

Control Plane Auditing

Date published: 2019-08-22

Date modified:



Legal Notice

© Cloudera Inc. 2024. All rights reserved.

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

Auditing Control Plane activity.....	4
Control Plane auditing data model.....	4
CDP service event.....	5
API request event.....	6
Interactive login event.....	6
CDP service event sources and names.....	6
Retrieving audit events.....	8
AWS setup for audit archiving.....	10
Configuring audit event archiving through the UI.....	10
Setting up an AWS policy and role to configure archiving using the CLI.....	12
Creating an AWS credential for audit event archiving using the CLI.....	14
Setting up audit archiving in AWS using the CLI.....	14
Azure setup for audit archiving.....	16
Creating a storage account in Azure.....	16
Creating a container in Azure.....	19
Configuring audit event archiving through the UI.....	20
Creating a CDP credential for audit archiving on Azure using the CLI.....	23
Configuring audit archiving for Azure using the CLI.....	24
GCP setup for audit archiving.....	25
Configuring audit event archiving through the UI.....	25
Configure GCP audit event archiving using the CLI.....	27
Pull-based audit archiving.....	28
What archiving looks like.....	30

Auditing Control Plane activity

Auditing is used to collect or log evidence of activity in a system that auditors can use to both track and analyze to answer questions such as: Who made a change to the system? When did a change happen? What exactly changed? Why was a change authorized?

Control Plane auditing is based on the concept of an audit event. An audit event is a record of an audited action which is typically a change in the system that is important enough to keep a record of. However, even some read-only actions are audited, because it might be important to know who was able to see information in the system, and not just who could alter it.

Control Plane auditing is scoped to actions that occur within the CDP Control Plane. Audit events are not collected from workload clusters; in fact, many Control Plane audit events are collected without the need for any workload clusters to exist.

The auditing system initially stores generated audit events into a cloud provider managed database. After a specific amount of time, audit logs are exported to customer-managed storage, in their own cloud provider.

The audit records are kept in the system for a maximum of six months. After the six-month period, the records are removed from the internal storage of CDP regardless of the archive status. In case archiving is enabled and the access to the destination is lost, the archiving process will be retried for any records that are not archived until the access is restored or the six-month limit is reached. The auto-archiving process can be enabled or disabled as necessary. Once the archiving is enabled after disabling, all of the records that have not been archived will be archived regardless of age. The pull-based audit archiving can be used in case the automatic archiving is disabled.

Related Information

[Control Plane auditing data model](#)

[CDP service event sources and names](#)

[Retrieving audit events](#)

[AWS setup for audit archiving](#)

[Azure setup for audit archiving](#)

[What archiving looks like](#)

Control Plane auditing data model

There are three categories of audit events.

CDP service events

For actions that a service within the CDP Control Plane undertakes. These actions are often as a result of human activity, but can also result from autonomous processes within the Control Plane.

API request events

For calls to public API endpoints. These events are analogous to access logs kept by web servers. Because an API call often leads to actions within the CDP Control Plane, an API request often connects to one or more CDP service events.

Interactive login events

For logins to the CDP Control Plane.

All three categories of audit events share the following common fields:

Field name	Description	Example
version	Version of the audit event model	1.0.0
ID	Unique identifier for the event	a random UUID

source	Control plane service that submitted the event	iam
name	Type of action being audited	createGroupServiceEvent
timestamp	Time when the action occurred	2020-03-18T01:02:03Z
actor identity	Who initiated the action	see below
account ID	Identifier for account within which the action occurred	a UUID
request ID	Identifier for API request that led to the action	a UUID
result code	A string describing the result of the action, whether successful or not	INVALID_ARGUMENT
result message	A short message further describing the result	The group already exists

Each event source (service) defines its own event names. So, two sources may emit events with the same name, but for different actions. Events for an action within one source always use the same name.

An actor is an entity that causes an action to occur. In an audit event, an actor may be specified one of two ways.

Actor CRN

For a human actor, or for the special "internal" actor

Actor service name

For an antonymous process initiated by a control plane service (reserved for future use)

It is possible for the actor service name and the source in an audit event to be different. For example, a high-level service A may kick off an autonomous process that makes calls to another service B to make changes; audit events from that process would have actor service name A but source B.

Every call to the Control Plane public API receives a request ID. The request ID propagates through the Control Plane to services that perform actions, and audit events from those services include the request ID. Therefore, a request ID can be used to tie together multiple audit events under the umbrella of a single API request.

Most audit events include result information, but sometimes that information may be missing. This indicates that the event source experienced a failure such that it could not submit the result information for an event after submitting its initial, known set of information.

For example, consider the action to grant a role to a user. The event source responsible for this action starts by submitting an audit event for role creation, including all the information known before attempting the action: the user CRN, the role CRN, and perhaps more. After role creation either succeeds or fails, the source appends result information to the event. However, if the source crashes, it cannot append the result information. When this happens, at least the initial, known information is recorded in an audit event.

CDP service event

A CDP service event contains additional fields.

Field name	Description	Example
details version	Schema version for the additional details content	2020-03-31
additional details	JSON containing additional event-specific information	{ "groupCrn" : "crn:..." }
resource CRNs	CRNs for an affected resource, if applicable (may be multiple values)	crn:altus:iam:us-west-1:altus:role:PowerUser

Usually there is much more information about an action than can fit into the common audit event fields. The additional details field holds that additional information in a structured way. Each event source defines the details structure for each type of event (by event source / name) it generates.

API request event

An API request event contains additional fields.

Field name	Description	Example
request parameters	JSON format of API request	{ "param1": ... }
response parameters	JSON format of API response	{ "param1": ... }
mutating	Boolean indicating if this API call changes resource data	true
apiVersion	Version of the API called	2020-03-31
source IP address	IP address from where the request originated	192.168.0.1
user agent	User agent string of the request	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:74.0) Gecko/20100101 Firefox/74.0

The request ID field can be used to tie an API request event to corresponding CDP service events, because CDP Control Plane services propagate the request ID initially generated for an API request through all resulting service activity.

Interactive login event

There are sufficient additional details about Control Plane login events that they merit their own category. An interactive login event contains the following additional fields. Here, an identity provider is an authentication system outside of the Control Plane that keeps identity information about users, such as Okta.

