Version** from the **Actions** drop-down menu. The **Add Version for Google Cloud Key** dialog box is displayed.
4. Select **Auto Rotate** tab. The **Auto Rotate** dialog box displays.
5. From **Enable autorotation on this key?**, select **On** to enable the autorotation of this key. By default, this toggle is initially set to **Off**.

6. From the **Schedule** drop-down menu, select a rotation schedule to which to apply to this key. Any key-rotation schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
7. From **Key Provider**, select the key provider (also known as key source) from which to generate the new key material for the new key version. Two options of the three displayed are available for selection (the current key source you are currently using and Native):
   - **DSM**—Key is to be generated from DSM.
   - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager.
   - **Native**—Key is to be generated natively in the designated Google Cloud KMS using the CCKM UI. Note that this option is equivalent to generating a key in the Google Cloud portal.
8. (Optional) From the **Version Algorithm** box, select the algorithm to use for the new key version. This box is prepopulated with the algorithm from the template version algorithm initially configured for the key. If you selected either **DSM** or **CipherTrust Manager** as the key provider, you have the option of changing the version algorithm. If you selected **Native**, you do not have the option to change the version algorithm.
9. Click **Submit**.

View All Key Versions

From the **Keys** page, you can access the **Key Versions** dialog box to view all the versions of a key and its associated information. At the top half of this dialog box, the following information about the selected key version is displayed: **Key**, **Purpose**, **Primary version**, **Template Protection Level**, **Template Algorithm**, **Rotation Period**, **Creation Date**, and **Next Rotation Time**. Note that the **Rotation Period** and **Next Rotation Time** refers to the key rotation that Google Cloud KMS manages and performs. At the bottom half of this dialog box, a list of all the key versions with the following columns is displayed: **Version**, **State**, **Algorithm**, **Created On**, **Key Material Origin**, and **Actions**. From the **Actions** column, you can select an action available to key versions to apply to a selected key version. See "Keys" on page 140 for more information about the available options for **Actions**. You can display the next 5, 10, 15, 20, 25, 50, or 100 entries at the bottom of the **Key Versions** page using the **Show entries** drop down menu. You can view each available page by scrolling through the pages by clicking the less than "**<**" or greater than "**>**" symbols or by clicking on a given page number.

To view all of the key versions of a key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key versions you wish to view.
3. Click the key name to expand the row to show the key versions underneath the key name.
4. Click **View all versions**. The **Key Versions** dialog box displays. You can scroll through list to view all of the key versions.
5. Click the **Close** button to close the **Key Versions** dialog box.

Synchronize Keys

When you initially log into the CCKM User Portal, CCKM retrieves all of the projects and project locations that meet the following criteria and saves this information to MongoDB:

- active projects
- where Google Cloud KMS is enabled
- billing is enabled
- belong to the user's current organization

CCKM *also* performs the *first* synchronization with Google Cloud KMS. During this synchronization, the key metadata of all of your Google Cloud keys from your key rings within the downloaded projects are also downloaded to CCKM and saved to MongoDB. Essentially, it is the key metadata for each of the keys that is downloaded during this first key synchronization. All of the downloaded information is available to you the next time you log back into CCKM as this information is saved to MongoDB.

If you wish to manually synchronize your keys from your projects (all or selected) and key rings from Google Cloud KMS with CCKM, you can do so using the **Synchronize** button in the **Google Cloud Keys** page. Included in the synchronizing of the keys are EKM keys (if these exists within your projects). Synchronizing your key rings is optional. By default, the **Also synchronize Key Rings** check box within the **Synchronize Keys** dialog box is deselected. When this box is deselected, synchronization does not include downloading your key rings. In this case, only key metadata of the keys from the key rings that were *previously* downloaded to CCKM during the last synchronization will be downloaded. When this box is selected, then all of your key rings are downloaded including new ones that may have been added to your projects within Google Cloud KMS since the last synchronization of key rings took place. In addition, key metadata of all of your keys are also downloaded as part of the synchronization.

> **Note**
> If you do not select the **Also synchronize Key Rings** check box, the permissions on the saved key rings are retrieved during key synchronization even though your key rings are not downloaded.

You have the option to schedule key synchronization, which you perform in the **Schedules** page. See "Schedule Key Synchronization" on page 160 for more information.

To synchronize your keys and (optionally) key rings from selected projects:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. From the **Google Cloud Keys** page, click **Synchronize**. The **Synchronize Key** dialog box displays.
3. Under **Select projects to synchronize**, select the **Select All** check box to select all of your projects. To deselect a project, click the **X** next to the project name. You can also click in the **Select projects to synchronize** box and bring up the list of projects from which you can select a project.
4. (Optional) If you wish to also synchronize your key rings, select the **Also synchronize Key Rings** check box.
5. Click **Synchronize**.

Disable or Enable Key Version

You have the option to disable a key version in CCKM. Any key version that is disabled has a version state of DISABLED as displayed in the **Version State** column on the **Google Cloud Keys** page. After a key version is disabled, any data that was encrypted using the given key version cannot be accessed. Essentially, the key version can no longer be used for cryptographic operations. However, the key material of the disabled key version is still available for use. You can re-enable a key version and set its version state back to ENABLED so that you can use this key version again.

For more information, see the following sections:

- "Disable Key Version" below
- "Enable Key Version" on the next page

Disable Key Version

To disable a key version:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key version you wish to disable.
3. Click the key name to expand the row to show the key versions underneath the key name.
4. Click **View all versions**. The **Key Versions** dialog box displays. Scroll through the list to view the information for each version.
5. Find the key version you wish to disable.
6. Select **Disable** from the **Actions** drop-down menu.

Enable Key Version

To re-enable a key version that is in a DISABLED version state:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key version you wish to enable.
3. Click the key name to expand the row to show the key versions underneath the key name.
4. Click **View all versions**. The **Key Versions**dialog box displays. Scroll through the list to view the information for each version.
5. Find the key version you wish to enable
6. Select **Enable** from the **Actions** drop-down menu.

Schedule (or Cancel) Destroy Key Version

CCKM supports the scheduling of a key version for destruction in Google Cloud KMS. A key version that is scheduled for destruction has a version state of **Scheduled for destruction** as displayed in the **Version State** column on the **Google Cloud Keys** page. Destroying a key version equates to destroying the version's key material. After destruction, the key version and the key material cannot be recovered. Google Cloud performs the key version destruction after 24 hours has elapsed. After a key version is destroyed, any data that was encrypted using the given key version cannot be accessed.

Note that you can cancel the destruction of a key version in CCKM before the 24 hours has elapsed to restore the key material for the given key version. Google Cloud KMS does not perform the destruction after this cancellation is made in CCKM. After the cancellation, the key version state changes from DESTROY_SCHEDULED to DISABLED. In this version state, you can enable the key. And from the ENABLED version state, the key version is available for use for cryptographic operations once again.

> **Note**
> If you schedule the destruction of a key version, be sure to synchronize your keys with the Google Cloud KMS from the CCKM **Google Cloud Keys** page after the 24 hours has elapsed. Otherwise, your CCKM keys will be out of sync with Google Cloud KMS keys, and the option to cancel the destruction of a key version will still display as being available in CCKM. If you attempt to cancel the key version destruction after the 24 hours has elapsed, the operation will fail because the key material of the key version has already been destroyed within Google Cloud KMS.

For more information, see the following sections:

- "Schedule Destroy Key Version" on the facing page
- "Cancel Schedule Destroy Key Version" on the facing page

Schedule Destroy Key Version

To schedule the destruction of a key version:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key version you wish to destroy and then expand the row to show the key versions underneath the key name.
3. To see all key versions underneath the key name, click **View all versions**. The **Key Versions** dialog box displays. Scroll through the list to find the key version you wish to destroy.
4. Select **Schedule Destroy** from the **Actions** drop-down menu for the selected key version. The **Schedule Destroy Key Material** dialog box is displayed. A warning displays indicating that this action will schedule key material destruction for "<path to key version>" from your Google Cloud Key Ring in 24 hours. The message also indicates to type the confirmation phrase (I wish to destroy the key material) in the text box if you wish to schedule the key material destruction.
5. Enter or copy/paste the confirmation phrase in the text box.
6. Click **Destroy**.

Cancel Schedule Destroy Key Version

To cancel the scheduled destruction of a key version and restore it:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key version you wish to restore and then expand the row to show the key versions underneath the key name.
3. Find the key version you wish to restore.
4. Select **Cancel Destroy** from the **Actions** drop-down menu. The **Cancel Schedule Destroy Key Material** dialog box is displayed. A warning displays indicating that this action will restore key material to key version "<path to key version>" in your Google Cloud Key Ring if key version has not been destroyed yet.
5. Click **Restore**.

Download Public Key (Asymmetric Key Only)

Before you can begin using a new key version of an asymmetric key after key rotation, you are required to take additional steps. Part of these additional steps include distributing the public key of the new key version. CCKM provides support for downloading a CCKM public key of the new key version.

To download the CCKM public key of a new asymmetric key version:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, find the name of the key whose key version you wish to download the public key and then expand the row to show the key versions underneath the key name.
3. Find the key version for which to download the public key.
4. Select **Download** from the **Actions** drop-down menu. The public key file is downloaded to your laptop or desktop. When prompted as to whether to keep or discard the file type, select **Keep**.

Export Keys

Export Keys allows you to export the contents of the **Google Cloud Keys** page into a report in a CSV format.

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Google Cloud Keys** page, click **Export Keys**. The report file is downloaded to your laptop or desktop.

Reports

CCKM includes the following reports:

- **Combined Key Activity Reconciliation Report**

  Provides the history of a key for a selected period using the following information (in column headings):
  - Key name
  - Project
  - Key Ring
  - Key Activity
  - Cloud Key Manager Key Activity
  - Timestamp
  - Cloud Key Manager Timestamp
- **Google Cloud Key Activity Report**

  Illustrates the activity type performed by a specific user and at the stated time. Information provided in the column headings of this report is the following:
  - Key name
  - Project
  - Key Ring
  - Key Activity
  - Region
  - Origin
  - Modified By
  - Modified Time
- **Google Cloud Key Aging Report**

  Provides information about key expiration using the following information (in column headings):
  - Key Name
  - Project
  - Key Ring
  - Region
  - Key Rotation
  - Schedule Key Deletion Date

- **Cloud Key Manager User Action Report**

  Lists the users and their actions including creating, uploading, rotating, and deleting keys in the CCKM portal. Information provided in the column headings of this report is the following:

  - User Name
  - Delete Key
  - Upload Key
  - Schedule Delete Key Material
  - Cancel Delete
  - Rotate Key

Generate a Report

1. In the CCKM portal, in the left navigation bar, select **Reports**.
2. Under **Report Name**, select the report you wish to run.
3. From **Date Range**, click **Last day** to run the report for the activities from the previous day or **Specific** to run the report using a date range. If you choose to run by a date range, enter the start and end dates in the format MM/DD/YY or select the dates from the calendar button at the right of the box.

   If you are running the **Google Cloud Key Aging Report**, click **Next week** to run the report for the activities for the next week or **Specific** to run the report using a date range.
4. Click **Run Report**. A link to the report is made available under the **Last Run** column. To view the report, click on the link and the report displays.
5. To download the report, click **Download** at the top right of the page and select the report format (PDF or CSV) to download.

Delete a Report

1. From the **All Reports** page, select the specific report you wish to delete. The report displays.
2. From the **Actions** column, click **Delete**. The **Delete Current Report** dialog box with a warning that you are about to delete a current report run on a specified date and time.
3. Click **Delete** to confirm. The report is deleted from CCKM.

Schedules

The **Schedules** page allows Cloud administrators to view and schedule the following job types:

- Key Rotation
- Key Synchronization

CCKM provides support for multiple schedules for each of the job types.

At the top right of the **Schedules** page, the **Add Schedule** button is available from which to add a scheduled job type.

Schedule Key Rotation

CCKM provides support for multiple schedules of the rotation (autorotation) of symmetric keys within Google cloud. During a scheduled key rotation, new key material is added to the key and the new version of the key is set as the primary key.

When you rotate a symmetric key, the following applies:

- Any data that was encrypted with previous versions of the key is not automatically re-encrypted.
- Existing key versions are not automatically disabled or destroyed.

Google Cloud KMS does *not* support automatic rotation of asymmetric keys. However, CCKM does support automatic rotation of these keys by adding new key material to the key to create a new version. To begin using the new key version for either signing or encryption, you are required to take additional steps. For information about these additional steps, refer to Considerations for Asymmetric Keys.

Scheduled key rotation (for both symmetric and asymmetric keys) requires that this feature be configured in the **Schedules** page prior to enabling the feature from the **Keys** page. For more information about enabling scheduled key rotation of symmetric and asymmetric keys, refer to "Rotate a Key" on page 150.

You can pause or delete a scheduled key rotation in the **Schedules** page.

Set up a Key Rotation Schedule

To set up a key rotation schedule:

> **Note**
> The access permissions of an Admin role (roles/cloudkms.admin) within Google Cloud KMS are required to set up a key rotation schedule.

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays. Your username associated with your Google Cloud account is automatically populated in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_ROTATION**.
4. From **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).

   If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. Set the repeat interval of the key rotation by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

   If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
8. Click **Save** to save the setting.

Pause a Key Rotation Schedule

To pause a key rotation schedule, select **Yes** for **Pause** within the **Add Schedule** dialog box and then click **Save**. The setting is saved. To resume the rotation schedule after pausing it, select **No** for **Pause** and then click **Save**. The setting is saved.

Delete a Key Rotation Schedule

To delete a key rotation schedule, select **Delete** from the **Actions** drop-down menu from the **Schedules** page.

Schedule Key Synchronization

CCKM allows for multiple schedules of the synchronization of Google Cloud keys. Synchronizing downloads the key metadata of any keys that were created in Google Cloud KMS, outside of the CCKM portal, into your CCKM. Scheduled key synchronization requires that this feature be configured in the **Schedules** page. You can also pause or delete a scheduled key synchronization in the **Schedules** page.

Set up a Key Synchronization Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays. Your username associated with your Google Cloud account is automatically populated in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_SYNCHRONIZATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the key synchronization by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).

   If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers. Set the repeat interval of the key synchronization by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

   If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
8. Click **Save** to save the setting.

Logs

CCKM maintains logs for events, such as login, key generation, synchronizing of keys, and deletion of keys, and displays these logs in the **All Logs** list of the **Logs** page. For each logged event listed on the **Logs** page, the following information is displayed:

- Event name
- Severity level
- Date and time of the event
- Event message
- User (name) associated with the event

You can search for a specific logged event from the **Logs** page using the **Search** box. Allowable filters on the search are event name, severity level, event message, and user (name).

Key Sources

The **Key Sources** page displays all of the keys that are generated from the key source you are using. Three tabs are available at the top of the page:

- **CipherTrust Manager:** Use this tab to use CipherTrust Manager as your key source. The keys are generated and stored within CipherTrust Manager.
- **DSM Key:** Use this tab to use DSM as your key source. The keys are generated and stored within DSM.
- **nShield Key:** This tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for Google Cloud.

> **Note**
> CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

Keys listed on the **Key Sources** page are listed by name, cloud ID, key type, key algorithm, creation date, key description, and actions you can take regarding that key. Note that the key algorithms currently supported for a Google Cloud key in CCKM are AES and RSA.

A **Search** box is available at the top left of the **Keys Sources** page allowing for a search of a specific key by entering any one of the following parameters:

- Name (key name)
- Cloud ID
- Key Type
- Algorithm
- Description
- Created By (a key created by a specific user)

Add (Generate) a Key

From the **Key Sources** page, you can add (or generate) keys in CCKM using CipherTrust Manager or DSM as a key source. These keys are of the AES and RSA algorithm types.

Add (Generate) a CipherTrust Manager Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **CipherTrust Manager Key** tab.
3. Select **Add Key > GoogleCloud Key**.
4. In the **Add CipherTrust Manager Key** dialog box, select the name of the user of the key from the **User** drop-down menu. Service is fixed to **GoogleCloud**.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. From the **Algorithm** box, select the algorithm (**AES256**, **RSA 2048**, or **RSA 4096**) to apply to the key.
8. Click **Save** to save the new key to CCKM. The new key is added to the keys list. (Reset clears the screen before it has been saved.)

Add (Generate) a DSM Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **DSM Key** tab.
3. Select **Add Key > GoogleCloud Key**.

4. In the **Add DSM Key** dialog box, select the name of the user of the key from the **User** drop-down menu.
5. In the **Source Key Tier**, select one of the following as a key source of the key:
   - **DSM**—Generate a key from DSM and import it into a Google cloud.
6. In the **Name** box, enter the name of the new key.
7. Enter a description of the key in the **Description** box.
8. From the **Algorithm** box, select the algorithm (**AES256**, **RSA 2048**, or **RSA 4096**) to apply to the key.
9. Click **Save** to save the new key to CCKM. The new key is added to the keys list. (Reset clears the screen before it has been saved.)

Add (Generate) an nShield Key

The **nShield Key** tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for Google cloud.

Delete a Key

Deleting a key from the **Key Sources** page deletes the key from CCKM. When deleting a key from CCKM, the key is deleted from the key source you used to generate the key.

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **CipherTrust Manager** or **DSM Key** tab depending on the key source you used to generate this key initially.
3. Click **Delete** next to the target key.
4. The **Delete Key** dialog box with a Warning display asking you to confirm the key deletion by entering the supplied phrase.
5. Enter or copy/paste the confirmation phrase.
6. Click **Delete** to confirm.

Manage Keys in Key Sources List Page

In addition to adding and deleting keys, the **Key Sources** page allows you to sort the existing keys by Name, Cloud ID, Key Type, Algorithm, and Created Date.

