Cloudera Runtime 7.3.2

Access control for Amazon S3-compatible object stores

Date published: 2020-07-28
Date modified: 2026-03-31

https://docs.cloudera.com/

Legal Notice

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Introduction to Ranger RAZ in Cloudera Base on premises clusters................. 4

Prerequisites for RAZ.. 5

Storage prerequisites.. 5

Distributing certificates to access S3 endpoint.. 6

Adding Knox IDBroker role... 8

Adding RAZ service... 9

Configuring Knox IDBroker role for S3 object stores...................................... 10

Configuring cloud credentials using Knox Alias... 10

Configuring cloud credentials using HashiCorp Vault... 11

Configuring RAZ service for S3 object stores..13

Configuring the endpoint for an S3 bucket in the HDFS cluster-wide configuration...13

Restarting all services...14

Post-installation tasks...14

Ranger Hive authorizer for S3 cloud storage..16

Accessing multiple buckets managed by the same cloud account credential...18

Limitations...19

Introduction to Ranger RAZ in Cloudera Base on premises clusters

Learn how to support Amazon S3-compatible object stores through the Ranger Remote Authorization Service (RAZ).

Cloudera Base on premises defaults to HDFS and natively supports the Ozone object store. There was limited support for Amazon S3-compatible object stores with respect to executing workloads like Hive and Spark. Not all workloads were supported from an authorization perspective on Amazon S3-compatible object stores. The inconsistent policy framework created challenges for the Policy Administrator in terms of applying corporate policies to secure sensitive data. Managing data access across teams and individuals was an architectural challenge.

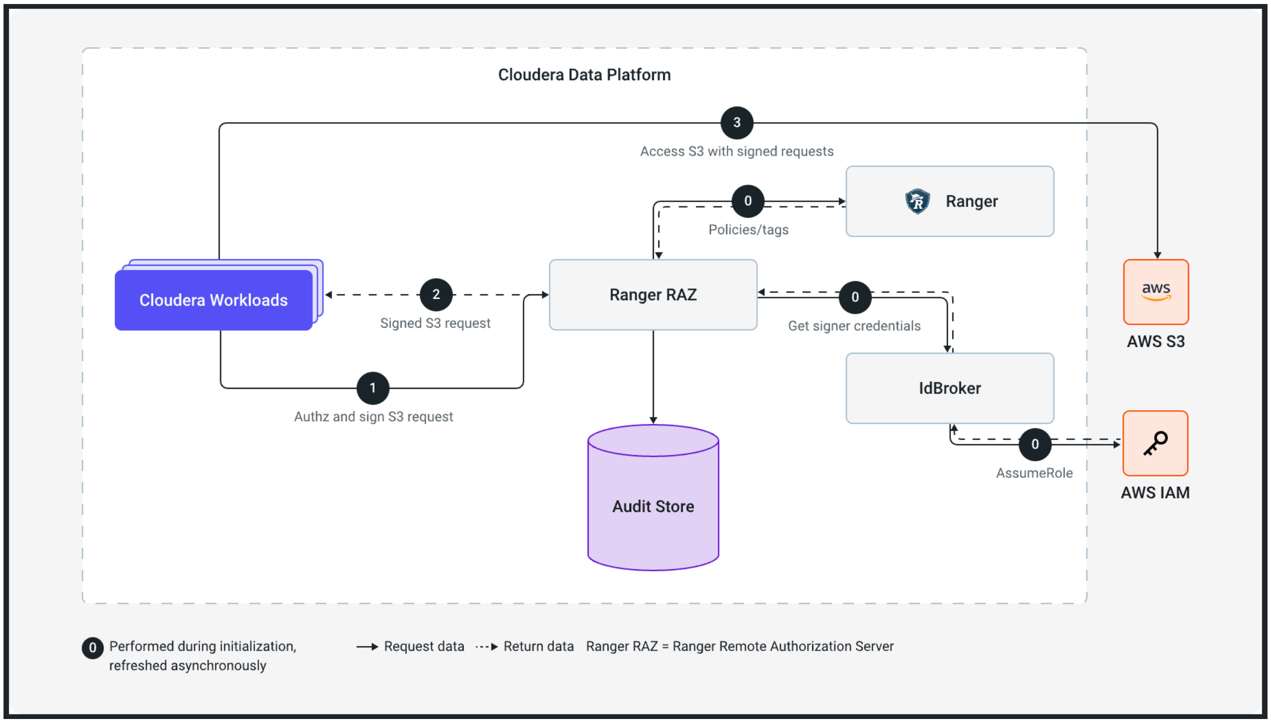
The Ranger RAZ resolves this challenge by using Apache Ranger policies to authorize access to Amazon S3-compatible object storage, similar to HDFS files.

Ranger RAZ allows Ranger access control policies to be applied to external object stores. This removes the need to maintain a separate set of policies for the external object stores. Currently, this applies to the following components:

- Hive or Impala executors
- Spark executors
- Data Explorer for accessing the users home and other accessible directories or transferring files.

RAZ server obtains cloud credentials from IDBroker during initialization. RAZ provides client tokens to access a cloud storage object, a file or directory. RAZ enforces access control using Cloudera cluster identities through Ranger policies. The following architectural diagram shows how RAZ integrates and interacts with other components in a RAZ-enabled Amazon S3 environment:

Figure 1: RAZ architecture



RAZ supports use cases that require access control on files or directories, including the following examples:

- Per-user home directories.
- Data engineering (Spark) efforts that require access to cloud storage objects and directories.
- Data warehouse queries (Hive, Impala, or Iceberg) that use external tables.

- Access to Ranger's rich access control policies, such as date-based access revocation and user, group, or role-based controls, along with corresponding audits.
- Tag-based access control using the classification propagation feature that originates from directories.

Prerequisites for RAZ

Before you enable RAZ in your environment, you must ensure that Cloudera has access to the Amazon S3 resources in your AWS account and that your account contains all the resources Cloudera requires.

Storage prerequisites

Prepare your Amazon S3 cloud storage for access by creating an IAM user, generating access keys, setting up a bucket and folder, and configuring IAM policies.

About this task

Important: The following steps are specific for accessing Amazon S3. Apply similar procedures to your Amazon S3-compatible object storage solution.

Procedure

1. Create an AWS IAM user.

   For instructions, see Creating IAM users in the IAM documentation.
2. Generate an access key and a secret access key for this user.

   For instructions, see Create new access keys for an IAM user in the AWS documentation.
3. Create a bucket and a storage folder.

   Note: The bucket is configured as a general-purpose type with versioning disabled, default encryption set to server-side encryption with Amazon S3-managed keys (SSE-S3), and all public access blocked.