Field name	Description	Example
identity provider CRN	CRN of the identity provider as known by the control plane	crn:altus:iam:us-west-1:altus:samlProvider:cloudera-ssso
identity provider session ID	identifier assigned to the login session by the identity provider	TBD
identity provider user ID	identifier of the user as stored in the identity provider	spongebob@cloudera.com (for Cloudera SSO, email is used)
email	email address of the user logging in	spongebob@cloudera.com
first name	first (given) name of the user logging in	Spongebob
last name	last name (family name / surname) of the user logging in	Squarepants
account admin	a Boolean flag indicating if the login is for an administrative user	true
groups	names of groups to which the user belongs in their account	TBD
source IP address	IP address of the user logging in	192.168.0.1
user CRN	CRN of the user logging in	crn:...

The account admin flag is only available if login succeeds. The user CRN is not known until login is attempted, and the CRN is not always recorded when login fails (for example, if the user's account cannot be determined by the Control Plane). Remaining fields are filled in before the user login attempt, and so should be present in every event.

CDP service event sources and names

CDP defines many service event sources and names.

The defined CDP service event sources and names will expand in the future.

Event source	Event name	Action
iam	AssignRoleServiceEvent	Assignment of a role to a user
	CreateUserServiceEvent	Creation of a new user in the control plane
	CreateGroupServiceEvent	Creation of a new group
	DeleteGroupServiceEvent	Deletion of an existing group
	InteractiveLogoutEvent	Interactive logout by a user in the control plane.
	UnassignRoleServiceEvent	Removal of a role from a user
datahub	CreateDataHubCluster	Creation of a Data Hub cluster
	DeleteDataHubCluster	Deletion of a Data Hub cluster
	InstanceDeleteDatahubCluster	Deletion of a Data Hub cluster instance
	MaintainDatahubCluster	Upscaling/downscaling/starting/stopping a Data Hub cluster
	ManualRepairDatahubCluster	Manual repair of a Data Hub cluster
	RetryDatahubCluster	Retrying a Data Hub cluster
	StartDatahubCluster	Starting of a Data Hub cluster
	StopDatahubCluster	Stopping of a Data Hub cluster
datalake	CreateDatalakeCluster	Creation of a Data Lake cluster
	DeleteDatalakeCluster	Deletion of a Data Lake cluster
	InstanceDeleteDatalakeCluster	Deletion of a Data Lake cluster instance
	ManualRepairDatalakeCluster	Manual repair of a Data Lake cluster
	ResizeDatalakeCluster	Resizing of a Data Lake cluster
	RetryDatalakeCluster	Retrying of a Data Lake cluster
	StartDatalakeCluster	Starting of a Data Lake cluster
	StopDatalakeCluster	Stopping of a Data Lake cluster

The schemas for the additional details JSON for each event are defined in [CDP Control Plane Audit Event Details Documentation](#).

Cloudera Data Warehouse audit events

You can retrieve the following audit events [using the CDP CLI](#) that occur in CDW:

Event source	Event name
dw	CreateEnvironment
	DeleteEnvironment
	UpdateEnvironment

Event source	Event name
	UpgradeEnvironment
	CreateDbCatalog
	DeleteDbCatalog
	StartDbCatalog
	StopDbCatalog
	UpdateDbCatalog
	CloneDbCatalog
	UpgradeDbCatalog
	CreateHiveVirtualWarehouse
	DeleteHiveVirtualWarehouse
	StartHiveVirtualWarehouse
	StopHiveVirtualWarehouse
	UpdateHiveVirtualWarehouse
	CloneHiveVirtualWarehouse
	UpgradeHiveVirtualWarehouse
	CreateImpalaVirtualWarehouse
	DeleteImpalaVirtualWarehouse
	StartImpalaVirtualWarehouse
	StopImpalaVirtualWarehouse
	UpdateImpalaVirtualWarehouse
	CloneImpalaVirtualWarehouse
	UpgradeImpalaVirtualWarehouse

For more information about experimental CLI commands for Cloudera Data Warehouse, go to [Version Mapping](#). Click the CDP CLI Reference link for your CDW version. Scroll to Available Commands, and click dw.

Retrieving audit events

The Control Plane provides access to view/retrieve audit events and archive history in two ways: through the Management Console UI and through the CDP API offered by the CDP Control Plane audit service.

Required role: PowerUser

Management Console UI

With the required permissions, you can view both the audit events and the audit archive history in the Management Console UI.

A list of both the audit events and the archive history are accessible from the main left-hand navigation menu under Audit:

CDP

CLOUDERA

Management Console

Dashboard

Environments

Data Lakes

User Management

Data Hub Clusters

Data Warehouses

ML Workspaces

Classic Clusters

Audit

Shared Resources

Global Settings

Audit

Audit Events

Archive History

Request ID

All

Event Source

All

Event Name

All

Result Code

All

Last 4 Hours

ID	Request ID	Event Source	Event Name	Actor	Result Code	Origin	Timestamp
...	...	df	/v1/environments/{environmentid}/deployments/{deploymentid}/event-history	...	SUCCESS		11/21/2022 11:00 PM CST
...	...	df	/v1/environments/{environmentid}/deployments/{deploymentid}/active-alerts	...	SUCCESS		11/21/2022 11:00 PM CST
...	...	df	/v1/environments/{environmentid}/deployments/{deploymentid}/system-metrics	...	SUCCESS		11/21/2022 11:00 PM CST

The **Audit Events** page contains a record of all the events that generate an audit record, along with their name, origin, source, and timestamp.

CDP

CLOUDERA

Management Console

Dashboard

Environments

Data Lakes

User Management

Data Hub Clusters

Data Warehouses

ML Workspaces

Classic Clusters

Audit

Shared Resources

Global Settings

Audit

Audit Events

Archive History

Archive ID

All

Status

All

Archive ID	Event Status	Summary	Details	Creation Time	Archive Time
...	SUCCEEDED	...	Archived 7585 events.	11/21/2022 9:59 PM CST	11/21/2022 9:59 PM CST
...	SUCCEEDED	...	Archived 2522 events.	11/21/2022 8:06 PM CST	11/21/2022 8:06 PM CST
...	SUCCEEDED	...	Archived 5999 events.	11/21/2022 6:54 PM CST	11/21/2022 6:53 PM CST
...	SUCCEEDED	...	Archived 5774 events.	11/21/2022 5:07 PM CST	11/21/2022 5:07 PM CST
...	SUCCEEDED	...	Archived 7098 events.	11/21/2022 3:00 PM CST	11/21/2022 3:00 PM CST

The **Archive History** page contains a record of each archive batch that is sent to cloud storage, along with details related to that batch, such as the status, number of events archived, and time of both creation and archive. The archive history gives greater visibility into the audit process and will alert you if an archive run fails to export to cloud storage.

