Cloudera Management Console

Fine-grained access control for Amazon S3

Date published: 2019-08-22
Date modified: 2025-08-18

https://docs.cloudera.com/

Legal Notice

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.

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 RAZ on AWS environments

Cloudera on cloud defaults to using cloud storage which might be challenging while managing data access across teams and individual users. The Ranger Authorization Service (RAZ) resolves this challenge by enabling Amazon S3 users to use fine-grained access policies and audit capabilities available in Apache Ranger similar to those used with HDFS files in an on-premises or IaaS deployment.

The core RAZ for AWS for Data Lakes and several Cloudera Data Hub templates are available for production use platform-wide.

Many of the use cases that RAZ for S3 enables are cases where access control on files or directories is needed. Some examples include:

- Per-user home directories.
- Cloudera Data Engineering (Spark) efforts that require access to cloud storage objects and directories.
- Cloudera Data Warehouse queries (Hive/Impala) that use external tables.
- Access to Ranger's rich access control policies such as date-based access revocation, user/group/role-based controls, along with corresponding audit.
- Tag-based access control using the classification propagation feature that originates from directories.

Prior to the introduction of RAZ, controlling access to S3 could be enforced at coarse-grained group level using IDBroker mappings. This required rearchitecting the implementation of important file-centric activities as well as admin-level access to both the AWS account and the Cloudera account.

⚠ **Important:** It is recommended that you do not setup IDBroker mapping for workload users in a RAZ-enabled AWS environment.

In HDP and CDH deployments, files and directories are protected with a combination of HDFS Access Control Lists (ACLs) (in CDH, HDP) and Ranger HDFS policies (in HDP). Similarly, in a Cloudera on AWS environment with RAZ for S3 enabled, Ranger's rich access control policies can be applied to Cloudera's access to S3 buckets, directories, and files and can be controlled with admin-level access to Cloudera alone.

You can backup and restore the metadata maintained in the Data Lake services of RAZ-enabled environments. For more information, see Data Lake Backup and Restore.

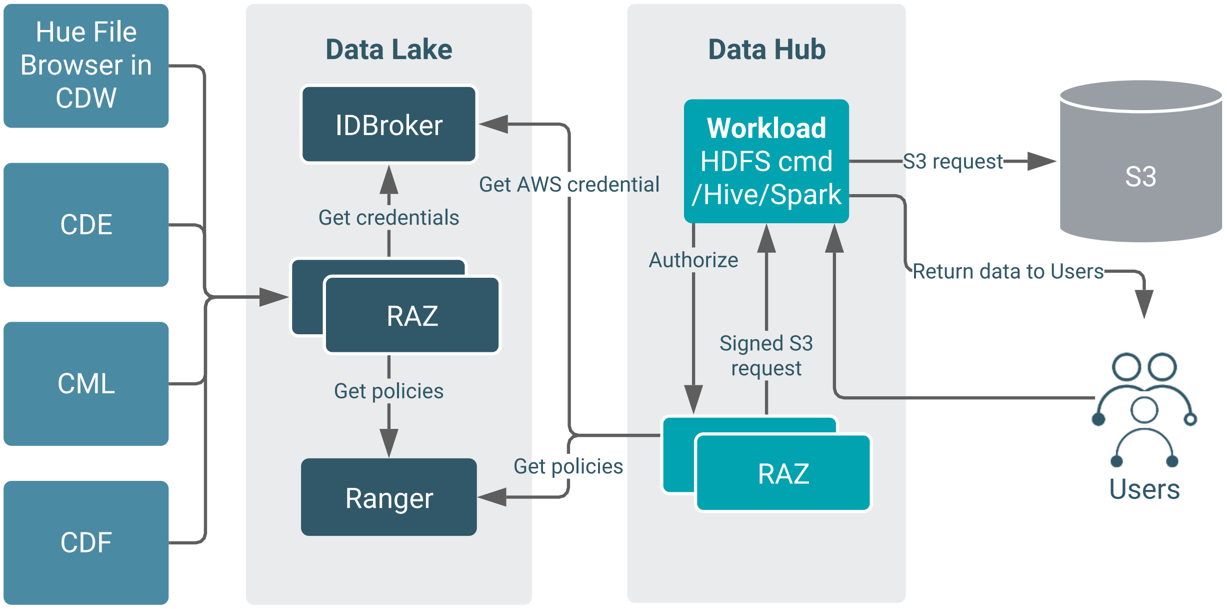
Limitations to use RAZ in AWS environments

The following limitations and known issues have been identified and are under development:

- Currently, there is no automated way to enable RAZ in an existing Cloudera environment that does not have RAZ enabled.

Architecture diagram

The following architecture diagram shows how RAZ integrates and interacts with other components in a RAZ-enabled AWS environment:



AWS requirements for RAZ-enabled AWS environment

Before you enable RAZ on an AWS environment, you must ensure that Cloudera has access to the resources in your AWS account and that your AWS account has all the necessary resources required by Cloudera.

For information about the requirements that must be met to use AWS environment, see AWS requirements documentation.

There are no additional requirements for a RAZ-enabled AWS environment, but RAZ can use the DATALAKE_ADMIN_ROLE IAM role. For more information about the IAM role, see Minimal setup for cloud storage.

Registering a RAZ-enabled AWS environment

You can use the Cloudera web interface or CDP CLI to register a RAZ-enabled AWS environment.

You can enable RAZ on the latest available version of Cloudera Runtime. The minimum version supporting RAZ for AWS environments is Cloudera Runtime 7.2.11.

When you register an AWS environment, enable the Fine-grained access control on S3 option, and then select the DATALAKE_ADMIN_ROLE role to use RAZ in your AWS environment. For more information about the role, see Minimal setup for cloud storage.

Using the Cloudera UI to register RAZ-enabled AWS environment

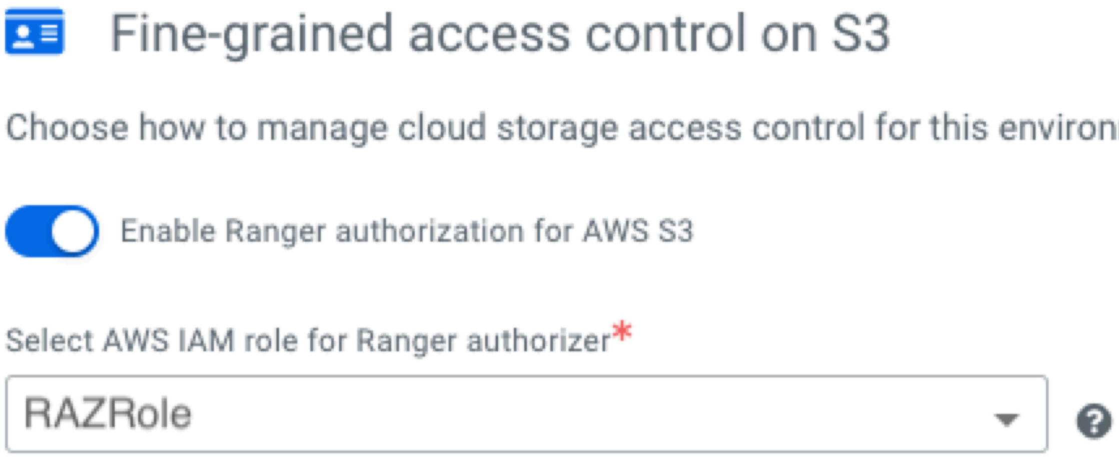
You can use the Cloudera web interface to register a RAZ-enabled AWS environment.

Procedure

1. Log in to the Cloudera web interface.
2. Click Register environment on the Cloudera Management Console Environments page.

3. Provide an Environment Name.
4. Select a provisioning credential.
5. Click Next.
6. Provide a Data Lake Name.
7. Make sure to select Runtime 7.2.11 or higher as the Data Lake version.
8. In the Data Access and Audit section, provide your data storage location and IAM roles created for minimal setup for cloud storage.
9. In the Fine-grained access control on S3 section, click on the toggle button to enable Ranger authorization for S3. Select DATALAKE_ADMIN_ROLE or RAZ_ROLE (if created) as the AWS IAM role.