   The storage folder is created without specifying an encryption key.

   ```
   s3a://<BUCKET-NAME>/storage/
   ```
4. Create an AWS IAM attached policy named S3RazPolicy by using the following permissions:

   For instructions, see Creating IAM policies in the IAM documentation.

   Note: Modify the bucket name according to your use case.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "AccessToBucketObjects",
         "Effect": "Allow",
         "Action": [
           "s3:AbortMultipartUpload",
           "s3:DeleteObject",
           "s3:DeleteObjectVersion",
           "s3:GetObject",
           "s3:GetObjectAcl",
           "s3:GetObjectVersion",
           "s3:GetObjectVersionAcl",
   ```

```
        "s3:PutObject",
        "s3:ListMultipartUploadParts"
      ],
      "Resource": "arn:aws:s3:::<BUCKET-NAME>/*"
    },
    {
      "Sid": "AccessToBucket",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketAcl",
        "s3:GetBucketLocation",
        "s3:GetBucketVersioning",
        "s3:GetEncryptionConfiguration",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads"
      ],
      "Resource": "arn:aws:s3:::<BUCKET-NAME>"
    }
  ]
}
```

5. Create an AWS IAM role named S3RazRole by using the following trust policy, and then attach the previously created S3RazPolicy AWS IAM policy:

   For instructions, see Creating a role to delegate permissions to an IAM user in the IAM documentation.

   Note: Modify the Amazon Resource Name (ARN) of the IAM user name in the trust policy according to your use case. This trust policy allows the newly created IAM user to assume S3RazRole.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<ARN>:user/<IAM-USER-NAME>"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Distributing certificates to access S3 endpoint

Distribute Amazon certificates to the required truststores to enable secure communication with Amazon S3 endpoints. This process requires importing necessary certificates into the Cloudera Manager global truststore and in Java cacerts files to ensure TLS handshake success across the environment.

About this task

Note: The following steps are specific to accessing the Amazon S3 endpoint. Please follow similar procedures using the certificate file for your Amazon S3-compatible object storage endpoint.

Perform the following steps to add Amazon certificates to the required truststores, including the Java cacerts file:

Procedure

1. **Add Amazon certificates into the truststore.**

   a) Download Amazon Trust Services certificates from https://www.amazontrust.com/repository/ to the truststore files in both JKS and PEM formats.

   Note: If you are using the Cloudera Manager Auto-TLS feature, you need to follow step b. If you are managing TLS manually, you need to follow step c.

   Note: For an Amazon S3-compatible object storage endpoint, generate a certificate in PEM format using OpenSSL as follows:

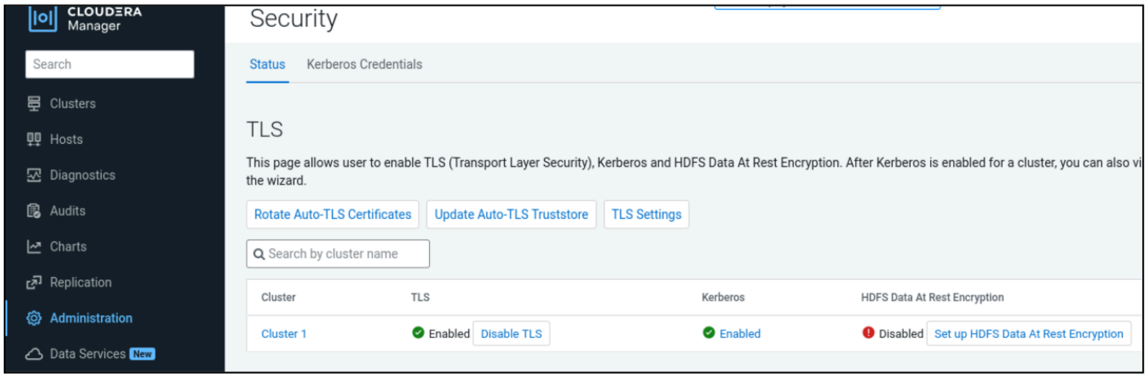
   ```
   openssl s_client -connect s3.myobjstore.com:443 <<<'' |
   openssl x509 -out /tmp/s3-endpoint.pem
   ```

   b) If you are using a Cloudera Manager managed truststore, you need to update the cm-auto-global_truststore.jks and cm-auto-global_cacerts.pem files.

   1. Create a combined certificate file of AmazonRootCA1.pem, AmazonRootCA2.pem, AmazonRootCA3.pem, AmazonRootCA4.pem, and SFSRootCAG2.pem files:

      ```
      cat AmazonRootCA1.pem AmazonRootCA2.pem AmazonRootCA3.pem AmazonRoot
      CA4.pem SFSRootCAG2.pem >> AmazonCombineCacerts.pem
      ```

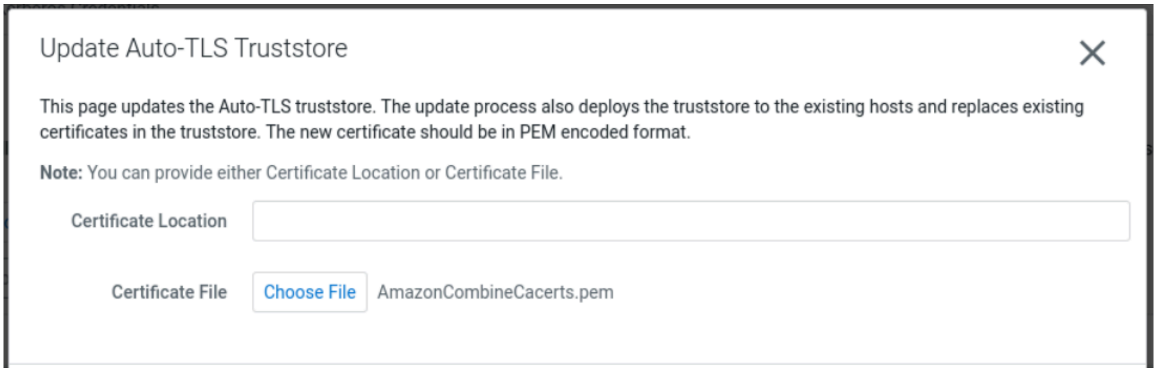
   2. Go to Cloudera Manager Administration Security .
   3. Click Update Auto-TLS Truststore.



This page allows user to enable TLS (Transport Layer Security), Kerberos and HDFS Data At Rest Encryption. After Kerberos is enabled for a cluster, you can also vi the wizard.

Cluster	TLS	Kerberos	HDFS Data At Rest Encryption
Cluster 1	✓ Enabled Disable TLS	✓ Enabled	● Disabled Set up HDFS Data At Rest Encryption

   4. Upload the combined AmazonCombineCacerts.pem certificate file, and click Save.



This page updates the Auto-TLS truststore. The update process also deploys the truststore to the existing hosts and replaces existing certificates in the truststore. The new certificate should be in PEM encoded format.

Note: You can provide either Certificate Location or Certificate File.

Certificate Location:

Certificate File: AmazonCombineCacerts.pem

Cloudera Manager will start the `Update The Global Truststore` commands on all the managed hosts. These commands update the cm-auto-global_truststore.jks and cm-auto-global_cacerts.pem files containing the required Amazon Trust Services certificates.

> **Note:** If you are using the Auto-TLS feature, you need to upload the certificate into the Cloudera Manager managed truststore for an Amazon S3-compatible object storage endpoint. To do so, follow step b described above.
>
> If you have multiple certificates to upload, create a single PEM format certificate by combining all the individual certificates, and upload it from Cloudera Manager.

c) If you are managing TLS settings manually, update the managed truststore files in the cluster manually by using the Java keytool command.

2. Add the Amazon Trust Services certificates into the java cacerts keystore.

If you are using Java 17, you can skip this step.

> **Note:** For an Amazon S3-compatible object storage endpoint, you must add the required endpoint certificate to the Java cacerts keystore. If multiple certificates are needed for your endpoint, add each certificate individually.

3. Export the Cloudera Manager CA global truststore into Java cacerts if you are using Cloudera Manager Auto-TLS.

> **Note:** If you are managing TLS settings manually, then update the Java cacerts to include the CA certificate used for creating the manual keystore.

a) Run the following command on the Cloudera Manager server host:

```
${JAVA_HOME}/bin/keytool -exportcert -keystore /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_truststore.jks -storepass <TRUSTSTORE-PASSWORD> -alias cmrootca-0 -file /tmp/cm-truststore.cer
${JAVA_HOME}/bin/keytool -import -file /tmp/cm-truststore.cer -alias cm-truststore -keystore <JAVA-CACERTS-PATH> -storepass changeit
```

b) Copy the updated Java cacerts to the other Cloudera Manager-managed hosts.

Adding Knox IDBroker role

When accessing cloud storage in Cloudera, credentials are provided by Knox IDBroker, an identity federation solution that exchanges cluster authentication for temporary cloud credentials. Add the Knox IDBroker role to your cluster.

About this task

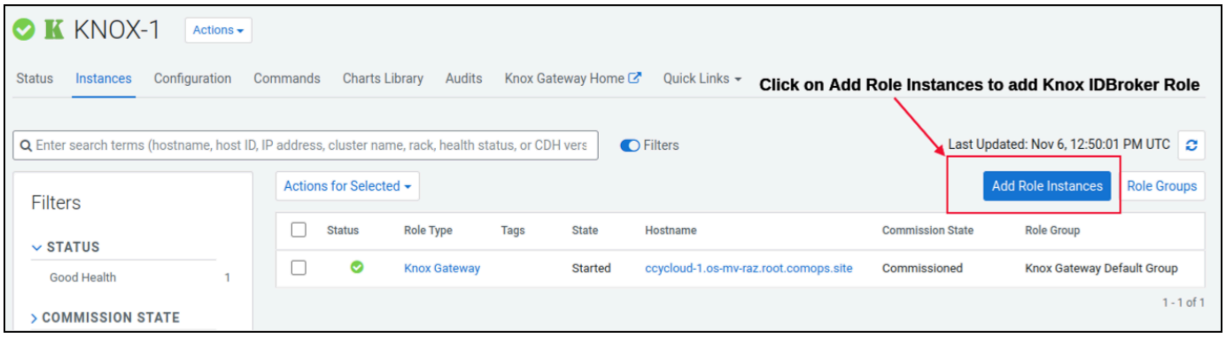
IDBroker is a REST API built as part of Apache Knox’s authentication services. It allows an authenticated and authorized user to exchange a set of credentials or a token for cloud vendor access tokens.

Perform the following steps to add the Knox IDBroker role to your cluster:

Procedure

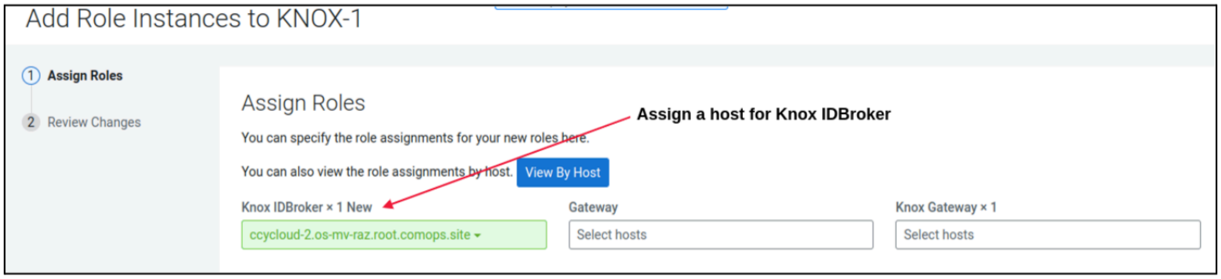
1. Log in to Cloudera Manager with admin credentials.
2. Go to the **Knox** service page.
3. Select the Instances tab.

4. Add the IDBroker role to the installed Knox service.



□	Status	Role Type	Tags	State	Hostname	Commission State	Role Group
□	✓	Knox Gateway		Started	ccycloud-1.os-mv-raz.root.comops.site	Commissioned	Knox Gateway Default Group

5. Assign a host for the Knox IDBroker role.



You can specify the role assignments for your new roles here.

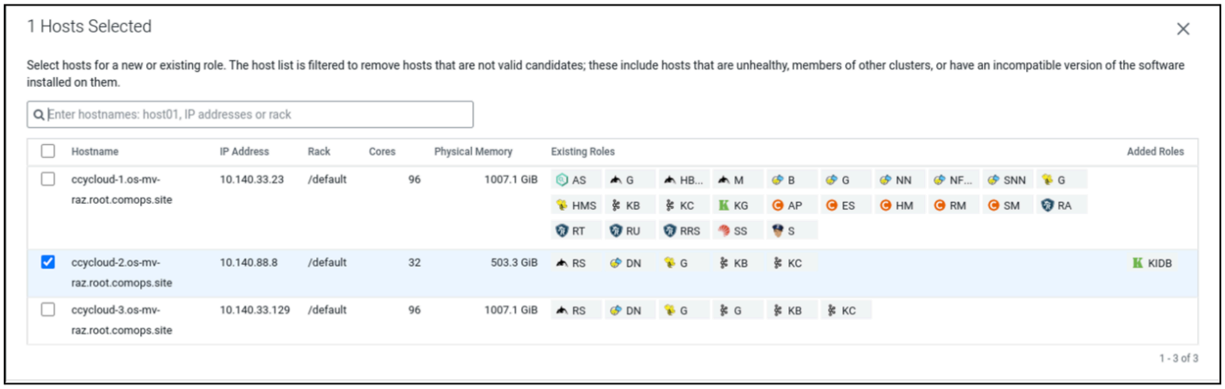
You can also view the role assignments by host.

Knox IDBroker × 1 New: ccycloud-2.os-mv-raz.root.comops.site

Gateway: Select hosts

Knox Gateway × 1: Select hosts

6. Select a host other than the Knox Gateway role and click Continue.



Select hosts for a new or existing role. The host list is filtered to remove hosts that are not valid candidates; these include hosts that are unhealthy, members of other clusters, or have an incompatible version of the software installed on them.

□	Hostname	IP Address	Rack	Cores	Physical Memory	Existing Roles	Added Roles
□	ccycloud-1.os-mv-raz.root.comops.site	10.140.33.23	/default	96	1007.1 GiB	AS G HB... M B G NN NF... SNN G HMS KB KC KG AP ES HM RM SM RA RT RU RRS SS S	
☑	ccycloud-2.os-mv-raz.root.comops.site	10.140.88.8	/default	32	503.3 GiB	RS DN G KB KC	KIDB
□	ccycloud-3.os-mv-raz.root.comops.site	10.140.33.129	/default	96	1007.1 GiB	RS DN G G KB KC	

7. On the **Review Changes** page, update the IDBroker Master Secret password configuration to start the service.
8. Click Continue.
9. Restart only the Knox service from Knox Actions Restart .
10. Deploy the client configuration for Knox from Cloudera Manager Knox Actions Deploy Client Configuration .

Adding RAZ service

Learn how to add Ranger Remote Authorization Service (RAZ) service to your cluster.

About this task

Note: If you are an upgraded user from Cloudera Runtime 7.1.9 or 7.1.7 to Cloudera Runtime 7.3.2.0, then before adding the RAZ service, ensure that you have the default cm_s3 service available in the Ranger Admin UI. If it is not available, you must create the default cm_s3 service in the Ranger Admin UI. To create this default service, you can run the `Setup Ranger Plugin Service` command from Cloudera Manager Ranger Actions Setup Ranger Plugin Service . This command will create the missing default cm_s3 service, or you can manually create the service by logging in to the Ranger Admin UI using admin user credentials.

Perform the following steps to add the RAZ service to your cluster:

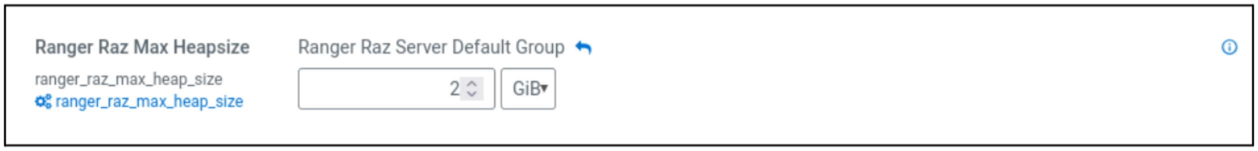
Procedure

1. On the Cloudera Manager home page, go to the Status tab.
2. Click the Actions icon beside the cluster name, and select the Add a Service action.

   A list of service types appears. You can select only one type of service from the list at a time.
3. Select the Ranger RAZ service, and click Continue.
4. On the **Select Dependencies** page, select a combination that includes at least the Ranger and ZooKeeper services.
5. Assign a host for the Ranger RAZ Server role on the **Assign Roles** page.
6. Restart only the RAZ service from Ranger RAZ Actions Restart .
7. Create the RAZ plugin by running the Create Ranger RAZ Plugin Audit Directory command from Cloudera Manager Ranger RAZ Actions .

What to do next

To ensure optimal performance, you must set the Java heap size to a minimum of 2GB for the Ranger RAZ service from Cloudera Manager. Update the Ranger Raz Max Heapsize property to 2GB from Ranger RAZ Configuration .



Ranger Raz Max Heapsize
ranger_raz_max_heap_size
ranger_raz_max_heap_size

Ranger Raz Server Default Group

2 GiB

Configuring Knox IDBroker role for S3 object stores

You can configure the Knox IDBroker role for S3 object stores either by using a Knox Alias or by using HashiCorp Vault.

Configuring cloud credentials using Knox Alias

Configure the Knox IDBroker role for S3 object stores using a Knox Alias. This process requires setting AWS IAM user credentials, configuring alternative STS endpoints for S3-compatible storage, and managing user or group mappings.

