

Hitachi Content Platform for Cloud Scale

S3 Console Guide

This document contains information and guidance on using the S3 Console to manage buckets and objects stored through Hitachi Content Platform for cloud scale (HCP for cloud scale).

© 2020 Hitachi Vantara LLC. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including copying and recording, or stored in a database or retrieval system for commercial purposes without the express written permission of Hitachi, Ltd., or Hitachi Vantara LLC (collectively "Hitachi"). Licensee may make copies of the Materials provided that any such copy is: (i) created as an essential step in utilization of the Software as licensed and is used in no other manner; or (ii) used for archival purposes. Licensee may not make any other copies of the Materials. "Materials" mean text, data, photographs, graphics, audio, video and documents.

Hitachi reserves the right to make changes to this Material at any time without notice and assumes no responsibility for its use. The Materials contain the most current information available at the time of publication.

Some of the features described in the Materials might not be currently available. Refer to the most recent product announcement for information about feature and product availability, or contact Hitachi Vantara LLC at https://support.hitachivantara.com/en_us/contact-us.html.

Notice: Hitachi products and services can be ordered only under the terms and conditions of the applicable Hitachi agreements. The use of Hitachi products is governed by the terms of your agreements with Hitachi Vantara LLC.

By using this software, you agree that you are responsible for:

1. Acquiring the relevant consents as may be required under local privacy laws or otherwise from authorized employees and other individuals; and
2. Verifying that your data continues to be held, retrieved, deleted, or otherwise processed in accordance with relevant laws.

Notice on Export Controls. The technical data and technology inherent in this Document may be subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Reader agrees to comply strictly with all such regulations and acknowledges that Reader has the responsibility to obtain licenses to export, re-export, or import the Document and any Compliant Products.

Hitachi and Lumada are trademarks or registered trademarks of Hitachi, Ltd., in the United States and other countries.

AIX, AS/400e, DB2, Domino, DS6000, DS8000, Enterprise Storage Server, eServer, FICON, FlashCopy, GDPS, HyperSwap, IBM, Lotus, MVS, OS/390, PowerHA, PowerPC, RS/6000, S/390, System z9, System z10, Tivoli, z/OS, z9, z10, z13, z14, z/VM, and z/VSE are registered trademarks or trademarks of International Business Machines Corporation.

Active Directory, ActiveX, Bing, Excel, Hyper-V, Internet Explorer, the Internet Explorer logo, Microsoft, the Microsoft Corporate Logo, MS-DOS, Outlook, PowerPoint, SharePoint, Silverlight, SmartScreen, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, the Windows logo, Windows Azure, Windows PowerShell, Windows Server, the Windows start button, and Windows Vista are registered trademarks or trademarks of Microsoft Corporation. Microsoft product screen shots are reprinted with permission from Microsoft Corporation.

All other trademarks, service marks, and company names in this document or website are properties of their respective owners.

Copyright and license information for third-party and open source software used in Hitachi Vantara products can be found at <https://www.hitachivantara.com/en-us/company/legal.html>.

EXPORT CONTROLS - Licensee will comply fully with all applicable export laws and regulations of the United States and other countries, and Licensee shall not export, or allow the export or re-export of, the Software, API or Materials in violation of any such laws or regulations. By downloading or using the Software, API, or Materials, Licensee agrees to the foregoing and represents and warrants that Licensee is not located in, under the control of, or a national or resident of any embargoed or restricted country.

Contents

Chapter 1: Getting started.....	9
Permissions.....	9
Logging in as an admin.....	9
Logging in as a user.....	10
Generating S3 credentials.....	10
Logging out.....	11
Chapter 2: S3 user credentials.....	12
Generating S3 credentials.....	12
Obtaining S3 credentials.....	12
S3 application instructions.....	13
Related REST API methods.....	13
Revoking S3 credentials.....	13
Logging out.....	14
Chapter 3: Bucket management.....	15
Buckets.....	16
Creating a bucket.....	16
Deleting a bucket.....	17
Rules.....	17
Adding rules to policies.....	17
Adding pre-existing rules to a policy.....	18
Editing a rule.....	18
Deleting rules from a policy.....	18
Filters.....	19
Prefixes.....	19
Adding a prefix filter to a policy.....	19
Editing a prefix filter.....	19
Removing a prefix filter from a policy.....	20
Tags.....	20
Adding tags to rules.....	20
Editing a tag.....	21
Deleting a tag filter.....	21