   The following image shows the Fine-grained access control on S3 section where you can enable the Ranger authorization for AWS S3 and choose an AWS IAM role for Ranger authorizer:

   Fine-grained access control on S3

   Choose how to manage cloud storage access control for this environment

   Enable Ranger authorization for AWS S3

   Select AWS IAM role for Ranger authorizer*

   RAZRole

10. Click Next.
11. Select your region, network, security groups, and provide an SSH key. If required, add tags.
12. Click Next.
13. In the Logs section, provide your logs storage location and managed identities created for minimal setup for cloud storage.
14. Click Register Environment.

Using the CDP CLI to register RAZ-enabled AWS environment

You can use the CDP CLI to register a RAZ-enabled AWS environment. You must download and install beta CDP CLI, and then use CDP CLI commands to register a RAZ-enabled AWS environment.

Procedure

1. To install beta CDP CLI, see Installing Beta CDP CLI.
2. To register a RAZ-enabled AWS environment, you can use the --ranger-cloud-access-authorizer-role *[***RAZ_ROLE***]* and --enable-ranger-raz CDP CLI commands.

   If you have CDP CLI templates to create an AWS environment, modify them by adding the additional parameters required for RAZ.

   The additional option is highlighted in the following sample snippet:

```
cdp environments create-aws-environment \
    --environment-name [***ENVIRONMENT_NAME***] \
    --credential-name [***CREDENTIAL_NAME***] \
    --region [***REGION***] \
    --security-access cidr=[***YOUR_CIDR***] \
    --authentication publicKeyId=[***SSH_PUBLIC_KEY***] \
    --log-storage storageLocationBase=[***S3-LOCATION***],instancePro
file=[***LOG_ROLE***] \
    --vpc-id [***VPC***] \
    --subnet-ids [***SUBNETS***]

 cdp environments set-id-broker-mappings \
    --environment-name [***ENVIRONMENT-NAME***] \
    --data-access-role [***DATALAKE-ADMIN-ROLE***] \
    --ranger-audit-role [***RANGER_AUDIT_ROLE***] \
    --ranger-cloud-access-authorizer-role [***RAZ_ROLE***] \
    --set-empty-mappings
cdp datalake create-aws-datalake \
    --datalake-name [***DATALAKE_NAME***] \
    --environment-name [***ENVIRONMENT-NAME***]\
    --cloud-provider-configuration instanceProfile=[***INSTANCE-
PROFILE***],storageBucketLocation=s3a://[***BUCKET***]/[***PATH***] \
   --enable-ranger-raz
```

Note:

RAZ on AWS environment requires S3Guard to be disabled. Do not pass --S3-guard-table-name in the cdp-environments create-aws-environments command.

You can obtain CDP CLI commands for environment creation from CDP CLI help on Cloudera web interface. For more information, see Obtain CLI commands for registering an environment.

Cluster templates available in RAZ-enabled AWS environment

After you register your RAZ-enabled AWS environment, you can create a Cloudera Data Hub cluster for the environment.

The following cluster templates have been tested and can be used with RAZ:

- Data Engineering
- Data Engineering HA
- Data Engineering Spark3
- Data Mart
- Operational Database with SQL

You can create custom variants of these templates in a RAZ-enabled AWS environment.

Warning: If you create a Cloudera Data Hub with custom paths for a RAZ-enabled AWS environment, your cluster might fail with an AccessDeniedException exception on the custom Zeppelin path. To fix the problem, you must use the workaround mentioned in Creating a Cloudera Data Hub cluster with custom path on page 7

Creating a Cloudera Data Hub cluster with custom path

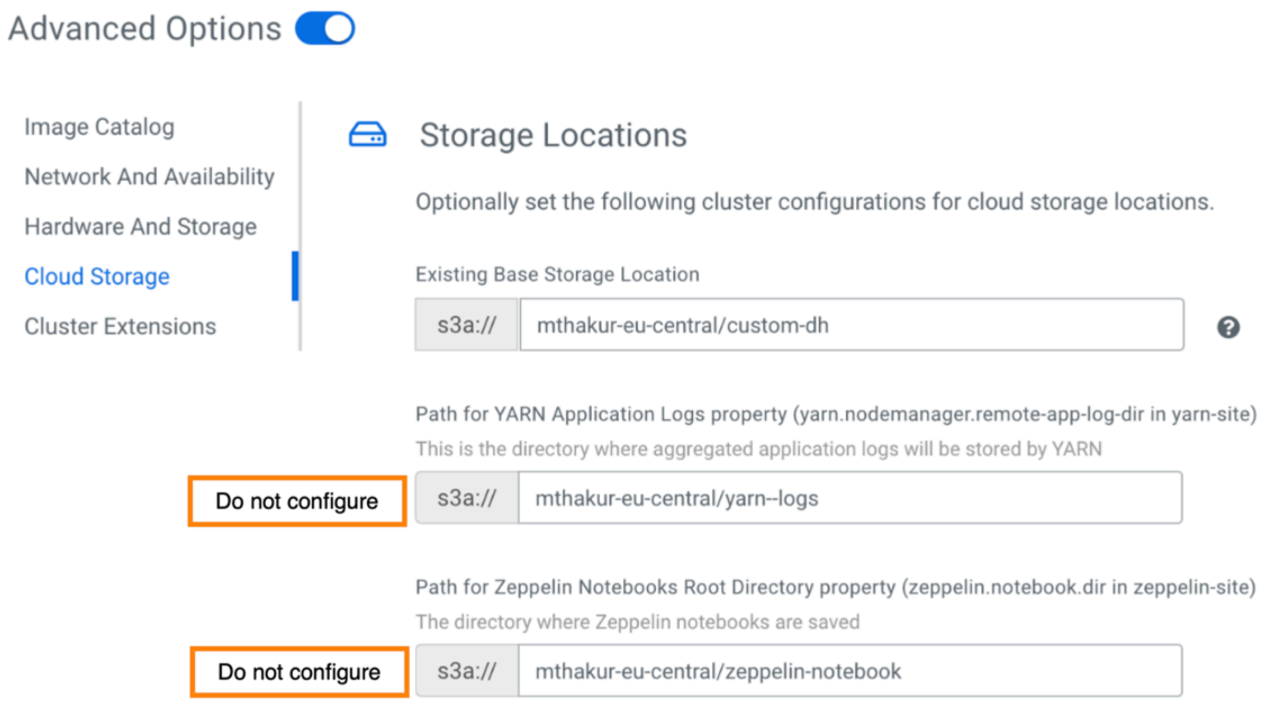
When you create a Cloudera Data Hub cluster with custom paths for Cloudera Data Hub clusters, your cluster might fail with an access exception. To resolve this issue, you must add the policy path to the default Zeppelin notebooks policy using Ranger.

About this task

Cloudera Data Hub cluster creation fails and a Zeppelin exception is logged in the Cloudera Manager logs, if you create a Cloudera Data Hub cluster with a custom path <bucket>/custom-dh, overwriting the YARN location and Zeppelin notebook location to <bucket>/yarn-logs and <bucket>/zeppelin-notebook respectively.

The following image shows the Create Cluster Advanced Options Cloud Storage page where you can optionally specify the base storage location used for YARN and Zeppelin:

Figure 1: Cloud Storage page



Storage Locations

Optionally set the following cluster configurations for cloud storage locations.

