Amazon Web Services KMS External Key Store (XKS)

KeyControl Cloud Key Management Vault (HYOK) Integration Guide

2026-04-15

© Entrust Corporation. All rights reserved.

Table of Contents

1. Introduction . . . 1
    1.1. Product configuration . . . 1
    1.2. Requirements . . . 2
    1.3. Overview . . . 2
2. Procedures . . . 4
    2.1. Prerequisites . . . 4
    2.2. Adding an Elastic Load Balancer . . . 4
    2.3. Configure certificates and DNS . . . 11
    2.4. Key Administrators - AWS IAM user . . . 13
    2.5. Create a Cloud Key Management Vault . . . 14
    2.6. Create a CSP Account in the Cloud Key Management Vault . . . 18
    2.7. Create the Key Set . . . 19
    2.8. Create an External Key Store in AWS . . . 22
    2.9. Test the integration . . . 26
3. Additional resources and related products . . . 32
    3.1. nShield HSMs . . . 32
    3.2. nShield as a Service . . . 32
    3.3. Entrust products . . . 32
    3.4. nShield product documentation . . . 32

Chapter 1. Introduction

Entrust KeyControl has been rebranded as the Entrust Cryptographic Security Platform (CSP) Key Manager.

The Entrust CSP Key Manager continues to provide a comprehensive solution for discovering and managing the lifecycles of cryptographic keys, secrets, certificates, tokens, libraries, protocols, and configurations:

- The KeyControl Compliance Manager is now the Entrust CSP Compliance Manager. It still integrates with Entrust nShield Hardware Security Modules (HSMs) to protect the master keys for the CSP.
- KeyControl Vault is now the Entrust Cryptographic Security Platform Vault. The Cryptographic Security Platform Vaults also still integrate with Entrust nShield HSMs to provide an optional HSM root of trust.

Because the Entrust integrations are tested against specific product versions, this guide is still branded as a "KeyControl" integration. It was tested against a pre-CSP version of KeyControl.

Exercise caution when using an Entrust Integration Guide with a product version that does not match the tested version, because your version might not function in exactly the same way.

Entrust cannot guarantee the success of integrations in configurations other than those indicated in the guide. This guide remains on the website for customers using pre-CSP versions of KeyControl.

This guide describes the integration of the Entrust KeyControl Cloud Key Management Vault with Amazon Web Services KMS External Key Store (XKS).

Entrust KeyControl Cloud Key Management Vault provides an External Key Store Proxy inside KeyControl Vault. This feature allows the KeyControl Vault administrator to protect their data within Amazon Web Services (AWS) with 256-bit AES keys residing in KeyControl Vault. KeyControl Vault generates the keys and the keys are stored in KeyControl Vault only.

1.1. Product configuration

Entrust has successfully tested the following software version:

Product	Version	Certification
KeyControl Vault	10.1.1 10.2	FIPS 140-2 Level 1

1.2. Requirements

To integrate Entrust KeyControl Cloud Key Management Vault and Amazon Web Services KMS External Key Store (XKS), the server must be set up as follows.

- You must have an AWS account with KMS access allowed.
- There is a minimum of 2 KeyControl instances within a cluster. These instances must be accessible through a load balancer, typically realized using Elastic Load Balancer in the AWS environment.

This integration uses a public endpoint connectivity for AWS XKS. The following are required:

- Your external key store proxy must be reachable at a publicly routable endpoint.
- You must obtain a TLS certificate issued by a public certificate authority supported for external key stores. For a list, see https://github.com/aws/aws-kms-xksproxy-api-spec/blob/main/TrustedCertificateAuthorities.
- The subject common name (CN) on the TLS certificate must match the domain name in the proxy URI endpoint for the external key store proxy. For example, if the public endpoint is `https://myproxy.xks.example.com`, the TLS, the CN on the TLS certificate must be `myproxy.xks.example.com` or `*.xks.example.com`.
- Ensure that any firewalls between AWS KMS and the external key store proxy allow traffic to and from port 443 on the proxy. AWS KMS communicates on port 443 and this value is not configurable.

Familiarize yourself with:

- Entrust KeyControl and AWS External Key Store (XKS) Overview
- The Amazon Web Services KMS External Key Store (XKS) Documentation

1.3. Overview

Entrust KeyControl Cloud Key Management Vault provides an External Key Store Proxy

within KeyControl. This feature allows KeyControl administrators to safeguard their data within Amazon Web Services (AWS) using 256-bit AES keys housed in the KeyControl Vault. KeyControl generates the keys, which are exclusively stored in KeyControl.

In this guide:

- BYOK (Bring Your Own Key): This approach involves generating and managing encryption keys within an external key management system, such as the Amazon Web Services Key Management Service (AWS KMS). BYOK allows you to maintain control over your encryption keys while utilizing the services provided by AWS.
- HYOK (Hold Your Own Key): This method takes data security a step further by enabling you to retain absolute control over encryption keys, even when data is processed in cloud environments. With HYOK, the encryption keys are stored outside the cloud provider's infrastructure.

Entrust supports both BYOK and HYOK approaches to data security. This integration is HYOK implementation, ensuring that encryption keys are held within our control while still harnessing the benefits of cloud services.

For more information about the BYOK approach, refer to *Bring Your Own Key for AWS Key Management Service and Entrust KeyControl Integration Guide*.

Chapter 2. Procedures

Integration steps:

1. Prerequisites
2. Adding an Elastic Load Balancer
3. Configure certificates and DNS
4. Key Administrators - AWS IAM user
5. Create a Cloud Key Management Vault
6. Create a CSP Account in the Cloud Key Management Vault
7. Create the Key Set
8. Create an External Key Store in AWS
9. Test the integration

2.1. Prerequisites

Before integrating Entrust KeyControl Vault server and AWS External Key Store (XKS), ensure the following:

- Entrust KeyControl Vault server is deployed and configured. For details, see KeyControl Installation.
- Entrust KeyControl Compliance Manager is deployed and configured.

For this integration, the KeyControl Vault servers were deployed using AWS EC2 instances. To learn more about deploying KeyControl Vault in Amazon Web Services, refer to Creating KC Cluster AWS.

However, KeyControl Vault servers can also be deployed outside of AWS EC2, provided they fulfill the requirements outlined in [procedures:::requirements].

2.2. Adding an Elastic Load Balancer

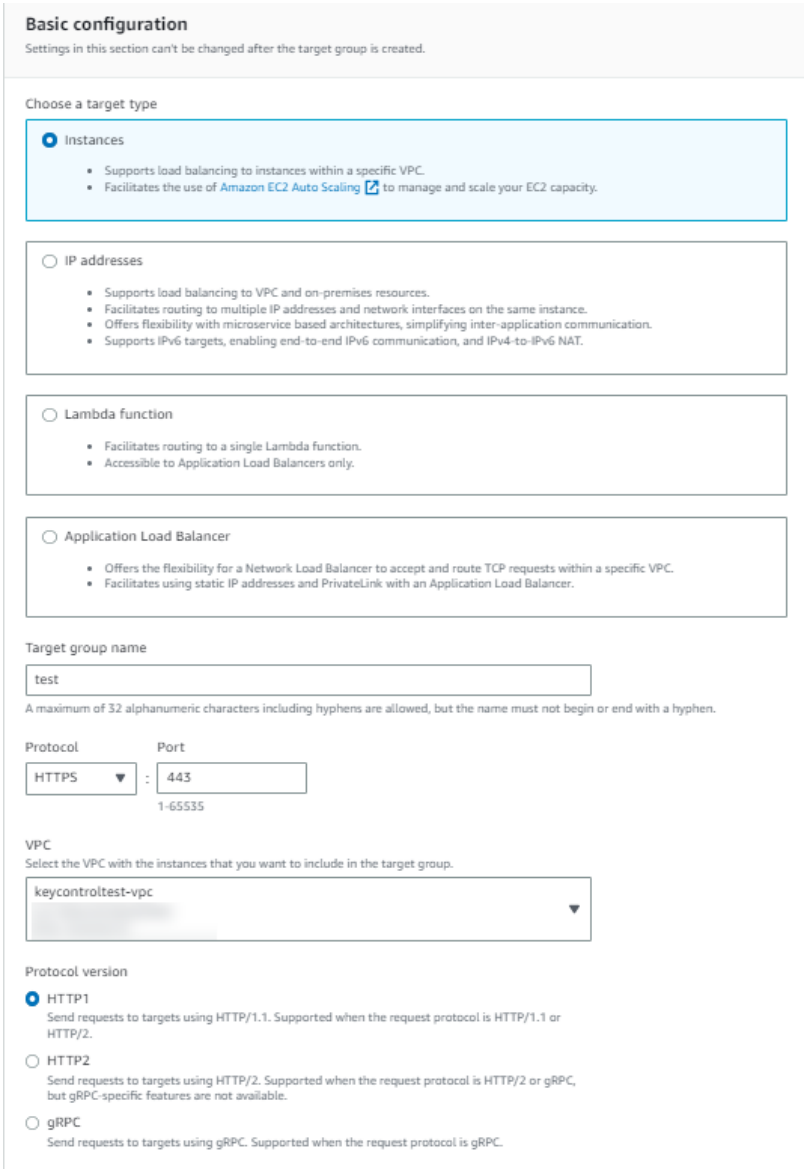
After cluster set-up is complete, you must use AWS elastic load balancing for the KeyControl load balancing.

For more information on AWS ELB, refer to AWS ELB Documentation.

2.2.1. Configure target group

To configure the target group:

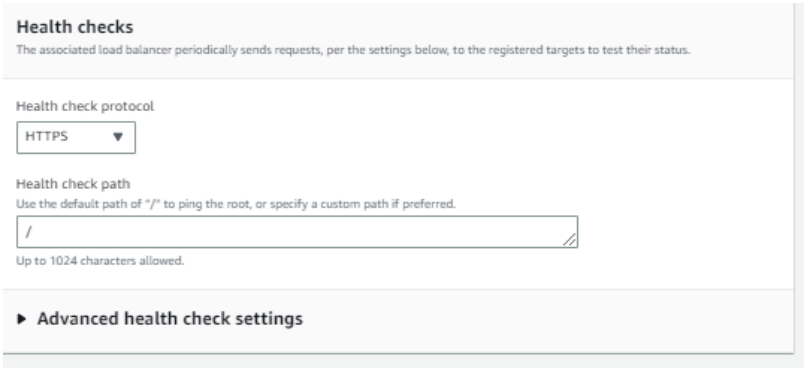
1. Sign in the Amazon EC2 console.
2. In the navigation pane, under Load Balancing, select **Target Groups**.
3. Select **Create target group**.
4. Under **Basic configuration**:
    a. Select **Instances** as target type.
    b. For **Target group name**, enter a name for the new target group.
    c. For **Protocol**, select **HTTPS**.
    d. For **Port**, select **443**.
    e. Select the VPC containing your instances.
    f. For **Protocol version**, retain the default.



5. Under **Health checks**:

    a. For **Health check protocol**, select **HTTPS**.

    b. Retain the default settings for other properties.