Chapter 4: Object management.....	22
Chapter 5: Monitoring.....	23
Bucket metrics.....	23
Chapter 6: Policy management.....	25
Bucket policies.....	25
Expiration Lifecycle policy.....	25
Adding an expiration lifecycle policy to a new bucket.....	26
Adding an Expiration Lifecycle policy to a pre-existing bucket.....	26
Adding actions to an Expiration Lifecycle policy.....	26
Editing a Expiration Lifecycle policy	27
Removing an Expiration Lifecycle policy.....	27
Sync-from Replication policy.....	28
Adding a Sync-from Replication policy to a new bucket.....	28
Adding a Sync-from Replication policy to a pre-existing bucket.....	28
Editing a Sync-from Replication policy.....	29
Removing a Sync-from Replication policy.....	29
Sync-to Replication policy.....	29
Adding a Sync-to Replication policy to a new bucket.....	29
Adding a Sync-to Replication policy to a pre-existing bucket.....	30
Editing a Sync-to Replication policy.....	30
Removing a Sync-to Replication policy.....	30
Chapter 7: Bucket synchronization.....	31
About bucket synchronization.....	31
Synchronization to an external bucket: high-level tasks.....	34
Synchronization from an external bucket: high-level tasks.....	35
Bucket synchronization configuration.....	36
Configure bucket synchronization (PUT bucket replication).....	37
Script to generate bucket sync-to JSON.....	43
Script to generate bucket sync-from JSON.....	45
Get bucket synchronization rules (GET bucket replication).....	48
Get object synchronization status.....	51
Delete bucket synchronization rules (DELETE bucket replication).....	52
Chapter 8: S3 event notification.....	53
About S3 event notification.....	53
Script to generate S3 event notification configuration JSON.....	54

Preface

This document describes and provides instructions for using the S3 Console software on Hitachi Content Platform for cloud scale systems.

Please read this document carefully to understand how to use these products, and maintain a copy for your reference.

Intended audience

This document is intended for consumers who use HCP for cloud scale as a way to store and retrieve objects in S3 buckets.

Product version

This document revision applies to S3 Console 2.0 or later.





Document conventions

This document uses the following typographic conventions:

Convention	Description
Bold	<ul style="list-style-type: none">Indicates text in a window, including window titles, menus, menu options, buttons, fields, and labels. Example: Click OK.Indicates emphasized words in list items.
<i>Italic</i>	<ul style="list-style-type: none">Indicates a document title or emphasized words in text.Indicates a variable, which is a placeholder for actual text provided by the user or for output by the system. Example: <pre>pairedisplay -g group</pre> (For exceptions to this convention for variables, see the entry for angle brackets.)
Monospace	Indicates text that is displayed on screen or entered by the user. Example: <code>pairedisplay -g oradb</code>

Convention	Description
< > angle brackets	<p>Indicates variables in the following scenarios:</p> <ul style="list-style-type: none"> Variables are not clearly separated from the surrounding text or from other variables. Example: <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"> <code>Status-<report-name><file-version>.csv</code> </div> Variables in headings.
[] square brackets	Indicates optional values. Example: [a b] indicates that you can choose a, b, or nothing.
{ } braces	Indicates required or expected values. Example: { a b } indicates that you must choose either a or b.
vertical bar	<p>Indicates that you have a choice between two or more options or arguments. Examples:</p> <p>[a b] indicates that you can choose a, b, or nothing.</p> <p>{ a b } indicates that you must choose either a or b.</p>

This document uses the following icons to draw attention to information:

Icon	Label	Description
	Note	Calls attention to important or additional information.
	Tip	Provides helpful information, guidelines, or suggestions for performing tasks more effectively.
	Caution	Warns the user of adverse conditions and/or consequences (for example, disruptive operations, data loss, or a system crash).
	WARNING	Warns the user of a hazardous situation which, if not avoided, could result in death or serious injury.

Accessing product documentation

Product user documentation is available on Hitachi Vantara Support Connect: <https://knowledge.hitachivantara.com/Documents>. Check this site for the most current documentation, including important updates that may have been made after the release of the product.