Existing Base Storage Location

s3a:// mthakur-eu-central/custom-dh

Path for YARN Application Logs property (yarn.nodemanager.remote-app-log-dir in yarn-site)

This is the directory where aggregated application logs will be stored by YARN

Do not configure

s3a:// mthakur-eu-central/yarn--logs

Path for Zeppelin Notebooks Root Directory property (zeppelin.notebook.dir in zeppelin-site)

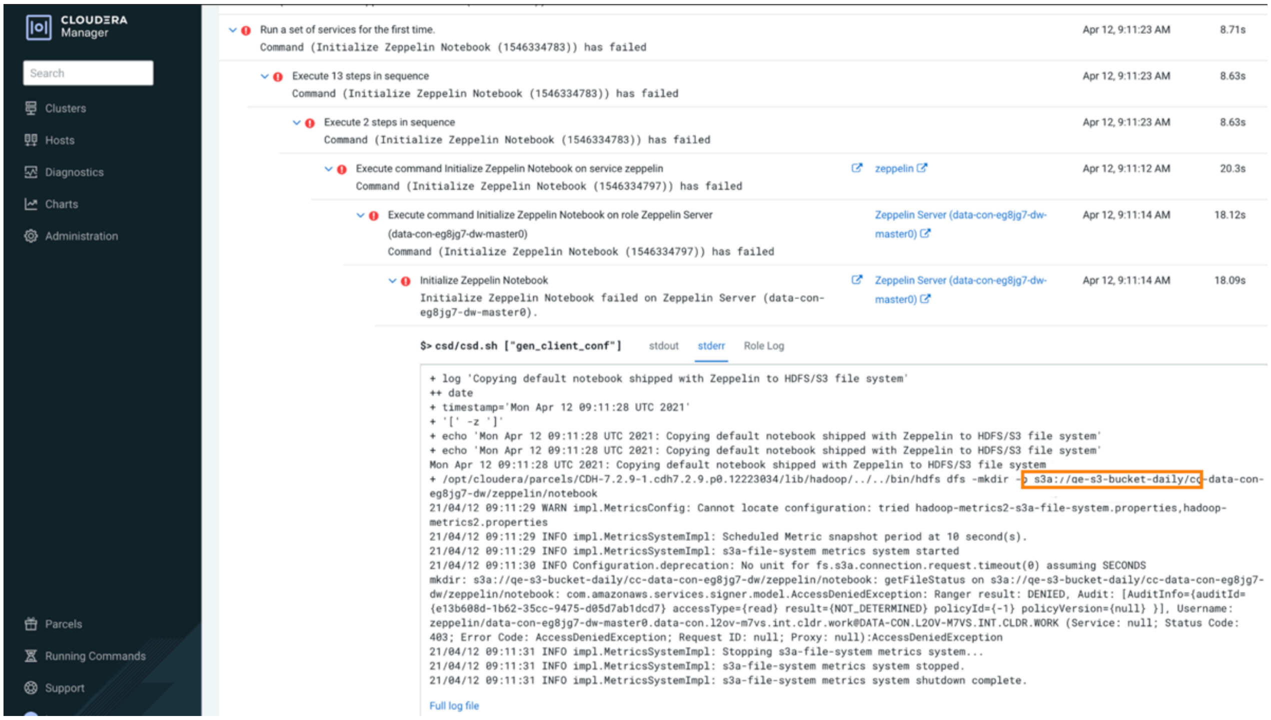
The directory where Zeppelin notebooks are saved

Do not configure

s3a:// mthakur-eu-central/zeppelin-notebook

The following image shows the Zeppelin exception in the Cloudera Manager logs:

Figure 2: Zeppelin exception in Cloudera Manager logs



Run a set of services for the first time. — Apr 12, 9:11:23 AM — 8.71s
Command (Initialize Zeppelin Notebook (1546334783)) has failed

Execute 13 steps in sequence — Apr 12, 9:11:23 AM — 8.63s
Command (Initialize Zeppelin Notebook (1546334783)) has failed

Execute 2 steps in sequence — Apr 12, 9:11:23 AM — 8.63s
Command (Initialize Zeppelin Notebook (1546334783)) has failed

Execute command Initialize Zeppelin Notebook on service zeppelin — zeppelin — Apr 12, 9:11:12 AM — 20.3s
Command (Initialize Zeppelin Notebook (1546334797)) has failed

Execute command Initialize Zeppelin Notebook on role Zeppelin Server (data-con-eg8jg7-dw-master0) — Zeppelin Server (data-con-eg8jg7-dw-master0) — Apr 12, 9:11:14 AM — 18.12s
Command (Initialize Zeppelin Notebook (1546334797)) has failed

Initialize Zeppelin Notebook — Zeppelin Server (data-con-eg8jg7-dw-master0) — Apr 12, 9:11:14 AM — 18.09s
Initialize Zeppelin Notebook failed on Zeppelin Server (data-con-eg8jg7-dw-master0).

$> csd/csd.sh ["gen_client_conf"]   stdout   stderr   Role Log

```
+ log 'Copying default notebook shipped with Zeppelin to HDFS/S3 file system'
++ date
+ timestamp='Mon Apr 12 09:11:28 UTC 2021'
+ '[' -z ']'
+ echo 'Mon Apr 12 09:11:28 UTC 2021: Copying default notebook shipped with Zeppelin to HDFS/S3 file system'
+ echo 'Mon Apr 12 09:11:28 UTC 2021: Copying default notebook shipped with Zeppelin to HDFS/S3 file system'
Mon Apr 12 09:11:28 UTC 2021: Copying default notebook shipped with Zeppelin to HDFS/S3 file system
+ /opt/cloudera/parcels/CDH-7.2.9-1.cdh7.2.9.p0.12223034/lib/hadoop/../../bin/hdfs dfs -mkdir -p s3a://qe-s3-bucket-daily/cc-data-con-
eg8jg7-dw/zeppelin/notebook
21/04/12 09:11:29 WARN impl.MetricsConfig: Cannot locate configuration: tried hadoop-metrics2-s3a-file-system.properties,hadoop-
metrics2.properties
21/04/12 09:11:29 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
21/04/12 09:11:29 INFO impl.MetricsSystemImpl: s3a-file-system metrics system started
21/04/12 09:11:30 INFO Configuration.deprecation: No unit for fs.s3a.connection.request.timeout(0) assuming SECONDS
mkdir: s3a://qe-s3-bucket-daily/cc-data-con-eg8jg7-dw/zeppelin/notebook: getFileStatus on s3a://qe-s3-bucket-daily/cc-data-con-eg8jg7-
dw/zeppelin/notebook: com.amazonaws.services.signer.model.AccessDeniedException: Ranger result: DENIED, Audit: [AuditInfo={auditId=
{e13b608d-1b62-35cc-9475-d05d7ab1dcd7} accessType={read} result={NOT_DETERMINED} policyId={-1} policyVersion={null} }], Username:
zeppelin/data-con-eg8jg7-dw-master0.data-con.l2ov-m7vs.int.cldr.work@DATA-CON.L2OV-M7VS.INT.CLDR.WORK (Service: null; Status Code:
403; Error Code: AccessDeniedException; Request ID: null; Proxy: null):AccessDeniedException
21/04/12 09:11:31 INFO impl.MetricsSystemImpl: Stopping s3a-file-system metrics system...
21/04/12 09:11:31 INFO impl.MetricsSystemImpl: s3a-file-system metrics system stopped.
21/04/12 09:11:31 INFO impl.MetricsSystemImpl: s3a-file-system metrics system shutdown complete.
```