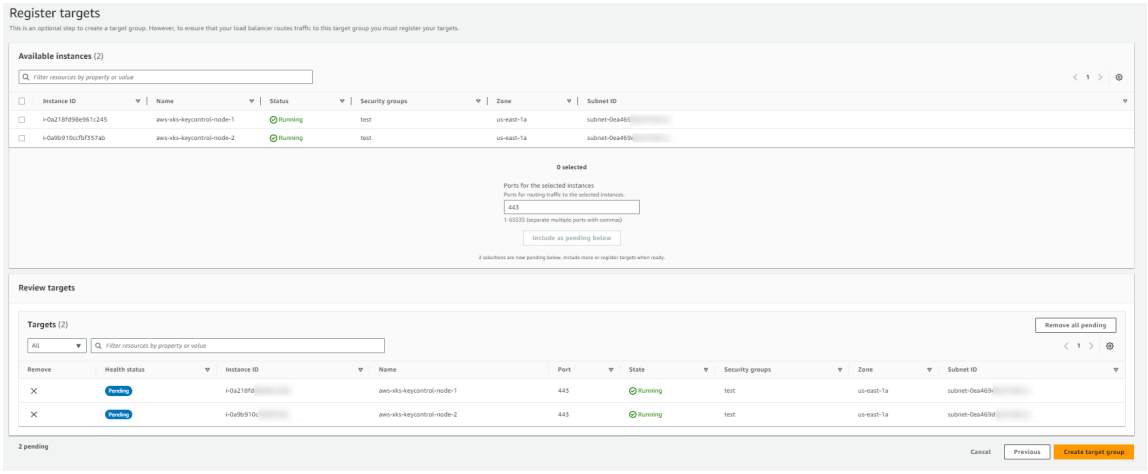


6. Select **Next**.
7. On the **Register Targets** page, complete the following steps. This is an optional step for creating the load balancer. However, you must register this target if you want to test your load balancer and ensure that it is routing traffic to this target.

    a. For **Available instances**, select the two KeyControl instances.

    b. For **Port for the selected instances**, enter **443**, and select **Include as pending** below.

    c. Select **Create target group**.



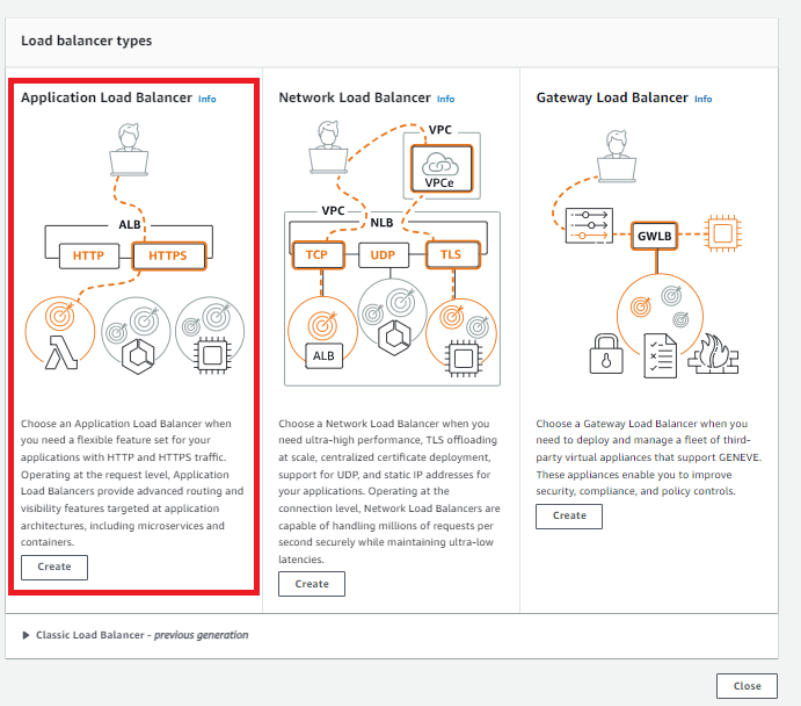
2.2.2. Create an Elastic Load Balancer

To create an Elastic Load Balancer:

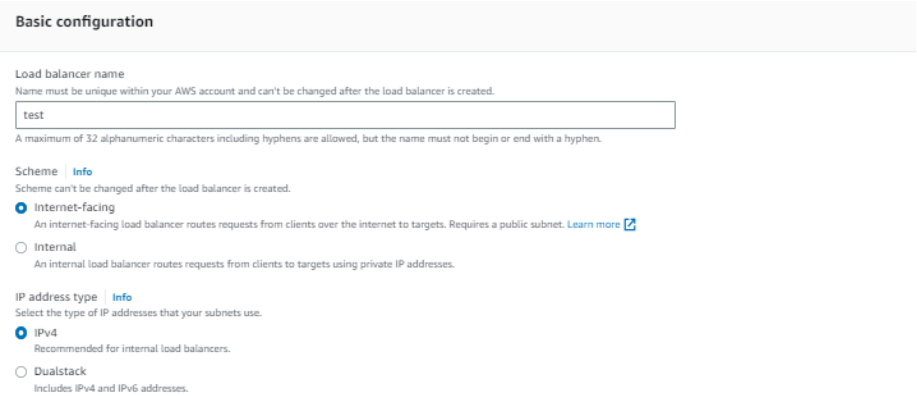
1. Sign in to the Amazon EC2 console.
2. On the navigation bar, select a region for your load balancer. You must select the same

region that you used for your EC2 instances.

3. In the navigation pane, under **Load Balancing**, select **Load Balancers**.
4. Select **Create Load Balancer**.
5. Select **Application Load Balancer**, select **Create**.



6. Under **Basic configuration**:
    a. For **Load balancer name**, enter a name for your load balancer.
    b. For **Scheme**, select **Internet-facing**.
    c. Retain the **IP address** type default.

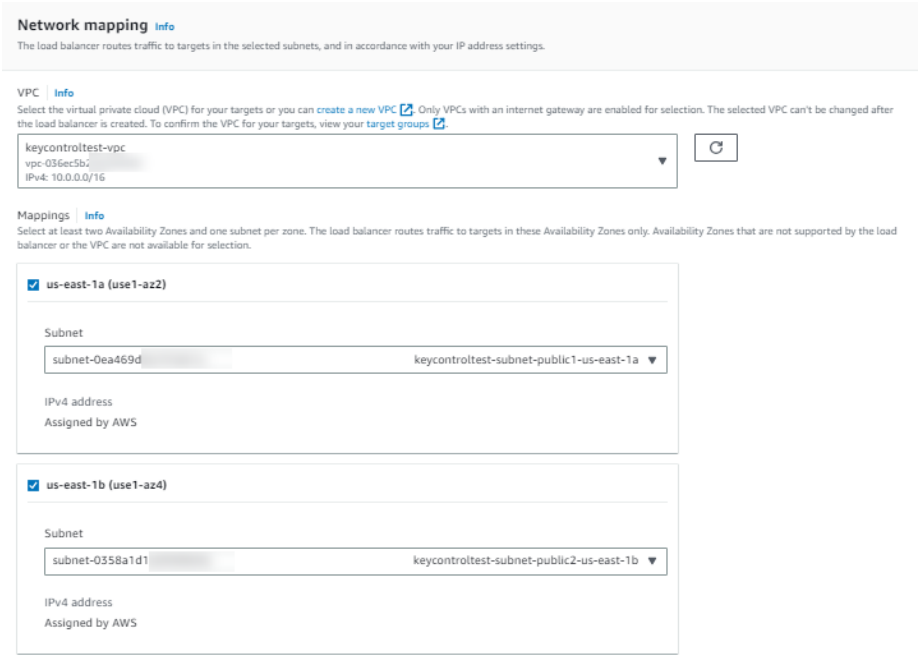


7. Under **Network mapping**:
    a. For **VPC**, select the VPC that you used for your EC2 instances.

b. For **Mappings**, select at least two Availability Zones and one subnet per zone.

c. For each Availability Zone that you used to launch your EC2 instances, select the Availability Zone and then select one public subnet for that Availability Zone.

d. You must select at least one Availability Zone that was used when launching your instances.

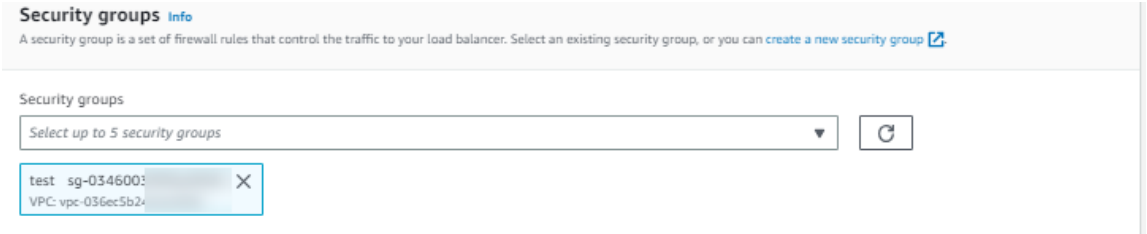


8. Under **Security groups**:

a. For **Security group**, select the default security group for the VPC that you selected in the previous step. Alternatively, you can select a different security group.

b. Ensure that the security group includes rules that allow the load balancer to communicate with registered targets on both the listener port and the health check port.

c. You must include the VPC source in the inbound rule to allow access to all ports or the port you are using as a listener.



9. Under **Listeners and routing**:

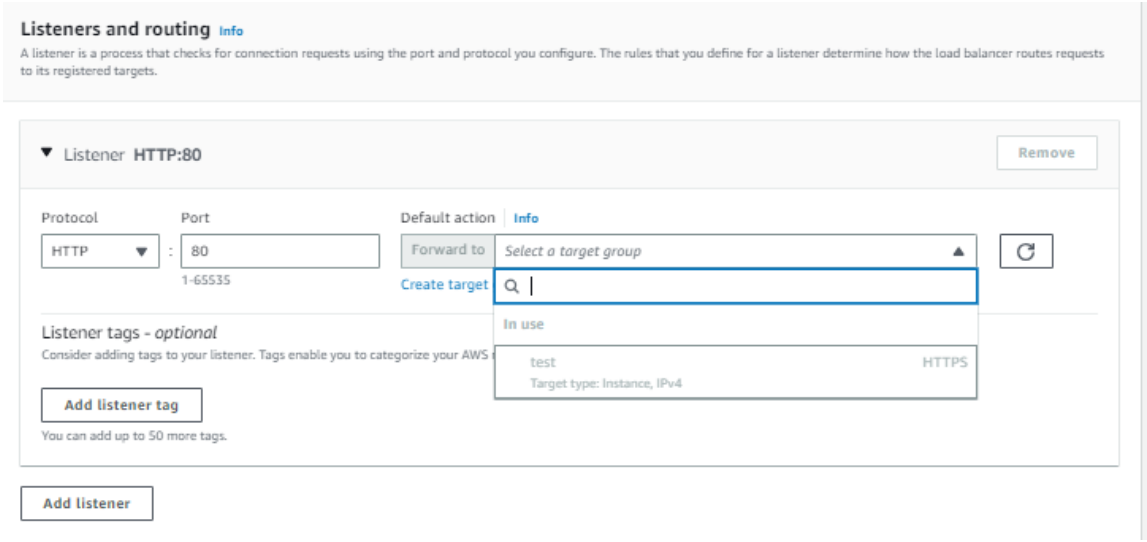
a. For **Protocol**, retain the default setting.

b. For **Port**, retain the default setting.