Accessing product downloads

Product software, drivers, and firmware downloads are available on Hitachi Vantara Support Connect: <https://support.hitachivantara.com/>.

Log in and select Product Downloads to access the most current downloads, including important updates that may have been made after the release of the product.

Getting help

[Hitachi Vantara Support Connect](#) is the destination for technical support of products and solutions sold by Hitachi Vantara. To contact technical support, log on to Hitachi Vantara Support Connect for contact information: https://support.hitachivantara.com/en_us/contact-us.html.

[Hitachi Vantara Community](#) is a global online community for Hitachi Vantara customers, partners, independent software vendors, employees, and prospects. It is the destination to get answers, discover insights, and make connections. **Join the conversation today!** Go to community.hitachivantara.com, register, and complete your profile.

Comments

Please send us your comments on this document to doc.comments@hitachivantara.com. Include the document title and number, including the revision level (for example, -07), and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara LLC.

Thank you!

Chapter 1: Getting started

The S3 Console provides Hitachi Content Platform for cloud scale (HCP for cloud scale) users with a place to manage and browse their buckets and objects. In addition, it can be used to manage bucket policies, such as Expiration Lifecycle, Sync-to Replication, and Sync-from Replication. In addition, metrics are provided for each bucket.

All completable actions have predefined roles, which are controlled by a system administrator who assigns them to users.



Important: Currently, S3 users can only generate the access and secret key pair needed to create, access, and manage buckets and their policies via the S3 API. To learn more, consult with your system administrator, or see [S3 user credentials \(on page 12\)](#).

Permissions

In the S3 Console app, the following rules apply to permissions:

- S3 users can view and browse buckets and objects they are given access to.
- S3 users who are not bucket owners do not have the ability to view or assign policies to buckets, even if they are provided access to those buckets.
- Bucket owners will need to have roles assigned to them by an admin to be able to configure and view bucket policies.

The following HCP for cloud scale roles can be applied and allow/disallow bucket owners to:

- Set sync-to replication policies: `data:bucket:sync:to:set`
- Set sync-from replication policies: `data:bucket:sync:from:set`
- View sync-to and sync-from replication policies: `data:bucket:sync:get`



Note: A user requires `s3:user:generate_credentials` permission to be logged into the S3 Console.

For more information on assigning roles, see the Administrator Help.

Logging in as an admin

A system admin is a user under the local *admin* account, or a user that has been assigned administrator permissions.



Important: The local *admin* user cannot log in to the S3 Console directly like basic users. The *admin* user can only login via the admin port.

To log in to the S3 Console as an administrator:

Procedure

1. Connect to the admin port:
`http://<clustername>:8000`
2. Select **S3 Console**.

Logging in as a user



Note: The local *admin* user is required to login through the admin port and can not access the S3 Console through the user login page, as it is a realmless account. To log in as an admin, see [Logging in as an admin \(on page 9\)](#).

To log in to the S3 Console as a user:

Procedure

1. Connect to the S3 port:
`https://s3.<clustername>`
2. If HCP for cloud scale is connected to a single realm (AD/LDAP), enter your **Username** and **Password**.
3. If HCP for cloud scale is connected to more than one realm, enter your **Username**, **Password**, and select the applicable **Security Realm**.

Generating S3 credentials

S3 credentials are used to connect to the S3 gateway for S3 operations and are created from the user menu. They are the credentials assigned to a bucket owner, allowing users to create and manage buckets and their objects from within HCP for cloud scale.



WARNING: Generating new credentials removes the previously generated S3 credentials. Additionally, the generated values for *Access Key* and *Secret Key* will not be viewable again beyond this point, so maintain them for your records.

To generate new S3 credentials:

Procedure

1. Click the user icon in the upper right corner of the screen.
2. Click **Generate credentials**.
A warning screen appears.

3. To create credentials, click **Generate**.
New values for **Access Key** and **Secret Key** appear.
4. To copy one of these values, click **Copy**.
5. Click **Done**.

Logging out

To log out of the S3 Console:

Procedure

1. In the upper right corner of the screen, click the user icon.
2. Click **Log out**.

Chapter 2: S3 user credentials

HCP for cloud scale includes an application to obtain S3 user credentials.