Settings

To access the **Settings** page in CCKM, select **Settings** from the left-hand navigation bar. From the **Settings** page, you can:

- Enter the emails of the recipients who are to receive alerts and reminders.
- Set up key alerts.
- Configure remote syslog servers to which to send syslog messages.
- Configure Service account to use for authentication to CCKM API.
- View information about the software release number of the CCKM you are running including the build and version number.

Adding Recipients' Emails (for Alerts and Reminders)

Use the **General** settings tab to enter the designated recipients' emails for email alerts and reminders to be sent from the system. Enter the recipients' email addresses separated by a comma in the **Alert Format** box.

Setting Keys Alerts

Use the **Keys Alert** tab from the **Settings** page to enable whether to send key alerts to specified recipients. When a key is deleted from Google cloud, restored or uploaded to Google cloud, a key alert can be sent. The recipients' emails are set up in the **General** tab. By default, no alerts or reminders are sent.

To set up key alerts or a reminder to manually rotate (or rekey) a key:

1. In the **Settings** page, select the **Keys Alert** tab.
2. Under **Alerts**, click **On** for each of the alert types and reminder you wish to enable:
   - Rekey Alert (reminder)
   - Delete Google Cloud Key Version
   - Restore Google Cloud Key Version
   - Upload Google Cloud Key
3. If you set a reminder for **Rekey Alert**, then in the **Days** box, enter the number of days for the frequency an email reminder to manually rotate a key is sent.
4. Click **Save** to save the setting.

Configuring Syslog Server

Use the **Syslog** tab from the **Settings** page to configure a remote syslog server to which to send syslog messages. Note that the default port number of 514 and the facility name of “LOCAL1” for the syslog server display in the **Port** and **Facility** boxes, respectively. These boxes are not available for modification.

Configure Remote Syslog Server

To configure a remote syslog server:

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.
2. Enter the hostname or IP address of the syslog server in the **Hostname** box.
3. (Optional) Enter a description of the syslog server in the **Description** box.
4. Click **Save** to save the setting.

Delete Remote Syslog Server Configuration

To delete the configuration of a remote syslog server, click **Delete** from the **Syslog** dialog box. No syslog messages are sent to the remote syslog server after the deletion of this configuration.

Adding Google Cloud Service Account to CCKM

Use the **Service Account** tab from the **Settings** page to add your Google Cloud service account to CCKM and create a password for it. Your service account must already be created in your Google Cloud Platform console before you proceed with adding your service account to CCKM.

After submitting the request to add your Google service account to CCKM in the **Add Service Account** dialog box, CCKM will call the Google Cloud API to add a key pair to the service account. After the key pair is added, CCKM downloads the private key and saves it in the database as an encrypted key. CCKM also saves your password hash in the database.

Later, you can use the service account email and password to authenticate to CCKM REST API.

The following columns display on the **Service Account** tab:

- Service Account Email
- Key ID
- Key Creation Date
- Key Expiration Date
- Actions

Add Service Account

To add a Google service account to CCKM:

1. In the **Settings** page, select the **Service Account** tab. The **Service Account** dialog box displays.
2. Click **Add Service Account**. The **Add Service Account** dialog box displays.
3. In the **Email** box, enter the email of your Google service account.
4. In the **Password** box, enter your password for the service account. The password must be at least 8 characters long and contain one upper case letter, one lower case letter, one digit, and one special character from a limited set of special characters "!@#$%^&*".
5. In the **Confirm Password** box, enter your password for the service account again to confirm the password.
6. Click **Submit**.

Update Service Account

You have the option to change the service account password and change the key pair for service account from the **Service Account** tab. To change the key pair, select to rotate the key. When Google Cloud rotates the key pair, the current key pair will be deleted, and a new key pair will be created. CCKM then encrypts the new private key and your password hash and saves this data in the CCKM database.

To update the information for Google Cloud service account:

1. In the **Settings** page, select the **Service Account** tab. The **Service Account** dialog box displays.
2. Click **Add Service Account**. The **Add Service Account** dialog box displays.
3. In the **Email** box, enter the email of your Google Cloudservice account.
4. In the **Password** box, enter your password for the service account. The password must be at least 8 characters long and contain one upper case letter, one lower case letter, one digit, and one special character from a limited set of special characters "!@#$%^&*".
5. In the **Confirm Password** box, enter your password for the service account again to confirm the password.
6. Click **Submit**.

Delete Service Account

To delete a Google Cloud service account:

1. In the **Settings** page, select the **Service Account** tab. The **Service Account** dialog box displays.
2. Select **Delete** from the **Actions** drop-down menu. The **Confirm Deletion of Service Account** dialog box is displayed.
3. Click **Confirm**.

About

Use the **About** tab to view the build number, version number, patch number (if applicable), and build number of the CCKM you are running.

Chapter 16: CCKM for IBM Key Management

Overview

CCKM provides support for IBM Key Protect, which is IBM’s encryption key management service (KMS) used to provision encrypted keys for apps across IBM Cloud. IBM Key Protect safeguards keys using FIPS 140-2 Level 3 certified cloud-based hardware security modules (HSMs).

With the support of IBM Cloud BYOK keys within IBM Key Protect, you can upload your BYOK keys within CCKM to IBM Key Protect.

Note: This documentation assumes that you are familiar with using IBM Key Protect and IBM Cloud Identity and Access Management (IAM). For details about these IBM technologies, refer to the IBM cloud documentation at https://cloud.ibm.com/docs.

Types of Encryption Keys Supported within IBM Key Protect

The following encryption keys are employed and managed within IBM Key Protect, which you can also manage from CCKM:

- Standard key—This is a symmetric key used to directly encrypt and decrypt secrets, such as passwords or encryption keys.
- Root key—This is a symmetric key-wrapping key used to protect other cryptographic keys that use envelope encryption.

Both an IBM native key and a BYOK (non-native) key can either be a standard or a root key.

IBM Cloud Identity and Access Management (IAM)

IBM Key Protect for IBM Cloud provides a centralized access control system using IBM IAM to assist you in managing users and access for your encryption keys within your organization. IBM Key Protect employs IAM roles in the management of your keys. These roles have different access permissions for the key management service. For detailed information about IBM Cloud IAM, refer to topics covering “IBM Cloud Identity and Access Management” within the IBM cloud documentation.

As an account administrator in IBM Cloud, the actions you can perform include the following:

- assign and manage access for users
- create resource groups
- create access groups
- create service instances

To provide access for users and/or access groups in IBM Key Protect, you can create policies that define the resource (s) the users and/or access groups are permitted to access and a role or roles that define the type of access permissions (actions allowed) on the given resources.

The following are the two types of roles supported within IBM IAM:

- Platform management role—used to grant permissions at the account level, such as creating or deleting instances in your IBM Cloud account. To access your Key Protect instances through CCKM, ensure to set the Platform management role of Viewer (at the least). The supported roles are Administrator, Operator, Editor, and Viewer.
- Service access role—used to define the actions a user or service can perform on an IBM Key Protect instance, such performing API calls. The following table provides the supported Service access roles along with the IBM Key Protect access permissions allowed for each:

Service Access Role	Access Permissions (Allowed Actions) in IBM Cloud	Notes
Reader	• View keys • Wrap keys • Unwrap keys	CCKM does not support wrap and unwrap keys.
Writer	• Create keys • List keys • Retrieve keys • Rotate keys • Wrap keys • Unwrap keys • Set keys for deletion • Unset keys for deletion • Create import tokens • Retrieve import tokens	CCKM does not support the following: • Set keys for deletion • Unset keys for deletion • Wrap keys • Unwrap keys
Manager	• All actions that a reader or a writer is allowed to perform • Update keys • Delete keys	

For detailed information about these roles, refer to the topics covering “Managing user access” and “Roles and Permissions” within IBM cloud documentation.

Action Permissions Mapping for CCKM and IBM Key Protect

A mapping of each of the supported CCKM actions (you can perform in the IBM cloud) to an IBM Key Protect action along with the required Service access role(s) to perform the action is provided in the following table:

CCKM Action	IBM Key Protect Actions Required	Required Service Access Role
Synchronize	List keys Retrieve keys	Reader/Writer/Manager
View keys	List keys Retrieve keys	Reader/Writer/Manager
Create native key	Create keys Retrieve keys	Writer/Manager
Upload standard key	Create keys Retrieve keys	Writer/Manager
Upload root key	Create keys Retrieve keys Create import tokens Retrieve import tokens	Writer/Manager
Update key	Retrieve keys Set key policies List key policies	Manager
Delete key	Delete keys	Manager
Rotate key	Rotate keys Retrieve keys	Writer/Manager
Export Keys		N/A

CCKM's Interaction with IBM Key Protect

CCKM uses IBM Cloud credentials to work with the IBM Cloud APIs. To log into the IBM key management service of CCKM, you are required to have the access credentials of IBM username and IBM Cloud API key. To obtain an IBM Cloud API key, generate it in the IBM portal. For information about generating an IBM Cloud API key, see section "Prerequisites on IBM Key Protect" on the next page. You can create a IBM credentials file in a JSON format to use when logging into the IBM key management service of CCKM. For more information, see section "Creating an IBM Credentials File in a JSON Format " on page 170.

You can access IBM Key Protect instances by setting access policies on a user or an access group with permissions to Key Protect service.

The following access policies are allowed for using CCKM:

- setting a service access policy for a single IBM key
- setting a service access policy for a Key Protect instance
- setting a service access policy for a resource group containing Key Protect instances

In order to manage IBM keys in CCKM, a user must have **Viewer** platform permission on every Key Protect instance that contains the keys.

After logging into CCKM, CCKM retrieves all access groups that a user belongs to (if any) and all resource groups to which the user has read access (if any).

For each IBM Key Protect instance, CCKM retrieves the following:

- Platform management role the user has on the instance (Administrator, Operator, Editor, or Viewer)
- Service access role the user has on the instance (Reader, Writer, or Manager)

Prerequisites on IBM Key Protect

In this section, it is assumed you will use access groups and resource groups when configuring the IBM key management service in CCKM.

Prior to configuring and accessing the IBM key management service in CCKM, do the following:

> **Note**
> You can perform the first two steps in either order.
>
> Steps 2 through 4 are covered in this section below.

1. Create an instance of IBM Key Protect in IBM cloud by using the IBM Cloud console or the IBM Cloud CLI. For information on how to create an instance in IBM Key Protect, refer to the topic of “Provisioning the service” in the IBM cloud documentation.
2. Generate an API key in the IBM portal.
3. Create an access group and then add users to your access group.
4. Assign an access policy to the access group.

Create an Instance of IBM Key Protect in IBM Portal

For information on how to create an instance in IBM Key Protect, refer to the IBM cloud documentation covering the topic of “Provisioning the service”.

Generate API Key in IBM Portal

To generate an API key in the IBM portal:

> **Note**
> Be sure to save the generated API key to a file as you will not be able to view or retrieve it later from the IBM portal.

1. From the menu bar, click **Manage > Access (IAM)**.
2. Select **IBM Cloud API Keys**.
3. Click **Create an IBM Cloud API key**.
4. Enter a name and description for the API key.
5. Click **Create**.
6. Click **Copy** to copy the API key or click **Download** to download it.
7. Save the API key.

Proceed to the next section, “Create Access Group”.

Create Access Group

To create an access group, you need to first set up the group and then add users to your access group.

To create an access group:

1. From the menu bar, click **Manage > Access (IAM)**, and select **Access Groups**.
2. Click **Create**.
3. Enter a name and description (optional) for your group
4. Click **Create**.

Proceed to the next section.

Add Users to Access Group

To add users to your access group:

1. Select the name of the access group to which to add users.
2. Click **Add users**.
3. Select the user to add from the list, and then click **Add to group**.

Proceed to the next section, “Assign access to your groups”.

Assign Access to your Access Groups

After you create your access group and add users to it, you can assign an access policy to the group. Note that any policy that you set for the access group is applicable to all entities within the group.

1. From the menu bar, click **Manage > Access (IAM)**, and select **Access Groups**.
2. From the row for the group that you want to assign access, select the **Actions List** of actions icon menu, and click **Assign access**.
3. Select the name of the access group that you want to assign access.
4. Select the **Access policies** tab, and then click **Assign access**.
5. Select the type of access that you want to assign.
6. Click **Add > Assign**.

Creating an IBM Credentials File in a JSON Format

When prompted for your IBM login credentials on the IBM login page within CCKM, you can either enter your IBM Username and API Key, or upload an IBM credentials file. The IBM credentials file should be in a .json format (can be written in Notepad and saved for upload).

The following is sample .json file (add your username and API key to the file):

```
{
    "ibm_username": "ibmusernamehere",
    "ibm_apikey": "XQq81eP78EHSG9EZbXwgAYJUfRMpC8wNiNkqiFudkZeT"
}
```

Access CCKM for IBM

For details about accessing CCKM as a CCKM user for the CCKM service for IBM Key Management, see Chapter 3: "About CCKM Admin and User Portals" on page 24. This chapter includes information on the ways in which to access the CCKM service for Key Management.

IBM CCKM User Portal

The CCKM User Portal for IBM contains the following menu options in the left-hand navigation bar:

- **Home**: **Home** page displays key operation and count metrics.
- **Keys**: **Keys** page displays key management overview and provides the upload, create, and synchronize functionality, and the manual rotation, auto-rotation and update key operations available in CCKM.
- **Reports**: **All Reports** overview page links to detailed individual reports.
- **Schedule**: In **Schedule**, you can schedule automatic key rotation and key synchronization.
- **Logs**: **Logs** page lists individual logs that can be searched, sorted, and viewed.
- **Key Sources**: In **Key Sources**, you can generate keys in CCKM to upload to IBM. CCKM provides support for the use of CipherTrust Manager and DSM as key sources to generate, store, and retrieve the encryption keys that CCKM uses and manages.

  > Note: CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

- **Settings**: The **Settings** page allows you to configure settings for email alert, key alerts, and remote syslog server. From this page, you can also enable CCKM reporting for IBM keys using IBM Cloud Activity Tracker and LogDNA and view the release number of the CCKM running.

To display the context of which Key Management service you are currently logged into within CCKM, the icon associated with the Key Management service displays at the top of the left-hand navigation bar.

The arrow button at the bottom of the navigation bar allows you to expand the bar to view the names of the menu options and collapse it to view only the icons associated with the menu options. Click the arrow button to either expand or collapse the navigation bar.

On the top-right side of the CCKM portal for IBM, the following options are available:

- **Azure Cloud Service Context Switcher**: This context switcher allows you to login into another Azure cloud service different from the current Azure cloud service into which you are currently logged. If you select the Azure cloud in which you are currently logged, the Multi-account Login/Logout window for the Azure account(s) displays. For more information about the Multi-account Login/Logout window, refer to "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.
- **AWS Login**: If you are not already logged into AWS, clicking on the AWS Login icon brings you to the AWS Password Login page where you can enter your AWS credentials to access the CCKM AWS cloud service (from the current cloud service in which you are currently logged). If you are already logged into AWS, clicking on the AWS Login icon brings you to the Multi-account Login/Logout window for the AWS account(s). For more information about the Multi-account Login/Logout window, refer to "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.
- **Google Cloud Service Context Switcher:** This context switcher allows you to login into another Google Cloud organization different from the current Google Cloud organization into which you are currently logged. If you select the Google Cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Google account(s) displays. For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.

- **Salesforce Cloud Service Context Switcher**: This context switcher allows you to login into another Salesforce cloud service different from the current Salesforce cloud service into which you are currently logged. This is either Salesforce or Salesforce Sandbox. If you select the Salesforce cloud in which you are currently logged, the Multi-account Login/Logout window for the Salesforce account(s) displays. For more information about the Multi-account Login/Logout window, refer to "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.
- **Logout**: Logout from CCKM.

Keys

The IBM Keys page allows IBM cloud administrators to view all of the keys and manage them within the boundary of granted permissions. Only the keys to which you have permissions to access display in Keys page.

At the top right of the Keys page, the following buttons are available:

- **Export Keys**: Allows you to export the contents of the Keys list into a report in a CSV format.
- **New Key**: Allows you to upload an external key (BYOK) or create a native key. For information on how to upload an external key or create a native key, see section "Create a Key" on page 174.
- **Synchronize**: Allows you to download any keys that were created in IBM Key Protect, outside of the CCKM portal, into CCKM. For information on how to synchronize keys, see section "Synchronize Keys" on page 177.

In this section, create, delete, update, rotate, synchronize, exports keys are described.

A **Search** box is available at the top the **IBM Keys** page allowing for a search of a specific key by entering any one of the following parameters:

- Name: The name of the key.
- Key ID: The globally unique identifier for the key.
- Region: The IBM Region to which the key is uploaded.
- Key Protect Name: The name of the Key Protect instance.
- Creation Date: Indicates the date on which the key was created.
- Type: Indicates whether the key of a Standard or Root Key type.
- Key State: The state of the key. Valid values are:
  - Pre-activation
  - Active
  - Deactivated
  - Destroyed

  > **Note:** The key states of Pre-activation and Destroyed transition to other states. For this reason, you may not see keys with these states displayed in the CCKM UI.