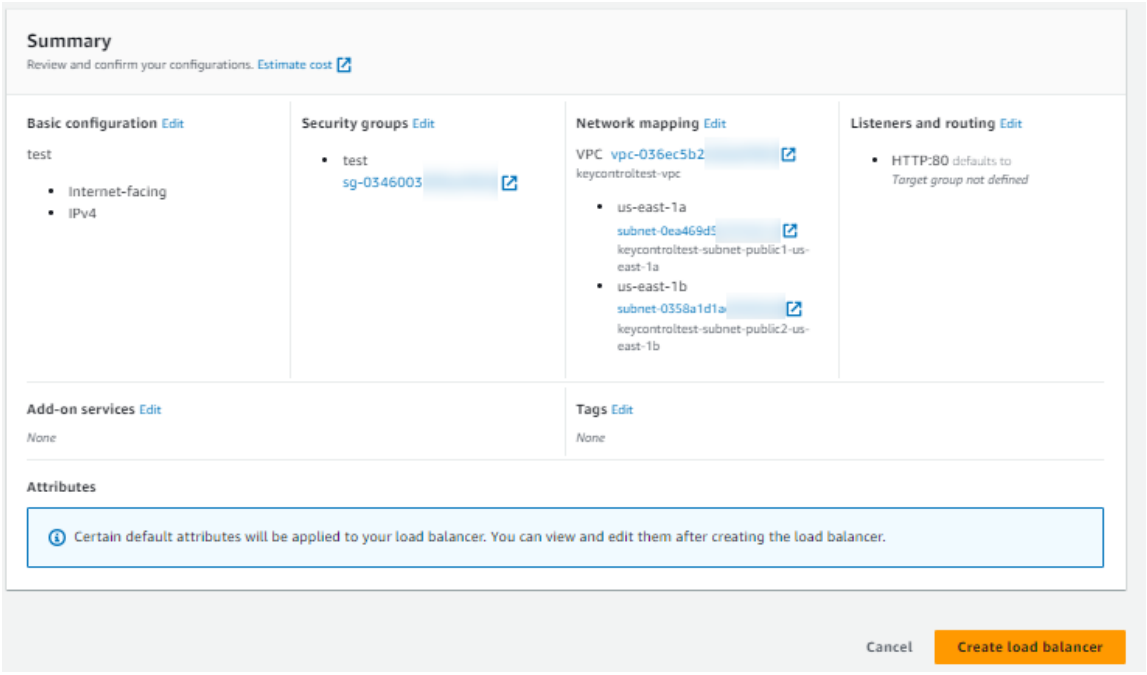
c. For **Default action**, select the **Forward to** action and select the target group that you created and registered.

d. Keep the **Add-on services** and **Load balancer** tags unchecked and left as default.

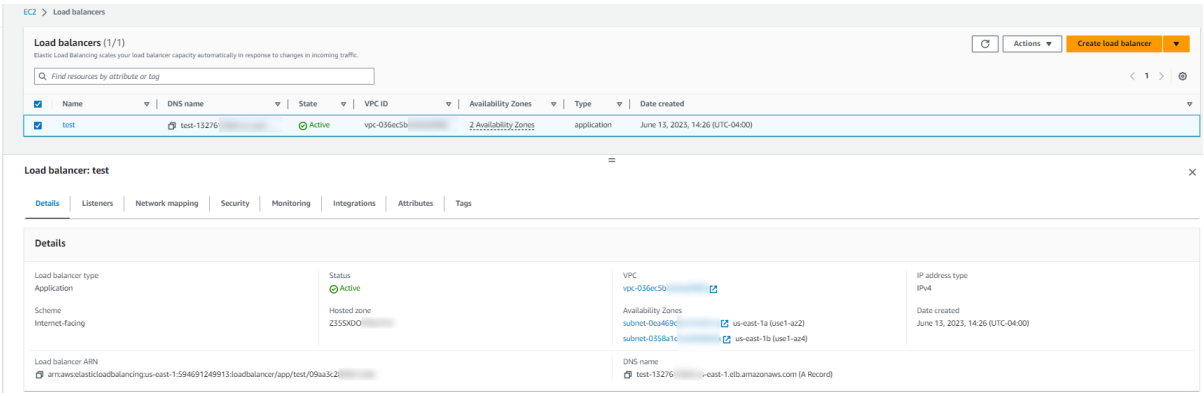
This configures a listener that accepts HTTP traffic on port 80 and forwards traffic to the selected target group by default.



10. Review your configuration and select **Create load balancer**. A few default attributes are applied to your load balancer during creation. You can view and edit them after creating the load balancer.
11. Select **Create load balancer**.

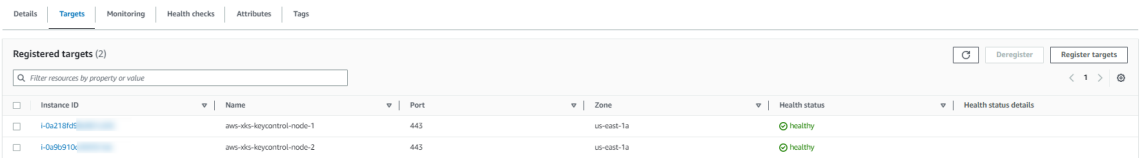


After you receive the notification confirming the successful creation of your load balancer, follow the steps below to verify the status of your instances and test the load balancer.



1. After you are notified that your load balancer was created successfully, select **Close**.
2. In the navigation pane, under **Load Balancing**, select **Target Groups**.
3. Select the newly created target group.
4. Select **Targets** and verify that your instances are ready.

   If the status of an instance is `Initial`, the instance is either in the process of being registered or has not passed the minimum number of health checks to be considered healthy. Wait until the status of at least one instance is `Healthy`. For example:



2.3. Configure certificates and DNS

For the successful set-up of AWS External Key Store (XKS), note that the DNS record and TLS certificate relate to the Fully Qualified Domain Name (FQDN) of the load-balanced endpoint utilized for accessing the service. This is distinct from the KeyControl instances.

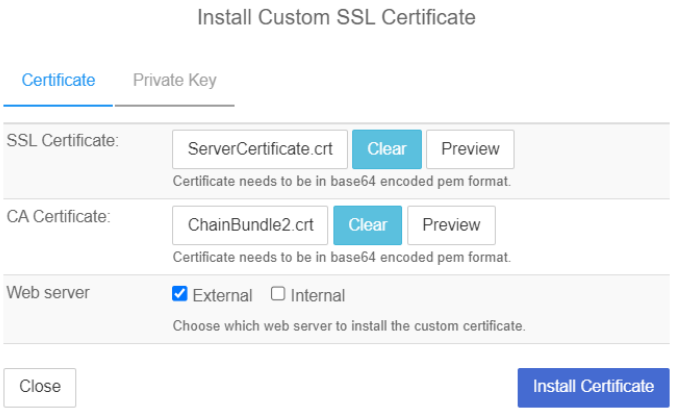
Ensure the KeyControl Vault server possesses a publicly accessible IP address and that a DNS record is in place for the designated common name within the public DNS server.

Amazon recommends a round-trip time latency of under 35 milliseconds between the AWS region and the KeyControl.

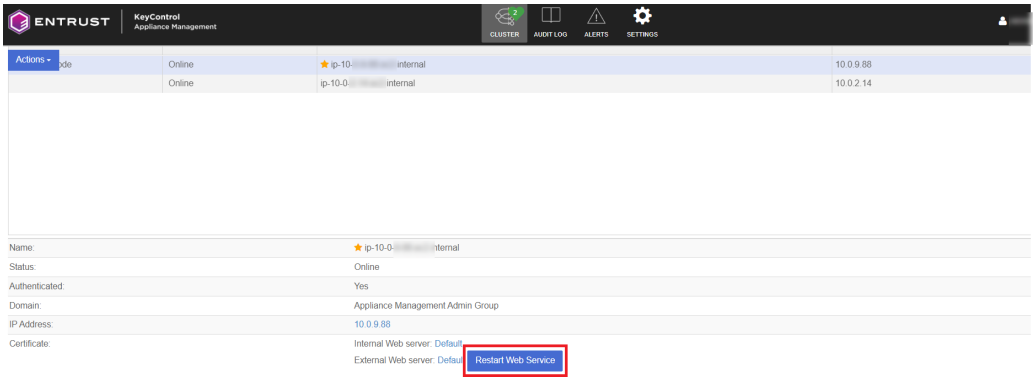
You must obtain a TLS certificate issued by a public certificate authority supported for external key stores. For a list, see https://github.com/aws/aws-kms-xksproxy-api-spec/blob/main/TrustedCertificateAuthorities

To ensure seamless access across the cluster nodes, you must install the TLS certificate on all nodes of the cluster, especially if users plan to access the cluster through other nodes. If an Elastic Load Balancer (ELB) is part of the set-up, the certificate handling process will differ. When using an ELB, the TLS certificate must be managed according to ELB requirements.

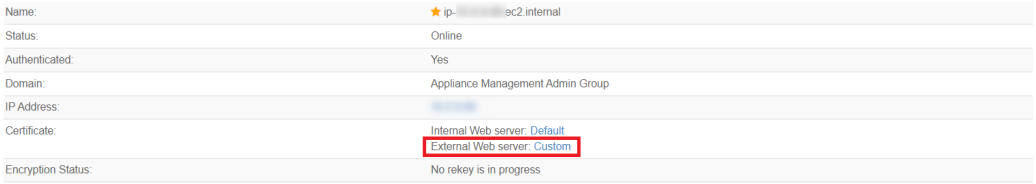
1. In the KeyControl Appliance Management:
    a. Navigate to **Cluster** > **Servers**.
    b. Select the server to install the certificate.
    c. Select **Actions** > **Install Certificate**. The **Install Custom SSL Certificate** dialog appears.
    d. Locate and select the **SSL Certificate** file.
    e. Locate and select the **CA certificate** chain file.
    f. Select **External** for the Web server.
    g. Select **Install Certificate**.



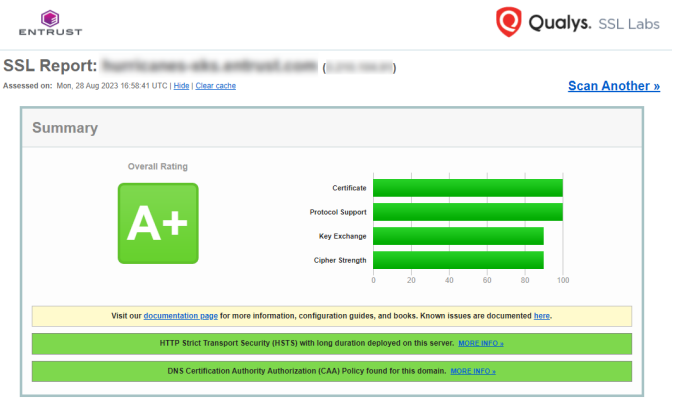
2. After installation, restart the Web service



3. Confirm the installation. The External Web server will now show as `Custom` for the certificate.



4. You can validate the certificate using https://entrust.ssllabs.com/ or a similar tool. For example:



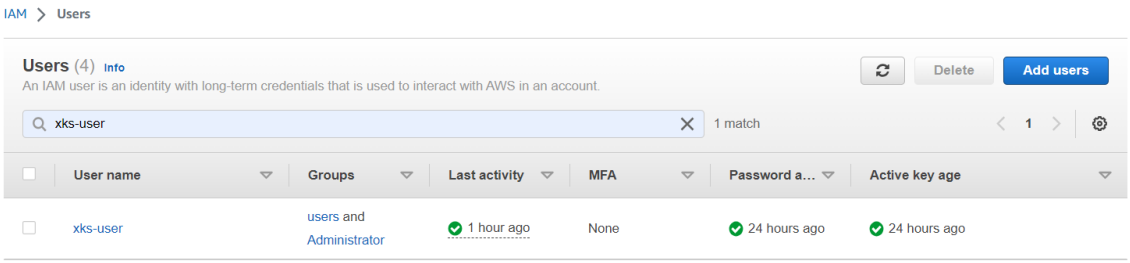
If you are not able to verify the server hostname, ensure that any firewalls between AWS KMS and the external key store proxy allow traffic to and from port 443 on the proxy.