Full log file

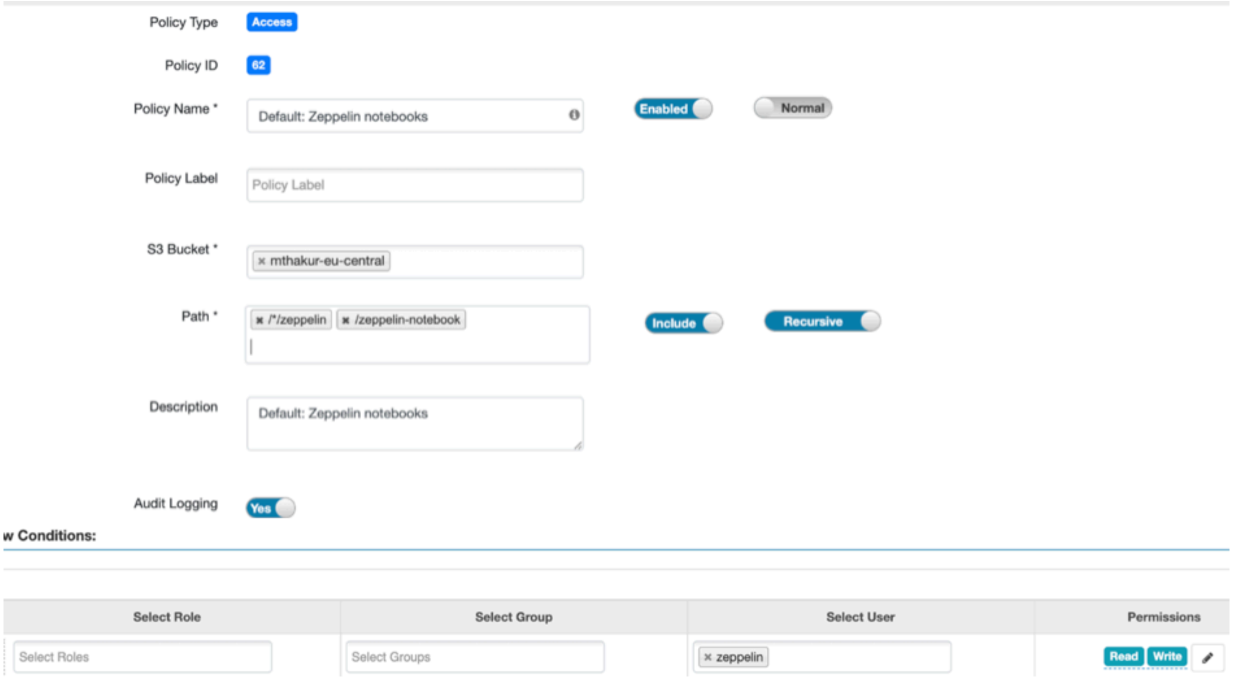
To resolve this issue, perform the following steps:

Procedure

1. Add the policy path to the default Zeppelin notebooks policy using Ranger.

   The following image shows the options to add the policy path to the Zeppelin notebooks policy in Ranger UI:

   Figure 3: Policy path in Ranger UI



Policy Type: Access

Policy ID: 62

Policy Name *: Default: Zeppelin notebooks — Enabled — Normal

Policy Label: Policy Label

S3 Bucket *: × mthakur-eu-central

Path *: × /*/zeppelin × /zeppelin-notebook — Include — Recursive

Description: Default: Zeppelin notebooks

Audit Logging: Yes

w Conditions:

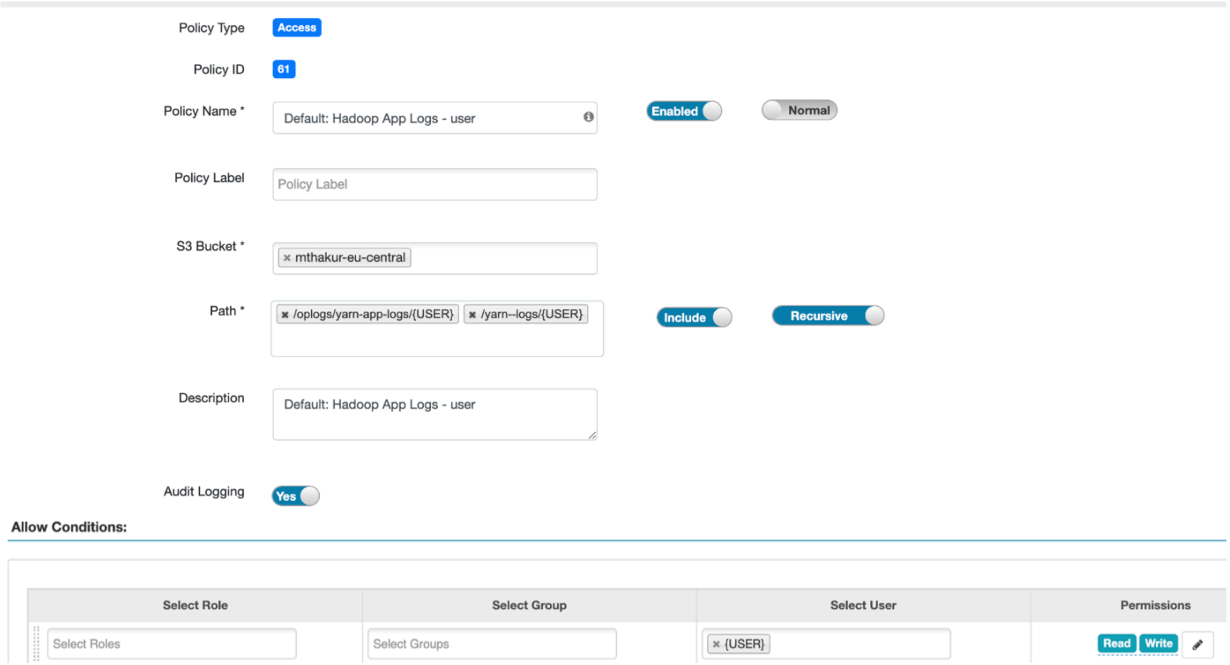
Select Role	Select Group	Select User	Permissions
Select Roles	Select Groups	× zeppelin	Read Write

2. For YARN logs aggregation, add the following using Ranger:

    - Add /yarn--logs/{USER} path to the Default: Hadoop App Logs - user default policy.
    - Add /yarn--logs path to the Default: Hadoop App Logs default policy.

    The following sample image shows the update required in the Default: Hadoop App Logs - user default policy:

    Figure 4: Default: Hadoop App Logs - user default policy



Policy Type: Access

Policy ID: 61

Policy Name *: Default: Hadoop App Logs - user

Policy Label: Policy Label

S3 Bucket *: × mthakur-eu-central

Path *: × /oplogs/yarn-app-logs/{USER} × /yarn--logs/{USER}

Description: Default: Hadoop App Logs - user

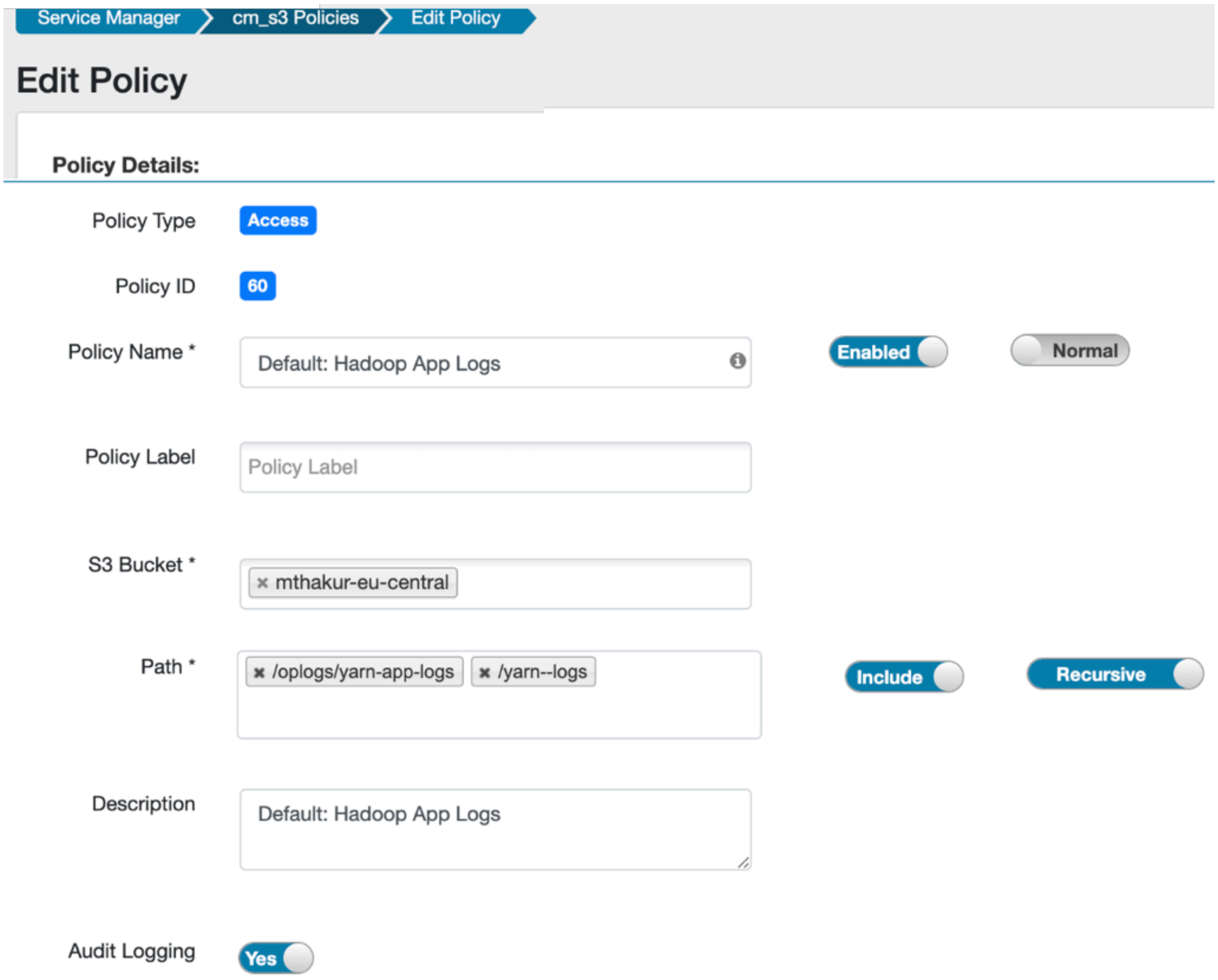
Audit Logging: Yes

Allow Conditions:

Select Role	Select Group	Select User	Permissions
Select Roles	Select Groups	× {USER}	Read Write

The following sample image shows the update required in the Default: Hadoop App Logs default policy:

Figure 5: Default: Hadoop App Logs default policy



Edit Policy

Policy Details:

Policy Type: Access

Policy ID: 60

Policy Name *: Default: Hadoop App Logs — Enabled — Normal

Policy Label: Policy Label

S3 Bucket *: mthakur-eu-central

Path *: /oplogs/yarn-app-logs, /yarn--logs — Include — Recursive

Description: Default: Hadoop App Logs

Audit Logging: Yes

3. After adding the policies, perform the following steps:

a. In Cloudera Manager, re-run the failed commands.

b. After the commands run successfully, go to the Cloudera Management Console Data Hub Clusters page, and click Actions Retry Yes option.

The operation continues from the last failed step.

Provisioning Cloudera AI workbench for RAZ-enabled AWS environment

If you require Cloudera AI, you must provision a Cloudera AI workbench. A Cloudera AI workbench enables teams of data scientists to develop, test, train, and ultimately deploy machine learning models for building predictive applications all on the data under management within the enterprise data cloud.

For information about provisioning a Cloudera AI workbench, see Provisioning Cloudera AI workbenches.

Ranger policy options for RAZ-enabled AWS environment

After you register the RAZ-enabled AWS environment, you can log in to Ranger to create the policies for granular access to the environment's cloud storage location.

Ranger includes a set of preloaded resource-based services and policies. You need the following additional policies for granular access to the environment's cloud storage location:

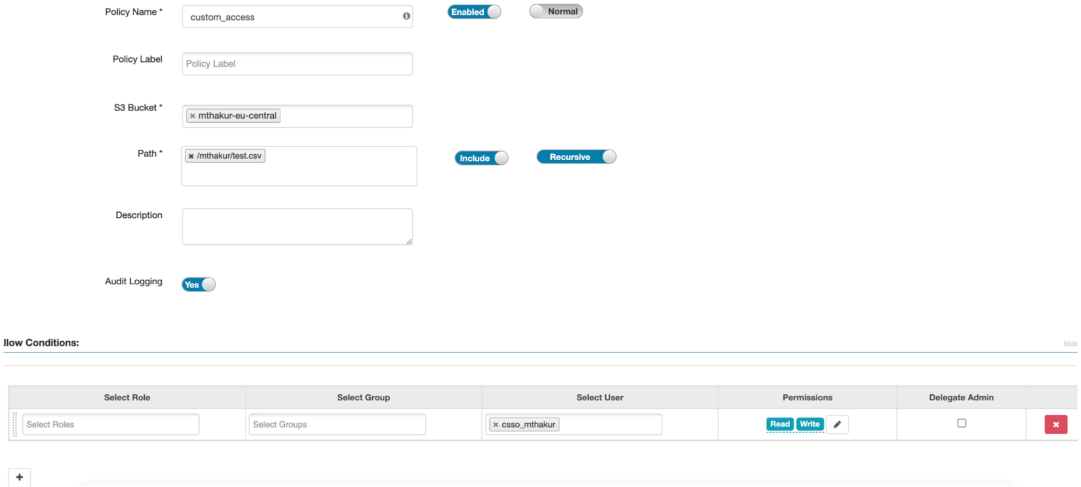
- Policies for Spark jobs
- Policies for Hive external tables and Spark jobs
- Policies for Hive managed tables and Spark jobs

Policies for Spark jobs

A Spark job on an S3 path requires an S3 policy for the end user on the specific path. For information to create the policies, see Creating Ranger policy to use in RAZ-enabled AWS environment on page 13.

For example, a Spark job on s3a://bucket/data/logs/tabledata requires an S3 policy in cm_s3 repo on s3a://bucket/data/logs/tabledata for end user.

The following sample image shows the S3 policy created in the cm_s3 repo for the user csso_abc to read and write data in s3a://abc-eu-central/abc/test.csv:



Policy Name * custom_access Enabled Normal

Policy Label

S3 Bucket * × mthakur-eu-central

Path * × /mthakur/test.csv Include Recursive

Description

Audit Logging Yes

llow Conditions:

Select Role	Select Group	Select User	Permissions	Delegate Admin	
Select Roles	Select Groups	× csso_mthakur	Read Write	□	×

Policies for Hive external tables and Spark jobs

Running the create external table *[***TABLE DEFINITION***]* location 's3a://bucket/data/logs/tabledata' command in Hive requires the following Ranger policies:

- An S3 policy in the cm_s3 repo on s3a://bucket/data/logs/tabledata for hive user to perform recursive read/write.
- An S3 policy in the cm_s3 repo on s3a://bucket/data/logs/tabledata for the end user.
- A Hive URL authorization policy in the Hadoop SQL repo on s3a://bucket/data/logs/tabledata for the end user.

Access to the same external table location using Spark shell requires an S3 policy (Ranger policy) in the cm_s3 repo on s3a://bucket/data/logs/tabledata for the end user.

For information to create the policies, see Creating Ranger policy to use in RAZ-enabled AWS environment on page 13.

Policies for Hive managed tables and Spark jobs

Operations such as create, insert, delete, select, and so on, on a Hive managed table do not require any custom Ranger policies.

For information to create the policies, see Creating Ranger policy to use in RAZ-enabled AWS environment on page 13.

Creating Ranger policy to use in RAZ-enabled AWS environment

After you register the RAZ-enabled AWS environment, you can log in to Ranger to create the policies for granular access to the environment's cloud storage location. To create the Ranger policy, you must first create the required S3 policy and then a Hive URL authorization policy, on an S3 path for the end user.

Procedure

1. To create the required S3 policy on an S3 path for end user, perform the following steps:

   a) Navigate to the Ranger UI.

   b) On the S3 tab, click cm_s3.