Procedure

1. Provide the AWS user access key and secret access key to the Knox IDBroker role.
   a) Go to Cloudera Manager Clusters Knox Configuration .
   b) Search for the Save Alias Command Input - IDBroker property to set the AWS IAM user credentials access key alias.
   c) Enter aws-cab.aws.credentials.key=<AWS user Access key>, and click Save Changes.
   d) Click Actions and run the Save Alias - IDBroker command.
   e) Again, search for the Save Alias Command Input - IDBroker property to set the AWS IAM user credential secret access key alias.
   f) Enter aws-cab.aws.credentials.secret=<AWS user Secret access key>, and click Save Changes.
   g) Click Actions and run the Save Alias - IDBroker command.
   h) Set the Save Alias Command Input - IDBroker property to empty.
   i) Click Save Changes.

   Note: You must perform the above steps again whenever access keys are rotated. Then restart Knox and Ranger RAZ services from Cloudera Manager.

2. If using an S3-compatible alternative to Amazon storage, configure the IDBroker topology to reference the alternative STS endpoint.
   a) Go to Cloudera Manager Clusters Knox Configuration .
   b) Add the following configuration to the Knox IDBroker Advanced Configuration Snippet (Safety Valve) for conf/cdp-resources.xml advanced configuration snippet:

   ```
   Name: aws-cab
   Value: providerConfigRef=cab-providers# IDBROKER:cloud.policy.config.p
   rovider=default# IDBROKER:cloud.client.provider=AWS# IDBROKER:aws.region
   .name=us-east-1# IDBROKER:org.apache.knox.idbroker.endpoint.override=htt
   ps://s3.myobjectore.com/#
   ```

   Note: Update the region name and endpoint in the above value as per your setup.

   c) Click Save Changes.

3. Provide user or group mapping administration.
   a) Set the Knox IDBroker AWS User Mapping configuration property to the following value:

   ```
   rangerraz=arn:aws:iam::<ARN>:role/S3RazRole
   ```

   Note: Modify the Amazon Resource Name (ARN) of the IAM role created earlier.

   b) Click Save Changes.

Configuring cloud credentials using HashiCorp Vault

Configure Knox IDBroker with HashiCorp Vault to securely manage AWS credentials. This process requires retrieving short-lived tokens, enhancing security, and eliminating long-term credential storage.

About this task

Knox IDBroker can be configured with HashiCorp Vault to enhance AWS credentials management. HashiCorp Vault enables Knox to retrieve short-lived AWS credentials from HashiCorp Vault, improving security and eliminating the need for long-term credential storage.

Before you begin

- HashiCorp Vault must be installed and properly configured.
- An AWS account must be available with the required IAM roles.
- Network connectivity must be established between Knox and the HashiCorp Vault server.

Procedure

1. Go to Cloudera Manager Knox Configuration .
2. Search for the Save Alias Command Input - IDBroker property.
3. Enter the following parameter:

   ```
   [***TOPOLOGY NAME***].vaultTokenAlias=[***VAULT TOKEN***]
   ```

   Note: Replace [***TOPOLOGY NAME***] with the name of your topology, and [***VAULT TOKEN***] with the specific token for the vault.

4. Click Save Changes(CTRL+S).
5. Select the Save Alias - IDBroker action from the Actions drop-down list.
6. Search for the Knox IDBroker Advanced Configuration Snippet (Safety Valve) for conf/cdp-resources.xml advanced configuration snippet.
7. Click the + icon and add the following entries:

   - Name= [***TOPOLOGY NAME***]
   - Value=

     ```
     providerConfigRef=cab-providers\#IDBROKER:cloud.policy.config.provider=d
     efault\#IDBROKER:cloud.client.provider=AWS\#IDBROKER:hashicorp.vault.ena
     bled=true\#IDBROKER:hashicorp.vault.address=[***VAULT ADDRESS***]\#IDBRO
     KER:hashicorp.vault.path=aws/sts\#IDBROKER:hashicorp.vault.server.cert.p
     ath=[***VAULT CERTIFICATE PATH***]
     ```

     The value string includes the following parameters:

     - hashicorp.vault.enabled=true – Enables the vault integration for that specific topology.
     - hashicorp.vault.address=[***VAULT ADDRESS***] – Replace [***VAULT ADDRESS***] with the actual address of the vault.
     - hashicorp.vault.path=aws/sts –Specifies the path for the STS credentials.
     - (Optional) hashicorp.vault.server.cert.path=[***VAULT CERTIFICATE PATH***] – Adds the path for the vault certificate. Required in case of self-signed certificates. Replace [***VAULT CERTIFICATE PATH***] with the path to the certificate.

8. Click Save Changes(CTRL+S).
9. Search for the Knox IDBroker AWS User Mapping property.
10. Enter the following parameter:

    ```
    rangerraz=arn:aws:iam::<ARN>:role/S3RazRole
    ```

    Note: Modify the Amazon Resource Name (ARN) of the IAM role created earlier.

11. Refresh the Knox instance configuration by clicking the Stale Configuration: Refresh needed indicator and wait until the refresh process completes.
12. Verify that the vault integration was successful by running the following command in the Command Line Interface (CLI):

    ```
    kinit rangerraz
    ```