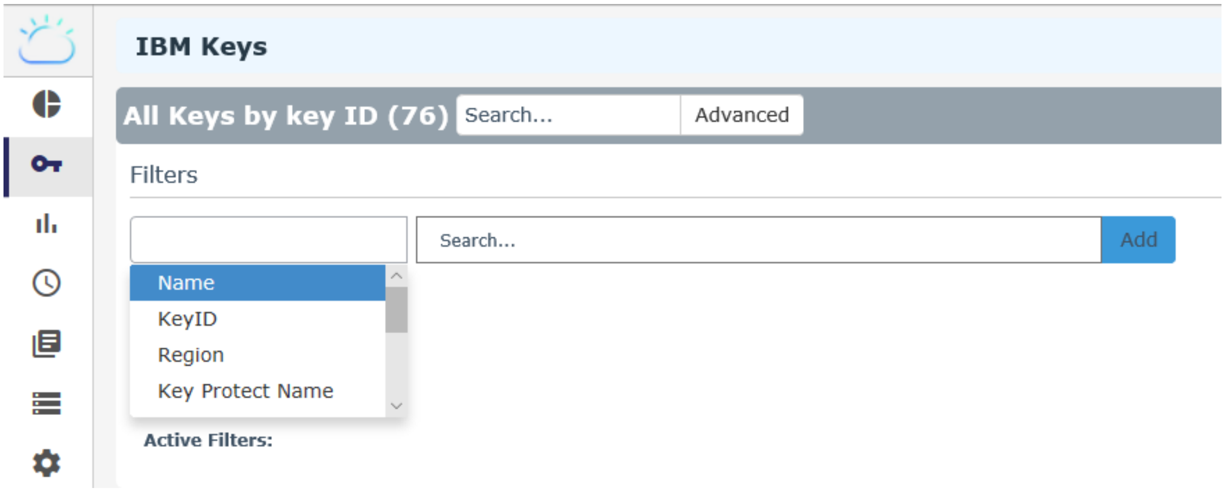
- Key Material Origin: Specifies the origin of the key material.
- Rotation Status: Available options are AUTO ON and AUTO OFF.

The Advanced tab allows you to filter your search on the each of the columns displayed on the IBM Keys page with the exception of the Action columns:

- Name
- Key ID
- Region
- Creation Date.
- Key Protect Name
- Key Protect ID
- Type
- Key State
- Key Expiration Date
- Key Material Origin
- Rotation Status

  Each of these columns are described in more detail below in this section.

  To select to filter based on a specific column from the **IBM Keys** page, click **Advanced**. The **Filters** section displays. From the column filter drop-down menu, select the name of the column on which to filter your search. For example, select **Region**. In the **Search** box, enter the value on which to search (for example, Key Protect Name) and click **Add**. For the **Enabled**, **Key State**, and **Rotation Status** columns, a drop-down menu is available from which to select a value for the search. Click the down arrow, select a value from the available options, and then click **Add**. For the **Creation Date** and **Key Expiration Date**, select a start and end date for the search and then click **Add**. The specified search filter is then added to **Active Filters**. The results of the search displays in the **IBM Keys** table. You can narrow your search by selecting more columns to add to the filter. To clear the filtered results, from **Active Filters**, click on the “X” (or the **Delete** button) for each of the named filters.



The following columns display on the **IBM Keys** page:

- Name: The name of the key.
- Region: The IBM Region to which the key is uploaded.
- Key Protect ID: The ID of the Key Protect instance in which the key stored.
- Creation Date: Indicates the date on which the key was created.
- Type: Indicates whether the key is a Root or a Standard key.

- Key State: The state of the key. The valid values are:
  - Pre-activation
  - Active
  - Deactivated
  - Destroyed
- Key Material Origin:
  - If the key was uploaded from CCKM as a BYOK key, then the value is INTERNAL(<source_key_name>).
  - If the key was not uploaded from CCKM as a BYOK key, then the value is EXTERNAL.
  - If the key material was created within IBM Key Protect, then the value is Key Protect.
- Rotation Status: Indicates whether the key is in an active rotation schedule or if it has been rotated in the past.
  - Auto—if the autorotation is currently enabled.
  - blank—if the autorotation is disabled.
- Actions: Allows you to select any of the following actions on a given key:
  - Rotate (Writer permission required)
  - Delete (Manager permission required)
  - Update (Manager permission required)
- Version Count: Reflects the number of versions the key currently has. To view information about all of the key versions, click on the number of the version count. The **Key Versions** dialog box displays. In this dialog box, the key name, version ID, and date on which the key version was created (Created On) displays. For more information, see "View All Key Versions" on page 178.

If you have the manager role, you have the permissions of writer and viewer. Writer has permissions of writer and viewer.

Create a Key

You have two options to create a new key in an IBM Key Protect instance:

- Upload a BYOK
- Create a native key (in IBM Key Protect)

Create a Native Key

Create a native key in IBM Key Protect using CCKM. In the process of creating a native key in IBM Key Protect, key material from IBM Key Protect along with key metadata from CCKM are used. Once created, the key displays as a Key Protect key under the Key Material Origin column within the Keys > All Keys list page in CCKM.

> **Note**
> The access permission of a Writer or a Manager role is required for creating native keys.

To create a native key in an IBM Key Protect instance:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, select the **New Key** button and then **Create**. The **Create Native Key** dialog box displays.
3. In the **Key Name** box, enter the name of the IBM native key to create.

4. From the **Key Protect** box, select the IBM Key Protect instance to which you will add the new key.
5. For **Key Type**, select whether this key is a **Standard Key** or a **Root Key**.
6. (Optional) Enter the expiration date of the key material in the **Key Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
7. (Optional) In the **Description** box, enter a description for the key you are uploading.
8. (Optional) In the **Tags** box, enter the tag name and then click the **+** button.
9. Click **Create**.
(Reset clears the page before it is created, if necessary.)
The new key is displayed in the list.

Upload Your Own Key (BYOK)

Upload the AES256 keys created from the **Key Sources** page of CCKM to an IBM Key Protect instance. These keys are your BYOK keys. The supported key sources for BYOK keys are CipherTrust Manager and DSM.

> **Note**
> The access permission of a Writer or Manager role is required for uploading a BYOK key.

To upload a BYOK key from CCKM to an IBM Key Protect instance:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **All Keys** list page, select the **New** button and then **Upload**. The **Upload a Key** dialog box displays.
3. In the **Key Name** box, enter the name of the IBM key to upload.
4. From the **Key Protect** box, select the Key Protect instance to which you will upload the new key.
5. For **Key Type**, select whether this key is a **Standard Key** or a **Root Key.**
6. From **Key Provider** box, select**CipherTrust Manager** or **DSM** depending on the key source you are using for the key.
7. From **Source Key** box, select the source key to use for the BYOK key. The keys available in the drop-down menu are the keys generated from CipherTrust Manager or DSM depending on the key source you are using.
8. (Optional) Enter the expiration date of the key material in the **Key Expiration Date** box in the format MM/DD/YY or select a day from the calendar button at the right of the box.
9. (Optional) In the **Description** box, enter a description for the key you are uploading.
10. (Optional) In the **Tags** box, enter the tag name and then click the **+** button.
11. Click **Upload to IBM**.
The uploaded key is displayed in the list.

Delete a Key

By default, IBM Key Protect blocks the deletion of a key that is protecting a cloud resource, such as a Cloud Object Storage bucket. To delete a key from IBM Key Protect, go to the IBM cloud console and delete the key from IBM Key Protect directly. You must have the access permissions of a Manager role to delete keys from IBM Key Protect.

> **Note**
> Currently, IBM does not provide the functionality to restore keys in IBM Key Protect that have been deleted. By default, the **Delete Key** function in the **Keys** page is disabled due to the absence of the restore functionality within IBM. This default behavior is implemented to safeguard your encrypted data. Without the ability to restore keys, there is the risk that you may delete ACTIVE keys that are still in use within the cloud. Without these keys, you cannot decrypt data that use these keys. As a result, the data is lost.

Update Key (or Enable Automatic Key Rotation Policy for Root Key)

In CCKM, you can enable an automatic rotation policy for a root key using IBM Key Protect. IBM Key Protect manages and performs the key rotation. After the key rotation takes place in an IBM Key Protect instance, you can manually synchronize your IBM keys in CCKM. You can also schedule key synchronization in CCKM to synchronize keys that have been rotated in IBM Key Protect. See section "Synchronize Keys" on the next page for more information.

> **Note**
> The access permission of a Manager role is required to enable an automatic rotation policy for a Root key.

To enable automatic key rotation policy for a root key:

1. On the **Keys** list page, click **Update** in the **Action** column for a particular key. The **Update key** dialog box displays.
2. From the **Update Key** dialog box, click **Enable autorotation in key policy** toggle to turn it to **On**. By default, this toggle is initially set to **Off**.
3. In the **Specify the key rotation time interval in months** box, enter the number of months for the key rotation interval. The minimum is 1, and the maximum is 12.
4. Click **Update**.

Rotate Key

CCKM allows for a manual rotation and a scheduled rotation (autorotation) of IBM Root keys.

> **Note**
> IBM does not support the rotation of IBM standard keys.

When you manually rotate a key in CCKM, use the **Rotate Now** tab in the **Rotate IBM Key** dialog box. Upon receiving the CCKM API call to rotate a specific key, IBM rotates the key. During the rotation of a key (manual or autorotation), new key material from the key source is used to replace the existing key material in the key while the key name and key ID remains the same. A new key version is created after the rotation. The supported key sources are CipherTrust Manager, DSM, and Native. If the key source is Native, then the key material originates from IBM Key Protect.

Auto rotating a key is simply rotating a key at a scheduled time. Key autorotation requires that you first schedule the autorotation in the **Schedules** page prior to enabling this feature in the **Auto Rotate** tab in the **Rotate IBM Key** dialog box. For information about scheduling autorotation, see "Schedules" on page 180.

Rotate a Key Manually

> **Note**
> The access permissions of a Writer or Manager role are required to manually rotate a key.

To rotate a key manually:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. From the **All Keys** list, find the row that holds the key you wish to rotate and select Rotate Key from the Actions column. The **Rotate** IBM **Key** dialog box is displayed.
3. Select the **Rotate Now** tab.
4. In **Key Provider** box, the key source that is preselected depends on the source (provider) of the key. The supported key sources are **CipherTrust Manager, DSM**, and **Native**.
5. Click **Rotate**.

Enable Autorotation of Key

> **Note**
> The access permissions of a Writer or Manager role are required to enable autorotation of a key.

To enable autorotation of a key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. From the **All Keys** list, find the row that holds the key you wish to rotate and select Rotate Key from the Actions column. The **Rotate** IBM **Key** dialog box is displayed.
3. Select the **Auto Rotate** tab.
4. From **Enable autorotation on this key?**, select **On** to enable the autorotation of this key. By default, this toggle is initially set to **Off**.
5. From the **Schedule** drop-down menu, select a rotation schedule to which to apply to this key. Any key-rotation schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
6. From **Key Provider** box, select **CipherTrust Manager, DSM**, or **Native** depending on the key source (provider) you are using for the key.
7. Click **Submit**.

Synchronize Keys

After the first login into the CCKM IBM cloud, all of your keys (Standard and Root) from all Key Protect instances to which you have access are synchronized with CCKM. IBM key metadata will be downloaded to CCKM during synchronization and displayed in the CCKM UI.

> **Note**
> If keys are auto-rotated in IBM Key Protect, be sure to synchronize your keys in CCKM.
>
> The access permissions of Viewer, Writer or Manager role are required to synchronize keys.

To synchronize your keys:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. Click **Synchronize**. The **Synchronize Key** dialog box displays.
3. Click **Synchronize** to proceed with the synchronization.

Export Keys

Export Keys allows you to export the contents of the **Keys** list into a report in a CSV format.

> **Note**
> No role access permissions are required to export keys.

To export your **Keys** list:

1. In the left navigation bar, click **Keys**.
2. Click **Export Keys**.
   This will download all of the keys in a report in a CSV format.

View All Key Versions

From the **Keys** page, you can view information about all of the key versions of a given key by accessing the **Key Versions** dialog box. A new key version is created after a key rotation (manual or autorotation). The **Key Versions** dialog box displays all the versions of a key including the key name, version ID (**Version**), and date on which it was created (**Created On**). You can display the next 5, 10, 15, 20, 25, 50, or 100 entries using the **Show entries** drop down menu. You can view each available page by scrolling through the pages by clicking the less than "**<**" or greater than "**>**" symbols or by clicking on a given page number.

To view all of the key versions of a key:

1. In the CCKM portal, in the left navigation bar, select **Keys**.
2. On the **Keys** page, find the name of the key whose key versions you wish to view.
3. Under the **Version Count** column, click on the number. The **Key Versions** dialog box displays. You can scroll through the list to view all of the key versions.
4. Click the **Close** button to close the **Key Versions** dialog box.

Reports

To run the **Combined Key Activity Reconciliation** and **IBM Key Activity** reports, you are required to provision an instance of IBM Activity Tracker with LogDNA with the Key Protect instances that are to be tracked before IBM keys will be represented in the CCKM reports. You also must also set these services up in CCKM using the **General** tab in the **Settings** page.

Choose Reports in the left-hand navigation to access the following reports:

- **Cloud Key Manager User Action Report**

  Lists the users and their actions in creating, deleting, updating and rotating keys in the CCKM portal.

- **Combined Key Activity Reconciliation Report**

  Click Run Report to generate a new list of events connected to a key. Reconciliation before and after key import the cloud service. The following actions can be re conciliated: creation, deletion, rotate, and policy update.

- **IBM Key Activity Report**

  Illustrates the activity type performed by a specific user at the stated time.

- **IBM Key Aging Report**

  Provides information about key expiration using the following columns:

  - Key Name
  - Region

  - Key expiration: Date and time at which this key expires.

- **Cloud Key Manager User Action Report**

  Lists the users and their actions in creating, deleting, updating and rotating keys in the CCKM portal.

IBM Cloud Activity Tracker with LogDNA Service

IBM Cloud Activity Tracker in conjunction with LogDNA allows you to do the following:

- Collect and store logs of API calls made to IBM cloud resources within your IBM cloud account.
- Monitor all of the API activities within your IBM cloud account.

CCKM uses these two applications to generate the Combined Key Activity Reconciliation Report and the IBM Key Activity Report. Prior to generating these two reports in CCKM, you must set up and configure IBM Cloud Activity Tracker and LogDNA. For more information about this setup and configuration, refer to section "Provisioning IBM Cloud Activity Tracker with LogDNA Service" below.

You must also configure the settings of the IBM Cloud Activity Tracker and LogDNA in the Settings page of CCKM so that communication between CCKM and IBM Cloud Activity Tracker with LogDNA can take place and enable reporting on the IBM keys. For more information about this configuration, refer to section "Enabling Reporting for IBM Key Activities Using IBM Cloud Activity Tracker and LogDNA" on page 1.

> **Note**
> Only the activities associated with the keys that are managed from CCKM are reflected in these reports. To ensure the Combined Key Activity Reconciliation Report and the IBM Key Activity Report, synchronize your keys with IBM cloud prior to running these reports. Otherwise, any activities that were performed from the IBM Key Protect portal (if any) will not be reflected in these reports. If you are using only CCKM to manage your keys for the IBM Key Protect instances then this note does not apply to you.

Provisioning IBM Cloud Activity Tracker with LogDNA Service

To set up and configure IBM Cloud Activity Tracker and LogDNA, refer to the "Getting Started Tutorial" under the topic of “IBM Cloud Activity Tracker with LogDNA” within *IBM Cloud Docs*.

The prerequisites to setting up and configuring IBM Cloud Activity Tracker and LogDNA are the following:

- Obtain a user ID (or username) that is a member or an owner of an IBM Cloud account.
- To manage access to the service, your user ID needs administrator platform permissions to manage the IBM Cloud Activity Tracker with LogDNA service.

The following are the high-level steps to take to set up and configure IBM Cloud Activity Tracker and LogDNA described in detail within the "Getting Started Tutorial":

1. Provision an instance of the IBM Cloud Activity Tracker with LogDNA service. (Note the name for the LogDNA instance, IBM cloud region of the LogDNA instance, and LogDNA Service Key for the instance. You will add this information in the Settings page of CCKM.)
2. Manage access to the monitoring service.
3. Generate IBM Cloud Activity Tracker with LogDNA events.
4. Launch the Web UI for the IBM Cloud Activity Tracker with LogDNA.
5. View events in the Web UI.

Schedules

The **Schedules** page allows cloud administrators to view and schedule the following job types:

- Key Rotation
- Key Synchronization

CCKM provides support for multiple schedules for each of the job types.

At the top right of the **Schedules** page, the Add Schedule button is available from which to add a scheduled job type.

Schedule Key Rotation

CCKM provides support for multiple schedules of the rotation (autorotation) of an IBM key. A new version of the key is added to the IBM key during a scheduled rotation. Scheduled rotation requires that this feature be configured in the **Schedules** page prior to enabling the feature in the **Auto Rotate Key** dialog box (from the **Actions** column of the All Keys list). Your IBM key credentials (IBM username and API key) are required as part of the configuration and are saved to the key source you are using (CipherTrust Manager or DSM).

> **Note**
> CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

During a scheduled rotation of a key, CCKM uses your associated IBM key credentials to rotate the key. If the IBM API key expired or changed, then a new IBM API key must be reentered. Otherwise, the scheduled rotation will fail.

You can pause or delete a scheduled key rotation in the **Schedules** page. Note that when a key rotation schedule is deleted from CCKM, then the IBM key credentials are also deleted from the key source you are using (CipherTrust Manager or DSM)

Delete a Key Rotation Schedule

> **Note**
> No role access permissions are required to delete a key rotation schedule.

To delete a key rotation schedule, select Delete from the Actions drop-down menu from the Schedules page. Note that the IBM username and IBM Cloud API key associated with the given key rotation schedule is also deleted from the key source you are using (CipherTrust Manager or DSM).

Pause a Key Rotation Schedule

> **Note**
> No role access permissions are required to pause a key rotation schedule.