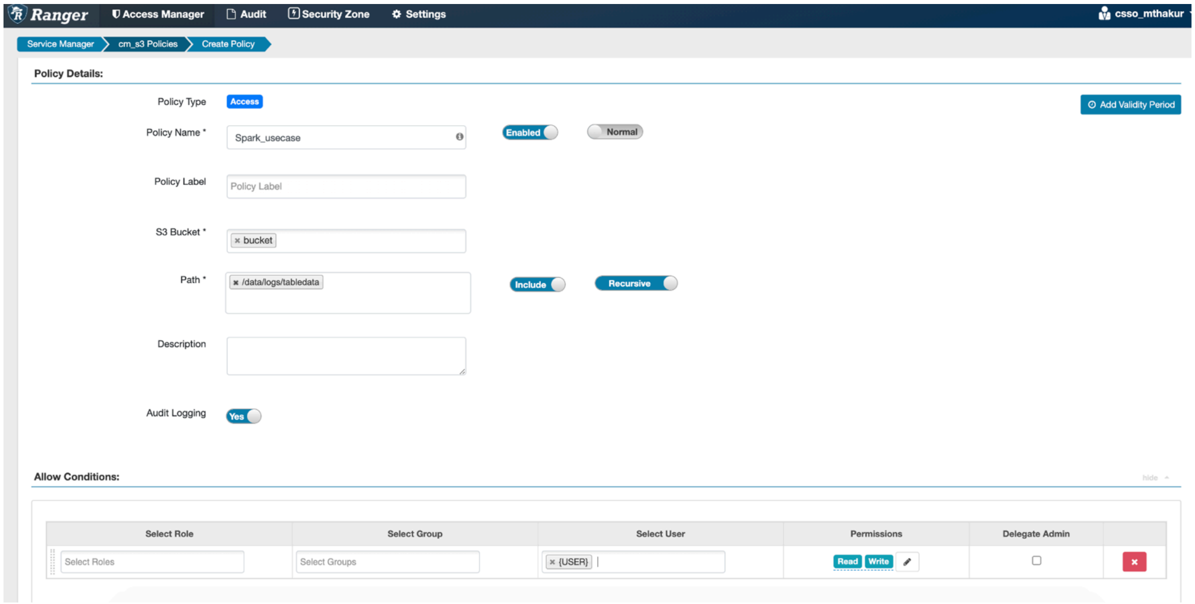
   c) Click Add New Policy in the top right corner.

   d) Provide the following policy details:

      1. Enter Policy Name.
      2. Enter an S3 Bucket name.
      3. Provide a Path within the S3 bucket.
      4. Select users and permissions to assign to the end user.

         Only Read and Write permissions can be assigned to the end user.

      The following sample image shows the **Create Policy** page in Ranger UI to create an S3 policy on an S3 path for an end user.



Policy Details:

Policy Type: Access

Policy Name *: Spark_usecase — Enabled — Normal

Policy Label: Policy Label

S3 Bucket *: × bucket

Path *: × /data/logs/tabledata — Include — Recursive

Description:

Audit Logging: Yes

Allow Conditions:

Select Role	Select Group	Select User	Permissions	Delegate Admin	
Select Roles	Select Groups	× {USER}	Read Write	□	×

   e) Click Add to save the policy.

2. To create a Hive URL authorization policy on an S3 path for the end user, perform the following steps:

   a) Navigate to the Ranger UI.

   b) On the Hadoop SQL tab, click Hadoop SQL.

   c) Click Add New Policy in the top right corner.

   d) Provide the policy details. The following sample image shows the policy details:

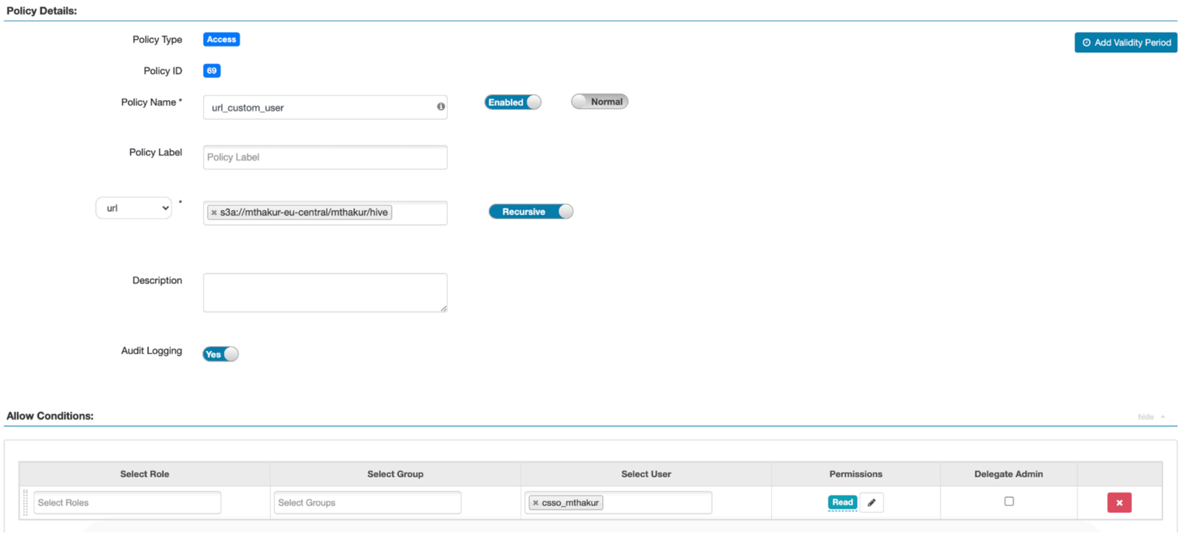
      1. Enter Policy Name.
      2. Enter the Hive URL authorization path in the url field, and enable the Recursive option.
      3. Provide a Path within the S3 bucket.

         Note: You can append *, also known as a "wildcard", to the path name. For example: s3a://bucket/* . Appending * to a URL path grants (or denies) access to the child directories in that path.

      4. Select users and permissions to assign to the end user.

         Note: Only Read permission can be assigned to the end user.

      The following sample image shows the Policy Details page in Ranger UI to create a Hive URL authorization policy on an S3 path for an end user.



Policy Details:

Policy Type: Access

Policy ID: 69

Policy Name *: url_custom_user

Enabled

Normal

url *: × s3a://mthakur-eu-central/mthakur/hive

Recursive

Policy Label: Policy Label

Description:

Audit Logging: Yes

Allow Conditions:

Select Role	Select Group	Select User	Permissions	Delegate Admin	
Select Roles	Select Groups	× csso_mthakur	Read	□	×

   e) Click Add to save the policy.

Troubleshooting for RAZ-enabled AWS environment

This section includes common errors that might occur while using a RAZ-enabled AWS environment and the steps to resolve the issues.

Why does the "AccessDeniedException" error appear? How do I resolve it?

Complete error snippet:

```
org.apache.hadoop.hive.ql.metadata.HiveException: java.nio.file.AccessDenied
Exception:
s3a://abc-eu-central/abc/hive: getFileStatus on s3a://abc-eu-central/abc/hiv
e:
```

```
com.amazonaws.services.signer.model.AccessDeniedException: Ranger result: DE
NIED, Audit:
[AuditInfo={auditId={null} accessType={read} result={NOT_DETERMINED} polic
yId={-1} policyVersion={null} }],
Username: hive/demo-dh-master0.abc.xcu2-8y8x.dev.cldr.work@abc.XCU2-8Y8X.
DEV.CLDR.WORK (Service: null;
Status Code: 403; Error Code: AccessDeniedException; Request ID: f137c7b2
-4448-4599-b75e-d96ae9308c5b;
Proxy: null):AccessDeniedException
```

Cause

This error appears when the hive user does not have the required S3 policy.

Remedy

Add the required policy for the user.

For more information, see Ranger policy options for RAZ-enabled AWS environment on page 11.

Why does the "Permission denied" error appear? How do I resolve it?

Complete error snippet:

```
Error: Error while compiling statement:
FAILED: HiveAccessControlException Permission denied: user [csso_abc] does
not have
[READ] privilege on [s3a://abc-eu-central/abc/hive] (state=42000,code=40000)
```

Cause

This error appears if you did not add the required Hive URL authorization policy to the user.

Remedy

Procedure

Add the required policy for the user.

For more information, see Ranger policy options for RAZ-enabled AWS environment on page 11.

Why does the "S3 access in hive/spark/mapreduce fails despite having an "allow" policy" error message appear? How do I resolve it?