```
hdfs dfs -ls s3a://[***RESOURCE***]
```

Note: Replace [***RESOURCE***] with the name of your S3 bucket.

Configuring RAZ service for S3 object stores

After you add the Ranger Remote Authorization Service (RAZ) service and configure the Knox IDBroker role for S3 object stores, you must configure the RAZ service for S3 object stores.

Procedure

1. Go to Cloudera Manager Ranger RAZ Configuration .
2. Add the following configurations in the Ranger Raz Server Advanced Configuration Snippet (Safety Valve) for ranger-raz-conf/ranger-raz-site.xml advanced configuration snippet:
    - Key: ranger.raz.bootstrap.servicetypes

      Value: s3
    - Key: ranger.raz.force.s3.region.lookup

      Value: FALLBACK

Configuring the endpoint for an S3 bucket in the HDFS cluster-wide configuration

Configure the endpoint for an S3 bucket in the HDFS cluster-wide configuration to manage bucket-specific settings and regional endpoints.

Procedure

1. Go to Cloudera Manager HDFS Configuration .
2. Add the following configuration in the Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml advanced configuration snippet:

```
Key: fs.s3a.bucket.<BUCKET-NAME>.endpoint
```

```
Value: s3.<REGION-NAME>.amazonaws.com
```

The configuration setting described above is used to specify a bucket-specific endpoint for Amazon S3. This configuration is also used in case where multiple buckets are managed in different regions. There are also global configuration settings that control the default behavior for all S3 buckets:

- fs.s3a.endpoint: Sets the default endpoint for all S3 buckets.
- fs.s3a.endpoint.region: Specifies the region used for signing when combined with fs.s3a.endpoint, and all S3 buckets point to the same region.

Example values for these configurations:

- f3.s3a.endpoint: s3.amazonaws.com
- f3.s3a.endpoint.region: us-east-1/us-east-2/us-west-1/us-west-2/eu-west-1/..

  For more information on the endpoints and regions supported, refer to Amazon Simple Storage Service endpoints and quotas.

For detailed information on configuration behavior and how to connect to an Amazon S3 bucket using the S3A Connector, please refer to Connecting to an Amazon S3 Bucket through the S3A Connector.

Restarting all services