To pause a key rotation schedule, select Yes for Pause within the Add Schedule dialog box and then click **Save**. The setting is saved. To resume the rotation schedule after pausing it, select No for Pause and then click **Save**. The setting is saved.

Schedule Key Synchronization

schedules of the synchronization Synchronizing downloads any keys that were created in IBM Key Protect, outside of the CCKM portal, into your CCKM. Scheduled key synchronization requires that this feature be configured in the **Schedules** page. You can also pause or delete a scheduled key synchronization in the **Schedules** page. Note that when a key synchronization schedule is deleted from CCKM, then the IBM username and IBM Cloud API key associated with the given key scheduled synchronization are also deleted from the key source you are using (CipherTrust Manager or DSM).

> **Note**
> CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

Set up a Key Rotation Schedule

To set up a key rotation schedule:

> **Note**
> No role access permissions are required to set up a key rotation schedule.

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_ROTATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or Advanced (to enter a Cron Expression for the schedule).
8. If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.

   Set the repeat interval of the key rotation by entering a number in the Repeat Interval and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.
9. If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
10. In the **Username** box, enter your username for the IBM account.
11. In the **IBM API key** box, enter your IBM API key for the IBM account.
12. Click **Save** to save setting.

Set up a Key Synchronization Schedule

To set up a key synchronization schedule:

> **Note**
> The access permissions of Viewer, Writer or Manager role required to set up a key synchronization schedule. However, it is possible to setup synchronization schedule with none of these permissions on any keys. In this case, synchronization will run on schedule but no key will be downloaded.

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_SYNCHRONIZATION**.

4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule).
8. If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.

   Set the repeat interval of the key rotation by entering a number in the Repeat Interval and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.
9. In the **Username** box, enter your username for the IBM account.
10. In the **IBM API key** box, enter your IBM API key for the IBM account.
11. Click **Save** to save setting.

Logs

CCKM maintains logs for events, such as login, key generation, synchronizing of keys, and deletion of keys, and displays these logs in the **All Logs** list of the **Logs** page. For each logged event listed on the **Logs** page, the following information is displayed:

- Event name
- Severity level
- Date and time of the event
- Event message
- User (name) associated with the event

You can search for a specific logged event from the **Logs** page using the **Search** box. Allowable filters on the search are event name, severity level, event message, and user (name).

Key Sources

The **Key Sources** page displays all of the keys that are generated from the key source you are using. Three tabs are available at the top of the page:

- **CipherTrust Manager**: Use this tab to use CipherTrust Manager as your key source. The keys are generated and stored within CipherTrust Manager.
- **DSM Key**: Use this tab to use DSM as your key source. The keys are generated and stored within DSM.
- **nShield Key**: This tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for IBM.

Keys are listed by name, key type (the cloud in which this key is used), the algorithm of the key, the creation date of the key, key description, and actions you can take regarding that key. Note that the key algorithm currently supported for an IBM key in CCKM is AES256.

A **Search** box is available at the top left of the **Keys Sources** page allowing for a search of a specific key by entering any one of the following parameters:

- Name (key name)
- Key Type
- Algorithm
- Description
- Created By (a key created by a specific user)

Add (Generate) a Key

From the **Key Sources** page, you can add (or generate) keys in CCKM using CipherTrust Manager or DSM as a key source. These keys are of the AES256 algorithm type.

Add (Generate) a DSM Key

1. In the CCKM portal, in the left navigation bar, select Key Sources.
2. Select the **DSM Key** tab.
3. Select **Add Key** > **IBM Key**.
4. In the **Add DSM Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to IBM and (key) **Algorithm** is fixed to AES256.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. Click **Save** to save the new key to CCKM. The new key is added to the key list.
   (Reset clears the screen before it has been saved.)

Add (Generate) a CipherTrust Manager Key

1. In the CCKM portal, in the left navigation bar, select Key Sources.
2. Select the **CipherTrust Manager Key** tab.
3. Select **Add Key** > **IBM Key**.
4. In the **Add CipherTrust Manager Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to **IBM** and (key) **Algorithm** is fixed to AES256.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. Click **Save** to save the new key to CCKM. The new key is added to the key list.
   (Reset clears the screen before it has been saved.)

Add (Generate) an nShield Key

The **nShield Key** tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for IBM Key Protect.

Delete a Key

Deleting a key from the **Key Sources** page deletes the key from CCKM. If you delete a key from **Key Sources** that has already been uploaded to IBM Key Protect, the key will not be deleted from IBM Key Protect.

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **DSM Key** or **CipherTrust Manager** tab depending on the key source you used to generate this key.
3. Click **Delete** next to the target key.
4. The **Delete Key** dialog box with a Warning displays asking you to confirm the key deletion by entering the supplied phrase.
5. Enter or Copy/paste the confirmation phrase.
6. Click **Delete** to confirm.

Manage keys in Key Sources List Page

In addition to adding and deleting keys, the **Key Sources** page allows you to sort the existing keys by Name, Key Type, Algorithm and Created Date.

Settings

To access the **Settings** page in CCKM, select **Settings** from the left-hand navigation bar. From the **Settings** page, you can:

- Enter the emails of the recipients who are to receive alerts and reminders.
- Enable reporting for IBM keys using IBM Cloud Activity Tracker in conjunction with LogDNA.
- Set up key alerts.
- Configure a remote syslog server to which to send syslog messages.
- View information about the software release number of the CCKM you are running including the build and version number.

Adding Recipients’ Emails (for Alerts and Reminders)

Use the **General** settings tab to enter the emails of the recipients who are to receive alerts and reminders from the system. Enter the recipients’ email addresses separated by a comma in the **Alert Format** box and then click **Save** to save setting.

Enabling Alerts

Use the **Keys Alerts** tab from the **Settings** page to specify whether to send an alert to the specified user email(s) when an IBM key is uploaded to IBM. The user emails are set up in the General tab. By default, no alerts or reminders are sent. From the **Keys Alerts** tab, you can also enable alerts to receive notifications when a key is uploaded to IBM Key Protect from CCKM.

To set up key alerts or a reminder to manually rotate (or rekey) a key:

1. In the **Settings** page, select the Keys Alert tab.
2. Under **Alerts**, click **On** for each of the alert types and reminder you wish to enable:

- Rekey Alert (reminder)
- Upload IBM Key

3. If you set a reminder for **Rekey Alert**, then in the **Days** box, enter the number of days for the frequency an email reminder to manually rotate a key is sent.

4. Click **Save** to save the setting.

Configuring Syslog Server

Use the **Syslog** tab from the **Settings** page to configure a remote syslog server to which to send syslog messages. Note that the default port number of 514 and the facility name of “LOCAL1” for the syslog server display in the **Port** and **Facility** boxes, respectively. These boxes are not available for modification.

Configure Remote Syslog Server

To configure a remote syslog server:

1. In the **Settings** page, select the Syslog tab. The **Syslog** dialog box displays.

2. Enter the hostname or IP address of the syslog server in the **Hostname** box.

3. (Optional) Enter a description of the syslog server in the **Description** box.

4. Click **Save** to save the setting.

Delete Remote Syslog Server Configuration

To delete the configuration of a remote syslog server, click Delete from the Syslog dialog box. No syslog messages are sent to the remote syslog server after the deletion of this configuration.

Enabling Reporting for IBM Keys

Use the **General** tab within the Settings page to enable CCKM reporting for IBM keys using IBM Cloud Activity Tracker and LogDNA. However, you are required to provision an instance of IBM Activity Tracker with LogDNA with the Key Protect instances that are to be tracked before IBM keys will be represented in the CCKM reports. Enabling reporting for IBM keys allows you to generate the Combined Key Activity Reconciliation and IBM Key Activity reports. See section "Provisioning IBM Cloud Activity Tracker with LogDNA Service" on page 179 for more information about provisioning an instance of IBM Activity Tracker with LogDNA.

Enabling Reporting for IBM Key Activities Using IBM Cloud Activity Tracker and LogDNA

To enable CCKM reporting for IBM key activities using IBM Cloud Activity Tracker and LogDNA:

1. In the **Settings** page, select the **General** tab. The **General** page displays.

2. From the **LogDNA Name** box, provide a name for the LogDNA instance.

3. From the **Select LogDNA Region** drop-down menu, select the IBM cloud region of the LogDNA instance (that you previously created) for which to enable the reporting.

4. From the **LogDNA Service Key** box, enter the LogDNA Service Key that is used to authenticate to the LogDNA instance.

5. Click **Add LogDNA Instance**.

6. Click **Save** to save setting.

About

Use the About tab to view the build number, version number, patch number (if applicable), and build number of the CCKM you are running.

Chapter 17: CCKM for Salesforce Key Management

Overview

CCKM provides support for Salesforce and Salesforce Sandbox. The Salesforce Sandbox is a copy of your production Salesforce environment used primarily for testing, development, or training purposes. Note that in this chapter, the information regarding Salesforce is applicable to both Salesforce and Salesforce Sandbox unless otherwise noted. The CCKM UI for Salesforce and Salesforce Sandbox is identical with the exception of the display of the names of the clouds. Your login credentials for Salesforce and Salesforce Sandbox are unique to each cloud. If you are configuring for Salesforce Sandbox, be sure to configure within the Salesforce Sandbox cloud.

Prerequisites on Salesforce

To use CCKM for Salesforce, you must:

- Have Salesforce's Shield Platform Encryption Service enabled
- Have at least one valid certificate in Salesforce with the following settings:
  - Key Size: 4096
  - **"Use Platform Encryption"** selected
  - **"Exportable Private Key"** deselected
- **Be familiar with different types of tenant secrets, Salesforce 'Bring Your Own Key' options, and certificates** in Salesforce.
- **Have access to a Salesforce account with "Manage Encryption Keys" permission enabled**, in order to handle tenant secrets. If you wish to disable Salesforce's key derivation process, and use your uploaded key material as the final data encryption key, then enable the "**Allow BYOK to Opt Out of Key Derivation**" option within the **Security** > **Platform Encryption** > **Advanced Settings** page in Salesforce. If you are using Salesforce cache-only keys, ensure your account also has **"Customize Application"** permission enabled. Also ensure the **"Allow Cache-Only Keys with BYOK"** option within the **Security > Platform Encryption > Advanced Settings** page in Salesforce is also enabled. If you wish to enable the Replay Detection for cache-only keys, ensure the "**Enable Replay Detection for Cache-Only Keys**" option within the **Security** > **Platform Encryption** > **Advanced Settings** page in Salesforce is also enabled.

Enable the permissions of “Manage Encryption Keys” and “Customize Application” (optional) in Salesforce

You must have the **“Manage Encryption Keys”** permission enabled to perform operations related to tenant secrets. If you are using Salesforce cache-only keys, then you must also have the **“Customize Application”** permission enabled to perform operations related to cache-only keys.

1. Log in with the administrator privileges to your Salesforce account.
2. Click **Setup**.
3. In the left navigation bar, expand **Manage Users**.
4. Click **Permission Sets**.

   Be sure that **Session Activation Required** is not selected.
5. Click **New**, and then select **New Permission Set**.
6. Click **System Permissions**, then click **Edit**.
7. Check **Manage Encryption Keys**. If you are using Salesforce cache-only keys, then also check **Customize Application**. Click **Save**.
8. Select **Manage Assignments > Add Assignments** and select the user account(s) which you want to enable to import and export keys.

For more information about managing permissions on encryption keys in Salesforce, see the “Permission Sets” documentation within the Salesforce Help documentation.

Salesforce Cache-only Keys

CCKM provides support for Salesforce’s cache-only key service as part of its extended BYOK capability. With the use of a cache-only key, Salesforce calls the CCKM REST APIs to fetch a tenant secret (or cache-only key) from CCKM. This tenant secret is used for encrypt and decrypt operations in Salesforce. With the initial versions of Salesforce BYOK capability, the upload of the tenant secret and other key operations in Salesforce are initiated from CCKM using the available Salesforce Rest APIs.

In the API call from Salesforce to CCKM to fetch a tenant secret for the use of Salesforce’s cache-only key service, a named credential is employed. The named credential is a combination of a URL of a CCKM callout endpoint along with the configured authentication parameters (CCKM username and password) associated with the URL. When CCKM receives the API call and successfully authenticates it, CCKM passes the tenant secret to Salesforce through a secure and authenticated connection. For more information about Salesforce cache-only keys, refer to the Salesforce Help documentation.

CCKM also provides support for Replay Detection for cache-only keys. With this feature enabled, Salesforce inserts a unique marker called a RequestIdentifier in each callout to get the cache-only key from CCKM. The request ID serves as a random, one-time identifier for each valid callout request. Once this feature is enabled from the Salesforce portal, CCKM will accept and return the request IDs. No configuration is required on CCKM. Any request without Request ID or mismatched Request ID is aborted.

> **Notes**
>
> - CCKM currently supports CipherTrust Manager and DSM as the key sources for use with Salesforce’s cache-only key service.
> - In this documentation and in the context of support for Salesforce’s cache-only key service, the term “tenant secret” is used interchangeably with “cache-only key”.

Configuring Cache-only Keys

The following is a summary of the steps to take to configure the use of cache-only keys within CCKM *and* Salesforce:

1. Configure CCKM with a SSL certificate that is signed by a Salesforce root Certificate Authority (CA) certificate. For more information, see "Configure CCKM with SSL Certificate Signed by Salesforce Root CA Certificate" below.
2. From the CCKM **Settings** page, configure the username and password to associate with the URL of the CCKM Rest API endpoint (or callout endpoint). For more information, see "Configure Cache-only Key Endpoint" on page 208.
3. Set up a named credential in Salesforce for use with the cache-only keys. For more information, see "Set up Named Credential in Salesforce" below.
4. Generate a key in DSM for use with the cache-only keys. For more information, see "Add (Generate) a Key" on page 206.
5. Configure a cache-only key in CCKM. For more information, see "Create a Tenant Secret/Key" on page 195.
6. Verify that Salesforce can fetch the key from CCKM using the call-out connection. For more information, see "Verify Call-Out Connection for Cache-only Key " on page 197.

Configure CCKM with SSL Certificate Signed by Salesforce Root CA Certificate

As part of cache-only key fetch process, Salesforce calls CCKM API over the HTTPS protocol, which requires CCKM to be configured with a valid SSL certificate. Salesforce cache-only key feature does not support self-signed certificates. CCKM must be configured with a SSL certificate that is signed by a Salesforce root Certificate Authority (CA) certificate. You can find the Salesforce root CA certificate list at https://INSTANCE.salesforce.com/cacerts.jsp where 'INSTANCE' is any upgraded instance. For example, https://cs32.salesforce.com/cacerts.jsp. You can use any one of these certificates from the list to sign the CCKM SSL certificate.

Set up Named Credential in Salesforce

Before setting up a named credential in Salesforce, ensure you have a CCKM deployed on a publicly accessible URL. In addition, ensure you have configured the username and password to be associated with the URL of the CCKM Rest API endpoint (or callout endpoint) in the **Endpoint for Cache-only Key** tab from the CCKM **Settings** page. (See "Configure Cache-only Key Endpoint" on page 208 for information about the configuration of the callout endpoint.) The username and password are used to set up the named credential in Salesforce, which in turn provides Salesforce a method for fetching the cache-only key from CCKM through a secure and authenticated connection.

For detailed information about how to set up a named credential in Salesforce, see the Salesforce Help documentation.

To set up a named credential in Salesforce, do the following:

1. Login into Salesforce.
2. Go to the **Setup** page.
3. In **Quick find** box, enter “Named Credentials” to search on this topic and then select **Named Credentials** in the results. The **Named Credentials Setup** page displays.
4. Select **New Named Credentials**.

5. In the **Label** box, enter the name of the named credential to use for CCKM. It is recommended you include "CCKM" in the name to reflect CCKM-associated information in the labels within Salesforce list views and reports. The name you provide in the **Label** box is automatically populated in the **Name** box.

6. In the **URL** box, enter the URL of the CCKM Rest API endpoint. To obtain the information to enter in this box, go to the **Endpoint for Cache-only Key** tab of the CCKM **Settings** page, and copy the URL from the **Call out URL** box and paste it into this box. Verify the hostname and port number are correct. For example, if CCKM URL is https://thalescckm.com:8443/kmaas/ and the org ID is 00DB00007731234MAK, then the callout URL will be: https://thalescckm.com:8443/kmaas/00DB00007731234MAK/

7. Under **Authentication**, do the following:

   a. Leave the **Certificate** box blank.

   b. In the **Identity Type** drop-down menu, select **Named Principal**.

   > **Note:** Named Principal is currently the only option supported for the Salesforce identify type within CCKM.

   c. In the **Authentication protocol** drop-down menu, select **Password Authentication**.

   > **Note:** Password Authentication is currently the only option supported for authentication protocol within CCKM.

   d. In the **Username** box, enter the username to associate with the URL of the CCKM Rest API endpoint. This username must match the username configured in the **Endpoint for Cache-only Key** tab of the CCKM **Settings** page.

   e. In the **Password** box, enter the password to associate with the URL of the CCKM Rest API endpoint. This password must match the password configured in the **Endpoint for Cache-only Key** tab of the CCKM **Settings** page.

   f. Under the **Callout Options**, ensure the **Generate Authorization Header** check box is selected. This check box is selected by default.

   g. Click **Save**.

Access CCKM for Salesforce

For details about accessing CCKM as a CCKM user for the CCKM service for Salesforce Key Management, see Chapter 3: "About CCKM Admin and User Portals" on page 24 on page 1. This chapter includes information on the ways in which to access the CCKM service for Key Management.