Complete error snippet:

```
S3 access in hive/spark/mapreduce fails despite having an "allow" policy def
ined for the path
with ‘java.nio.file.AccessDeniedException: s3a://sshiv-cdp-bucket/data/ex
ttable/000000_0: getFileStatus on
s3a://sshivalingamurthy-cdp-bucket/data/exttable/000000_0: com.amazonaws.
services.signer.model.AccessDeniedException:
Ranger result: NOT_DETERMINED, Audit: [AuditInfo={auditId={a6904660-1a0f-3
149-8a0b-c0792aec3e19} accessType={read}
result={NOT_DETERMINED} policyId={-1} policyVersion={null} }] (Service: nu
ll; Status Code: 403; Error Code: AccessDeniedException;
Request ID: null):AccessDeniedException’
```

Cause

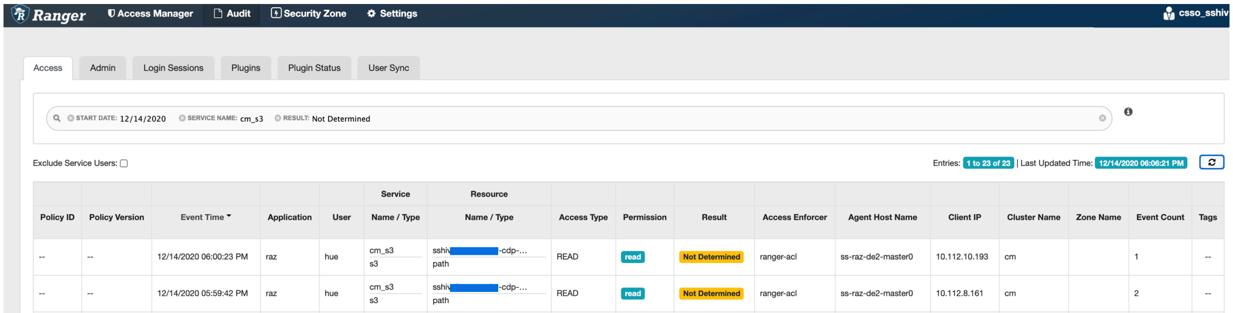
In some cases, RAZ S3 requires a non-recursive READ access on the parent S3 path. This error appears when the non-recursive READ access is not provided to the parent S3 path.

Remedy

Procedure

1. On the Ranger Audit page, track the user and the path where authorization failed.
2. Add the missing Ranger policy to the end user.

   The following sample image shows the **Access** tab on the Ranger Audit page where you can track the user and the path:



START DATE: 12/14/2020 SERVICE NAME: cm_s3 RESULT: Not Determined

Exclude Service Users: □

Entries: 1 to 23 of 23 | Last Updated Time: 12/14/2020 06:06:21 PM

					Service	Resource										
Policy ID	Policy Version	Event Time	Application	User	Name / Type	Name / Type	Access Type	Permission	Result	Access Enforcer	Agent Host Name	Client IP	Cluster Name	Zone Name	Event Count	Tags
--	--	12/14/2020 06:00:23 PM	raz	hue	cm_s3 s3	sshiv[redacted]-cdp-... path	READ	read	Not Determined	ranger-acl	ss-raz-de2-master0	10.112.10.193	cm		1	--
--	--	12/14/2020 05:59:42 PM	raz	hue	cm_s3 s3	sshiv[redacted]-cdp-... path	READ	read	Not Determined	ranger-acl	ss-raz-de2-master0	10.112.8.161	cm		2	--

An error related to Apache Flink appears after the job fails. How do I resolve this issue?

Complete error snippet:

```
2021-06-01 00:00:18,872 INFO  org.apache.flink.runtime.entrypoint.ClusterEnt
rypoint
- Shutting YarnJobClusterEntrypoint down with application status FAILED. Di
agnostics java.nio.file.AccessDeniedException:
s3a://jon-s3-raz/data/flink/araujo-flink/ha/application_1622436888784_0031/b
lob: getFileStatus on
s3a://jon-s3-raz/data/flink/araujo-flink/ha/application_1622436888784_0031/
blob: com.amazonaws.services.signer.model.AccessDeniedException:
Ranger result: DENIED, Audit: [AuditInfo={auditId={84aba1b3-82b5-3d0c-a05b-8
f5512e0fd36} accessType={read} result={NOT_DETERMINED}
policyId={-1} policyVersion={null} }], Username: srv_kafka-client@PM-AWS-R.A
465-4K.CLOUDERA.SITE (Service: null; Status Code: 403;
Error Code: AccessDeniedException; Request ID: null; Proxy: null):AccessDeni
edException
```

Cause

This error appears if there is no policy granting the necessary access to the /data/flink/araujo-flink/ha path.

Remedy

Procedure

Add the policy to grant the required access to the /data/flink/araujo-flink/ha path.

To configure the policy, see Configuring Ranger policies for Flink.

What do I do to display the Thread ID in logs for RAZ?

When you enable the DEBUG level for RAZ, a detailed verbose log is generated. It is cumbersome and tedious to identify the issue pertaining to a specific RAZ authorization request; therefore, Thread ID is good information to capture in the debug log.

Remedy

Procedure

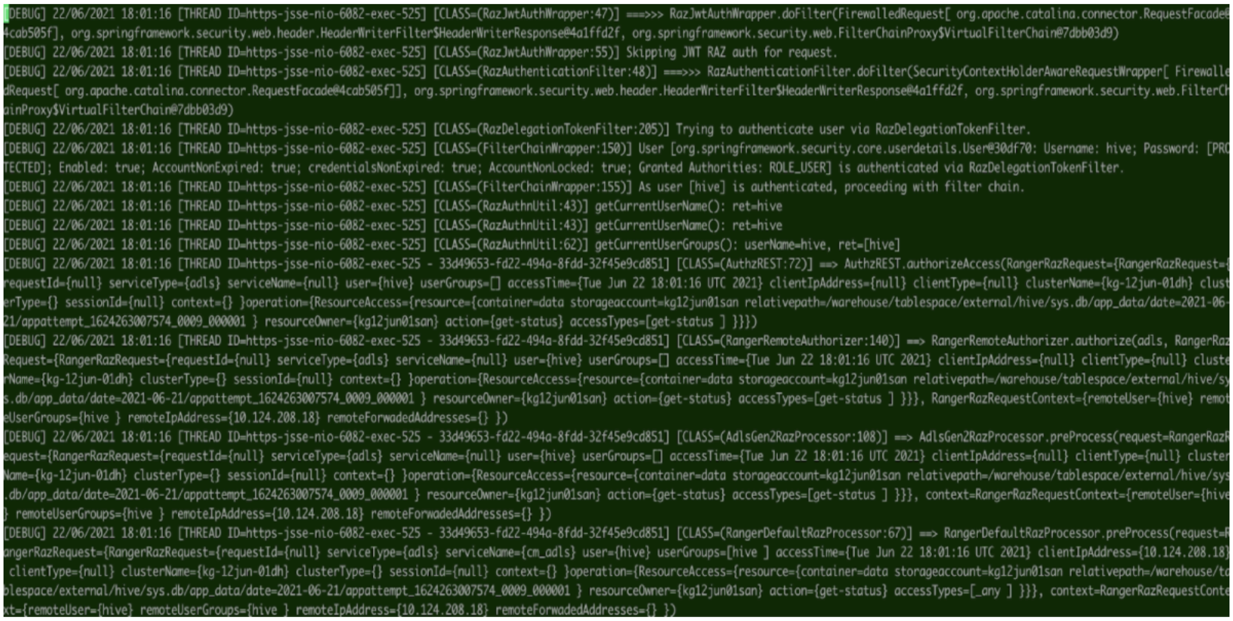
1. In Cloudera Manager, go to the Ranger RAZ service on the Configuration tab.
2. Search for the Ranger Raz Server Logging Advanced Configuration Snippet (Safety Valve) property, and enter the following information:

```
loggers=AUDIT, METRICS, RANGERRAZ
logger.RANGERRAZ.name=<PACKAGE_OR_FULLY_QUALIFIED_CLASS_NAME>
logger.RANGERRAZ.level=DEBUG
logger.RANGERRAZ.additivity=false
logger.RANGERRAZ.appenderRef.RANGERRAZ.ref=DRFA
```

3. Search for the Ranger Raz Server Max Log Size property, enter 200, and choose MiB.
4. Click Save Changes.
5. Restart the Ranger RAZ service.

The service prints the debug log details with Thread ID. To debug an issue, you can filter the logs based on the Thread ID.