Public listing API

The Control Plane audit service offers a basic API endpoint for listing audit events. The API accepts the following parameters:

Parameter name	Description	Required?
from timestamp	Beginning of time range within which to retrieve audit events	Y
to timestamp	End of time range within which to retrieve audit events	Y
request ID	Request ID to filter on	N
event source	Event source to filter on	N
page size	Number of audit events to return in one response; maximum and default value = 50	N
page token	Opaque value from previous listing call used to retrieve the next page	N

The result of a listing call is a set of audit events, possibly along with a page token value. In order to get a listing of the next page of audit events, pass the token value in another call, leaving other parameters the same.

Calls to the listing API endpoint are protected as follows:

- The caller must have the `audit/listAuditEvents` right.

When an account is first created in the control plane, it is not configured with the information needed to archive audit events to cloud storage. You can use the control plane without configuring archiving, but audit events are subject to purging after 90 days. So, it is important to configure archiving to avoid data loss.

Access to the listing API endpoint is strictly throttled, to prevent the critical audit service from becoming too busy to accept audit events from Control Plane services. The primary means of retrieving audit events is intended to be through cloud storage.

CDP CLI Syntax

To call the API endpoint for listing audit events, use a command line like the following, which includes examples of all of the required options:

```
cdp audit list-events
--from-timestamp 2020-03-01T00:00:00Z
--to-timestamp 2020-04-01T00:00:00Z
```

To additionally filter by request ID, use the `--request-id` option. To additionally filter by event source, use the `--event-source` option. Paging options work the same way as with other CDP CLI commands.

You can also use the CDP CLI to view the audit archive history. The following command returns the most recent audit archive exports:

```
cdp audit list-recent-archive-runs
```

AWS setup for audit archiving

While the auditing system stores generated audit events initially into a cloud provider managed database, after a specific amount of time, audit logs are exported to customer-managed storage, in their own cloud provider. As a result, you need to set up audit archiving for your platform.

Configuring audit event archiving through the UI

To configure archiving, you must create an AWS IAM role specifically for audit event archiving. Then create an audit credential using the cross-account role ARN.

Before you begin

In AWS, create a new S3 bucket or designate an existing bucket for audit archiving. Be sure to block all public access. The audit logs are written to the S3 bucket even without public access as the writing process is based on the AWS role and not on the internet access. Audit events will be archived under the `/cdp/cp` folder, which will be created automatically by CDP.

Required Role: PowerUser



Procedure

1. Log in to the CDP interface.
2. In the left-side navigation menu, click **Global Settings** **Audit Data Configuration** and then click **Create**.

3. Select the AWS icon.

Settings / Credential

Audit Data Configuration



Create Cross-account Access Policy

Copy the following JSON to create an [AWS IAM policy](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::<your-bucket-name-for-audit-archiving>"
      ]
    }
  ]
}
```

Create Cross-account Access Role

Use Service Manager Account ID and External ID to create an [AWS IAM role](#)

Service Manager Account ID*

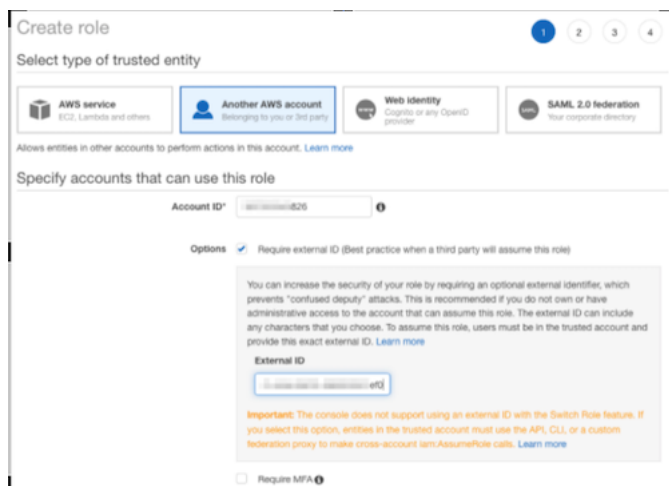
External ID*

Cross-account Role ARN*

Create

4. Copy the cross-account access policy provided to you in the **Create Cross-account Access Policy field, substituting your bucket name where indicated.**

- From the AWS account hosting the bucket, create an IAM role for another AWS account, in this case, for the account running the CDP Control Plane.



Include the policy that you copied as the only one in the role.

For detailed instructions on creating an AWS IAM cross-account role, see [Create a cross-account IAM role](#), starting with "1. Log into the AWS Management Console." Although the audit event archiving credential requires a unique policy and cross-account Role ARN, the process is largely the same as creating a role-based credential during environment registration.

- When you finish creating the new cross-account policy and role in AWS, copy the Role ARN from the Role **Summary** page in the AWS Management Console and return to CDP. Paste the Role ARN into the Cross-account role ARN field.
- Click Create.
- Configure the audit data location with the name of the S3 bucket that you designated as the audit archive bucket.
- Select the AWS region where storage services should be accessed.
- Use the toggle button to enable or disable audit log export to the configured storage location.
- Click Save.

Results

Audit event archiving configuration is complete.

Setting up an AWS policy and role to configure archiving using the CLI

To configure archiving through the CDP CLI, you must first set up a policy and cross-account role in AWS IAM and obtain the cross-account role ARN.

Before you begin

In AWS, create a new S3 bucket or designate an existing bucket. Be sure to block all public access. The audit logs are written to the S3 bucket even without public access as the writing process is based on the AWS role and not on the internet access. Audit events will be archived under the /cdp/cp folder, which will be created automatically by CDP.

Procedure

- From the AWS account hosting the bucket, create an IAM policy that permits read and write access to the bucket. For example, substituting your bucket name where indicated:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

{
  "Sid": "cdpauditb",
  "Effect": "Allow",
  "Action": [
    "s3:ListBucket",
    "s3:ListBucketMultipartUploads",
    "s3:GetBucketLocation"
  ],
  "Resource": [
    "arn:aws:s3:::<your-bucket-name-for-audit-archiving>"
  ]
},
{
  "Sid": "cdpauditbo",
  "Effect": "Allow",
  "Action": [
    "s3:ListMultipartUploadParts",
    "s3:GetObject",
    "s3:GetObjectVersion",
    "s3:PutObject"
  ],
  "Resource": [
    "arn:aws:s3:::<your-bucket-name-for-audit-archiving>/*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeRegions"
  ],
  "Resource": [
    "*"
  ]
}
]
}