Salesforce CCKM User Portal

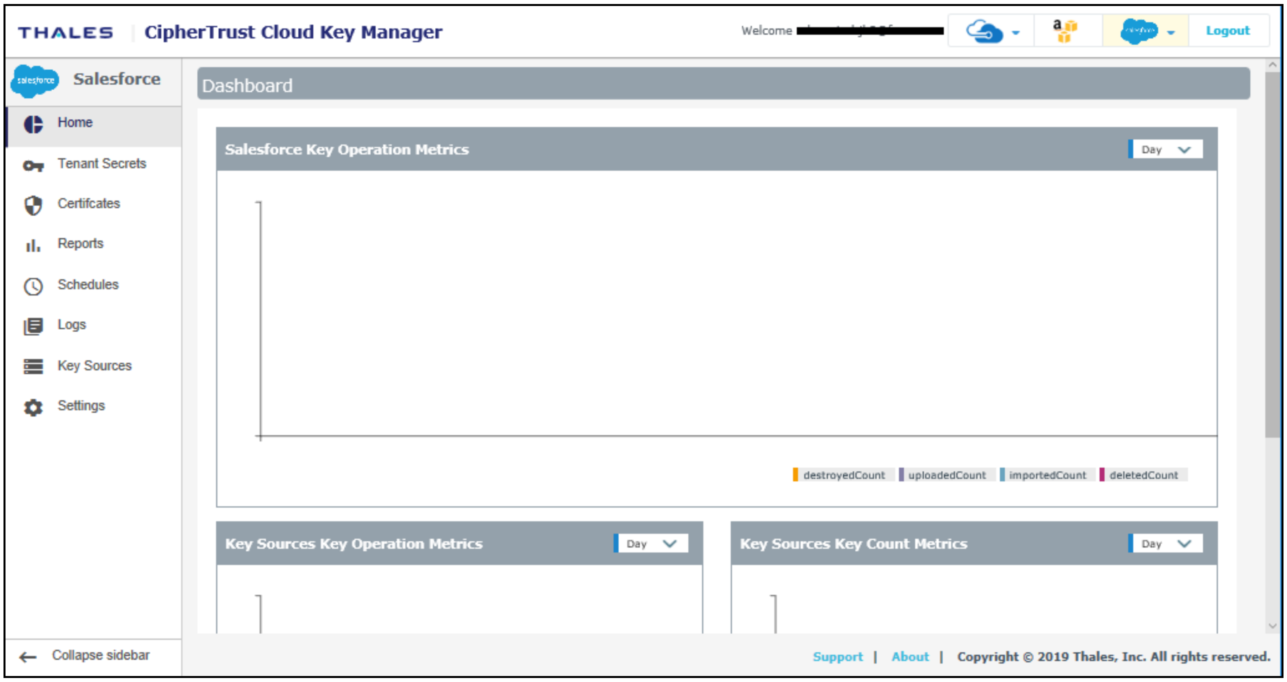
The CCKM User Portal for Salesforce contains the following menu options in the left-hand navigation bar:

- **Home:** Home page displays key operation and count metrics
- **Tenant Secrets:** Salesforce terminology for "keys". The **All Tenant Secrets** page displays key management overview and provides the Upload and Synchronize functionality, and the Create, Destroy, Import, Delete Backup key operations available in CCKM.
- **Certificates:** The **All Certificates** page displays certificate management overview, plus Synchronize functionality.
- **Reports:All Reports** overview page links to detailed individual reports.

- **Schedule:** In **Schedule**, you can schedule automatic key rotation, key rotation based on key expiration, and key synchronization.
- **Logs: Logs** page lists individual logs that can be searched, sorted, and viewed.
- **Key Sources:** In **Key Sources**, you can create tenant secrets in CCKM to upload to Salesforce. CCKM provides support for the use of either CipherTrust Manager or DSM as a key source to generate, store, and retrieve the encryption keys that CCKM uses and manages.

  > **Note:** CCKM does not support the concurrent use of CipherTrust Manager, DSM, and nShield Connect as key sources.

- **Settings:** The **Settings Management** page allows you to configure email alert settings, tenant secret alerts, remote syslog server, and proxy server settings. You can also view the release number of the CCKM running from this page.



To display the context of which Key Management service you are currently logged into within CCKM, the icon associated with the Key Management service displays at the top of the left-hand navigation bar.

The arrow button at the bottom of the navigation bar allows you to expand the bar to view the names of the menu options and collapse it to view only the icons associated with the menu options. Click the arrow button to either expand or collapse the navigation bar.

On the top-right side of the CCKM portal for Salesforce, the following options are available:

- **Azure Cloud Service Context Switcher:** This context switcher allows you to login into another Azure cloud service different from the current Azure cloud service into which you are currently logged. If you select the Azure cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Azure account(s) displays. For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 1.

- **AWS Login:** If you are not already logged into AWS, clicking on the **AWS Login** icon brings you to the **AWS Password Login** page where you can enter your AWS credentials to access the CCKM AWS cloud service (from the current cloud service in which you are currently logged). If you are already logged into AWS, clicking on the **AWS Login** icon brings you to the **Multi-account Login/Logout** window for the AWS account(s). For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.
- **Google Cloud Service Context Switcher:** This context switcher allows you to login into another Google Cloud organization different from the current Google Cloud organization into which you are currently logged. If you select the Google Cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Google account(s) displays. For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.
- **Salesforce Cloud Service Context Switcher:** This context switcher allows you to login into another Salesforce cloud service different from the current Salesforce cloud service into which you are currently logged. This is either Salesforce or Salesforce Sandbox. If you select the Salesforce cloud in which you are currently logged, the **Multi-account Login/Logout** window for the Salesforce account(s) displays. For more information about the **Multi-account Login/Logout** window, see "Support for Multiple accounts or Multiple Users per Cloud Service in CCKM UI" on page 29.
- **Logout:** Logout from CCKM.

Tenant Secrets

The **Tenant Secrets** page allows Cloud administrators to view all the keys and manage them within the boundary of granted permissions. Those keys or key vaults that do not have List permission for the respective Cloud administrator will not be listed.

Within the **Tenant Secrets** page, the following types of Salesforce tenant secrets are available to specify the kind of data to encrypt using a tenant secret:

- Data in Salesforce: Encrypts data using the probabilistic encryption scheme.
- Data in Salesforce (Deterministic): Encrypts data using the deterministic encryption scheme.
- Search Index: Encrypts search index files.
- Analytics: Encrypts Einstein Analytics data.
- Event Bus (Developer Preview): Encrypts data changes and the associated change event.

You specify the key type on the **Tenant Secrets** page when creating a new key using the **Create Tenant Secret** dialog box or uploading an existing key using the **Upload a Tenant Secret** dialog box. The existing key is one you previously created from the **Key Sources** page. For more information about the types of tenant secrets, see the appropriate Salesforce documentation.

The option to opt out of using the Salesforce key derivation mode when uploading tenant secrets from the **Tenants Secret** page is available in support of this Salesforce feature. The key derivation mode in CCKM allows you to derive a data encryption key based on the tenant secret you create or upload to Salesforce from the **Tenants Secret** page. Opting out of this mode means the tenant secret that you create or upload will be the key used to encrypt and decrypt your data.

From the **Tenant Secrets** page, a **Filter by** box is also available allowing you to filter on the types of tenant secrets to display in the list of tenant secrets.

A **Search** box is available at the top left of the **Tenant Secrets** page allowing for a search of a specific key by entering any one of the following parameters:

- Key ID
- Version Key ID
- Keyvault name
- Keyvault location
- Source key
- Subscription ID

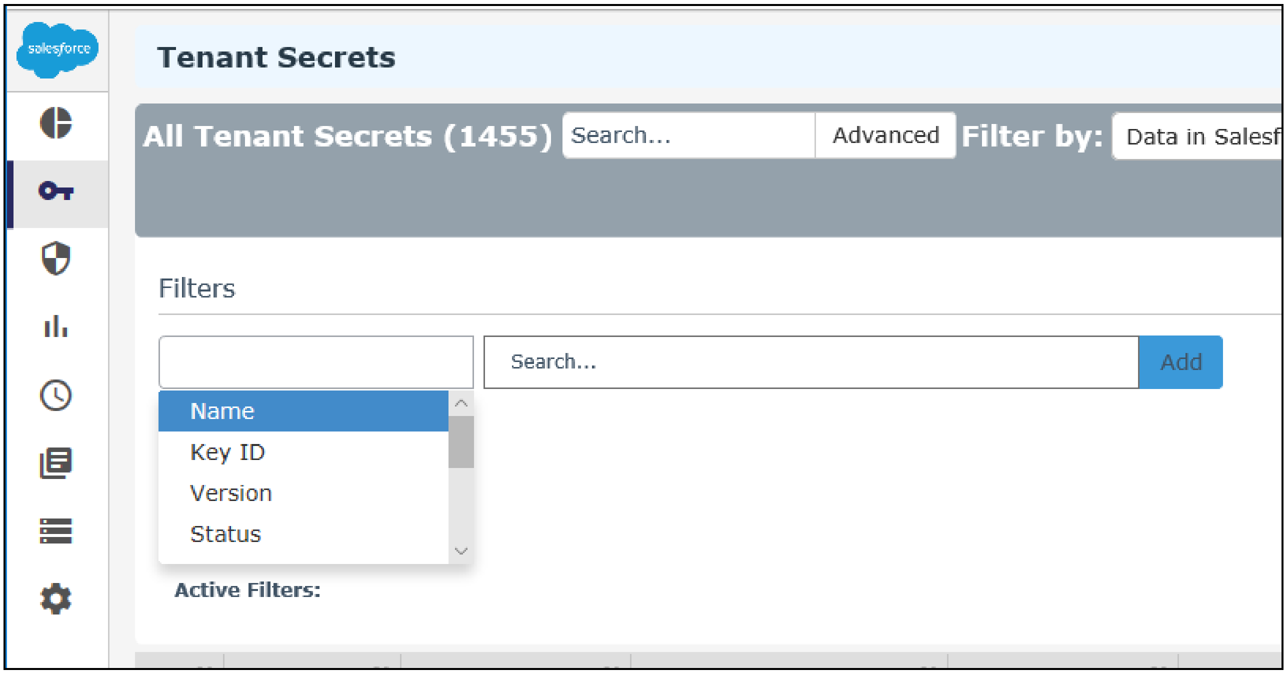
The **Advanced** tab allows you to filter your search on the each of the columns displayed on the **Tenant Secrets** page with the exception of the **Action** column:

- Key Name
- Version—Min and Max
- Status— Available options from the drop-down menu are ACTIVE, ARCHIVED, and DESTROYED.
- Generated By
- Generated On—Enter START DATE and END DATE.
- Key Derivation
- Backup—Available options from the drop-down menu are Backed Up and Not Backed UP.
- Uploaded—Available options from the drop-down menu are Uploaded and Not Uploaded
- Key Material Origin

In addition, you can also search on **Key ID**. A **Key ID** for a given key is only available in the **Tenant Secret** box. To view the **Key ID** of a key, from the **Tenant Secrets** page, select the name of the key listed under the **Name** column. The **Tenant Secret** box displays and includes the name of the key, the Salesforce Key ID, key version, description of the key, and other properties.

Each of these columns are described in more detail below in this section.

To select to filter based on a specific column or Key ID from the **Tenant Secrets** page, click **Advanced**. The **Filters** section displays. From the filter drop-down menu, select the name of the column on which to filter your search or Key ID. For example, select **Key Name**. For the **Status**, **Backup**, and **Uploaded** columns, a drop-down menu is available from which to select a value for the search. Click the down arrow, select a value from the available options, and then click **Add**. For the **Version** column, enter a minimum and maximum version number for the search and then click **Add**. For the **Generated On** column, enter a start and end date for the search and then click **Add**. The specified search filter is then added to **Active Filters**. In the example, **Key Name: MK-createkey** is added to **Active Filters**. The results of the search displays in the **Tenant Secrets** table. You can narrow your search by selecting more columns to add to the filter. To clear the filtered results, from **Active Filters**, click on the “X” (or the **Delete** button) for each of the named filters.



At the top right of the **Tenant Secrets** page, the following buttons are available:

- Export Secrets: Allows you to export the contents of the Tenants Secrets list into a report in a CSV format. For information on how to export your tenant secrets, see "Export Tenant Secrets" on page 201.
- New Key: Allows you to upload an external key (BYOK) or create a native key. For information on how to upload an external key or create a native key, see "Create a Tenant Secret/Key" on the facing page.
- Synchronize: Allows you to download any keys that were created in Salesforce, outside of the CCKM portal, into CCKM. For information on how to synchronize your keys, see "Synchronize Tenant Secrets" on page 200. The following columns display in the **Tenant Secrets** page:
  - Name: The name of the key.
  - Version: The version number of the key.
  - Status:
    - ACTIVE
    - DESTROYED
    - ARCHIVED
  - Generated By: The username associated with generating the key.
  - Generated On: The date on which the key was generated.
  - Key Derivation: Indicates whether the tenant secret is in key derivation mode. A green check icon indicates that it is in this mode.
  - Backup: Indicates whether a backup of the key exists in CCKM. The status can be one of the following:
    - A green check icon indicates the backup exists in CCKM.
    - A red exclamation icon indicates the backup key does not exist in CCKM.
  - Uploaded: Indicates whether the key is uploaded to Salesforce. A green check icon indicates that it is uploaded. Cache-only keys that are uploaded to Salesforce also display with the green check icon.

- Key Material Origin:
  - If the key was uploaded from CCKM, then the value is **Internal(<source_key_name>)**. For a cache-only key, the value is **FETCHED(<source_key_name>)**.
  - If the key was not uploaded from CCKM as a BYOK key, then the value is **EXTERNAL**.
- Actions: Allows you to select any of the following actions on a given key:
  - Destroy
    Note: For a cache-only key, **Destroy** is the only action available within the **Actions** column. After the key is destroyed, the **Actions** column displays as blank with no action that can be taken.
  - Edit key

    > **Note:** This action is only applicable to a cache-only key.

  - Import
  - Delete Backup

In this section, create, upload, destroy, import, synchronize, and manage tenant secrets, as well as delete a backup key are described.

Create a Tenant Secret/Key

You have three options to create a new tenant secret or key:

- Upload an external key (BYOK) (see "Upload your Own Tenant Secret" below)
- Create a native key (see "Create a Tenant Secret/Key" on the next page)
- Create cache-only key (see "Create a Cache-only Key" on the next page)

Upload your Own Tenant Secret

Salesforce supports only one active tenant secret per organization/OrgID. When you upload your own tenant secret, that key becomes active and the previously active key is automatically archived. Use Upload as part of your organization’s key rotation policy. You can only upload a tenant secret once every 24 hours, as per Salesforce’s policy.

The uploaded key is displayed in the Tenant Secrets list. As part of the upload process, Salesforce generates a corresponding backup key, which CCKM downloads and stores in the key source you are using (CipherTrust Manager or DSM). For more information about the backup key, see "Backup Key" on page 201.

> **Note**
> A valid certificate is required, as described in "Prerequisites on Salesforce" on page 187. If no certificate is listed in the certificate drop-down, you must go to Salesforce, create an appropriate certificate, and re-synchronize with CCKM.

To upload your BYOK tenant secret to CCKM:

1. In the CCKM portal, in the left navigation bar, select **Tenant Secrets**. The **Tenant Secrets** list page displays.
2. From the **Tenant Secrets** list page, within the **New Key** box, select **Upload**. The **Upload a Tenant Secret** dialog box displays.

3. From the **Type** box, select the type of tenant secret to upload. The following are the options (in the order of selection):
   - Data in Salesforce
   - Search Index
   - Data in Salesforce (Deterministic)
   - Analytics
   - Event Bus
4. From the **Tenant Secret** drop-down menu, select the tenant secret to upload. Enter the first three characters of the name of the tenant secret to bring it up for selection.
5. Select whether to apply the Salesforce key derivation mode to the tenant secret. If you choose to apply it, ensure that the **Use Salesforce key derivation** check box is selected. By default, the check box is selected. Otherwise, deselect the check box to opt out of applying the key derivation mode to the tenant secret.
6. From the **Certificate** drop-down menu, select a valid certificate (used to wrap the key for secure upload).
7. Click **Upload to Salesforce** to upload the tenant secret to Salesforce.

Create a Tenant Secret/Key

Create a native key in Salesforce using CCKM. After CCKM creates a native key, it displays the key as **EXTERNAL** under the **Key Material Origin** column within the **Tenant Secrets > All Tenant Secrets** list page in CCKM.

As part of creating a native key, Salesforce generates a corresponding backup key, which CCKM downloads and stores in the key source you are using (CipherTrust Manager or DSM).

To create a native tenant secret:

1. From the **Tenant Secrets** list page, within **New Key** box, select **Create**.
2. From the **Type** box of the **Create a Tenant Secret** dialog box, select the type of tenant secret to create.
3. Select whether to apply the Salesforce key derivation mode to the tenant secret. If you choose to apply it, ensure that the **Salesforce key derivation** check box is selected. By default, the check box is selected. Otherwise, deselect the check box to opt out of applying the key derivation mode to the tenant secret.
4. Click **Create** to create the new tenant secret.

Create a Cache-only Key

In CCKM, create a tenant secret or cache-only key for use with Salesforce's cache-only key service. In creating this key, select the key you generated using DSM as your key source. Also, select a named credential that matches the named credential you previously defined in Salesforce. (For more information, see "Add (Generate) a DSM Key" on page 206.) After this key is created, the name of the key is displayed in the **Name** column, a green checkmark icon is displayed in the **Uploaded** column, and a status of FETCHED(<key_name>) is displayed in the **Key Material Origin** column of the **Tenant Secrets** page. In addition, a log message in the **Logs** page is recorded and displayed indicating a call-out connection to Salesforce has been created.