Amazon Web Services uses security credentials, called S3 credentials, to authenticate and authorize data requests. The credentials consist of an access key and a secret key. Client applications that make S3 requests to perform actions, such as uploading documents, reading documents, and adding buckets, to Hitachi Content Platform for cloud scale (HCP for cloud scale) must include these credentials.

HCP for cloud scale includes a simple application, S3 User Credentials, to obtain these credentials for registered users of the system. The application obtains an OAuth token from system services when you log in and thereafter gives credentials on demand.

Generating S3 credentials

S3 credentials are used to connect to the S3 gateway for S3 operations and are created from the user menu. They are the credentials assigned to a bucket owner, allowing users to create and manage buckets and their objects from within HCP for cloud scale.



WARNING: Generating new credentials removes the previously generated S3 credentials. Additionally, the generated values for *Access Key* and *Secret Key* will not be viewable again beyond this point, so maintain them for your records.

To generate new S3 credentials:

Procedure

1. Click the user icon in the upper right corner of the screen.
2. Click **Generate credentials**.
A warning screen appears.
3. To create credentials, click **Generate**.
New values for **Access Key** and **Secret Key** appear.
4. To copy one of these values, click **Copy**.
5. Click **Done**.

Obtaining S3 credentials

You can use the S3 User Credentials application or API to obtain S3 credentials.

The S3 User Credentials application retrieves credentials (an access key and a secret key) to access Amazon S3 bucket services. These credentials are linked to the username and password supplied in the API request. Thus, each unique user retrieves a unique set of credentials.

If a user makes multiple, repeated API requests, only the last set of credentials remains active. Previously retrieved credentials no longer work. Credentials expire automatically if a user changes his or her password stored by the identity provider.

S3 application instructions

Use the S3 User Credentials application to obtain S3 user credentials.

Obtaining credentials nullifies any pre-existing S3 credentials you already have.

To obtain S3 user credentials:

Procedure

1. From the **Applications** page, select the application **S3 User Credentials**.
2. Click **Generate S3 Credentials**.
You are warned that any existing credentials for the logged-in user will be nullified.
3. Click **Generate**.



Note: If this step fails, your session might have timed out.

The application generates and displays an **Access Key** and a **Secret Key**.

4. Click **Copy**, next to the **Access Key** field, and paste the credential into the client application that you use to make S3 requests to HCP for cloud scale.
5. Click **Copy**, next to the **Secret Key** field, and paste the credential into the client application that you use to make S3 requests to HCP for cloud scale.

Related REST API methods

```
POST /s3/user/generate_credentials
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

Revoking S3 credentials

Amazon S3 credentials can be revoked by the associated user or by other users with appropriate permissions. If you have permissions you can revoke all Amazon S3 credentials belonging to a specific user. Use the method `/user/list` to look up the ID of the user for whom you want to revoke credentials.

Related API methods