```

- From the AWS account hosting the bucket, create an IAM role for another AWS account, in this case, for the account running the CDP Control Plane.

Create role

Select type of trusted entity

Another AWS account (Selected)

Specify accounts that can use this role

Account ID: 123456789012

Options: ☒ Require external ID (Best practice when a third party will assume this role)

External ID: 123456789012-external-id

Important: The console does not support using an external ID with the Switch Role feature. If you select this option, entities in the trusted account must use the API, CLI, or a custom federation proxy to make cross-account `iam:AssumeRole` calls.

Include the policy you just created as the only one in the role.

For detailed instructions on creating an AWS IAM policy and cross-account role, see [Create a cross-account IAM role](#), starting with "1. Log into the AWS Management Console." Although the audit event archiving credential requires a unique policy and Role ARN, the process is largely the same as creating a role-based credential during environment registration.

3. To finish creating the role you will need the account ID (service manager account ID) and external ID. Run the following CDP CLI command:

```
cdp environments get-audit-credential-prerequisites \
--cloud-platform AWS
```



Note: The control plane uses a different external ID for each account or tenant. So, be sure to look up the account ID and external ID using a user in the account which is being configured.

4. When you finish the role creation process, copy the Role ARN from the role **Summary** page in the AWS Management Console. You will need it to create the audit event archiving credential in the next task.

What to do next

Follow the process in the next topic, *Creating an AWS credential for audit event archiving using the CLI*, to create the audit archive credential.

Creating an AWS credential for audit event archiving using the CLI

To configure archiving, you must set up a credential for audit event archiving and then configure CDP.

Before you begin

You must have the cross-account role ARN obtained in the previous task to finish setting up an AWS credential for audit event archiving.

About this task

Complete this task before you configure audit archiving. The audit credential that you create here is not tied to an environment, and exists outside of any environment, like the control plane itself. The associated role / permissions require write access to the storage location, including the ability to create files and folders.

Required Role: PowerUser

Procedure

1. Use the following commands to create a new audit credential:

```
cdp environments set-aws-audit-credential \
--role-arn arn:aws:...
```

The role-arn information was provided when you created an IAM role.

You can view audit credentials with this command:

```
cdp environments list-audit-credentials
```

2. Make note of the credential name created by the command.

What to do next

Proceed to *Setting up audit archiving in AWS using the CLI* to complete the audit archiving setup.

Setting up audit archiving in AWS using the CLI

After you set up a credential for audit event archiving, you need to configure CDP. This process is not necessary if you used the UI to create the audit archiving credential and configure the data storage location.

About this task

Archiving configuration includes the following fields.

Field name	General description	Details
storage location	Where in cloud storage to save audit events	S3 bucket name. Use only the bucket name and not the "s3://" prefix or any sub-folders.
credential	Name of the credential to use when writing to cloud storage	credential name as saved in the control plane
enabled	A Boolean indicating whether archiving is enabled	true
storage region	Region where storage services should be accessed	AWS region, e.g., us-west-2

Create the credential for audit event archiving before configuring archiving itself. The credential is not tied to an environment, and exists outside of any environment, like the control plane itself. The associated role / permissions require write access to the storage location, including the ability to create files and folders.

The storage region is the region where the audit service (the control plane) accesses the cloud provider's storage service. For best results, this should be the same as the region where the control plane is running. Ideally, the bucket should be created in the same region.

Required Role: PowerUser

Procedure

1. To call the API endpoint for configuring archiving, run the `cdp audit configure-archiving` command. To begin, include the optional `--verify-only` flag, which the service uses to first verify that the configuration works:

```
cdp audit configure-archiving \
  --storage-location <$MYBUCKET> \
  --credential-name myauditcredential \
  --storage-region <$HOME_REGION>
  --enabled
  --verify-only
```

For the `credential-name` parameter, use the CRN of the credential that you noted when you set up the credential for audit event archiving.

If successful, the command will archive a test audit event to the container.

2. Run `cdp audit configure-archiving` again, omitting the `--verify-only` option, to apply the configuration to the account of the caller.

```
cdp audit configure-archiving \
  --storage-location <$MYBUCKET> \
  --credential-name myauditcredential \
  --storage-region <$HOME_REGION>
  --enabled
```

The command returns the archiving configuration for the account of the caller.

You can also include the following options:

- To disable archiving while retaining the other configuration information, use the `--no-enabled` option instead of the `--enabled` option.
- To retrieve the current archiving configuration, use a command like the following (there are no required options):

```
cdp audit get-archiving-config
```

Azure setup for audit archiving

While the auditing system stores generated audit events initially into a cloud provider managed database, after a specific amount of time, audit logs are exported to customer-managed storage, in their own cloud provider. As a result, you need to set up audit archiving for your platform.

Creating a storage account in Azure

To set up audit archiving, you must first designate a storage account and container for audit archiving in the Azure portal. While you can use an existing storage account, Cloudera recommends that you create a dedicated storage account for audit archiving, especially if you are archiving any sensitive data.

Before you begin

When creating a new storage account, you'll need to know the following pieces of information for creating a new Service Principal in Azure later on:

- Resource group name
- Storage account name

Although it is not required, it can be helpful to create a new resource group when you create a new storage account, so that future management becomes easier. You can create a new resource group in an Azure shell in the Azure Portal by running the following command:

```
az group create --location <location> --name <resource-group-name>
```


Procedure

1. In Azure, create a storage account.

In the **Basics** page, you should be able to accept the defaults for all settings other than the name. You should also be able to accept the defaults on the **Advanced** and **Networking** pages.