You must restart all services in the cluster, including the Re-deployClientConfig, to ensure that IDBroker and Ranger RAZ services are functional for supporting S3 object stores.

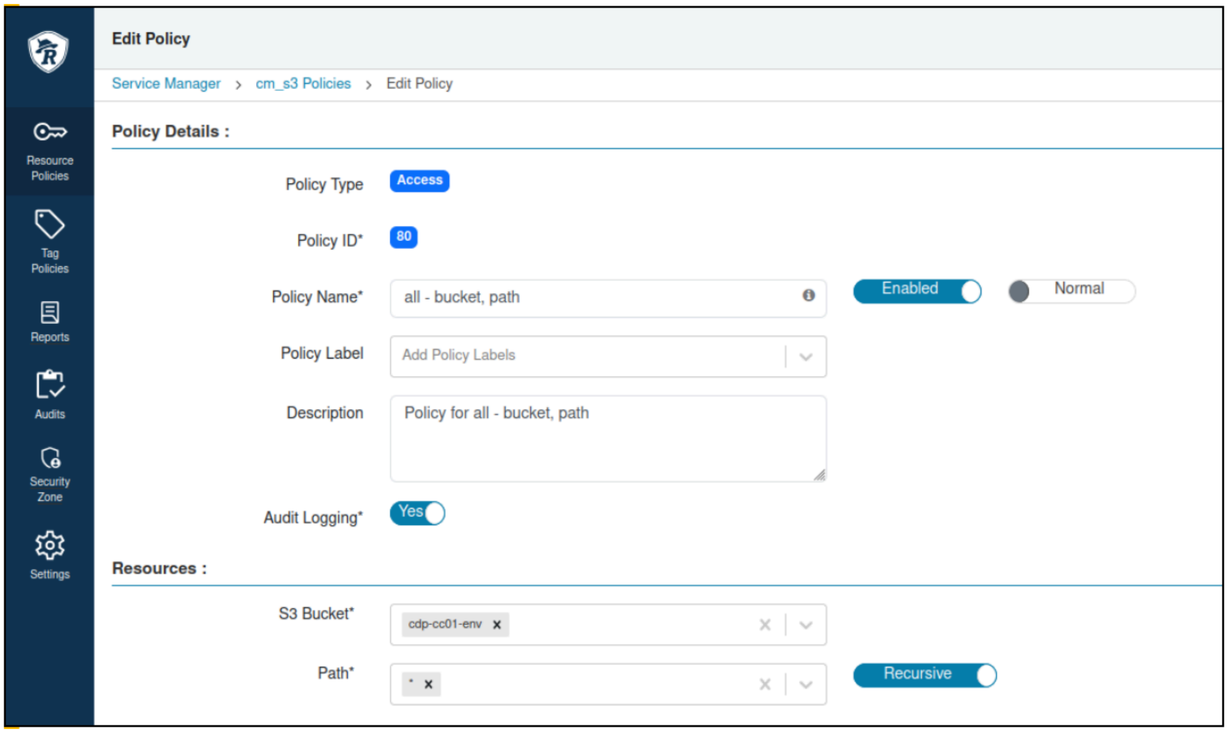
Post-installation tasks

After you restart all services, you must perform post-installation tasks, including updating the cm_s3 policies, creating policies for Hive and Impala, verifying authorization for RAZ, and updating a Spark property.

Procedure

1. Log in to Ranger Admin UI using admin user credentials.
2. After a successful login, go to the cm_s3 policy listing page.

3. Update the all - bucket, path default policy with the bucket being used.



Edit Policy

Service Manager > cm_s3 Policies > Edit Policy

Policy Details :

Policy Type: Access

Policy ID*: 80

Policy Name*: all - bucket, path — Enabled — Normal

Policy Label: Add Policy Labels

Description: Policy for all - bucket, path

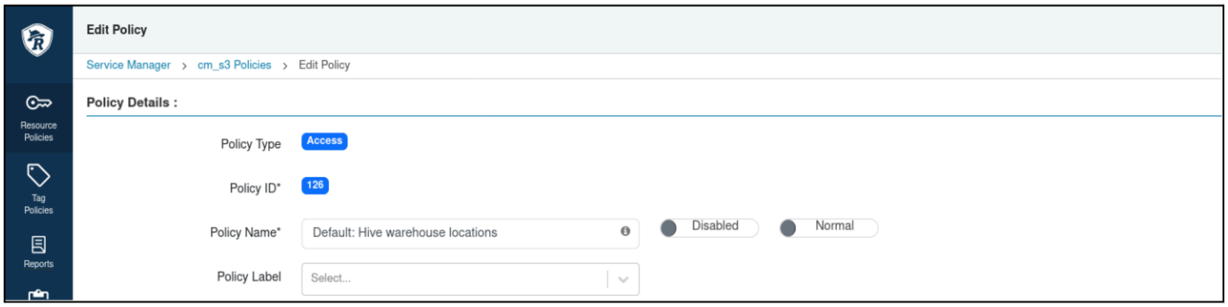
Audit Logging*: Yes

Resources :

S3 Bucket*: cdp-cc01-env

Path*: * — Recursive

4. Disable all the other policies in cm_s3 except for all - bucket, path.



Edit Policy

Service Manager > cm_s3 Policies > Edit Policy

Policy Details :

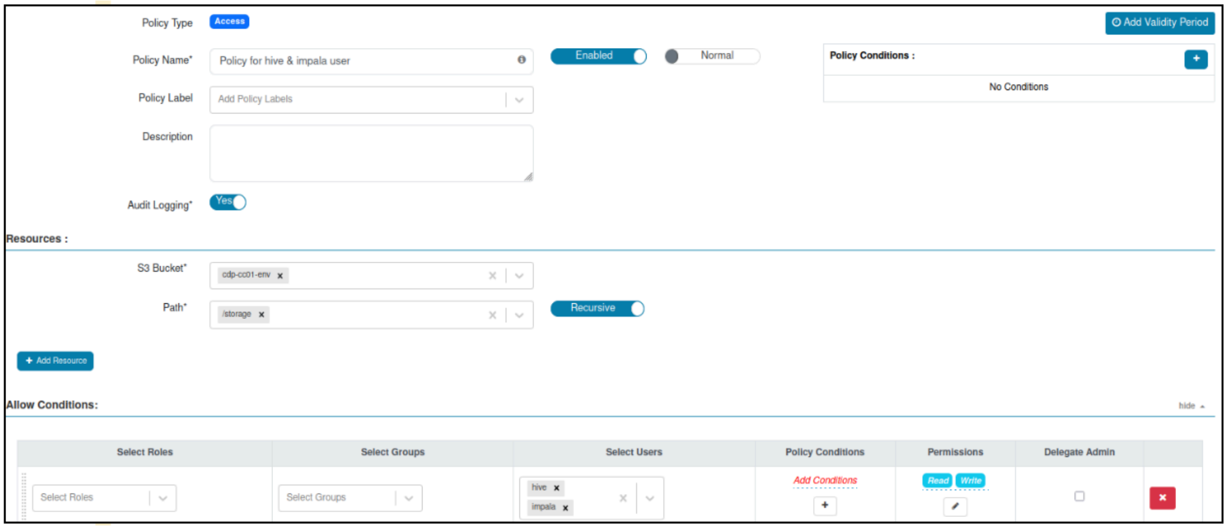
Policy Type: Access

Policy ID*: 126

Policy Name*: Default: Hive warehouse locations — Disabled — Normal

Policy Label: Select...

5. To disable the policies, edit the individual policy and click the Enabled toggle next to the policy name.
6. Create a policy for Hive and Impala users to have read and write access on the S3 bucket and path or storage.



Policy Type: Access

Policy Name*: Policy for hive & impala user — Enabled — Normal

Policy Label: Add Policy Labels

Description:

Audit Logging*: Yes

Policy Conditions :

No Conditions

Resources :

S3 Bucket*: cdp-cc01-env

Path*: /storage — Recursive

Allow Conditions:

Select Roles	Select Groups	Select Users	Policy Conditions	Permissions	Delegate Admin
Select Roles	Select Groups	hive impala	Add Conditions	Read Write	□

7. Verify that RAZ authorization is working by executing the following command:

   a) SSH to the node where Ranger RAZ Server is installed from Cloudera Manager and run the following commands:

   ```
   ps -ef | grep rangerraz | awk '{split($0, array,"classpath"); print arra
   y[2]}' | cut -d: -f1 #Note down RAZ_CONF_DIR directory