```
POST /user/list
```

```
POST /user/revoke_credentials
```

For information about specific API methods, see the *MAPI Reference* or, in the Object Storage Management application, click the profile icon and select REST API.

Logging out

To log out of the S3 Console:

Procedure

1. In the upper right corner of the screen, click the user icon.
2. Click **Log out**.

Chapter 3: Bucket management

Buckets are containers that store your data on HCP for cloud scale. They contain objects, the files and documents that you intend on storing on the cloud. Each bucket can also be assigned its own custom configuration and can be set with a unique set of permissions.


Buckets				+ Create bucket
Bucket name	Owner	Storage class	Access level	
tts1	Jimmy	STANDARD	Private	⋮
tts2	Jimmy	STANDARD	Private	⋮

Show 10 rows

To further define objects in a bucket, policies can be applied to them, which contain rules where both tags and prefixes can be added as filters to further define your virtual storage.

For more information about the different available policies, see [Bucket policies \(on page 25\)](#). For more information about tags, see [Tags \(on page 20\)](#). For more information about prefixes, see [Prefixes \(on page 19\)](#).

On the Buckets page, the following information is displayed:

- Bucket name: The name of the bucket
- Owner: The user that owns the bucket
- Storage class: The selected storage class for the bucket
-  **Note:** *S3 Standard* is currently the only supported storage class.
- Access level: The level of authentication required to use and view the bucket.
 - Private: Only you have access to this bucket.
 - Authenticated: Lets you grant access to this bucket for any user with an account on the system.
 - Unauthenticated: Lets you grant public access to this bucket for anyone. You can choose to assign Read or Read/Write privileges.

By clicking a bucket's more icon, users can find additional options for using their buckets:

- **View:** Takes users directly to the bucket's Overview page.
- **Browse:** Takes users directly to the bucket's Browse page.
- **Properties:** Takes users directly to the bucket's Properties page.
- **Delete:** Deletes the bucket.

Buckets

Creating a bucket



Tip: Bucket names can only contain lowercase letters, numbers, periods, and hyphens.

To create a bucket:

Procedure

1. From the **Buckets** page, click **+ Create Bucket**.
The **Create bucket** page appears.
2. In the **Name** field, enter a name for your bucket.
3. In the **Access level** section, select your required level of security.
 - **Private:** Only you have access to this bucket.
 - **Authenticated:** Lets you grant access to this bucket for any user with an account on the system.
 - **Unauthenticated:** Lets you grant public access to this bucket for anyone. You can choose to assign Read or Read/Write privileges.
4. In the **Bucket policies** section, chose your preferred policy using the selection slider.
 - **Expiration Lifecycle policy:** Lets you define when objects expire.
 - **Sync-to Replication:** Lets you enable automatic copying of objects to remote buckets. You cannot apply both Sync-to and Sync-from replications to the same bucket.
 - **Sync-from Replication:** Lets you enable automatic copying of objects from remote buckets. You cannot apply both Sync-to and Sync-from replications to the same bucket.
5. To add rules to your selected policy:
 - a. On your selected policy, click **Configure**.

Note: **Configure** is only displayed on policies that are selected.

 - b. On the **Configure** page, click **+ Add rule**.
 - c. To add prefixes or tags to your rule, click **Filter object**.

- d. When you are finished configuring your rule, click **Done**.
6. When you are finished configuring your policy, click **Create**.
You are returned to the **Buckets** page and a message confirming the creation of the new bucket is displayed.
7. To view your new bucket, select it by clicking its name from the **Bucket name** column.
The bucket page is displayed and an overview of your bucket is provided.

Deleting a bucket

A bucket can only be deleted if it is emptied of all objects contained within it.



WARNING: Bucket deletion is permanent.

To delete a bucket:

Procedure

1. From the **Buckets** page, navigate to the bucket you want to delete.
2. Click the more icon for your bucket and then select **Delete**.