2.4. Key Administrators - AWS IAM user

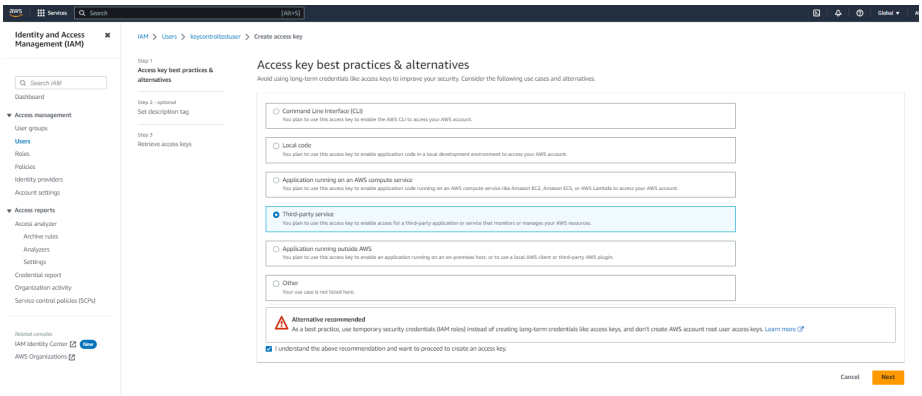
To enable the integration, you must designate an IAM user as a Key Administrator. This user is required to generate an access key that will be used in a later step.

This user must have permissions to manage and use the KMS key for cryptographic operations.

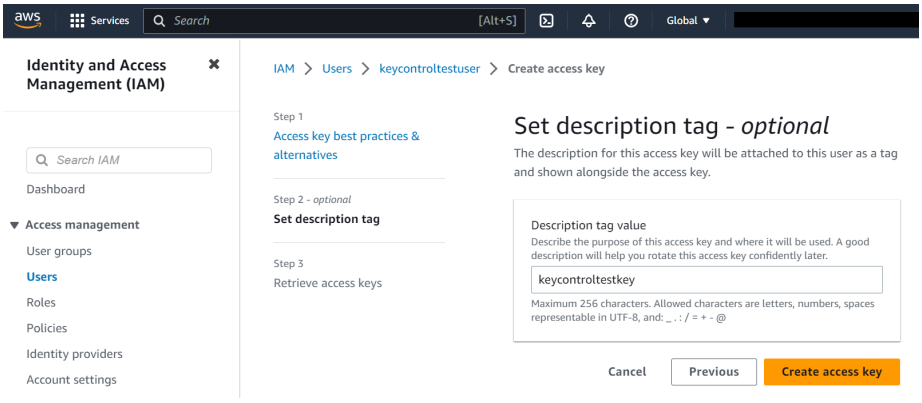
1. Sign in to the AWS Management Console.
2. Search for the **Identity and Access Management (IAM)** service and select it.
3. In the IAM console, select **Access Management** in the left tab and then select **Users**.
4. Create a new user or use an existing user to generate an access key. In this example integration, a new user named `xks-user` is created as the Key Administrator.



5. In the user settings, select **Create access key** and select **third-party service**.



6. Create the access key.



Ensure that you securely store the Access ID and Secret Access Key, as they are required for accessing and managing your AWS resources.

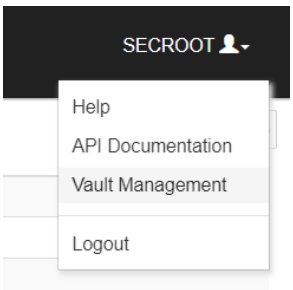
2.5. Create a Cloud Key Management Vault

The KeyControl Vault appliance supports the following types of vaults:

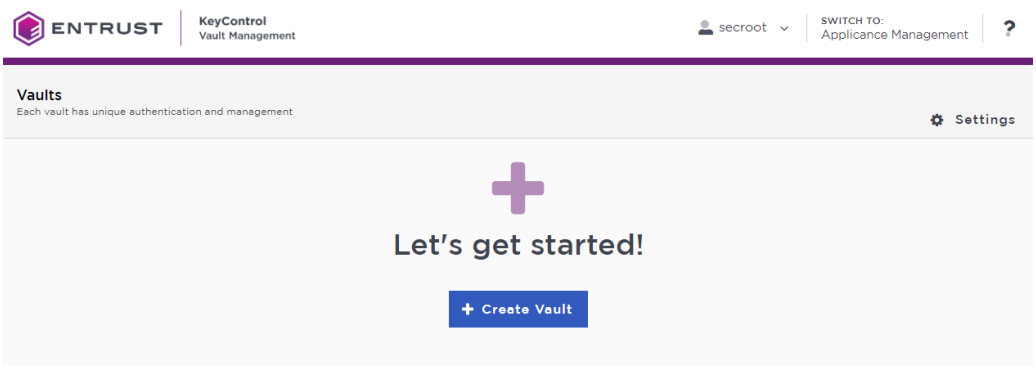
- **Cloud Key Management** - Vault for cloud keys such as BYOK and HYOK.
- **KMIP Vault** - Vault for KMIP Objects.
- **PASM** - Vault for objects such as passwords, files, SSH keys, and so on.
- **Database** - Vault for database keys.
- **Tokenization** - Vault for tokenization policies.
- **VM Encryption** - Vault for encrypting VMs.

To create a Cloud Key Management Vault:

1. Sign in to the KeyControl Vault Server Appliance Manager.
2. Open the drop-down menu and select **Vault Management**.



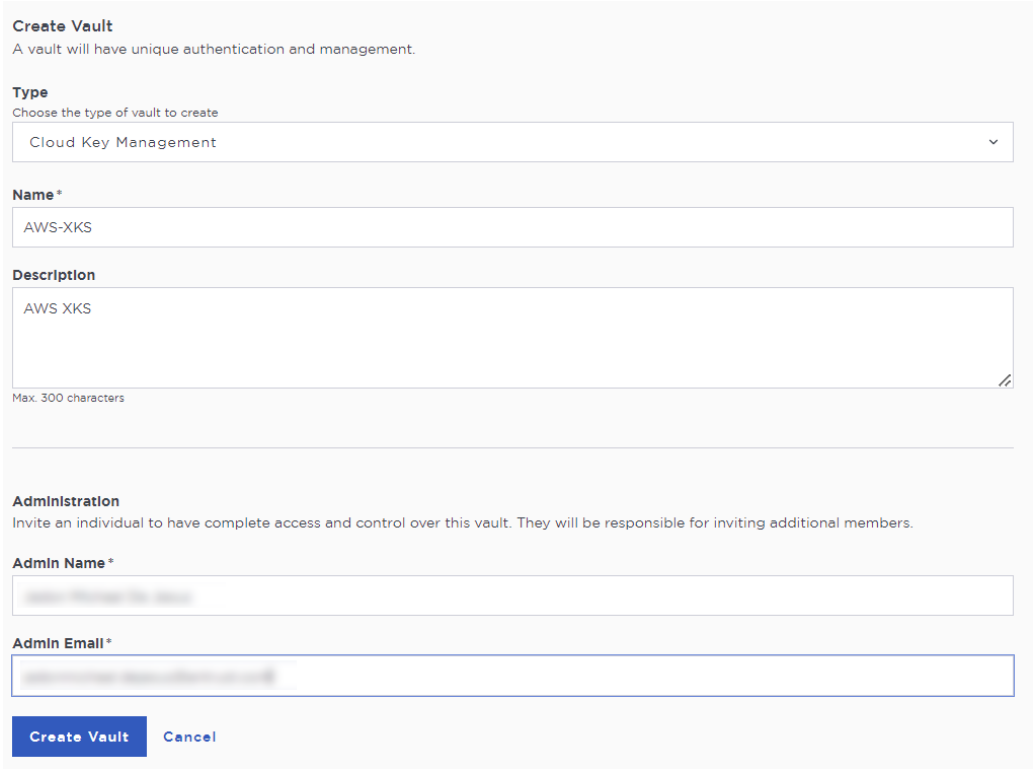
The KeyControl Vault Management interface appears.



3. Select **Create Vault**.

   The **Create Vault** page appears.

4. On the **Create Vault** page:
   a. For **Type**, select **Cloud Key Management**.
   b. Enter a **Name** for the vault.
   c. Provide a **Description** for the vault.
5. Under **Administration**:
   a. Enter the **Admin Name** who will be responsible for the vault.
   b. Enter a valid **Admin Email** address.



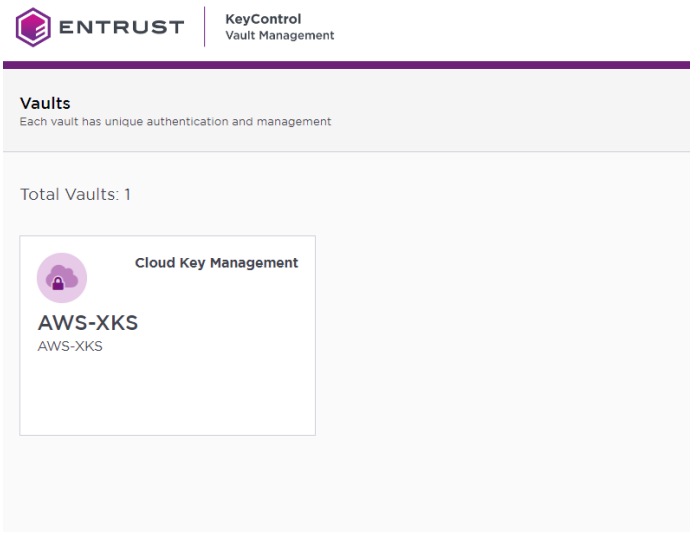
6. Select **Create Vault**.

If you set up an administrator email address when you logged in for the first time, a temporary password is mailed to that address. This is the password you must use when you sign in for the first time to Vaults space in KeyControl.

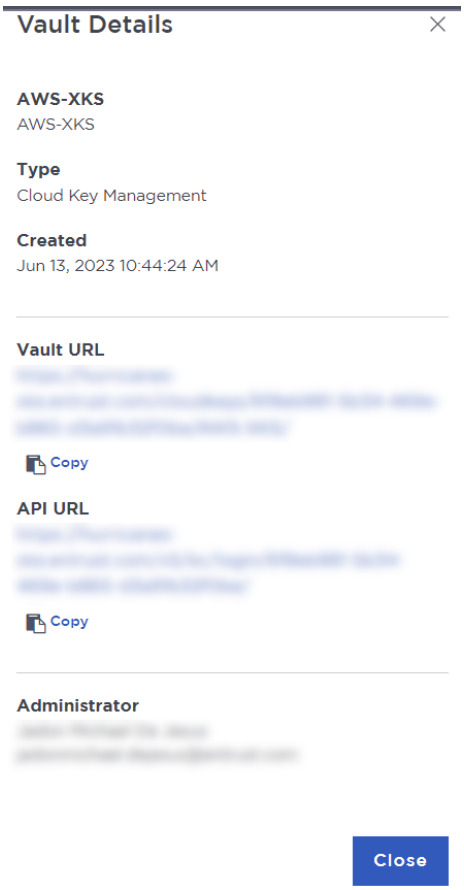
If you did not set up an email configuration when you logged in for the first time, a password is shown in the Vault Details when you create a Vault for the first time. You must make a note of the password at this time, as it will not be included in the Vault Details afterwards.