Note: Ensure that the Enable public access on all network setting is enabled on the **Networking** page for the Storage account as the audit archiving can fail if the default setting is changed.

The screenshot shows the 'Create a storage account' page in the Microsoft Azure portal, specifically the 'Basics' tab. The page has a blue header with the Microsoft Azure logo and a search bar. Below the header, there's a breadcrumb trail: 'Home > Storage accounts >'. The main title is 'Create a storage account'. Below the title, there are tabs for 'Basics', 'Advanced', 'Networking', 'Data protection', 'Tags', and 'Review + create'. The 'Basics' tab is selected. The page content includes a description of Azure Storage, followed by 'Project details' where a subscription ('azure-pm-sandbox') and resource group ('gpatel-rg') are selected. The 'Instance details' section includes fields for 'Storage account name' (cdpauditsetup), 'Region' ((US) East US), 'Performance' (Standard), and 'Redundancy' (Geo-redundant storage (GRS)). A checkbox for 'Make read access to data available in the event of regional unavailability' is checked. At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next : Advanced >'.

Microsoft Azure Search resources, services, and docs (G+)

Home > Storage accounts >

Create a storage account

Basics Advanced Networking Data protection Tags Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more about Azure storage accounts](#)

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription * azure-pm-sandbox

Resource group * gpatel-rg [Create new](#)

Instance details

If you need to create a legacy storage account type, please click [here](#).

Storage account name * cdpauditsetup

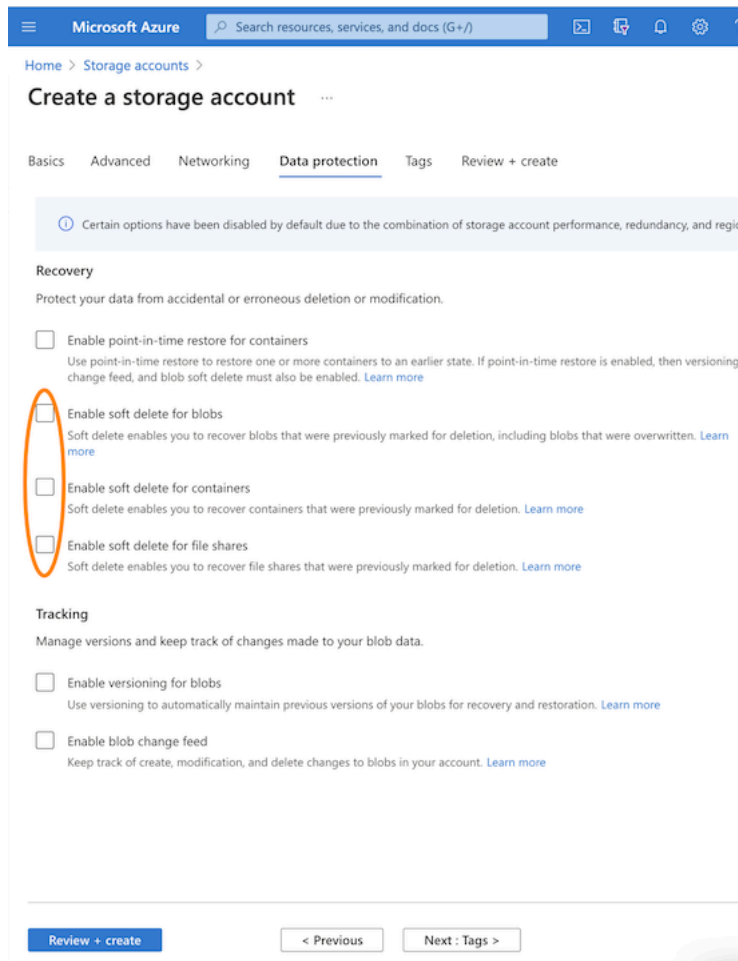
Region * (US) East US

Performance *
 ☒ Standard: Recommended for most scenarios (general-purpose v2 account)
 ☐ Premium: Recommended for scenarios that require low latency.

Redundancy * Geo-redundant storage (GRS)
 ☒ Make read access to data available in the event of regional unavailability.

[Review + create](#) < Previous Next : Advanced >

2. When you reach the **Data Protection** page, you must disable the soft delete options for blobs, containers, and file shares. These options are enabled by default, so be sure to uncheck the selection boxes for these options, as shown below:



Microsoft Azure Search resources, services, and docs (G+)

Home > Storage accounts >

Create a storage account

Basics Advanced Networking **Data protection** Tags Review + create

ⓘ Certain options have been disabled by default due to the combination of storage account performance, redundancy, and region.

Recovery

Protect your data from accidental or erroneous deletion or modification.

- ☐ Enable point-in-time restore for containers
Use point-in-time restore to restore one or more containers to an earlier state. If point-in-time restore is enabled, then versioning, change feed, and blob soft delete must also be enabled. [Learn more](#)
- ☐ Enable soft delete for blobs
Soft delete enables you to recover blobs that were previously marked for deletion, including blobs that were overwritten. [Learn more](#)
- ☐ Enable soft delete for containers
Soft delete enables you to recover containers that were previously marked for deletion. [Learn more](#)
- ☐ Enable soft delete for file shares
Soft delete enables you to recover file shares that were previously marked for deletion. [Learn more](#)

Tracking

Manage versions and keep track of changes made to your blob data.

- ☐ Enable versioning for blobs
Use versioning to automatically maintain previous versions of your blobs for recovery and restoration. [Learn more](#)
- ☐ Enable blob change feed
Keep track of create, modification, and delete changes to blobs in your account. [Learn more](#)

[Review + create](#) < Previous Next : Tags >

- Review your changes in the next screen and verify that the soft delete options are disabled. If everything looks correct, click the Create button.

Home > Storage accounts > Create a storage account ...