In order for Salesforce to use this key for its cache-only key service, Salesforce must be able to reach the endpoint (specified by the named credential) using the call-out connection that is created after you create the cache-only key in CCKM. If Salesforce can reach the endpoint (meaning the call-out connection between CCKM and Salesforce was set up successfully), then the key specified for the Unique Key Identifier is fetched from CCKM and becomes the active key within Salesforce. After performing the steps in this section, verify that Salesforce can reach the endpoint. See "Verify Call-Out Connection for Cache-only Key " on the facing page for more information.

> **Note**
> CCKM currently only supports DSM as the key source for use with Salesforce’s cache-only key service.

To create a cache-only key:

1. From the **Tenant Secrets** list page, within **New Key** box, select **Cache Only**. The **Create cache-only key** dialog box displays.
2. From the **Type** box, select the type of cache-only key to create. The following are the options (in the order of selection):
   - Data in Salesforce
   - Search Index
   - Data in Salesforce (Deterministic)
   - Analytics
   - Event Bus
3. From the **Tenant Secret** drop-down menu, select the key you generated using DSM as your key source. Enter the first three characters of the name of the key to bring it up for selection.
4. In the **Certificate** box, select a valid certificate (used to wrap the key for secure upload).
5. In the **Named Credentials**, select the named credential to associate with this cache-only key.
6. Click **Create cache-only key** to create the new cache-only key.

Verify Call-Out Connection for Cache-only Key

After creating the cache-only key, verify that Salesforce can fetch the key from CCKM using the call-out connection. When the CCKM receives the API call from Salesforce and successfully authenticates it, CCKM passes the tenant secret to Salesforce through the call-out connection.

To verify that Salesforce can fetch the key from CCKM using the call-out connection:

1. Login to Salesforce.
2. Click **Setup** and from the Search box of the Setup page, enter **Platform Encryption**.
3. Select **Key Management** and the Key Management page displays. You will see a list of keys.
4. From the list of keys, find the row that contains the cache-only key you just created (using the version number to identify it) and then click **Details** button next to it. The **Cache-Only Key: Callout Check** page displays.
5. Click the **Check** button to check the connection for your cache-only key callout. A callout connection starts to test the connection. Details about the results displays on the page.
6. Under the **Callout Collection Details**, verify that the values for **Unique Key identifier**, **Certificate Unique Name**, and **Named Credential** match the values selected during the creation of cache-only key.
7. Verify that the message in the results pane displays the following:
   `"The callout was successful"`.
8. Login to CCKM User Portal for Salesforce.
9. In the left navigation bar, select **Logs**. The **Logs** page displays.
10. Verify that two events named "Get_Cache_Only_Key" is listed on the page. (The first event indicates a Salesforce cache-only key call was made. The second indicates that the key was successfully read from DSM and returned to Salesforce.

Destroy a Tenant Secret

On the **Tenant Secret** page, destroying a tenant secret permanently deletes it from Salesforce. However, a backup key will remain in the key source you are using (CipherTrust Manager or DSM). After deletion, two new actions are displayed in the Action column: *Import* and *Delete Backup*.

> **Note**
> You can destroy only an archived tenant secret. If you want to destroy the currently active tenant secret, first change its status to ARCHIVED by uploading a new tenant secret (see "Create a Tenant Secret/Key" on page 195).

1. In the CCKM portal, in the left navigation bar, select **Tenant Secrets**.
2. Find the name of the tenant secret and click **Destroy** under the **Actions** column. The **Destroy Tenant Secret** dialog box displays with a warning asking you to confirm the key destruction by entering the supplied phrase.
3. Enter or copy/paste the confirmation phrase.
4. Click **Destroy**.

   The **Status** column for this tenant secret shows **DESTROYED**, and *Import* and *Delete Backup* actions are listed under the **Actions** column.

   > **Note:** If you simultaneously deleted the backup key using the **check box** option, the **Action** column will be blank.

Destroy a Cache-only Key

On the **Tenant Secret** page, destroying a cache-only key removes it from the Salesforce cache. However, a backup of the key still exists in DSM. After the key is destroyed, no actions are displayed under the **Action** column for the key.

> **Note**
> CCKM currently only supports DSM as the key source for use with Salesforce's cache-only key service.

1. In the CCKM portal, in the left navigation bar, select **Tenant Secret**s.
2. Find the name of the key and click **Destroy**. The **Destroy Tenant Secret** dialog box displays with a warning asking you to confirm the key destruction by entering the supplied phrase.
3. Enter or copy/paste the confirmation phrase.
4. Click **Destroy**.

   The Status column for this key shows **DESTROYED**.

Edit a Cache-only Key

If a certificate configured for use with a cache-only key has expired or you want to use another named credential that is configured with another CCKM hostname in Salesforce other than the one currently in use, edit the cache-only key configuration within the **Tenant Secret** page.

1. In the CCKM portal, in the left navigation bar, select **Tenant Secrets**.
2. Find the name of the key and click **Edit Key**. The **Edit <key_name> Cache-only Key** dialog box displays.
3. To update the certificate replacing the current certificate with another one, in the **Certificate** box, select the certificate you wish to use.
4. To update the named credential replacing the current named credential with another one, in the **Named Credentials** box, select the named credential you wish to use.
5. Click **Update cache-only key**.

Import a Tenant Secret

In the event that someone deletes a Salesforce tenant secret directly in Salesforce, it is possible to restore it from the CCKM as follows:

> **Notes**
> - The Import action is available for tenant secrets with the status DESTROYED.
> - If anyone deletes the backup key from CCKM, it will not be possible to import/restore the key to Salesforce.

1. In the CCKM portal, in the left navigation bar, select **Tenant Secrets**.
2. From the **Tenant Secrets** list, select **Import** in the **Action** column of a Destroyed key. The key and metadata are restored to Salesforce. The Success message is displayed and the key is listed at the top of the **All Tenant Secrets** page with the status **ARCHIVED**.

Rotate a Tenant Secret

CCKM allows for a manual or a scheduled rotation of a Salesforce tenant secret by the tenant secret type. When you rotate a tenant secret, that key becomes active and the previously active key is automatically archived. Scheduling a rotation of a key is adding a new version of the key at a scheduled time. Scheduling the rotation of a tenant secret requires that you first configure this feature from the **Schedules** page prior to enabling it using the **Auto-rotate** button at the top of the **Tenant Secrets** page.

> **Notes**
> - If you provide your Salesforce user password when configuring the scheduled rotation of a tenant secret, which is an optional step, the password is saved to the key source you are using (CipherTrust Manager or DSM).
> - The **Auto-rotate** button is set to **On** or **Off** depending on the key's autorotation status. If the key is scheduled for autorotation, this button is set to **On**. Otherwise, it is set to **Off** indicating the key is not scheduled for autorotation. The **Auto-rotate Tenant Secret for Type** dialog box displays when you click on the **Auto-rotate** button.

Manually Rotate a Tenant Secret (Add a New Version of a Tenant Secret)

Creating and uploading a tenant secret in CCKM is equivalent to manually rotating a tenant secret in CCKM.

Enable a Scheduled Tenant Secret Rotation

Enabling a scheduled rotation of a tenant secret requires that you first schedule to rotation in the **Schedules** page. Be sure to schedule the rotation of a tenant secret prior to enabling this feature.

During the scheduled date and time for the rotation, the tenant secret is rotated. The new tenant secret becomes active and the previously active tenant secret is automatically archived.

> **Note**
> A tenant secret that is a cache-only key remains as a cache-only key after rotation.

To enable a scheduled rotation of a tenant secret:

1. In the CCKM portal, in the left navigation bar, select **Tenant Secrets**. The **Tenant Secrets** list page displays.
2. From the **Tenant Secrets** list page, within the **Filter by** box, select the type of tenant secret to enable for a scheduled rotation. The following are the options:
   - Data in Salesforce
   - Search Index
   - Data in Salesforce (Deterministic)
   - Analytics
   - Event Bus
3. Click the **Auto-rotate** button. The **Auto-rotate Tenant Secret for Type** dialog box displays.
4. For **Enable autorotation on this type?**, select **On** to enable the autorotation of this tenant-secret type. By default, this toggle is initially set to **Off**.
5. From the **Schedule** drop-down menu, select a rotation schedule to which to apply to this key. Any key-rotation schedule you set in the **Schedules** page displays as an option for selection in the drop-down menu.
6. From the **Key Provider** box, select one of the following as the key source to which to add a new version:
   - **DSM**—Key is to be generated from DSM.
   - **CipherTrust Manager**—Key is to be generated from CipherTrust Manager.
7. (Optional) Select whether to apply the Salesforce key derivation mode to the tenant secret. If you choose to apply it, ensure that the **Use Salesforce key derivation** check box is selected. By default, the check box is selected. Otherwise, deselect the check box to opt out of applying the key derivation mode to the tenant secret.
8. From the **Certificate** drop-down menu, select a valid certificate (used to wrap the key for secure upload).
9. Click **Submit**.

Synchronize Tenant Secrets

Synchronization downloads tenant secrets that were directly created in the Salesforce portal (not in CCKM). It does not upload anything from CCKM to Salesforce. Periodically synchronize CCKM with Salesforce to get any newly created tenant secrets from Salesforce.

To synchronize tenant secrets:

1. In the left navigation bar, click **Tenant Secrets**.
2. Click **Synchronize**.

   Any new tenant secrets created in Salesforce are retrieved and displayed on the **Tenant Secrets** list.

Export Tenant Secrets

Export Secrets allows you to export the contents of the Tenant Secrets list into a report in a CSV format.

To export your Tenant Secrets list:

1. In the left navigation bar, click **Tenant Secrets**.
2. Click **Export Secrets**.

   This will download all of the tenant secrets in a report in a CSV format.

Backup Key

When a tenant secret is uploaded to Salesforce from CCKM, Salesforce creates a backup tenant secret (or key) as an encrypted blob, which contains information about the Salesforce tenant secret. The backup key can be used to restore a tenant secret that has been deleted in Salesforce. After Salesforce creates a backup key, CCKM automatically downloads and stores the tenant secret in the key source you are using (CipherTrust Manager or DSM). In addition, when Salesforce tenant secrets are synchronized, the backup key of any tenant secret that was created directly in the Salesforce portal is also downloaded and stored in the key source you are using.

Delete a Backup Key

Delete a backup key when you definitely do not want the key and have no corresponding data that will ever need decryption. For a tenant secret that has been deleted:

1. In the CCKM portal, in the left navigation bar, select **Tenant Secrets**.
2. From the list, select **Delete Backup** from the **Action** column of a Destroyed tenant secret. The **Destroy Key** warning page is displayed.
3. Enter or copy/paste the confirmation phrase.
4. Click **Delete**.

   The key is removed from CCKM.

Certificates

The **All Certificates** list page allows you to view all Salesforce certificates, including Name, Type, Key Size and Expiration Date. Use the **Synchronize** button to download newly created certificates from Salesforce.

> **Note**
> See "Prerequisites on Salesforce" on page 187 for the minimum required parameters for creating Salesforce certificates compatible with CCKM.

Synchronize Certificates

Synchronization downloads certificates that were directly created in the Salesforce portal (not in CCKM). Periodically synchronize CCKM with Salesforce to get any newly created tenant certificates from Salesforce.

1. In the left navigation bar, click **Certificates**.
2. Click **Synchronize**.

   Any new certificates created in Salesforce are retrieved and appear in the UI.

Reports

Before the Salesforce keys can be represented in the CCKM reports, Event monitoring and Event log files must be activated in Salesforce. In addition, the log files validity must be set to 30 days in Salesforce.

> **Note**
> Currently, there is no support for cache-only keys within CCKM reports.

The **All Reports** page displays links to the following detailed individual reports:

- **Key Activity Report**

  Displays the key activity type (such as TS (tenant secret) stored or TS generated) performed by a specific user along with the associated key ID, key version number, key name, and time at which the activity took place.

- **Cloud Key Manager User Action Report**

  Lists the users and their actions in creating, uploading, importing, destroying, and deleting keys in the CCKM portal.

For each report displayed on the **All Reports** page, the report name, its description, the email address of the user who had run it (Created By), and the date on which it was previously run (Last Run) is displayed. For a specific report you select to view, the date on which it was previously run (Last Run), the start and end dates selected for the report, the email address of the user who had run it (Created By), and the total number of records available in the report (Report Count) is displayed. In addition, an **Actions** column is displayed, which provides the **Delete Report** link to allow for the deletion of the selected report from CCKM.

Delete a Report

1. From All **Reports** page, select the specific report you wish to delete. The report displays.
2. From the **Actions** column, click **Delete Report**. The **Delete Current Report** dialog box with a warning that you are about to delete a current report run on a specified date and time.
3. Click **Delete** to confirm. The report is deleted from CCKM.

Generate a Report

1. In the CCKM portal, in the left navigation bar, select **Reports**.
2. Under **Report Name**, select the report you wish to run.
3. From **Date Range**, click **Last day** to run the report for the activities from the previous day or **Specific** to run the report using a date range. If you choose to run by a date range, enter the start and end dates in the format MM/DD/YY or select the dates from the calendar button at the right of the box.
4. Click **Run Report**. The report displays.
5. To download the report, click **Download** at the top right of the page and select the report format (PDF or CSV) to download.

Schedules

The **Schedules** page allows Cloud administrators to view and schedule the following job types:

- Key Rotation
- Key Synchronization

CCKM provides support for multiple schedules for each of the job types.

At the top right of the **Schedules** page, the **Add Schedule** button is available from which to add a scheduled job type.

> **Note**
> Currently, there is no support for cache-only keys within CCKM schedules.

Schedule Key Rotation

CCKM provides support for multiple schedules of the rotation (autorotation) of a Salesforce tenant secret. When you rotate a tenant secret, that key becomes active and the previously active key is automatically archived. Scheduled rotation is configured in the **Schedules** page. Note that if you provide your Salesforce user password when configuring a scheduled key rotation, which is an optional step, the password is saved to the key source you are using (CipherTrust Manager or DSM).

During a scheduled rotation of a key, CCKM uses the associated username and password to rotate the key. If the password expired or changed, then the new password must be reentered. Otherwise, the scheduled rotation will fail.

You can pause or delete a scheduled key rotation in the **Schedules** page. Note that when a key rotation schedule is deleted from CCKM, then the Salesforce user password is also deleted from the key source you are using (CipherTrust Manager or DSM).

> **Note**
> CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

Set up a Key Rotation Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The Schedules page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_ROTATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule). If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.

   Set the repeat interval of the key rotation by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

   If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
8. (Optional) Enter your Salesforce user password in the **User Password** box.
9. Click **Save** to save the setting.

Pause a Key Rotation Schedule

To pause a key rotation schedule, select **Yes** for **Pause** within the **Add Schedule** dialog box and then click **Save**. The setting is saved. To resume the rotation schedule after pausing it, select **No** for **Pause** and then click **Save.**The setting is saved.

Delete a Key Rotation Schedule

To delete a key rotation schedule, select **Delete** from the **Actions** drop-down menu from the **Schedules** page. Note that the Salesforce user password associated with the given key rotation schedule is also deleted from the key source you are using (CipherTrust Manager or DSM).

Schedule Key Synchronization

CCKM allows for multiple schedules of the synchronization of Salesforce tenant secrets. Synchronizing downloads any keys that were created in Salesforce, outside of the CCKM portal, into your CCKM. Scheduled key synchronization requires that this feature be configured in the **Schedules** page. You can also pause or delete a scheduled key synchronization in the **Schedules** page. Note that when a key synchronization schedule is deleted from CCKM, then the Salesforce user password is also deleted from the key source you are using (CipherTrust Manager or DSM).

> **Note**
> CCKM does not support the concurrent use of CipherTrust Manager and DSM as key sources.

Set up a Key Synchronization Schedule

1. In the CCKM portal, in the left navigation bar, select **Schedules**. The **Schedules** page displays.
2. On the **Schedules** page, select the **Add Schedule** button. The **Add Schedule** dialog box displays including your username in the **Username** box.
3. From the **Job Type** drop-down menu, select **KEY_SYNCHRONIZATION**.
4. In the **Name** box, enter the name of the schedule. Ensure this name is unique for this job type.
5. (Optional) Enter a description of the schedule in the **Description** box.
6. For **Pause**, select **No**.
7. For **Schedule**, set a schedule for the rotation by selecting **Basic** (to enter a date and interval for the schedule) or **Advanced** (to enter a Cron Expression for the schedule). If you selected **Basic**, select a start date and time from a date and time selector by clicking the calendar button to the right of the box. The time is based on a 12-hour clock and the A.M./P.M. modifiers.

   Set the repeat interval of the key rotation by entering a number in **Repeat Interval** and then selecting Day, Week, or Month from the drop-down menu. The supported ranges are 1 through 30 days, 1 through 4 weeks, and 1 through 12 months. You can increase or decrease the number you enter.

   If you selected **Advanced**, enter the cron expression in the **Cron Expression** box.
7. Enter your Salesforce user password in the **User Password** box.
8. Click **Save** to save the setting.

Logs

CCKM maintains logs for events, such as login, key generation, synchronizing of tenant secrets, calls from Salesforce to CCKM for cache-only keys, and deletion of keys, and displays these logs in the **All Logs** list of the **Logs** page. If Replay Detection for cache-only keys is enabled, then the request ID is included in logged events for cache-only keys. For each logged event listed on the **Logs** page, the following information is displayed:

- Event name
- Severity level
- Date and time of the event

- Event message
- User (name) associated with the event

You can search for a specific logged event from the **Logs** page using the **Search** box. Allowable filters on the search are event name, severity level, event message, and user (name).

Key Sources

The **Key Sources** page displays all of the keys that are generated from the key source you are using. Three tabs are available at the top of the page:

- **DSM Key:** Use this tab to use DSM as your key source. The keys (including cache-only keys) are generated and stored within DSM.