7. Select **Close**.

   The newly created vault is displayed in the **Vaults** dashboard.

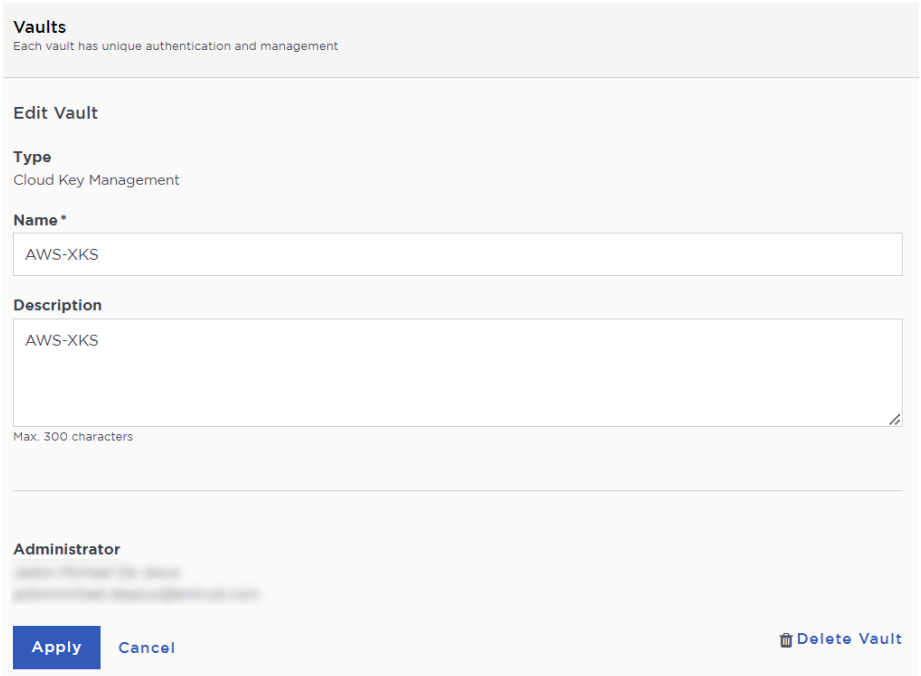


8. To view the details of a vault, hover over the vault and select **View Details**.



To edit the details of a vault:

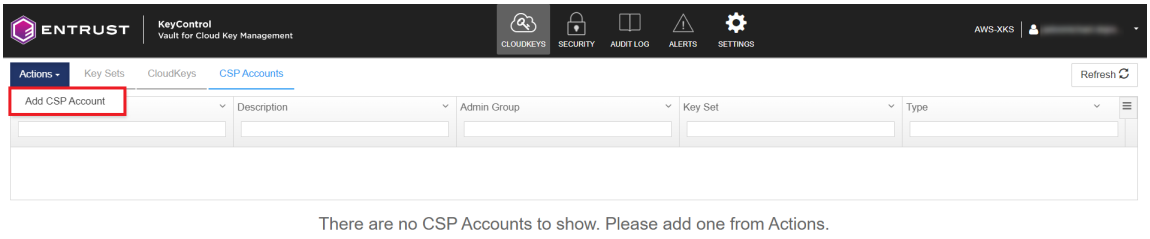
1. Hover over the vault and select **Edit**.
2. Make the required changes and select **Apply**.



2.6. Create a CSP Account in the Cloud Key Management Vault

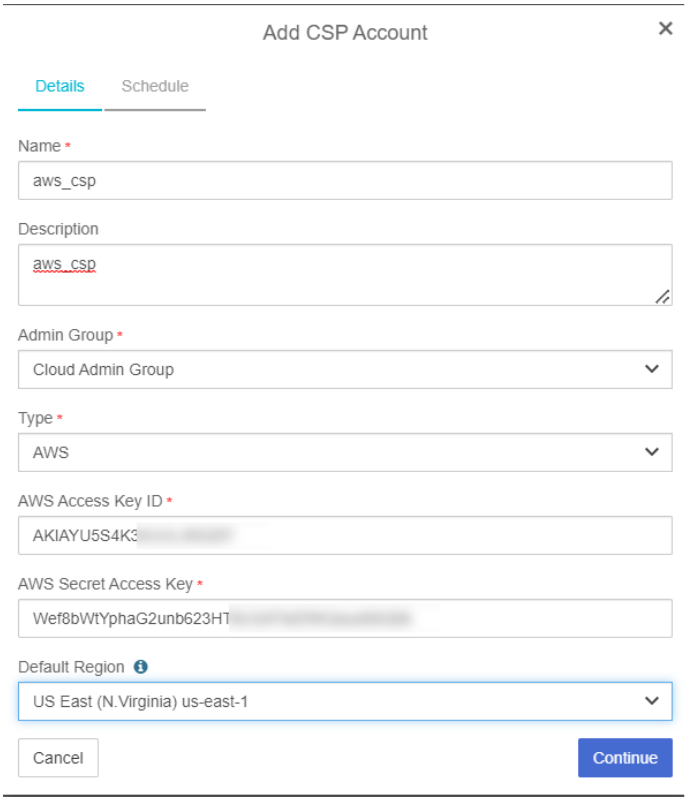
To create a CSP Account in the Cloud Key Management Vault:

1. Sign into the newly created vault.
2. Select **Cloud Keys** > **CSP Accounts** > **Actions** > **Add CSP Account**.



The **Add CSP Account** dialog appears.

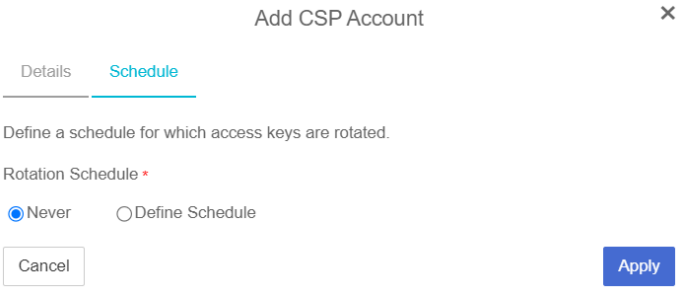
3. In the **Details** page:
   a. For **Name**, enter a name for the CSP account.
   b. Add a **Description**.
   c. For **Admin Group** select **Cloud Admin Group**.
   d. For **Type** select **AWS**.
   e. Enter the **AWS Access Key ID** and **AWS Secret Access Key** from earlier.
   f. Select the target region as the default region.
   g. Select **Continue**.



4. In the **Schedule** page:

    a. Select the required **Rotation Schedule**.

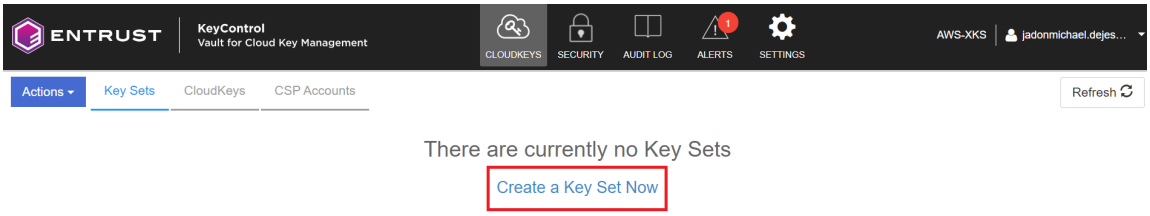
    b. Select **Apply**.



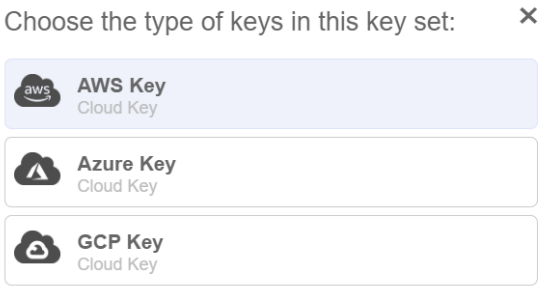
2.7. Create the Key Set

To create the Key Set:

1. Under **CloudKeys**, select **Key Sets** > **Create a Key Set Now**.



2. Select **AWS Key** for the type of keys in key set.



The **Create Key Set** dialog appears.

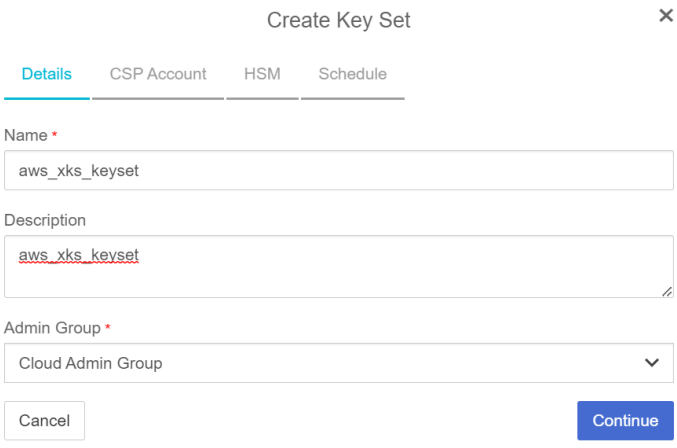
3. In the **Details** page:

   a. Enter a **Name**.

   b. Enter a **Description**.

   c. For **Admin Group**, select **Cloud Admin Group**.

   d. Select **Continue**.



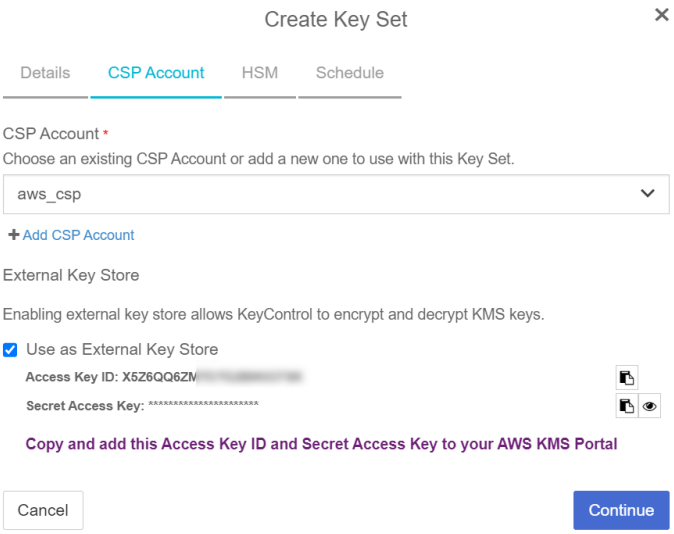
4. In the **CSP Account** page:

   a. For **CSP Account**, select the `aws_csp` account created earlier.