Validation passed

Basics Advanced Networking Data protection Tags **Review + create**

Advanced

Secure transfer	Enabled
Allow storage account key access	Enabled
Default to Azure Active Directory authorization in the Azure portal	Disabled
Infrastructure encryption	Disabled
Blob public access	Enabled
Minimum TLS version	Version 1.2
Enable hierarchical namespace	Disabled
Enable network file share v3	Disabled
Access tier	Hot
Large file shares	Disabled

Networking

Network connectivity	Public endpoint (all networks)
Default routing tier	Microsoft network routing

Data protection

Point-in-time restore	Disabled
Blob soft delete	Disabled
Container soft delete	Disabled
File share soft delete	Disabled
Versioning	Disabled
Blob change feed	Disabled

Create < Previous Next > Download a template for automation

Alternatively, you can create a storage account using the Azure shell by running the following command:

```
az storage account create --name <storage-account-name> --kind <kind> --location <location> --resource-group <resource-group-name>
```

Creating a container in Azure

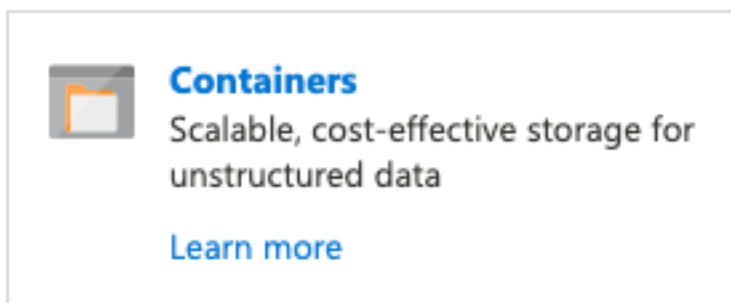
To set up audit archiving, you must designate a container for audit archiving in the Azure portal. While you can use an existing container, Cloudera recommends that you create a dedicated container for audit archiving.

Before you begin

Before you create a container, take note of the location property for the storage account at the top of the page in the Storage Account overview section. You will use this later as the storageRegion value when you configure archiving using the CDP CLI or UI.

Procedure

1. Navigate to your newly created storage account resource and click the Containers button:



2. In the next screen, click the + Container button, then enter a name for the container and the default Private access level.

You will use the container name when you create a new service principal in Azure.

Alternatively, you can create a new container using the Azure shell by running the following command:

```
az storage container create --name <container-name> --account-name <storage-account-name>
```

3. Obtain the URL for the newly created container.

To retrieve the URL, open the container and click the Properties link in the left navigation bar under Settings. Copy the URL shown. You will need this value when you configure audit archiving as the value for the storage location URL. The format is: `https://<storage-account-name>.blob.core.windows.net/container-name{ }}`

Configuring audit event archiving through the UI

To configure audit event archiving for Azure through the CDP UI, create a new audit archiving credential and then configure the audit data storage and location.

About this task

The process of creating an audit archiving credential is largely the same as creating a credential for an Azure environment, however, this credential is a unique credential used solely for audit event archiving.

Before you begin

You must have an Azure resource group as well as an Azure storage account and a container that will be used to store the audit archive logs. You can use an existing resource group, storage account, and container, but Cloudera recommends creating a new storage account and container dedicated to audit archiving. See the topic *Creating ADLS Gen2 storage account and container*.

Required Role: PowerUser

Procedure

1. In the left-side navigation menu, click Global Settings Audit Data Configuration and then click Create.
2. Select the Azure icon.
3. Use the provided command in an Azure shell to identify your subscription ID and tenant ID:

```
az account list | jq '.[] | { "name": .name, "subscriptionId": .id, "tenantId": .tenantId, "state": .state }'
```

4. In the corresponding fields, enter the Subscription ID and Tenant ID that are returned by the command.

Settings / Credential

Audit Data Configuration

Telemetry

Tags



Paste the following command into [Azure Shell](#) to identify your Subscription Id and your Tenant Id:

```
az account list | jq '.[] | {"name": .name, "subs
```



Subscription Id*



Tenant Id*



In order to create an application, you could use following command in [Azure Shell](#) or you could create it on [Azure Portal](#).

```
az ad sp create-for-rbac \      --name http://{
```



App Id*



Password*



Create

5. Use the provided command in an Azure shell to register a new application in Azure. Substitute your application name (whatever you chose to give it), subscription ID, resource group name, storage account name, and container name where indicated:

```
az ad sp create-for-rbac \      --name http://{app-name} \      --role "Storage Blob Data Contributor" \      --scopes /subscriptions/{subscriptionId}/resourceGroups/{resource-group-name}/providers/Microsoft.Storage/storageAccounts/{storage-account-name}/blobServices/default/containers/{container-name}
```

Save the App ID and password from the command output to enter in the Audit Data Configuration **Credential** screen.

Alternatively, you can follow steps 1-8 in the topic *Create an app-based credential* to register the new app through the Azure portal. When you add a role assignment, use the more limited "Storage Blob Data Contributor" role.

6. Back in the CDP **Audit Data Configuration** screen, enter the App Id and Password in the corresponding fields and click Create.
7. Configure the audit data location with the path to the ADLS container that you designated as the audit archive container. Use the following format:

```
https://<storage-account-name>.blob.core.windows.net/<container-name>
```

8. Select the region where storage services should be accessed.
9. Use the toggle button to enable or disable audit log export to the configured storage location.
10. Click Save.

Results

Audit event archiving configuration is complete.

Related Information

[Creating ADLS Gen2 storage account and container](#)

[Create an app-based credential \(Azure\)](#)

Creating a CDP credential for audit archiving on Azure using the CLI

After you create (or designate) a storage account and container in Azure, you need to create a CDP credential for audit archiving on Azure.

Before you begin

Before proceeding with this task, open the Azure shell in the Azure Portal.

Required Role: PowerUser

Procedure

1. Use the provided command in an Azure shell to identify your subscription ID and tenant ID:

```
az account list | jq '.[] | { "name": .name, "subscriptionId": .id, "tenantId": .tenantId, "state": .state }'
```

Take note of the subscription ID (the subscription used to create the storage account that you designated for auditing) and the tenant ID for later use.

2. Use the provided command in an Azure shell to create a new service principal in Azure. Substitute your service principal name (whatever you chose to give it, for example the container name), subscription ID, resource group name, storage account name, and container name where indicated:

```
az ad sp create-for-rbac \      --name http://{app-name} \      --role "Storage Blob Data Contributor" \      --scopes /subscriptions/{subscriptionId} /resourceGroups/{resource-group-name} /providers/Microsoft.Storage/storageAccounts/{storage-account-name} /blobServices/default/containers/{container-name}
```

Save the App ID and password from the command output to use in the credential creation command.