The sample snippet shows the log generated with Thread ID:



```
[DEBUG] 22/06/2021 18:01:16 [THREAD ID=https-jsse-nio-6082-exec-525] [CLASS=(RazJwtAuthWrapper:47)] ===>>> RazJwtAuthWrapper.doFilter(FirewalledRequest[ org.apache.catalina.connector.RequestFacade@
4cab505f], org.springframework.security.web.header.HeaderWriterFilter$HeaderWriterResponse@4a1ffd2f, org.springframework.security.web.FilterChainProxy$VirtualFilterChain@7dbb03d9)
[DEBUG] 22/06/2021 18:01:16 [THREAD ID=https-jsse-nio-6082-exec-525] [CLASS=(RazJwtAuthWrapper:55)] Skipping JWT RAZ auth for request.
[DEBUG] 22/06/2021 18:01:16 [THREAD ID=https-jsse-nio-6082-exec-525] [CLASS=(RazAuthenticationFilter:48)] ===>>> RazAuthenticationFilter.doFilter(SecurityContextHolderAwareRequestWrapper[ Firewalle
dRequest[ org.apache.catalina.connector.RequestFacade@4cab505f]], org.springframework.security.web.header.HeaderWriterFilter$HeaderWriterResponse@4a1ffd2f, org.springframework.security.web.FilterCh
ainProxy$VirtualFilterChain@7dbb03d9)
[DEBUG] 22/06/2021 18:01:16 [THREAD ID=https-jsse-nio-6082-exec-525] [CLASS=(RazDelegationTokenFilter:205)] Trying to authenticate user via RazDelegationTokenFilter.
[DEBUG] 22/06/2021 18:01:16 [THREAD ID=https-jsse-nio-6082-exec-525] [CLASS=(FilterChainWrapper:150)] User [org.springframework.security.core.userdetails.User@30df70: Username: hive; Password: [PRO
TECTED]; Enabled: true; AccountNonExpired: true; credentialsNonExpired: true; AccountNonLocked: true; Granted Authorities: ROLE_USER] is authenticated via RazDelegationTokenFilter.
[DEBUG] 22/06/2021 18:01:16 [THREAD ID=https-jsse-nio-6082-exec-525] [CLASS=(FilterChainWrapper:155)] As user [hive] is authenticated, proceeding with filter chain.
[DEBUG] 22/06/2021 18:01:16 [THREAD ID=https-jsse-nio-6082-exec-525] [CLASS=(RazAuthnUtil:43)] getCurrentUserName(): ret=hive
[DEBUG] 22/06/2021 18:01:16 [THREAD ID=https-jsse-nio-6082-exec-525] [CLASS=(RazAuthnUtil:43)] getCurrentUserName(): ret=hive
[DEBUG] 22/06/2021 18:01:16 [THREAD ID=https-jsse-nio-6082-exec-525] [CLASS=(RazAuthnUtil:62)] getCurrentUserGroups(): userName=hive, ret=[hive]
[DEBUG] 22/06/2021 18:01:16 [THREAD ID=https-jsse-nio-6082-exec-525 - 33d49653-fd22-494a-8fdd-32f45e9cd851] [CLASS=(AuthzREST:72)] ==> AuthzREST.authorizeAccess(RangerRazRequest={RangerRazRequest={
requestId={null} serviceType={adls} serviceName={null} user={hive} userGroups=[] accessTime={Tue Jun 22 18:01:16 UTC 2021} clientIpAddress={null} clientType={null} clusterName={kg-12jun-01dh} clust
erType={} sessionId={null} context={} }operation={ResourceAccess={resource={container=data storageaccount=kg12jun01san relativepath=/warehouse/tablespace/external/hive/sys.db/app_data/date=2021-06-
21/appattempt_1624263007574_0009_000001 } resourceOwner={kg12jun01san} action={get-status} accessTypes=[get-status ] }}})
[DEBUG] 22/06/2021 18:01:16 [THREAD ID=https-jsse-nio-6082-exec-525 - 33d49653-fd22-494a-8fdd-32f45e9cd851] [CLASS=(RangerRemoteAuthorizer:140)] ==> RangerRemoteAuthorizer.authorize(adls, RangerRaz
Request={RangerRazRequest={requestId={null} serviceType={adls} serviceName={null} user={hive} userGroups=[] accessTime={Tue Jun 22 18:01:16 UTC 2021} clientIpAddress={null} clientType={null} cluste
rName={kg-12jun-01dh} clusterType={} sessionId={null} context={} }operation={ResourceAccess={resource={container=data storageaccount=kg12jun01san relativepath=/warehouse/tablespace/external/hive/sy
s.db/app_data/date=2021-06-21/appattempt_1624263007574_0009_000001 } resourceOwner={kg12jun01san} action={get-status} accessTypes=[get-status ] }}}, RangerRazRequestContext={remoteUser={hive} remot
eUserGroups={hive } remoteIpAddress={10.124.208.18} remoteForwadedAddresses={} })
[DEBUG] 22/06/2021 18:01:16 [THREAD ID=https-jsse-nio-6082-exec-525 - 33d49653-fd22-494a-8fdd-32f45e9cd851] [CLASS=(AdlsGen2RazProcessor:108)] ==> AdlsGen2RazProcessor.preProcess(request=RangerRazR
equest={RangerRazRequest={requestId={null} serviceType={adls} serviceName={null} user={hive} userGroups=[] accessTime={Tue Jun 22 18:01:16 UTC 2021} clientIpAddress={null} clientType={null} cluster
Name={kg-12jun-01dh} clusterType={} sessionId={null} context={} }operation={ResourceAccess={resource={container=data storageaccount=kg12jun01san relativepath=/warehouse/tablespace/external/hive/sys
.db/app_data/date=2021-06-21/appattempt_1624263007574_0009_000001 } resourceOwner={kg12jun01san} action={get-status} accessTypes=[get-status ] }}}, context=RangerRazRequestContext={remoteUser={hive
} remoteUserGroups={hive } remoteIpAddress={10.124.208.18} remoteForwadedAddresses={} })
[DEBUG] 22/06/2021 18:01:16 [THREAD ID=https-jsse-nio-6082-exec-525 - 33d49653-fd22-494a-8fdd-32f45e9cd851] [CLASS=(RangerDefaultRazProcessor:67)] ==> RangerDefaultRazProcessor.preProcess(request=R
angerRazRequest={RangerRazRequest={requestId={null} serviceType={adls} serviceName={cm_adls} user={hive} userGroups=[hive ] accessTime={Tue Jun 22 18:01:16 UTC 2021} clientIpAddress={10.124.208.18}
 clientType={null} clusterName={kg-12jun-01dh} clusterType={} sessionId={null} context={} }operation={ResourceAccess={resource={container=data storageaccount=kg12jun01san relativepath=/warehouse/ta
blespace/external/hive/sys.db/app_data/date=2021-06-21/appattempt_1624263007574_0009_000001 } resourceOwner={kg12jun01san} action={get-status} accessTypes=[_any ] }}}, context=RangerRazRequestConte
xt={remoteUser={hive} remoteUserGroups={hive } remoteIpAddress={10.124.208.18} remoteForwadedAddresses={} })
```

What do I do when a long-running job consistently fails with the expired token error?

Sample error snippet - "Caused by: com.amazonaws.services.s3.model.AmazonS3Exception: The provided token has expired. (Service: Amazon S3; Status Code: 400; Error Code: ExpiredToken; Request ID: xxxx; S3 Extended Request ID:xxxx..."

Cause

This issue appears when the S3 Security token has expired for the s3 call.

Remedy

Procedure

1. Log in to the Cloudera Management Console as an Administrator and go to your environment.
2. From the Data Lake tab, open Cloudera Manager.
3. Go to the Clusters *HDFS SERVICE* Configurations tab.
4. Search for the core-site.xml file corresponding to the required Cloudera Data Hub cluster.
5. Open the file and enter fs.s3a.signature.cache.max.size=0 to disable the signature caching in Ranger RAZ.
6. Save and close the file.
7. On the Home Status tab, click Actions to the right of the cluster name and select Deploy Client Configuration.
8. Click Deploy Client Configuration.
9. Restart the HDFS service.

Note: Ensure that there are no stale configurations in Ranger RAZ and HDFS services.

Managing a RAZ-enabled AWS environment

You can manage a RAZ-enabled environment in a similar manner as any other Cloudera environment.

For information on how to manage and monitor Cloudera environments running in AWS, refer to Working with AWS environments.