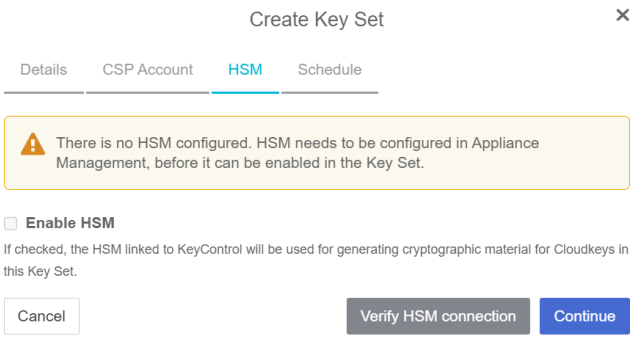
   b. Select **Use as External Key Store**.

   c. Make a note of the XKS credentials, as these are required later.

   d. Select **Continue**.

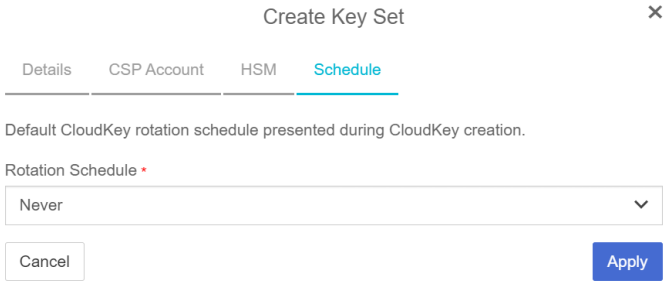


5. In the **HSM** page:
    a. Optionally select **Enable HSM**.
    b. Select **Continue**.



To set up an HSM linked to KeyControl, follow the installation and set-up instructions in the *Entrust KeyControl nShield HSM Integration Guide*.

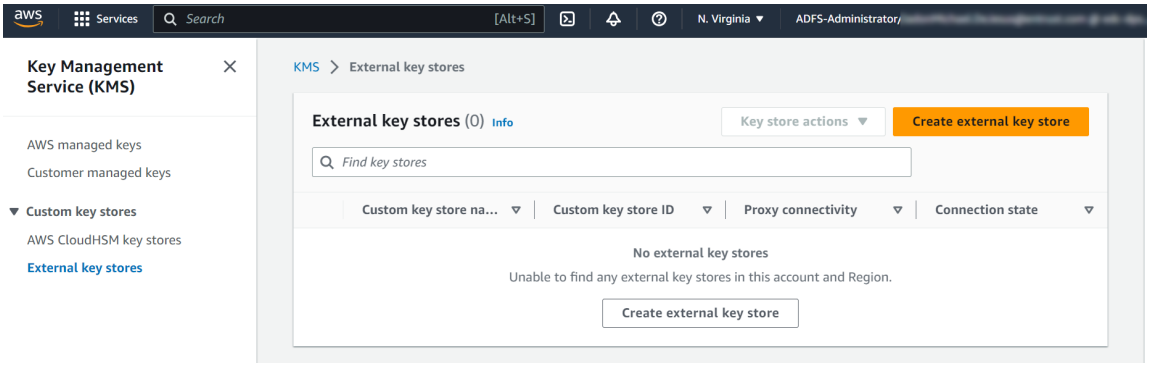
6. In the **Schedule** page:
    a. For **Rotation Schedule**, select your required CloudKey rotation.
    b. Select **Apply**.



2.8. Create an External Key Store in AWS

To create an External Key Store in AWS:

1. Sign in to the AWS console and navigate to **Key Management Service (KMS)**.
2. In the left panel, select **Custom key stores** > **External key stores**.
3. For **Key store name**, enter the required name.
4. Select **Create external key store**.



The **Create external key store** page appears.

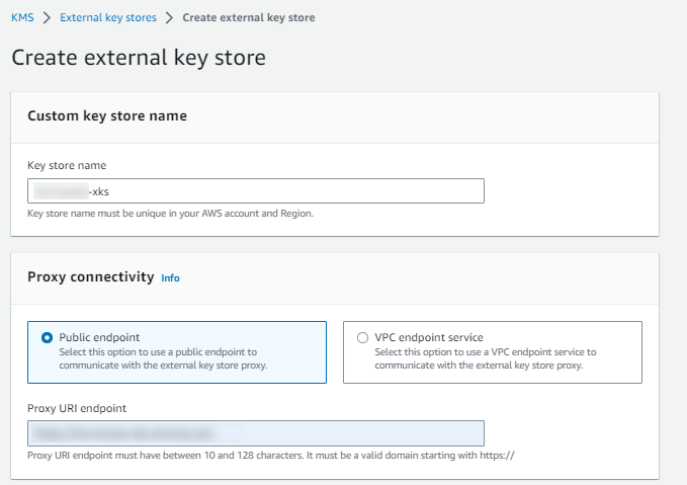
5. Under **Custom key store name**, provide a descriptive name for the external key store.
6. Under **Proxy connectivity**:

    a. Select **Public endpoint**.

    b. For **Proxy URI endpoint**, enter the Proxy URI endpoint in the following format:

    ```
    https://<FQDN of Load Balanced Endpoint>
    ```

    Substitute `<FQDN of Load Balanced Endpoint>` with the fully qualified domain name of the load-balanced endpoint utilized for accessing the service, distinct from any of the KeyControl instances.



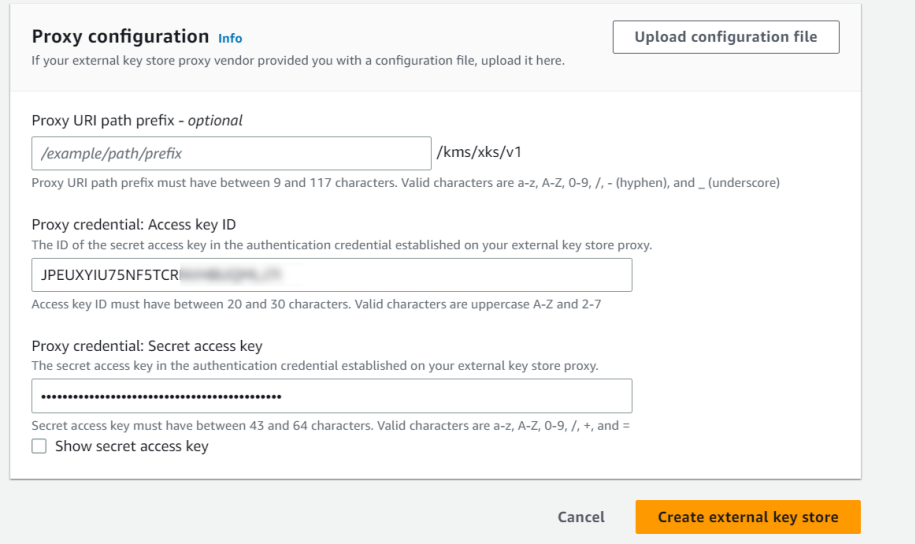
7. Under **Proxy configuration**:

   a. Leave **Proxy URI path prefix** empty.

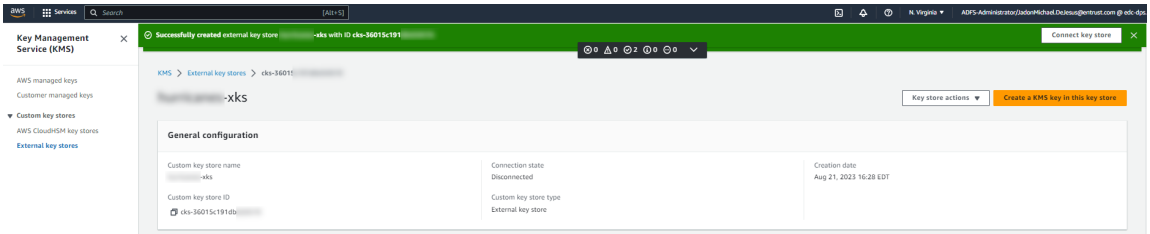
   b. For **Proxy credential: Access key ID**, enter the previously-saved proxy access key ID.

   c. For **Proxy credential: Secret access key**, enter the previously-saved proxy secret access key.

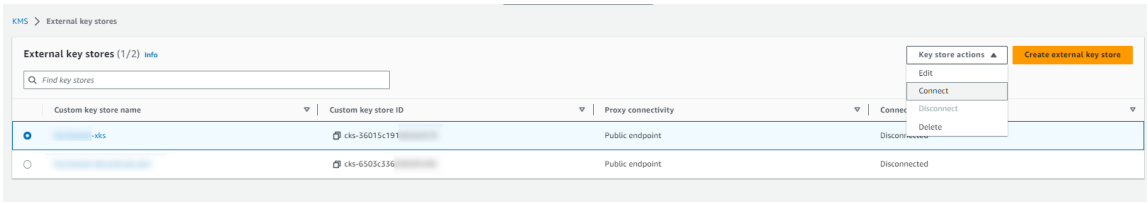
   d. Select **Create external key store**.



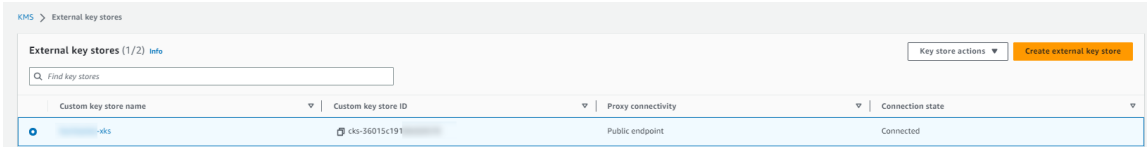
A details page for the new external key store appears.



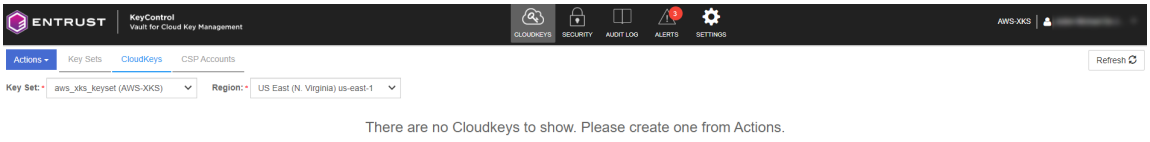
8. Select **External key stores** to view all external key stores.
9. Select **Key store actions** > **Connect** to connect to the external key store.



10. Wait for the Connection state to display as `Connected`.



11. Return to KeyControl Cloud Key Management Vault and select **CLOUDKEYS** > **CloudKeys**.
12. Select the **Key Set** created earlier along with the **Region**.

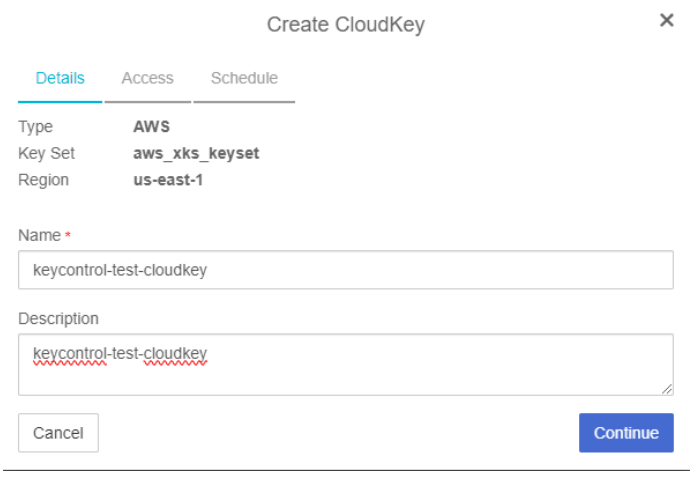


13. Select **Actions** > **Create CloudKey**.

    The **Create CloudKey** dialog appears.