3. In the CDP CLI, use the following command to create the CDP credential:

```
cdp environments set-azure-audit-credential --profile <profile-name> --subscription-id <subscription-id> --tenant-id <tenant-id> --app-based applicationId=<application-id>,secretKey=<password>
```

The value of the secretKey parameter is the password returned in the output of the previous command that you ran to create the Azure service principal.

Running this command creates the CDP credential for audit archiving on Azure. Take note of the crn returned in the command output, as you will need it to configure archiving in the next task.

What to do next

After you create the credential for auditing, proceed to the next task: *Configuring audit archiving for Azure using the CLI*.

Configuring audit archiving for Azure using the CLI

After you create a CDP credential, configure CDP for audit archiving on Azure.

Before you begin

Required Role: PowerUser

Procedure

1. Test the audit configuration by running the following command:

```
cdp audit configure-archiving --profile <profile-name> --storage-location <storage-location-url> --credential-name <credential-name> --enabled --storage-region <storage-region> --verify-only
```

For the credential-name parameter, pass the CRN returned in the command output when you created the credential in the previous task.

For the storage location, use the URL obtained from the **Properties** page of the container in the Azure Management Console. The format is: `https://<storage-account-name>.blob.windows.core.net/<container-name>`

If successful, the command will archive a test audit event to the container.

2. When the response to the previous command is successful, use the command again without the verify-only flag to set the audit archive configuration:

```
cdp audit configure-archiving --storage-location <storage-location-URL> --credential-name <credential-Name> --enabled --storage-region <storage-Region> --profile <profile-name>
```


Results

Audit archive configuration is complete.

GCP setup for audit archiving

While the auditing system stores generated audit events initially into a cloud provider managed database, after a specific amount of time, audit logs are exported to customer-managed storage, in their own cloud provider. As a result, you need to set up audit archiving for your platform.

Configuring audit event archiving through the UI

To configure archiving for GCP, you must create a GCP service account specifically for audit event archiving, download the service account private key in JSON format, and then upload the service account private key to CDP.

Before you begin

In GCP, create a new GCS bucket or designate an existing bucket for audit archiving. Be sure to block all public access. Audit events will be archived under the /cdp/cp folder, which will be created automatically by CDP.

Required Role: PowerUser

Procedure

1. Log in to the CDP interface.
2. In the left-side navigation menu, click Global SettingsAudit Data Configuration and then click Create.


3. Select the GCP icon.


Settings / Create Audit Credential


Audit Data Configuration

Telemetry

Tags







Create Service Account

Copy the script into your Terminal (you need [Google Cloud SDK](#) to be installed on your machine) or into the [Google Cloud Shell](#) to create a [Service Account](#)

```
gcloud init

SERVICE_ACCOUNT_NAME=cdp-audit-credential
PROJECT_ID=$(gcloud config get-value project)

echo "Enabling Compute and Runtimeconfig APIs"
gcloud services enable
compute.googleapis.com
runtimeconfig.googleapis.com

echo "Creating service account for CDP Audit"
```

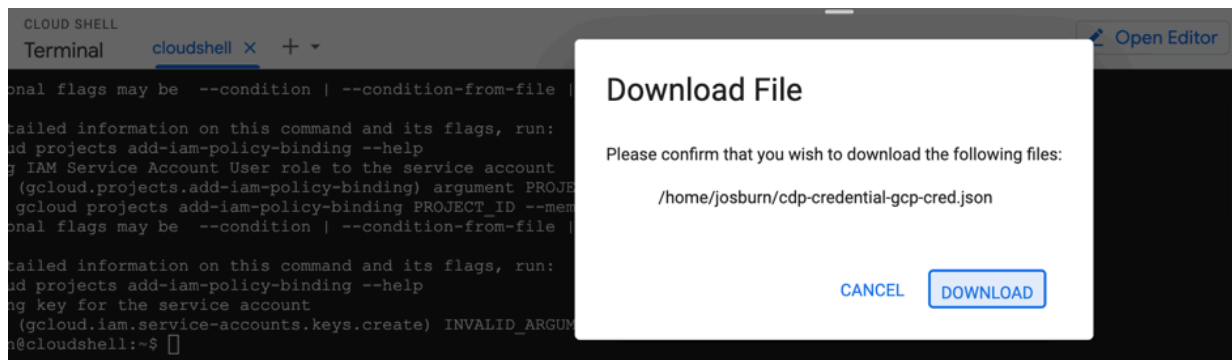
Service Account Private Key (json format)*

Upload file

No file chosen ?

Create Credential

4. Copy the script provided into your terminal or Google Cloud Shell to create a new service account and generate the service account private key.



When the script finishes running, it begins a download of the service account private key.

5. When you have finished creating the new service account and have the service account private key, click Upload file on the **Create Audit Credential** page in CDP to upload the private key JSON to CDP.
6. Click Create Credential.
7. After the credential has been created, you must configure the audit data location. In the Storage location field, provide the full path to the GCS bucket that you created or designated to be the audit log bucket.
8. Select the bucket region, then decide whether or not to export the audit logs to the configured storage location. You can also verify the configuration before saving.
9. Click Save Configuration.

Results

Audit event archiving configuration is complete.

Configure GCP audit event archiving using the CLI

Use the CDP CLI to configure audit event archiving in GCP.

Procedure

1. Use the CDP CLI to get the GCP command that creates the GCP service account and private key required for audit credential generation. In the CDP CLI, run the following command:

```
cdp environments get-audit-credential-prerequisites --cloud-platform gcp
```

The CDP CLI returns the service account creation command (in base64 decoded format) for GCP credential creation. Run this command in your terminal (with Google SDK installed) or Google Cloud Shell to create the new service account and download the private key that is required in the next step.

2. In the CDP CLI, run the command to upload the service account private key:

```
cdp environments set-gcp-audit-credential --credential-key <path to private key JSON file>
```

For example:

```
cdp environments set-gcp-audit-credential --credential-key file:///Users/jo/cdp_tools/artifacts/gcpauditcred.json
```

This command returns the credential name/CRN that you will need for the next step.