   klist -kt <RAZ_CONF_DIR>/../ranger_raz.keytab #Note down RAZ_PRINCIPAL,
   it must starts with rangerraz

   kinit -kt <RAZ_CONF_DIR>/../ranger_raz.keytab <RAZ_PRINCIPAL>
   ```

   b) List all items available under the S3 storage path by running the following HDFS command:

   ```
   hdfs dfs -ls s3a://<BUCKET-NAME>/storage/
   ```

   c) Log in to the Ranger Admin UI Audits Plugins tab.

   d) Use the Service Name filter to get the result for cm_s3.

   A successful 200 response record for the cm_s3 service appears.

8. Update the Spark service property.

Property Name	Value
Spark 3 Client Advanced Configuration Snippet (Safety Valve) for spark3-conf/spark-defaults.conf	• spark.kerberos.access.hadoopFileSystems=s3a://<BUCKET-NAME> **Note:** You can specify multiple buckets using a comma-separated list, but this action requires a restart. For example, spark.kerberos.access.hadoopFileSystems=s3a://<BUCKET_FIRST>,s3a://<BUCKET_SECOND>,..... • spark.hadoop.fs.s3a.ssl.channel.mode=default • spark.hadoop.mapreduce.fileoutputcommitter.algorithm.version=1

9. Restart cluster services for their staleness.

Ranger Hive authorizer for S3 cloud storage

Hive allows you to create tables based on existing or new files and directories within a storage location. To use the storage location, the end user must have authorized access to that location. If the storage location is an S3 cloud object, there are two primary options for granting permissions via Ranger.

Option 1: Use the Ranger RAZ S3 plugin policy

Use this option if you want fine-grained access control specifically managed through the Ranger RAZ S3 plugin policy.

1. Configure Hive.

   a. Go to Cloudera Manager Hive Configuration .

   b. Search for the Ranger Plugin URL Auth Filesystem Schemes configuration.

   c. Append s3a: to the existing values.

   Permissible values are hdfs:,file:,wasb:,adl:,s3a:.

   d. Restart the Hive service.

2. Configure Ranger.

a. Log in to the Ranger Admin UI.

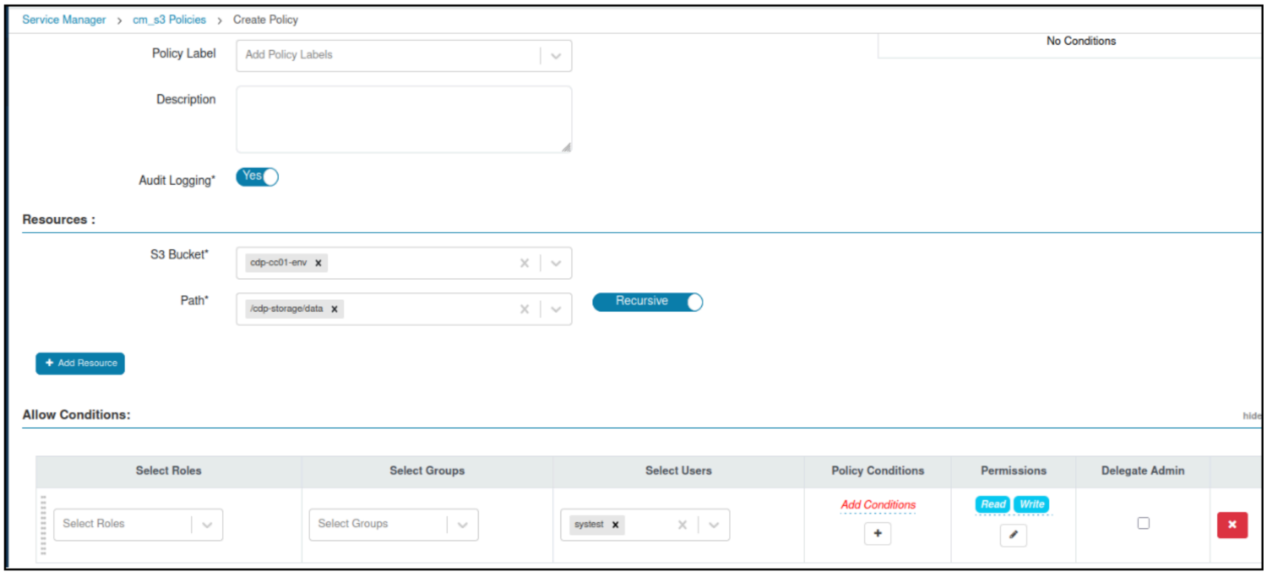
b. Navigate to Resource Policies cm_s3 and click Add New Policy.

c. Define the policy for your specific S3 location.

Note: Repeat these steps for the Hive On Tez and Impala services if they require S3 access.

S3 Plugin Policy give access to resource bucket and path:

```
S3 Bucket - cdp-cc01-env
Path - /cdp-storage/data
```



Service Manager > cm_s3 Policies > Create Policy

No Conditions

Policy Label: Add Policy Labels

Description

Audit Logging*: Yes

Resources :

S3 Bucket*: cdp-cc01-env

Path*: /cdp-storage/data — Recursive

Allow Conditions:

Select Roles	Select Groups	Select Users	Policy Conditions	Permissions	Delegate Admin	
Select Roles	Select Groups	systest	Add Conditions	Read Write	☐	✖

Option 2: Use the Hive URL policy

Use this option to authorize access requests against standard Hive URL policies rather than the RAZ S3 plugin.

1. Configure Hive.

a. Go to Cloudera Manager Hive Configuration .

b. Search for the Ranger Plugin URL Auth Filesystem Schemes configuration.

c. Remove s3a: from the configuration.

This ensures authorization is handled by the Hive URL policy and bypasses the Ranger RAZ S3 plugin.

d. Restart the Hive service.

2. Configure Ranger.

a. Log in to the Ranger Admin UI.

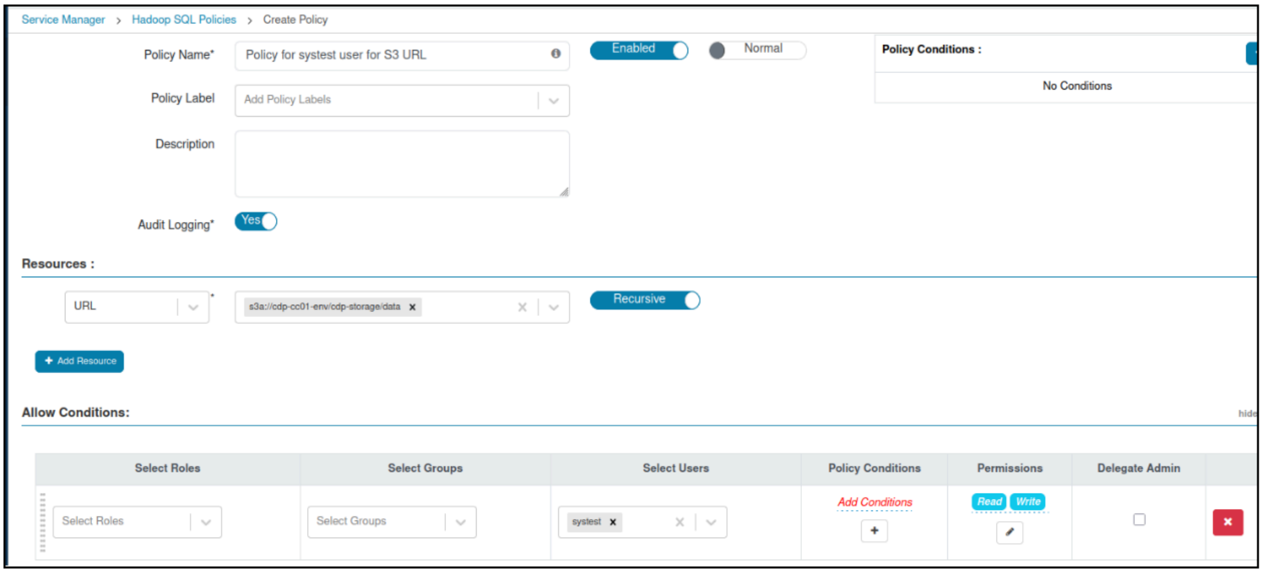
b. Navigate to Resource Policies Hadoop SQL and click Add New Policy.

c. Define a new Hive policy using the URL resource type to authorize the specific S3 location.

Note: Repeat these steps for the Hive On Tez and Impala services if they require S3 access.

Hive URL Policy giving access to S3 url full path:

```
URL - s3a://cdp-cc01-env/cdp-storage/data
```



Policy Name*: Policy for systest user for S3 URL

Policy Conditions : No Conditions

Policy Label: Add Policy Labels

Description:

Audit Logging*: Yes

Resources :

URL: s3a://cdp-cc01-env/cdp-storage/data

Allow Conditions:

Select Roles	Select Groups	Select Users	Policy Conditions	Permissions	Delegate Admin
Select Roles	Select Groups	systest	Add Conditions	Read Write	□

Accessing multiple buckets managed by the same cloud account credential

You can also access multiple buckets present in your Amazon S3 or Amazon S3-compatible object stores.

Procedure

1. Assign at least the following set of AWS permissions to an AWS IAM role used by IDBroker for Ranger RAZ support in a multiple buckets scenario:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessToBucketObjects",
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectVersion",
                "s3:GetObjectVersionAcl",
                "s3:PutObject",
                "s3:ListMultipartUploadParts"
            ],
            "Resource": [
                "arn:aws:s3:::${BUCKET-01}/*",
                "arn:aws:s3:::${BUCKET-02}/*",
                "arn:aws:s3:::${BUCKET-03}/*"
            ]
        },
        {
            "Sid": "AccessToBucket",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketAcl",
                "s3:GetBucketLocation",
                "s3:GetBucketVersioning",
                "s3:GetEncryptionConfiguration",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::${BUCKET-01}",
                "arn:aws:s3:::${BUCKET-02}",
                "arn:aws:s3:::${BUCKET-03}",
            ]
        }
    ]
}
```

2. In Cloudera Manager HDFS Configuration , add the following S3 endpoint and region configurations to the Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml advanced configuration snippet for the buckets you defined in your AWS IAM policy:

```
Key : fs.s3a.bucket.<BUCKET-01-NAME>.endpoint
Value : s3.<BUCKET-01-REGION-NAME>.amazonaws.com

Key : fs.s3a.bucket.<BUCKET-02-NAME>.endpoint
Value : s3.<BUCKET-02-REGION-NAME>.amazonaws.com
```

Limitations

This topic describes the limitations of using Ranger RAZ to authorize access to Amazon S3-compatible object stores.

In a multiple Kerberos realm/domain setup (for example, a Cloudera hybrid setup), the RAZ service fails to process requests coming from any realm/domain other than the one in which the RAZ service is deployed. The root cause of this issue is the hard-coded DEFAULT value of the ranger.raz.auth.method.dt.params.kerberos.name.rules configuration in the ranger-raz-site.xml file, which does not reflect the actual auth-to-local rules.
Workaround:

1. Log in to Cloudera Manager with admin access.
2. Go to HDFS Instances NameNode Processes .
3. Search for the hadoop.security.auth_to_local property in the core-site.xml file and copy the value.

   Example

   This example shows a sample configuration and value.

   ```
   <property>
   <name>hadoop.security.auth_to_local</name>
   <value>RULE:[2:$1@$0](rangeradmin@ROOT.COMOPS.SITE)s/(.*)@ROOT
   .COMOPS.SITE/ranger/
   RULE:[2:$1@$0](rangertagsync@ROOT.COMOPS.SITE)s/(.*)@ROOT.C
   OMOPS.SITE/rangertagsync/
   RULE:[2:$1@$0](rangerusersync@ROOT.COMOPS.SITE)s/(.*)@ROOT.COM
   OPS.SITE/rangerusersync/
   RULE:[2:$1@$0](rangerkms@ROOT.COMOPS.SITE)s/(.*)@ROOT.COMOP
   S.SITE/keyadmin/
   RULE:[2:$1@$0](atlas@ROOT.COMOPS.SITE)s/(.*)@ROOT.COMOPS.SITE/
   atlas/
   DEFAULT</value>
   </property>
   ```
4. Go to Ranger RAZ Configuration .

5. Search for the ranger.raz.auth.method.dt.params.kerberos.name.rules property, and add the same value you entered for the hadoop.security.auth_to_local property.
6. Save and restart the cluster to remove stale configurations.

   Note: If the custom principle is used for a user who has a Ranger policy, you must update those Ranger policies as well.