14. In the **Details** page:

    a. For **Name**, enter a name for the CloudKey.
    b. Enter a **Description**.
    c. Select **Continue**



15. In the **Access** page:

    a. For **Administrators**, select AWS IAM users who will have administrative rights.

    b. For **Users**, select AWS IAM users who will be able to use the key to encrypt/decrypt.

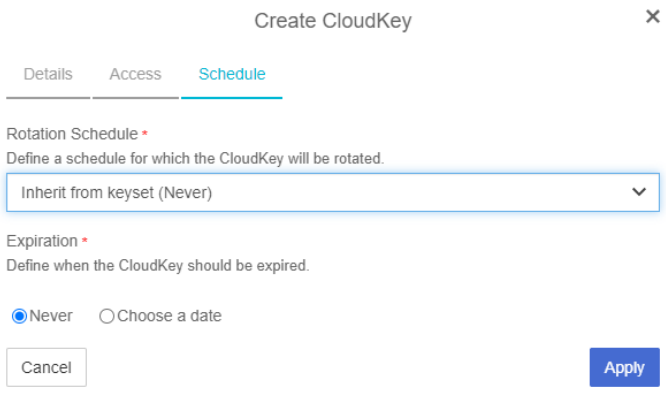
    c. Select **Continue**.

16. In the **Schedule** page:

    a. For **Rotation Schedule**, select a rotation schedule for the CloudKey.

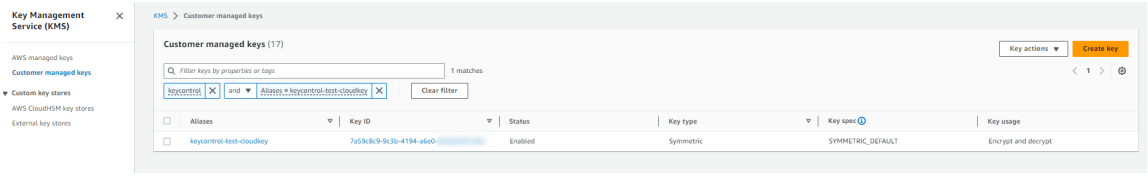
    b. For **Expiration**, select the required condition.

    c. Select **Apply** to finish the process.



After the XKS CloudKey is created in KeyControl, a KMS key pointer is automatically created in AWS KMS with a key alias that matches the KeyControl CloudKey name. This KMS key pointer can be utilized by AWS services to encrypt or decrypt user objects.

17. Return to AWS KMS > **Customer managed keys** to find the created CloudKey.



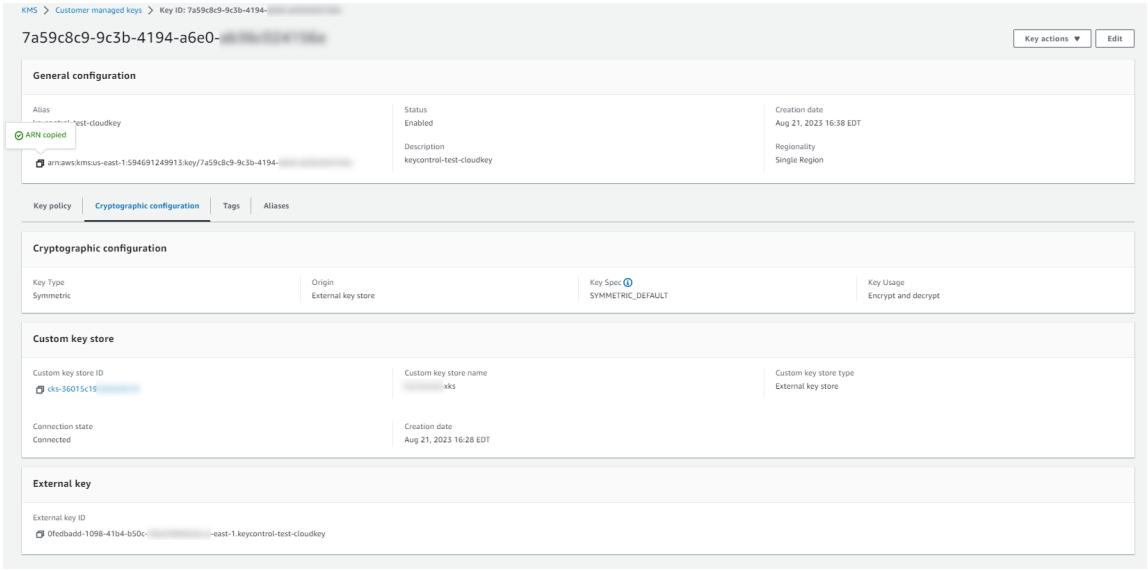
18. Select either the **Aliases** or **Key ID** hyperlink for the CloudKey.

    A details page for the CloudKey appears.

19. Select **Cryptographic configuration**.

    Note that under **Custom key store**, the **Custom key store name** appears and the **Custom key store type** as listed as **External**.

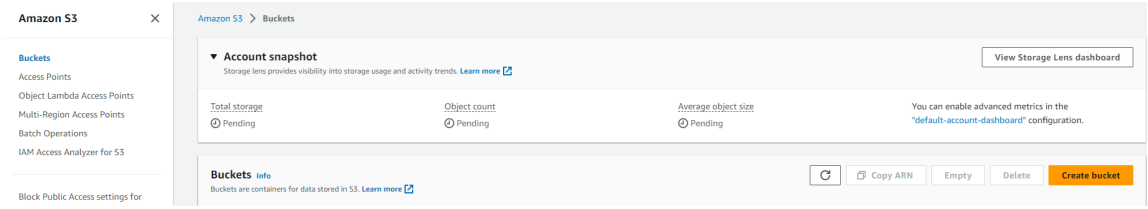
20. Under **General configuration**, copy the AWS KMS ARN for a later step.



2.9. Test the integration

To test the integration:

1. Sign in to the AWS Console and access **S3** services.
2. From the left panel, select **Buckets** and then select **Create bucket**.

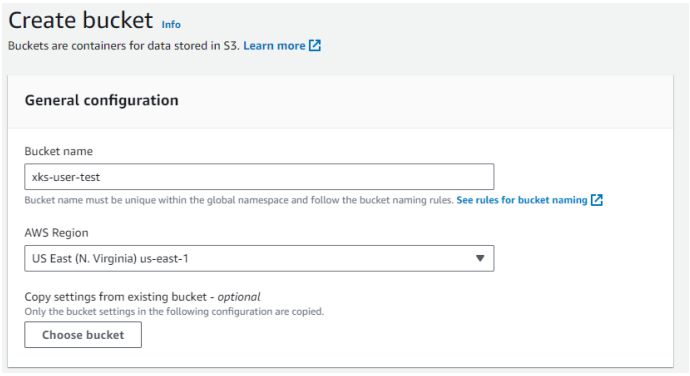


The **Create bucket** page appears.

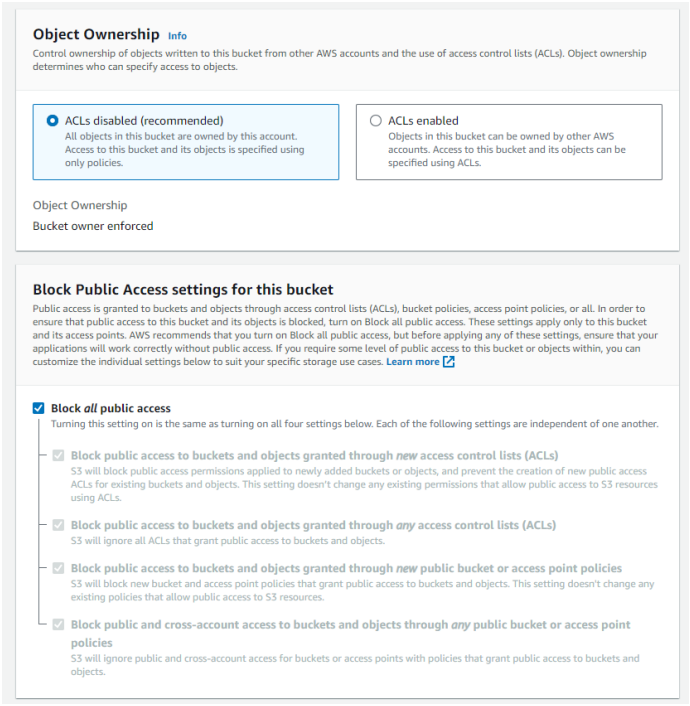
3. Under **General configuration**:

    a. For **Bucket name**, enter the required name for the bucket.

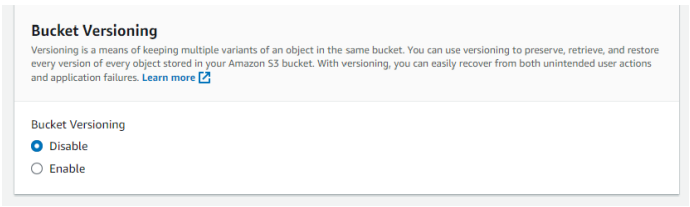
    b. Select an appropriate **AWS Region**.



4. Under **Object Ownership**, select **ACLs disabled**.



5. Under **Bucket Versioning**, set **Bucket Versioning** to **Disable**.



6. Under **Default encryption**:

a. For **Encryption type**, select **Server-side encryption with AWS Key Management Service keys (SSE-KMS)**.

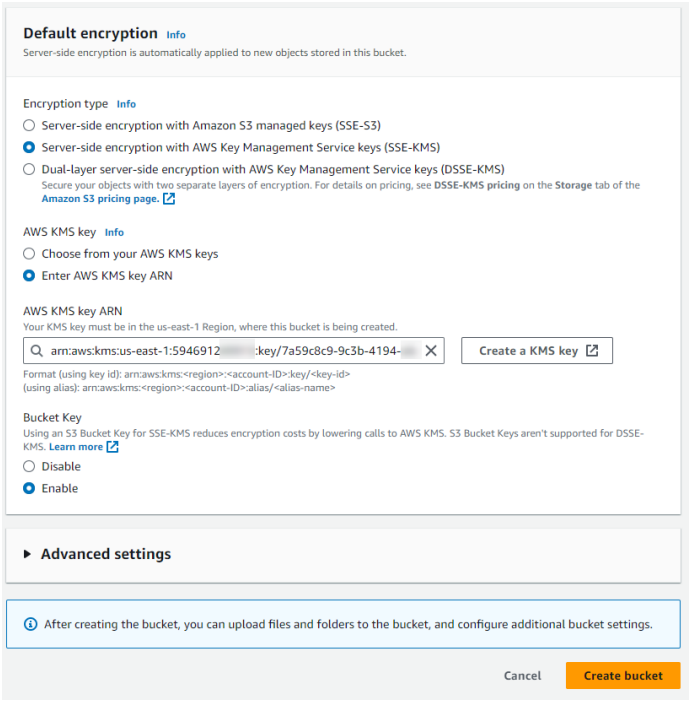
b. For **AWS KMS key**:

   i. Select **Enter AWS KMS key ARN**.