A confirmation message appears.
3. Click **Confirm Delete**.

Rules

Rules are conditions added to policies which apply certain actions to objects containing specific prefixes. They help users to further define object expiration for current and non-current versions of objects. Policies can support up to 1,000 rules at a time.

Tags can also be added to help further define rules and categorize your storage. To learn more, see [Tags \(on page 20\)](#).

Adding rules to policies

To add a new rule to a policy:

Procedure

1. From the **Buckets** page, select the bucket by clicking its name from the **Bucket name** column.
2. Click the **Properties** tag.
3. Click **Configure** on the policy.
4. Click **+ Add rule**.
The **Add Rule** page appears.
5. To add tags to your rule, click **+ Add tag**.
6. To apply actions (for the Expiration Lifecycle policy) or to change configuration settings for your rule, scroll to the bottom of the page.

7. When finished, click **Done**.
The rule is added to the selected policy's **Configure policy** page.
8. From the **Rules** table, select the rule(s) you want to add to your policy using the checkbox column.
9. Click **Done**.
You are returned to the **Create bucket** page and your rules are noted in the **Configured rules** section of your selected policy.

Adding pre-existing rules to a policy

To add a pre-existing rule to a policy:

Procedure

1. From the **Buckets** page, select the bucket by clicking its name from the **Bucket name** column.
2. Click the **Properties** tab.
3. Click **Configure** on your selected policy.
4. From the **Rules** table, select the rule(s) you want to add to your policy using the checkbox column.
5. Click **Done**.
You are returned to the **Create bucket** page and your rules are noted in the **Configured rules** section of your selected policy.

Editing a rule

To edit a rule:

Procedure

1. From the **Buckets** page, select the bucket by clicking its name from the **Bucket name** column.
2. Click the **Properties** tag.
3. Click **Configure** on the policy.
4. Click the more button for your rule and then select **Edit**.
5. Click **Done**.

Deleting rules from a policy

To delete a rule from a policy:

Procedure

1. From the **Buckets** page, select the bucket by clicking its name from the **Bucket name** column.
2. Click the **Properties** tag.
3. Click **Configure** on the policy.
4. Click the more button for your rule and then select **Delete**.

5. Click **Okay**.

Filters

Each rule can be assigned a filter. Filters help users identify a subset of objects in a bucket in to which the rule applies. They can be assigned as a prefix or tags and help you to further define and categorize your storage.

Prefixes

Prefixes are an S3 concept that represent the path to a virtual folder. They are a way to help users visualize the concept of Cloud storage and storage, given that no actual folders truly exist, and allow users to group objects by using common strings.

For example, setting a rule that applies to the `/foo` prefix would apply to all objects that start with `/foo`, such as `/foo/bar`, `/foo/bar1/bar2`, etc.



Note: A rule can only be assigned a single prefix.

Adding a prefix filter to a policy



Important: When adding a prefix filter to an Expiration Lifecycle policy, users must enable at least one action on the policy.

To add a prefix filter to a policy:

Procedure

1. From the **Buckets** page, select the bucket by clicking its name from the **Bucket name** column.
2. Click the **Properties** tag.
3. Click **Configure** on the policy.
4. Click **+ Add rule**.
The **Add Rule** page appears.
5. Click **Filter objects**.
The **Prefix** field and **Tags** list appear.
6. In the **Prefix** field, enter your prefix.
7. Click **Done**.

Editing a prefix filter

To edit a prefix filter on a policy:

Procedure

1. From the **Buckets** page, select the bucket by clicking its name from the **Bucket name** column.

2. Click the **Properties** tag.
3. Click **Configure** on the policy.
The **Rules** page appears.
4. Click the more icon for your prefix filter and select **Edit**.

Removing a prefix filter from a policy

To remove a prefix filter from a policy:

Procedure

1. From the **Buckets** page, select the bucket by clicking its name from the **Bucket name** column.
2. Click the **Properties** tag.
3. Click **Configure** on the policy.