3. Run the following command to configure GCP audit event archiving in CDP:

```
cdp audit configure-archiving --storage-location <full path to bucket> --
credential-name <audit credential name or CRN> --enabled
```

Optionally, you can use the `--storage-region` parameter to designate the storage region or the `--verify-only` parameter to verify the audit configuration but not update it.

For example:

```
cdp audit configure-archiving --storage-location gs://cdp/auditbucket --
credential-name audit-credential-5617y894 --enabled --verify-only
```

Pull-based audit archiving

Pull-based audit archiving allows you to pull audit events for archiving purposes without any extra configuration beyond Control Plane API usage.

About this task

The Control Plane auditing system archives auditing events by writing them to cloud storage that you configure and manage yourself. If you do not want to provide the network access or credentials required for the Control Plane to automatically export audit logs to cloud storage in your cloud provider, you can use pull-based audit archiving to retrieve the events yourself in the same format (see *What archiving looks like*). Using pull-based audit archiving, you can group the audit events into batches, list the event batches that have not been marked as archived, retrieve those batches, and then mark them as ready for purging from the Control Plane database.



Important: Pull-based audit archiving commands will return an error if automated audit archiving is enabled.

Procedure

1. On a command line, run `cdp audit batch-events-for-archiving` to begin the asynchronous process of grouping the audit events into batches. For example:

```
cdp audit batch-events-for-archiving \
--from-timestamp 2020-03-01T00:00:00Z \
--to-timestamp 2020-04-01T00:00:00Z
```

Note that if there is already a batch event operation in-progress, running the command again is not allowed. If you run the command when an operation is already in progress, you will receive an error.

If successful, the command returns a task ID for tracking the status of the process. For example:

```
{
  "taskId": "0b67c29c-bce9-4bbd-ac3e-8445df029f4f"
}
```

2. Use the `cdp audit get-batch-events-for-archiving-status` command and the task ID to poll the asynchronous task repeatedly, until it completes successfully or with an error:

```
cdp audit get-batch-events-for-archiving-status \
--task-id 0b67c29c-bce9-4bbd-ac3e-8445df029f4f
```

While the task is running, the status will be "OPEN":

```
{
  "status": "OPEN",
```

```
    "eventBatches": [ ]
  }
```

When it completes successfully, the status will be "COMPLETED," and identifiers will be returned for the event batches:

```
{
  "status": "COMPLETED",
  "eventBatches": [
    {
      "accountId": "37t8i20c-cd82-4e8b-39e4-dcaelf9cd7ef",
      "eventCount": 11,
      "archiveId": "c5b57c79-6721-4e27-9hr9-67f5d299b1gq",
      "archiveTimestamp": 0
    }
  ]
}
```

3. Run `cdp audit list-outstanding-archive-batches` to determine event batches which have not yet been marked as archived. The output appears similar to the example below:

```
{
  "eventBatches": [
    {
      "accountId": "37t8i20c-cd82-4e8b-39e4-dcaelf9cd7ef",
      "eventCount": -1,
      "archiveId": "c5b57c79-6721-4e27-9hr9-67f5d299b1gq",
      "archiveTimestamp": 0
    }
  ]
}
```

4. For each batch that has not been marked as archived, run `cdp audit list-events-in-archive-batch` to retrieve the batch of events:

```
cdp audit list-events-in-archive-batch \
  --archive-id c5b57c79-6721-4e27-9hr9-67f5d299b1gq
```

The output of which will be similar to:

```
{
  "auditEvents": [
    {
      "version": "1.0.0",
      ...
    },
    ...
  ]
}
```

Optionally, you can use shell builtins and utilities to convert the output to a gzipped JSON lines format, like the archives produced by automated audit archiving, with a file name that includes the account ID, a timestamp, and the batch archive ID. For example:

```
cdp audit list-events-in-archive-batch \
  --archive-id c5b57c79-6721-4e27-9hr9-67f5d299b1gq \
  | jq -c '.auditEvents[]' \
  | gzip > 37t8i20c-cd82-4e8b-39e4-dcaelf9cd7ef_`date -u +%Y%m%dT%H%M`Z_
c5b57c79-6721-4e27-9hr9-67f5d299b1gq.json.gz
```

5. Once you have saved an archive to your storage destination, use `cdp audit mark-archive-batches-as-successful` to mark a batch as successfully archived, so that it can later be purged automatically from the Control Plane database. You can provide one or more batches for the `--archive-ids` parameter. For example:

```
cdp audit mark-archive-batches-as-successful \
  --archive-ids c5b57c79-6721-4e27-9hr9-67f5d299b1gq

{
  "archiveIds": [
    "c5b57c79-6721-4e27-9hr9-67f5d299b1gq"
  ],
  "archiveTimestamp": "2021-08-10T21:54:59.223000+00:00"
}
```

Related Information

[What archiving looks like](#)

What archiving looks like

An audit event is eligible to be archived once it is complete, including all result data for the associated activity. The audit service will allow up to one hour for missing result data to become available for an event before archiving it. After that timeout period, however, the incomplete event will become eligible for archiving.

Audit events are not saved in individual files. Instead, groups of audit events from a continuous time span are saved into a single file, with one JSON object / audit event per line. The file is compressed using gzip for efficiency. The Control Plane does not encrypt the files. If you would like the files encrypted, you will need to configure your cloud storage service to perform it.

Files are organized in cloud storage based on the dates of their timestamps, as follows:

```
/cdp/cp
# yyyy (four-digit year)
# MM (two-digit month)
# dd (two-digit day of month)
```

Audit events are not saved to cloud storage immediately after they are generated, to enable efficiency of storage and to alleviate performance concerns for the audit service. On an hourly basis, the audit service batches up audit events that are eligible for archiving and writes them to cloud storage.

An audit event is eligible to be archived once it is complete, including all result data for the associated activity. The audit service will allow up to one hour for missing result data to become available for an event before archiving it. After that timeout period, however, the incomplete event will become eligible for archiving.

The Control Plane (audit service) periodically saves audit events to your cloud storage facility, for example, an S3 bucket. Cloud storage is intended to be the primary location for storage, retrieval, and analysis of audit events in the long term.