> **Note:** CCKM currently only supports DSM as the key source for use with Salesforce's cache-only key service.

- **CipherTrust Manager:** Use this tab to use CipherTrust Manager as your key source. The keys are generated and stored within CipherTrust Manager.
- **nShield Key:** This tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for Salesforce.

The **Key Sources** page displays all currently posted tenant secrets in your key vault for Salesforce.
Keys are listed by name, key type (the cloud in which this key is used), the algorithm of the key, the creation date of the key, key description, and actions you can take regarding that key. Note that the key algorithm currently supported for a Salesforce key in CCKM is AES.

A **Search** box is available at the top left of the **Keys Sources** page allowing for a search of a specific key by entering any one of the following parameters:

- Name (key name)
- Key Type
- Algorithm
- Description
- Created By (a key created by a specific user)

> **Note**
> The Salesforce term "tenant secrets" is used interchangeably with "keys" in CCKM. Manage tenant secrets from two places in the CCKM portal: the Tenant Secrets page and the **Key Sources** page.

Key Sources

The **Key Sources** page displays all of the keys that are generated from the key source you are using. Three tabs are available at the top of the page:

- **DSM Key:** Use this tab to use DSM as your key source. The keys (including cache-only keys) are generated and stored within DSM.

> **Note:** CCKM currently only supports DSM as the key source for use with Salesforce's cache-only key service.

- **CipherTrust Manager:** Use this tab to use CipherTrust Manager as your key source. The keys are generated

and stored within CipherTrust Manager.

- **nShield Key:** This tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for Salesforce.

The **Key Sources** page displays all currently posted tenant secrets in your key vault for Salesforce.
Keys are listed by name, key type (the cloud in which this key is used), the algorithm of the key, the creation date of the key, key description, and actions you can take regarding that key. Note that the key algorithm currently supported for a Salesforce key in CCKM is AES.

A **Search** box is available at the top left of the **Keys Sources** page allowing for a search of a specific key by entering any one of the following parameters:

- Name (key name)
- Key Type
- Algorithm
- Description
- Created By (a key created by a specific user)

> **Note**
> The Salesforce term “tenant secrets” is used interchangeably with “keys” in CCKM. Manage tenant secrets from two places in the CCKM portal: the Tenant Secrets page and the **Key Sources** page.

Add (Generate) a Key

From the **Key Sources** page, you can add (or generate) a key in CCKM using CipherTrust Manager or DSM as a key source. These keys are of the AES algorithm type.

> **Note**
> The nShield Key tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for Salesforce.

Add (Generate) a DSM Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **DSM Key** tab.
3. Select **Add Key > Salesforce Key**.
4. In the **Add DSM Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to **Salesforce Key** and (key) **Algorithm** is fixed to **AES256**.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. Click **Save** to save the new key to CCKM. The new key is added to the key list.

   (Reset clears the screen before it is saved.)

Add (Generate) a CipherTrust Manager Key

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Select the **CipherTrust Manager Key** tab.
3. Select **Add Key > Salesforce Key**.

4. In the **Add CipherTrust Manager Key** dialog box, select the name of the user of the key from the **User** drop-down menu. **Service** is fixed to **Salesforce Key** and (key) **Algorithm** is fixed to **AES256**.
5. In the **Name** box, enter the name of the new key.
6. Enter a description of the key in the **Description** box.
7. Click **Save** to save the new key to CCKM. The new key is added to the key list.

   (Reset clears the screen before it is saved.)

Add (Generate) an nShield Key

The **nShield Key** tab is currently unavailable for use as CCKM does not currently support nShield Connect as a key source for Salesforce.

Delete a Key

Deleting a key from the Key Sources page deletes the key from CCKM. If you delete a key from the Key Sources page that has already been uploaded to Salesforce, the key will not be deleted from Salesforce.

1. In the CCKM portal, in the left navigation bar, select **Key Sources**.
2. Click **Delete** next to the target key. The **Delete Key** dialog box with a Warning displays asking you to confirm the key deletion by entering the supplied phrase.
3. Enter or copy/paste the confirmation phrase.
4. Click **Delete** to confirm.

Manage Keys in Key Sources List Page

In addition to adding and deleting keys, the **Key Sources** page allows you to sort the existing keys by Name, Key Type, Algorithm and Created Date.

Settings

To access the **Settings** page in CCKM, select **Settings** from the left-hand navigation bar. From the **Settings** page, you can:

- Enter the emails of the recipients who are to receive alerts and reminders.
- Set up key alerts.
- Configure a remote syslog server to which to send syslog messages.
- Configure the username and password to associate with the CCKM Rest API endpoint (or callout endpoint) for use with cache-only keys.
- View information about the software release number of the CCKM you are running including the build and version number.

Adding Recipients' Emails (for Alerts and Reminders)

Use the **General** settings tab to enter the designated recipients' emails for email alerts and reminders to be sent from the system. Enter the recipients' email addresses separated by a comma in the **Alert Format** box.

Enabling Alerts and Reminders

Use the **Tenant Secret Alerts** tab from the **Settings** page to specify whether to send an alert to the specified user email(s) when a Salesforce key is deleted, restored, or uploaded in Salesforce. You can also use this tab to set up a reminder to manually rotate (or rekey) a key along with the frequency (in days) at which the reminder is sent to the specified user email(s). The user emails are set up in the **General** tab. By default, no alerts or reminders are sent.

To set up key alerts or a reminder to manually rotate (or rekey) a key:

1. In the **Settings** page, select the **Tenant Secret Alerts** tab.
2. Under **Alerts**, click **On** for each of the alert types and reminder you wish to enable:
   - Rekey Alert (reminder)
   - Destroy Tenant Secret
   - Import Tenant Secret
   - Upload Tenant Secret
3. If you set a reminder for **Rekey Alert**, then in the **Days** box, enter the number of days for the frequency an email reminder to manually rotate a key is sent.
4. Click **Save** to save the setting.

Configuring Syslog Server

Use the **Syslog** tab from the **Settings** page to configure a remote syslog server to which to send syslog messages. Note that the default port number of 514 and the facility name of “LOCAL1” for the syslog server display in the **Port** and **Facility** boxes, respectively. These boxes are not available for modification.

Configure a Remote Syslog Server

1. In the **Settings** page, select the **Syslog** tab. The **Syslog** dialog box displays.
2. Enter the hostname or IP address of the syslog server in the **Hostname** box.
3. (Optional) Enter a description of the syslog server in the **Description** box.
4. Click **Save** to save the setting.

Delete Remote Syslog Server Configuration

To delete the configuration of a remote syslog server, click **Delete** from the **Syslog** dialog box. No syslog messages are sent to the remote syslog server after the deletion of this configuration.

Configure Cache-only Key Endpoint

Use the **Endpoint for Cache-only Key** tab from the **Settings** page to configure the CCKM username and password to associate with the URL of the CCKM Rest API endpoint (or callout endpoint) for use with cache-only keys.

> **Note**
> **Note the call-out URL, username, and password when performing the steps in this section. These values are required when you create the Named credential in Salesforce. See "Set up Named Credential in Salesforce" on page 189 for more information.**

To configure the username and password to associate with the URL of the CCKM Rest API endpoint:

1. In the **Settings** page, select the **Endpoint for Cache-only Key** tab. The **Cache-only Key Configuration** page displays.
2. The URL of the CCKM Rest API endpoint automatically displays in the **Call out URL** box.

   > **Note:** When defining the named credential for use with cache-only keys within Salesforce's **Named Credentials Setup** page, copy the URL from the **Call out URL** box and paste it into the **URL** box in **Named Credentials Setup** page. Verify that URL points to public facing CCKM instance. Verify hostname and port number is correct. For example, if CCKM URL is https://thalescckm.com:8443/kmaas/ and the org ID is 00DB00007731234MAK, then the callout URL will be: https://thalescckm.com:8443/kmaas/00DB00007731234MAK/

3. In the **Username** box, enter the username to associate with the URL of the CCKM Rest API endpoint. This username must match the username configured in Salesforce's **Named Credentials Setup** page.
4. In the **Password** box, enter the password to associate with the URL of the CCKM Rest API endpoint. This password must match the password configured in Salesforce's **Named Credentials Setup** page.
5. Click **Save Settings** to save the setting.

About

Use the **About** tab to view the release number of the CCKM you are running.

Chapter 18: Backup and Restore

In the event of a software crash or system changes, a backup of the CCKM configuration data (such as usernames, encryption keys, key attributes, local CAs, external CAs, certificates, signatures, and syslog connections) stored within the key source you are using (CipherTrust Manager or DSM) and MongoDB can be used to restore to your most recent backup created.

Backup and Restore Recommendations

The following are the recommendations for backing up and restoring the CCKM configuration data within the key source you are using and MongoDB:

- CCKM should be inactive when the backups are performed to avoid data corruption.
- Backups of the key source and MongoDB should be performed during the same period (ensuring that the same snapshot of the CCKM configuration is backed up in both).
- Daily backups of your key source and MongoDB are recommended.

Backup and Restore for DSM

See the *Data Security Manager (DSM) Administration Guide* for information on how to backup and restore the CCKM configuration data stored within DSM, in addition to how to configure DSM for high availability.

Backup and Restore for CipherTrust Manager

See the *CipherTrust Manager Administrator Guide* for information on how to backup and restore the CCKM configuration data stored within CipherTrust Manager.

Backup and Restore for MongoDB

1. Make sure not to use CCKM during backup.
2. Use following commands to backup data using the mongodb dump utility:

   ```
   mongodump -u kmaas_user -p {password} --db kmaas --out kmaas.dump
   tar cvf backup.tar kmaas.dump
   ```

   Move this tar file to reliable storage.

3. Use following commands to restore data to new MongoDB instance.
   - Deploy new MongoDB instance.
   - Configure mongodb by following same steps that you used to configure earlier MongoDB
     - Steps for configuring MongoDB
   - Empty contents of the kmaas database (if it already exists).
     - You can use db.dropDatabase() to remove db from MongoDB.
   - Use following command to restore previously backed-up data.

     ```
     tar xvf backup.tar
     ```

     ```
     mongorestore -u kmaas_user -p {password} --db kmaas kmaas.dump/kmaas
     ```

Chapter 19: Troubleshooting

Troubleshooting nShield Configurations

This section describes a few of the common issues that may arise during the configuration of the connection from CCKM to nShield Connect HSM(s), and the associated RFS and Security World within the Settings page and the steps to take to attempt to resolve these issues.

For more information on troubleshooting the configuration for nShield Connect, and the associated RFS and Security World, refer to the *nShield Connect User Guide for Unix*.

Table 19-1: Basic Troubleshooting nShield Configurations

Issue	How to Resolve
You entered an incorrect IP address for an nShield Connect during "Configure nShield Connect (for Azure and AWS Clouds)" on page 74 procedure.	Verify the electronic serial number (ESN) of each nShield Connect. To verify this, ensure that the output of the "`rfs-sync --setup`" command indicates "Configuration successfully written". From the **nShield 1 IP Address** box of the **Key Sources** page > **nShield** tab, enter the correct IP address of the nShield Connect and then click **Add to Security World** again.
A network connection from CCKM to nShield Connect cannot be detected.	Verify the ESN of each nShield Connect. To verify this, ensure that the output of the "`rfs-sync --setup`" command indicates "Configuration successfully written". Also, ensure that the output of the "`rfs-sync --update`" command includes "Updated module ..." or "Finished synchronization".
You entered an incorrect IP address for the RFS computer during the "Configure nShield Connect (for Azure and AWS Clouds)" on page 74 procedure.	The output of the "`rfs-sync --update`" command does not include "Updated module ...". From the nShield RFS IP Address box of the **Key Sources** page > **nShield** tab, enter the correct IP address of the RFS computer and then click **Add to Security World** again.

Only after you have verified that the output of all commands (displayed on the right side of the nShield Info page) are correct should you proceed to saving the nShield Connect configurations during the Install and Configure nShield Connect as a Key Source procedure. Saving the configuration restarts the hard server (Security World Software server) on which the CCKM is installed and configured.

Troubleshooting Logs Files for Thales Support Team

Thales Support will request for you to download the debug logs from CCKM in order to assist you in troubleshooting your issue(s).

To download the debug logs, click on **Download Debug Logs** on the **Logs** page within the CCKM Admin Portal. You can also download the debug logs through the **upload** command within the CCKM **applog** commands.

Restrictions

The following restrictions apply to the current version of CCKM:

- Azure key vault name and key name combined cannot exceed 64 characters.
- The DSM domain name cannot exceed 22 characters.

Appendix A: CCKM CLI Commands

The CCKM Command Line Interface (CLI) enables you to configure the CCKM network and perform other system-level tasks using the CCKM CLI commands.

This appendix provides a high-level overview of each of the command categories that are available in the CCKM. For information about the details of each command and its options, run the command with the "--help" option within the CCKM CLI.

CCKM CLI Navigation

The following are the supported categories of CCKM CLI commands:

- Maintenance
- Network
- System
- User
- Applog

As a CCKM CLI administrator, log on to the CLI, and then enter a command category by typing the category name at the command line prompt. For example, type `system` to enter the system category. While in the category, you can execute the commands for that category.

Enter the entire category name, command, or argument, or enter just enough characters to identify uniquely the category, command, or argument. You can use the <Tab> key to complete a category, command, or argument. Enter enough characters to uniquely identify a category, command, or argument, and then press the <Tab> key. The CLI will complete it for you.

For example, at the top level, enter `n` and press <Tab>, the CLI expands it to network.

Inside the network category, you can enter `d`<Tab> and it expands to `dns`. Type `n`<Tab> and it expands to `netstat`. Note that there are other commands in the network category that start with n, such as nslookup, and ntpservice. To ensure to expand to netstat command, enter `ne` to identify it uniquely.

Other supported CLI navigation methods are:

- Enter a question mark (?) to display the next command or argument that is expected. Think of it is as a shorthand form of help.
- Enter "`up`" to return to the top level so that you can enter another category. You can enter another category only from the top level.
- Enter "`exit`" at any time to end the current CLI session.

Maintenance Commands

The CCKM maintenance commands enable you to upgrade the CCKM. Table A-1: "CCKM Maintenance Commands" below lists the supported CCKM maintenance commands:

Table A-1: CCKM Maintenance Commands

Command	Description
resetwebadmin	Resets the CCKM administrator password to the default value of 'admin123'.
updatedbpassword	Updates the MongoDB password in CCKM configuration. Provide the new password to replace the current password stored in the CCKM conf file.
upgrade	Upgrades the CCKM. Note: A backup of the CCKM data is saved to the CCKM as part of the upgrade process.
quit|q|up|<ctrl-d>	Quits or returns to the previous menu.
Exit	Exits the application.

Network Commands

The CCKM network commands are used to set, modify, or delete IP addresses on the CCKM, set up DNS servers, and perform other network-related tasks. Table A-2: "CCKM Network Commands" below lists the supported CCKM network commands:

Table A-2: CCKM Network Commands

Command	Description
checkport	Checks port connection status of a host.
dns	Shows or configures the DNS settings for the CCKM.
ip	Shows the CCKM network interface settings.
netstat	Prints network connections.
nslookup	Queries Internet name servers.
ntpservice	Configures NTP service settings.
ping	Pings an IP address, host name, or FQDN.
route	Sets a static route.
service	Starts, stop, or restart a network service or gets the status of a network service.
set	Configures a network device interface.
setup	Sets up a network configuration.
show	Shows network device configuration.
traceroute	Traces route to an IP address or hostname.
quit|q|up|<ctrl-d>	Quits or returns to the previous menu.
exit	Exits the application.

System Commands

The CCKM system commands enable you to configure the CCKM host settings (such as, date and time or timezone) and security features, reboot or shut down the CCKM, show current CCKM version information, and perform other system administration tasks. Table A-3: "CCKM System Commands" below lists the supported CCKM system commands:

Table A-3: CCKM System Commands

Command	Description
date	Prints or sets the system date and time.
host	Performs DNS lookups.
hostname	Sets or shows the CCKM hostname.
hosts	Updates the host file.
lastlogin	Shows the last logged-in date for each user on the system.
port	Configures the ports of the CCKM Admin Portal and User Portal.
reboot	Reboots the CCKM appliance.
security	Configures the CCKM security features including the certificates.
server	Manages the CCKM server.
shutdown	Stops the CCKM software and powers off the virtual appliance.
timezone	Sets or shows the system timezone.
uptime	Shows how long the CCKM has been running.
version	Shows the CCKM version information.
who	Shows information about the users who are currently logged into the CCKM appliance using CLI.
quit|q|up|<ctrl-d>	Quits or returns to the previous menu.
exit	Exits the application.

User Commands

The CCKM user commands enable you to add, modify, delete CCKM CLI users, and display information about these users. Table A-4: "CCKM User Commands" below lists the supported CCKM user commands:

Table A-4: CCKM User Commands

Command	Description
add	Adds a new CLI user.
delete	Deletes a CLI user.
modify	Changes the password of a CLI user.
show	Lists all of the configured CLI users.

Table A-4: CCKM User Commands (continued)

Command	Description
quit|q|up|<ctrl-d>	Quits or returns to the previous menu.
Exit	Exits the application.

Applog Commands

The CCKM applog commands enable you to upload debug logs from the CCKM to the remote servers using the secure copy protocol (SCP) to transfer the log files between the two hosts. Table A-5: "CCKM Applog Commands" below lists the supported CCKM applog commands:

Table A-5: CCKM Applog Commands

Command	Description
upload	Uploads logs to remote server using SCP.
quit|q|up|<ctrl-d>	Quits or returns to the previous menu.
Exit	Exits the application.