The **Rules** page appears.
4. Click the more icon for your prefix filter and select **Delete**.

Tags

Tags are independent of prefixes. Users can create a rule with only tag(s) without a prefix. A filter with tag(s) means that a rule applies only to objects that have tags matching the tags in the rule. A rule applies AND operator to all filter conditions, so all specified tags must match with tags in the object for the rule to apply to that object. Tags are used to help further define rules and categorize your storage. They are represented as a key-value pair and are added to rules representing a specific value for that prefix.

For example, you could apply a tag to the `/foo` prefix as a rule for your policy with a key of `Classification` and a value of `Internal` to help further classify and define the permissions of objects in the `/foo` directory.

Adding tags to rules

To add a tag to a rule:

Procedure

1. From the **Buckets** page, select the bucket by clicking its name from the **Bucket name** column.
2. Click the **Properties** tag.
3. Click **Configure** on your selected policy.
4. Click **+ Add rule**.
5. Click **Filter objects**.
The **Prefix** field and **Tags** list appear.
6. To add tags to your rule, click **+ Add tag**.
The **Add tag** window appears.
7. In the **Key** and **Value** fields, enter your tagging information.
8. When finished, click **Save**.

9. (Optional) To add additional tags, click the **Add another tag** box and then click **Save**.
10. On the **Add rule** page, click **Done**.

Editing a tag

To edit a tag which is part of a current rule:

Procedure

1. From the **Buckets** page, select the bucket by clicking its name from the **Bucket name** column.
2. Click the **Properties** tag.
3. Click **Configure** on your selected policy.
4. On the **Configure policy** page, click the more icon for the rule containing your tag and then select **Edit**.
The **Add Rule** page appears.
5. Click **Filter objects**.
The **Prefix** field and **Tags** list appear.
6. To edit a tag, click its more icon from the **Tags** list and then select **Edit**.
7. In the **Key** and **Value** fields, update your tagging information.
8. When finished, click **Save**.
9. To finalize your changes, on the **Add rule** page, click **Done**.

Deleting a tag filter

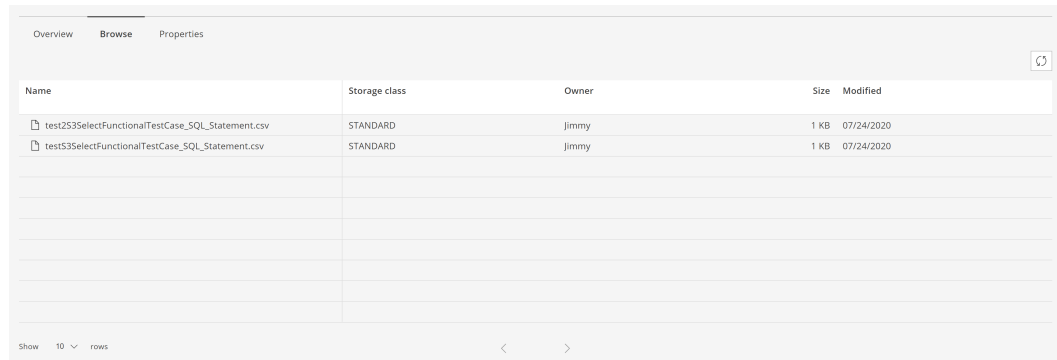
To delete a tag from a rule:

Procedure

1. From the **Buckets** page, select the bucket by clicking its name from the **Bucket name** column.
2. Click the **Properties** tag.
3. Click **Configure** on your selected policy.
4. On the **Configure policy** page, click the more icon for the rule containing your tag and then select **Delete**.

Chapter 4: Object management

An object is any file or document contained within a bucket. All objects that are currently stored in a bucket can be viewed from the Browse tab.



The screenshot shows the 'Browse' tab of an S3 console interface. It features a table with columns for Name, Storage class, Owner, Size, and Modified. Two objects are listed, both named 'testS3SelectFunctionalTestCase_SQL_Statement.csv', with a storage class of 'STANDARD', owned by 'Jimmy', and a size of '1 KB', both modified on '07/24/2020'. The interface includes tabs for 'Overview', 'Browse', and 'Properties', a refresh button, and pagination controls at the bottom.

Name	Storage class	Owner	Size	Modified
testS3SelectFunctionalTestCase_SQL_Statement.csv	STANDARD	Jimmy	1 KB	07/24/2020
testS3SelectFunctionalTestCase_SQL_Statement.csv	STANDARD	Jimmy	1 KB	07/24/2020

On the object page, the users can view the following information:

- Bucket name: The name of the object
- Size: The size of the object (in MB, GB, TB, or PB)
- Storage class: The selected storage class for the bucket



Note: Currently, the S3 Standard class is the only supported storage class.

- Owner: The user that owns the object



WARNING: Changes made by Delete all versions can not be reversed.

Chapter 5: Monitoring

The S3 Console provides powerful metrics that let you track input/output operations, the loading of objects (ingest), the number of objects stored, and the disk usage of stored objects.

Bucket metrics

All available buckets provide users with a dashboard to view its performance. They can be found on the Overview tab of any bucket, which is automatically displayed upon selecting a bucket.