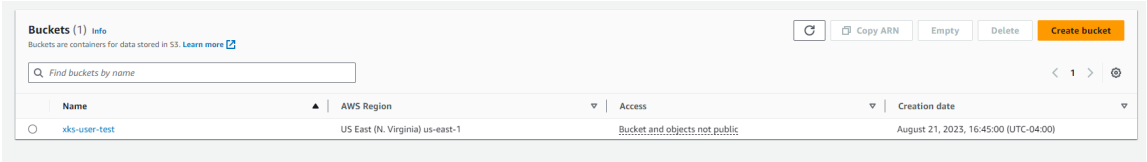
   ii. Paste the AWS KMS ARN from the previously created CloudKey.

c. For **Bucket Key**, select **Enable**.

d. Select **Create bucket** to complete the process.



The bucket is created.

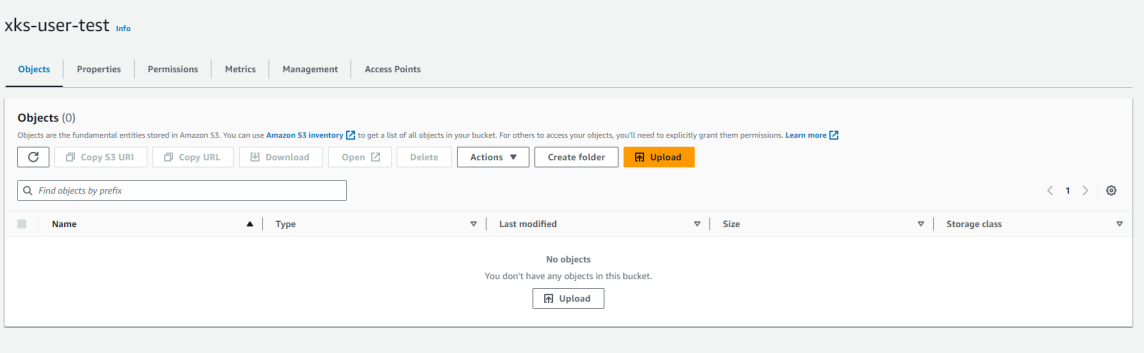


7. Select the hyperlink for the bucket.

   A details page for the bucket appears.

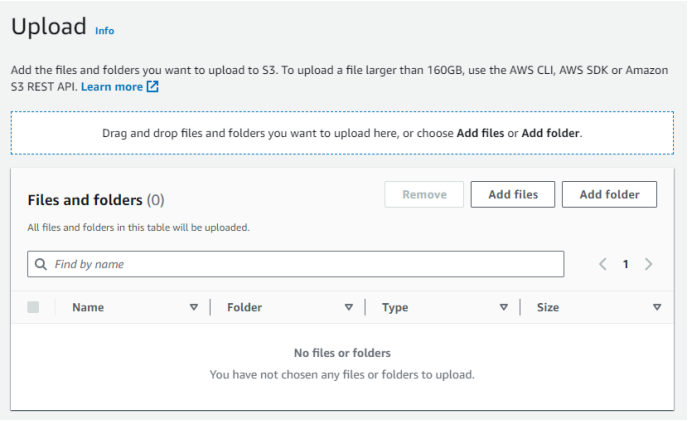
8. Select **Objects**.

9. To test the encryption, select **Upload**.



The **Upload** dialog appears.

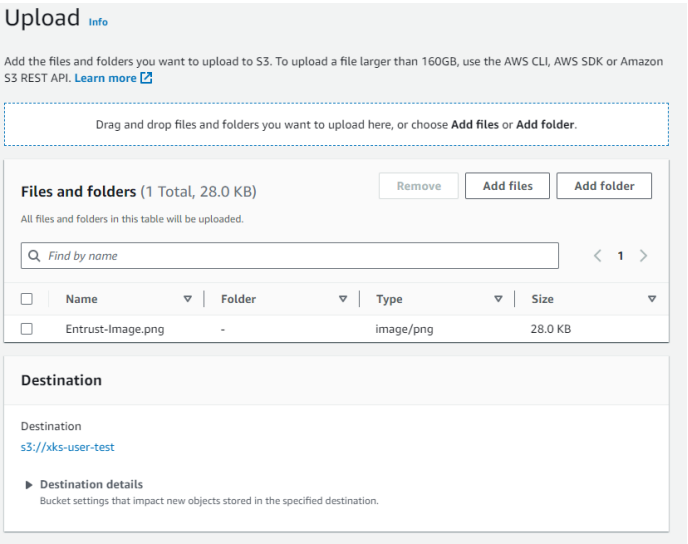
10. Select **Add files**.



11. Locate and select an image to upload.

    The file is added to the list of available images.

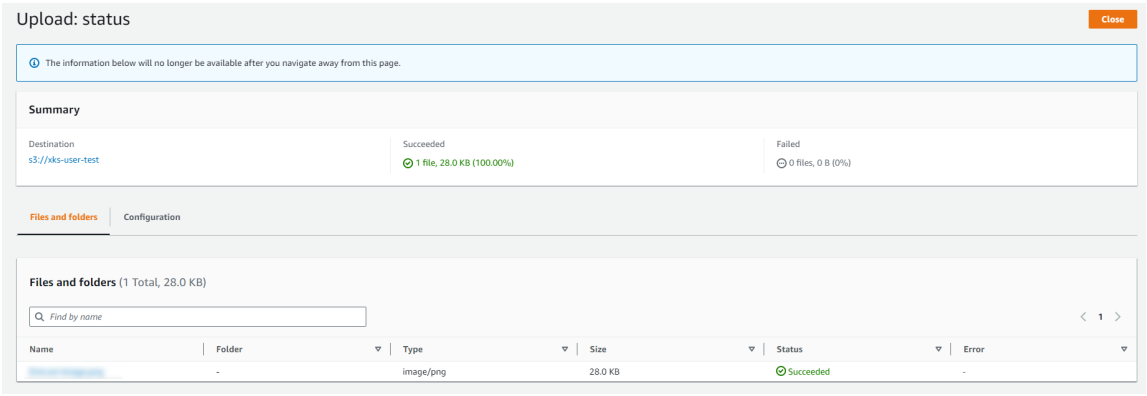
12. Select the check box for the image file and select **Upload**.



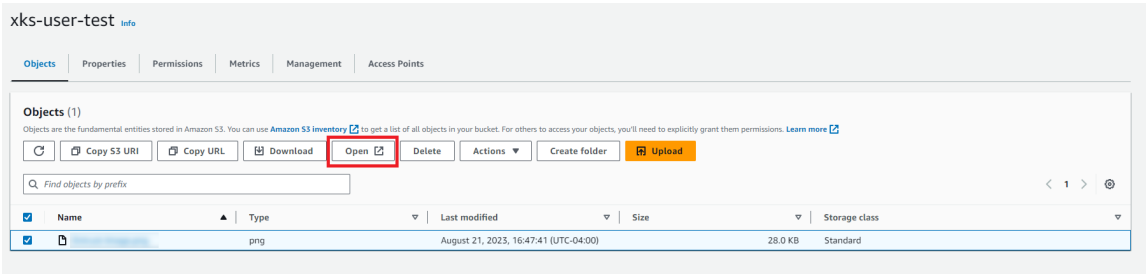
In this example, the `Entrust-Image.png` file was added and can be selected and

uploaded.

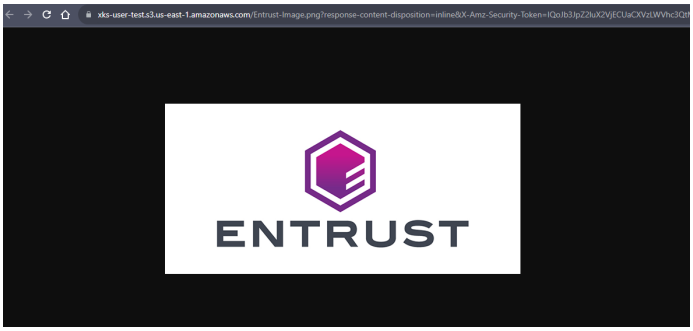
The newly uploaded image is listed within the bucket.



13. Select the new image and select **Open** to view it.

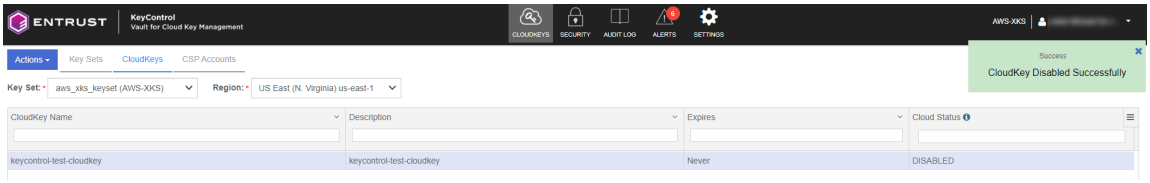


The image starts in a browser window.



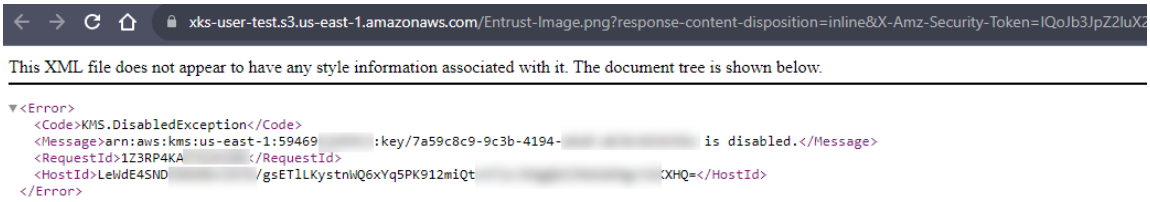
14. Return to KeyControl Cloud Key Management Vault and select **CLOUDKEYS** > **CloudKeys**.
15. Select the CloudKey and then select **Actions** > **Disable CloudKey**.

The CloudKey is disabled.

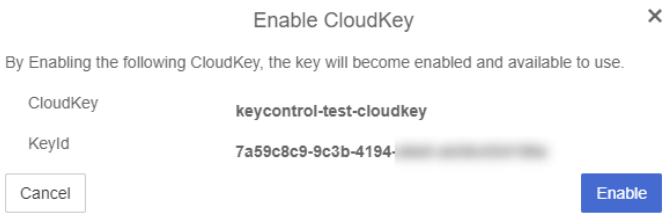


16. Return to the AWS S3 bucket and attempt to open the uploaded image.

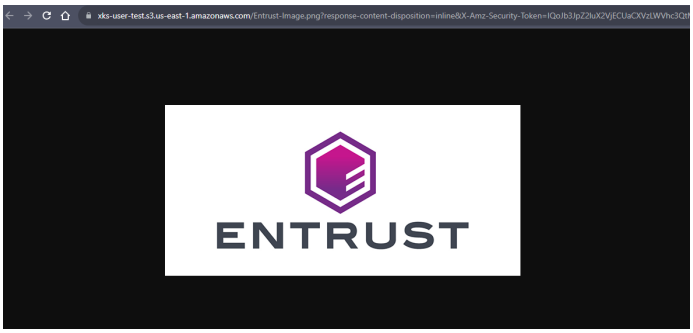
    The image is not viewable as the CloudKey was disabled.



17. Re-enable the **CloudKey** in the KeyControl CloudKey Management Vault.



18. Return to the AWS S3 bucket.
19. Open the uploaded image again. It is now viewable.



This concludes the integration process.

Chapter 3. Additional resources and related products

3.1. nShield HSMs

3.2. nShield as a Service

3.3. Entrust products

3.4. nShield product documentation