Appendix B: How to Export Your DSM Server Certificate from DSM

CCKM for a supported cloud performs key operations through REST APIs exposed in the key source (CipherTrust Manager, DSM, or nShield) you are using. If you are using DSM as a key source for CCKM, CCKM requires the DSM server certificate to invoke the APIs. You can download the certificate through the use of a browser pointing to the DSM. Download the file in a Base64 format. When configuring the DSM as your key source in the **Key Sources** page of the CCKM Admin Portal, you are required to upload the DSM certificate to CCKM.

This appendix describes how to export a DSM certificate file from DSM to your laptop or desktop using the Chrome Internet browser. This file is then available to upload to the CCKM. For more information, see "Configure DSM as Key Source" on page 73.

> **Note**
> The steps to export a DSM server certificate using the Firefox and IE browsers may differ slightly from the steps for the Chrome browser.

Exporting your DSM Server Certificate

To export your DSM server certificate from DSM using the Chrome browser, do the following:

1. Enter “https://<DSM IP address>” in a browser. The DSM user login page is displayed.
2. Log on to the web-based management console of the DSM.
3. In the web URL, click on the **Not Secure** section and click **View Certificate**.
4. Click the **Details** tab of the certificate.
5. Click **Copy to File**. The **Certificate Export Wizard** dialog box displays. This wizard is used to export your certificate in the appropriate format.
6. In the **Export File Format** page, select the **Base-64 encoded X.509 (.CER)** format option.
7. Click **Next**.
8. In the **File to Export** page, enter the directory to which to save the file within your laptop or desktop and specify the file name (for example, “DSM.cer”).
9. Click **Finish** to complete the export.

Appendix C: Configure SSL/TLS Certificate Chain of Trust for CCKM

The chain of trust of an SSL or TLS certificate is an ordered list of certificates. The following key and certificates are contained within the certificate chain of trust:

- Server.key—Private key of end-user certificate. This key is private to CCKM and never exposed publicly.
- Server.crt—End-user certificate, which would be signed by Intermediate CA.
- IntermediateCA.crt—Certificate of CA that certifies end-user certificate.
- RootCA.crt—Certificate of CA that certifies Intermediate CA certificate. This is the root CA.

> Note that there can be more than one intermediate CA certificate forming a chain of trust.

The certificate chain of trust enables the receiver of the certificate to verify that the sender and all intermediate certificates are trustworthy and that the certificate is valid.

CCKM provides a way for you to upload a chain of trust for an SSL/TLS certificate, which is sent to the web browser when you open up a browser window to login into the CCKM Admin and kmaas Portal.

Create and Configure SSL/TLS Certificate Chain of Trust

To create a chain of trust certificate and then configure a new CCKM server certificate based on the new chain of trust certificate, do the following:

1. Create a certificate chain of trust using the following `cat` command in a Bash shell outside of CCKM CLI:
   ```
   $cat Server.crt IntermediateCA.crt RootCA.crt >> chain_of_trust.crt.
   ```
2. Log into the CCKM CLI to access CCKM CLI.
3. At CCKM command prompt, enter the following commands to configure a new CCKM server certificate:
   ```
   cckm> system
   system> security --setservercert
   ```
4. The following message displays:
   ```
   WARNING: Setting a new server certificate operation will implicitly restart the
   application.
   Do you wish to continue? <y/n> [n]:y
   ```
5. Type `y` to continue.
6. You are prompted to enter the contents of your private key and then press **Enter** and **Ctrl+d**. Copy the contents of the Server.key file (your private key) into the CLI and then press **Enter** and **Ctrl+d**:

   ```
   Enter private key content and press Enter and Ctrl+d:
   ```
7. You are prompted to enter the contents of your server certificate and then press **Enter** and **Ctrl+d**. (Copy the contents of the chain_of_trust.crt file (from Step 1), paste it into CLI, and then press **Enter** and **Ctrl+d** :
   ```
   Enter server certificate content and press Enter and Ctrl+d:
   ```
8. After the Server key and chain_of_trust.crt is uploaded, CCKM restarts the CCKM Admin and User portals. After these portals are back online, open a new browser and login into the CCKM Admin Portal.
9. Verify that the certificate chain of trust for CCKM is valid:

> **Note**
> The following steps are based on the use of the Chrome Internet browser. The steps based on the use of the Firefox and IE browsers may differ slightly.

a. In the web URL, click on the Padlock icon in the address bar, and click **Certificate**.

b. Click the **General** tab to view that the certificate is valid.

Appendix D: Sample Python Scripts for Authenticate and Call CCKM REST API

This Appendix demonstrates how to authenticate and call CCKM REST API using a Python script. After launching and configuring a CCKM instance, you can access CCKM REST API at *https://<your cckm hostname* or *<ip address>:8443/kmaas/doc*. You can use your preferred tools or scripts to call CCKM REST API. In this appendix, a Python script is used to demonstrate these API calls.

To call CCKM REST API, the script must first authenticate to CCKM using your cloud credentials. The authentication call will differ slightly depending on the following:

- Cloud service you are using
- Whether you are using Azure service principal
- Whether you are using AWS long-term credentials or temporary credentials

After successful authentication, you will get a cookie JSESSIONID and start a new user session. Subsequent calls to CCKM REST API within the user session must include this cookie. In addition, all requests must have a valid "X-XSRF-TOKEN" header to prevent a Cross Site Request Forgery (CSRF) attack. The following is the authentication flow:

1. Call *https://<your cckm>:8443/kmaas*. Get the X-XSRF-TOKEN from the response header, and use it in the request header.
2. Call proper auth API, passing cloud credentials in the Content-Type "application/x-www-form-urlencoded".
3. If auth call succeeds, save the JSESSIONID cookie, and attach it for subsequent calls.
4. Call CCKM REST API, using Content-Type "application/json"
5. Call logout.

The following are the current limitations when using CCKM REST API:

1. If your Azure tenant uses federated authentication, or enable MFA, you cannot use your Azure user credentials in CCKM REST API. You will have to configure CCKM as a service principal and provide admin consent to grant permissions to the CCKM app to access your Azure resources. During the admin consent process, you will choose a tenant password. Thereafter your REST API client or script can use your tenant name and password to authenticate to CCKM REST API.
2. You can use the temporary credentials of the AWS Security Token Service (STS) to authenticate CCKM REST API. However, when you call the add schedule REST API, you have to pass long-term credentials. If you do not have long-term credentials, then create your schedule jobs outside of CCKM instead of using the CCKM schedules.

The sections that follow provide four sample Python scripts used to authenticate and call CCKM REST API. These scripts are used to manage the following:

- Azure keys
- AWS keys
- Google Cloud keys
- Salesforce keys

These scripts use a Python Requests module to handle the session cookie. To see how each script works, enter your cloud credentials and remove commented out calls.

cckm-azure.py

```
import requests
import json
import sys


requests.packages.urllib3.disable_warnings()
session = requests.Session()
session.verify = False
resp = session.get(cckm_url)

# For Azure user login.
# Replace with your host, username and password
cckm_url = "https://host:8443/kmaas/"
username = ""
password = ""

session.headers = {'X-XSRF-TOKEN': resp.headers['X-XSRF-TOKEN']}
data = {'username': username, 'password': password}
resp = session.post(cckm_url + "auth/azure", data=data)
if resp.status_code != 200:
     print resp.content
     sys.exit()

"""
# For Azure service principal login
# Replace your tenant name and tenant password
tenant = "mytenant.onmicrosoft.com"
password = ""

session.headers = {'X-XSRF-TOKEN': resp.headers['X-XSRF-TOKEN']}
data = {'tenant': tenant, 'password': password}
resp = session.post(cckm_url + "auth2/azure", data=data)
if resp.status_code != 200:
     print resp.content
     sys.exit()
"""

session.headers['Content-Type'] = 'application/json'

def call_api(method, api, data=None):
     if method == 'post':
        resp = session.post(cckm_url + api, data=json.dumps(data))
     elif method == 'put':
        resp = session.put(cckm_url + api, data=json.dumps(data))
     else:
        resp = session.get(cckm_url + api)
print "%s %s" % (method, api)
```

```
if data:
     print data
print json.dumps(resp.json(), indent=4)
print
return resp.json()

# Following are example REST API calls, uncomment needed lines

#call_api("get", "me")
#call_api("post", "rest/azureKeys/synchronize")
#call_api("get", "rest/azureKeyVaults")
#call_api("get", "rest/azureKeys")
#call_api("get", "rest/keyvaultkey")

"""
# Add a source key
name_of_new_source_key = "A-new-Azure-source-key"
data = {"name": name_of_new_source_key, "algorithm": "RSA2048", "service": "Azure",
"description": "a new key", "cckmId": username}
call_api("post", "rest/keyvaultkey", data)
"""

"""
# Upload the new source key to Azure
# First search source key by name and find its ID
source_key = call_api("get", "rest/keyvaultkey/keys?search=%s" % name_of_new_source_key)
source_key_id = source_key['content'][0]['id']

# Prepare Azure key, replace with your key vault and key names
azure_key_vault = "jimmy-kv-west-europe"
azure_key_name = "A00cl1-vceg"
data = {
     "keyVaultKeyId": source_key_id,
     "keyVaultName": azure_key_vault,
     "keyVaultDisplayName": azure_key_vault,
     "keyName": azure_key_name ,
     "attributes": {
        "enabled": True,
        "exp": 1546329600
     },
     "hsm": False,
     "key_ops": ["encrypt"],
     "tags": {
        "tag1": "test tag"
     }
}
call_api("post", "rest/azureKeys/upload", data)

# Find Azure key by name
azure_key = call_api("get", "rest/azureKeys/search/findByName?name=%s" % azure_key_name )
"""

#call_api("get", "rest/azureAudits")

resp = session.post(cckm_url + "logout")
```

cckm-aws.py

```
import requests
import json
import sys

# Replace with your host, username and password
cckm_url = "https://host:8443/kmaas/"
aws_access_key_id = ""
aws_secret_access_key = ""
# To use AWS STS temporary credentials, provide session_token
aws_session_token = ""

requests.packages.urllib3.disable_warnings()
session = requests.Session()
session.verify = False
resp = session.get(cckm_url)
session.headers = {'X-XSRF-TOKEN': resp.headers['X-XSRF-TOKEN']}

def call_api(method, api, data=None):
     if method == 'post':
        resp = session.post(cckm_url + api, data=json.dumps(data))
     elif method == 'put':
        resp = session.put(cckm_url + api, data=json.dumps(data))
     else:
        resp = session.get(cckm_url + api)
     print "%s %s" % (method, api)
     if data:
        print data
     print json.dumps(resp.json(), indent=4)
     print
     return resp.json()

# Authenticate CCKM REST API with AWS credentials
if aws_session_token:
     data = {'username': aws_access_key_id , 'password': aws_secret_access_key, 'session_token'
: aws_session_token}
else:
     data = {'username': aws_access_key_id , 'password': aws_secret_access_key }

resp = session.post(cckm_url + "login/aws", data=data)
print resp.status_code
if resp.status_code != 200:
     print resp.content
     sys.exit()

# Example CCKM REST API calls
session.headers['Content-Type'] = 'application/json'

# Example REST API calls, uncomment needed lines
#me = call_api("get", "me")
#call_api("post", "rest/awsKeys/synchronize", [ "us-east-1" ])
#call_api("get", "rest/awsKeys/aliases")

"""
# REST API calls for creating and uploading DSM key to AWS
# Add a source key in DSM
name_of_new_source_key = "A-new-AWS-source-key"
user_arn = me["aws"][0]["arn"]
```

```
data = {"name": name_of_new_source_key, "algorithm": "AES256", "service": "AWS", "description":
"a new key", "cckmId": user_arn, "provider": "DSM"}
call_api("post", "rest/keyvaultkey", data)

# Upload the new source key to AWS
# First search source key by name and find its ID
source_key = call_api("get", "rest/keyvaultkey/keys?search=%s" % name_of_new_source_key)
source_key_id = source_key['content'][0]['id']

# Prepare AWS key, replace with your key alias
aws_key_alias = "cckm-aws-test"
data = {
     "sourceKeyId": source_key_id,
     "aliasName": aws_key_alias,
     "region": "us-east-1",
     "tags": {
        "description": "Imported from CCKM source key " + name_of_new_source_key
     },
     "allowedExternalAccounts": []
}
call_api("post", "rest/awsKeys/upload", data)

# Find AWS key by alias
aws_key = call_api("get", "rest/awsKeys/aliases?aliasName=%s" % aws_key_alias )
"""

resp = session.post(cckm_url + "logout")
```

cckm-google.py

```
import requests
import json
import sys

# For Google Cloud service account login.
# Replace with your host, email and password
cckm_url = "https://localhost:8443/kmaas/"
email = ""
password = ""

requests.packages.urllib3.disable_warnings()
session = requests.Session()
session.verify = False
resp = session.get(cckm_url)

session.headers = {'X-XSRF-TOKEN': resp.headers['X-XSRF-TOKEN']}
data = {'email': email, 'password': password}
resp = session.post(cckm_url + "auth/googleCloud", data=data)
if resp.status_code != 200:
       print resp.content
       sys.exit()

session.headers['Content-Type'] = 'application/json'

def call_api(method, api, data=None):
       if method == 'post':
               resp = session.post(cckm_url + api, data=json.dumps(data))
       elif method == 'put':
               resp = session.put(cckm_url + api, data=json.dumps(data))
       else:
               resp = session.get(cckm_url + api)
       print "%s %s" % (method, api)
       if data:
               print data
       print json.dumps(resp.json(), indent=4)
       print
       return resp.json()

# Following are example REST API calls, uncomment needed lines

#call_api("get", "me")
#call_api("post", "rest/googleCloudKeys/synchronize", {"skipKeyRingSynchronization": False})
#call_api("get", "rest/googleCloudKeys")

resp = session.post(cckm_url + "logout")
```

cckm-salesforce.py

```
import requests
import json
import sys

# Example REST API for Salesforce keys
# Replace with your Salesforce username and password
cckm_url = "https://host:8443/kmaas/"
username = ""
password = ""

requests.packages.urllib3.disable_warnings()
session = requests.Session()
session.verify = False
resp = session.get(cckm_url)

session.headers = {'X-XSRF-TOKEN': resp.headers['X-XSRF-TOKEN']}
data = {'username': username, 'password': password}
resp = session.post(cckm_url + "auth/salesforce", data=data)
if resp.status_code != 200:
     print resp.content
     sys.exit()

def call_api(method, api, data=None):
     if method == 'post':
        resp = session.post(cckm_url + api, data=json.dumps(data))
     elif method == 'put':
        resp = session.put(cckm_url + api, data=data)
     else:
        resp = session.get(cckm_url + api)
     print "%s %s" % (method, api)
     if data:
        print data
     print json.dumps(resp.json(), indent=4)
     print
     return resp.json()

session.headers['Content-Type'] = 'application/json'

# Example REST API calls, uncomment needed lines
#call_api("get", "me")
```

```
#call_api("post", "rest/salesforceKeys/synchronize")
#call_api("post", "rest/salesforceCertificates/download")
#call_api("get", "rest/salesforceKeys")

"""
# Generate a source key in KeySecure
name_of_new_source_key = "A-new-Salesforce-source-key4"
data = {"name": name_of_new_source_key, "algorithm": "AES256", "service": "Salesforce",
"description": "a new key for uploading to Salesforce", "cckmId": username, "provider":
"KeySecure"}
source_key = call_api("post", "rest/keyvaultkey", data)

# Upload the new source key to Salesforce
source_key_id = source_key['id']

# get a certificate for upload wrapping, assume the first certificate
certs = call_api("get", "rest/salesforceCertificates")
cert_label = certs['_embedded']['salesforceCertificates'][0]['name']

# Upload
data = {"certificateName": cert_label, "keyVaultKeyId": source_key_id, "keyDerivationMode":
"PBKDF2", "type": "Data"}
call_api("post", "rest/salesforceKeys/upload", data)
"""

resp = session.post(cckm_url + "logout")
```

Appendix E: Authenticate your REST API Client to CCKM REST APIs for Google Cloud Keys

This appendix shows how to authenticate your API client to the CCKM REST APIs using API calls before making calls to CCKM REST APIs to manage Google Cloud keys.

After submitting a POST call to authenticate to CCKM REST APIs, CCKM will first check your password hash against the password hash in its database. If it matches, CCKM will use the private key of the service account to obtain an access token from Google Cloud. Your REST API client is successfully authenticated to CCKM REST APIs when CCKM receives an access token from Google Cloud . At this point, you will receive a JSESSIONID cookie. Once you have this cookie, you can make calls to CCKM REST APIs to manage Google Cloud keys. You will need to add the JSESSIONID cookie and X-XSRF-TOKEN header to all subsequent REST API calls.

Refer to "cckm-google.py" on page 225 for a sample Python script that demonstrates how to authenticate and call CCKM REST API to manage Google Cloud keys.

To authenticate your API client to the CCKM REST APIs using API calls:

1. Make a GET call:

   ```
   GET https://<cckm hostname>:8443/kmaas
   ```

2. From the response header, copy the session x-xsrf-token.
3. Make a POST call (paste the session x-xsrf-token in the request header):

   ```
   POST https://<cckm hostname>:8443/kmaas/auth/googleCloud
   ```

   Request body:

   ```
   {
   "email": "you service account email",
   "password": "your service account password",
   }
   ```

   Request headers:

   ```
   Content-Type: application/x-www-form-urlencoded
   X-XSRF-TOKEN: <your session x-xsrf-token>
   ```

THALES

Contact us

For office locations and contact information,
visit cpl.thalesgroup.com/contact-us

> cpl.thalesgroup.com <