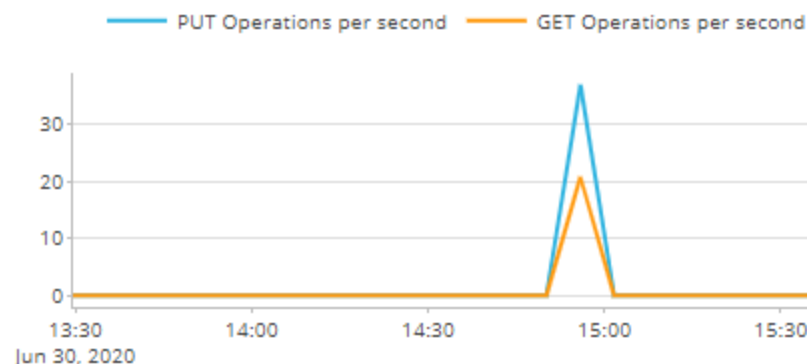
Tip: Using the key above the graphs, double-clicking a specific metric will enable or disable its visualization on the graph.

The dashboard can be customized to display different ranges of time by clicking Range. You can then filter your metrics by selecting Live, 1 Hour, 1 Day, 1 Week, 4 Weeks, and 1 Year views of the graphs.

IOPS dashboard

Displays the input/output operations per second (IOPS). Users can hover over the PUT Operations per second or GET Operations per second and DELETE Operations per second lines to be provided with specific metrics for specific data points.

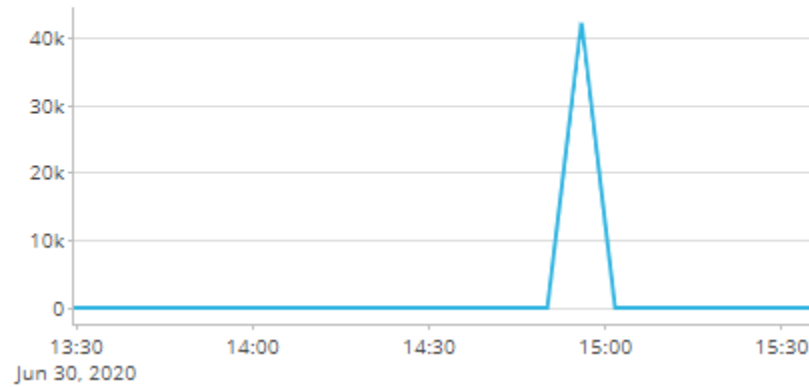
IOPS



Ingest throughput dashboard

The size (in bytes) of files and documents that have been taken into the bucket. Users can hover over the *Bytes ingested* line to be provided with specific metrics for specific data points.

Ingest throughput



Chapter 6: Policy management

Policies allow users to apply specific retention and permissions to buckets and the objects contained within.

Bucket policies

Currently, the S3 Console supports the following policies:

- Expiration Lifecycle policy
- Sync-from Replication policy
- Sync-to Replication policy



Tip: When navigating through policies, users can use the breadcrumbs found under the bucket's name to quickly navigate back to previous screens.

Expiration Lifecycle policy

The *Expiration Lifecycle* policy sets an expiration date on the objects within a bucket.

A set of rules is applied to this policy that define actions across groups of objects. These rules can apply to current versions, non-current versions, incomplete multi-part uploads, and expired delete markers.

Each policy can contain up to 1,000 rules. Additionally, each rule contains filters (such as prefixes and tags), as well as actions.



Important: When adding a rule to this policy, the All objects filter is selected by default. To add tags, a prefix, or both, click Filter objects.

Actions apply to all objects in the bucket and are specific to the Expiration Lifecycle policy. They can be added as individually set rules and do not require tags or a prefix. Currently, the following four expiration actions are supported:

- **Current versions:** Permanently deletes an object after a set number of days from object creation, or on a specific date. The default is 365 days.
- **Non-current versions:** Permanently deletes an object after a set number of days from having been made a previous version, or on a specific date. The default is 30 days.
- **Incomplete multi-part uploads:** Removes partial MPU uploads if they are not successfully completed within a set number of days.
- **Expired delete markers:** Retains an expired delete marker in the event that all previous versions of an object expire after the deletion of a versioned object. The default is 7 days.



Important: The Expired delete markers policy cannot be set if the Current versions policy is enabled.



Note: Actions are applied as to your Expiration Lifecycle policy as rules and can be implemented from the Add rule page of a given bucket.

Adding an expiration lifecycle policy to a new bucket

To add an expiration lifecycle policy to a new bucket:

Procedure

1. From the **Buckets** page, click **Create bucket**.
2. Enable the **Expiration Lifecycle** policy by clicking its selection toggle.
3. Configure your policy by clicking **Configure**.
4. Click **Create**.

Adding an Expiration Lifecycle policy to a pre-existing bucket

To add an Expiration Lifecycle policy to a pre-existing bucket:

Procedure

1. From the **Buckets** page, navigate to your respective bucket and click its name in the **Bucket name** column to select it.
2. Enable the **Expiration Lifecycle** policy by clicking its selection toggle.
3. Edit your policy by clicking **Configure**.
4. Click **Create**.

Adding actions to an Expiration Lifecycle policy

To add actions to an Expiration Lifecycle policy:

Procedure

1. From the **Buckets** page, select the bucket by clicking its name from the **Bucket name** column.
2. Click the **Properties** tag.
3. Click **Configure** on the policy.
4. Click **+ Add rule**.
5. From the **Actions** section, select your preferred action by clicking its selection slider.
 - To place a hold on current versions, enable **Current versions**. You can then set a number of days to hold these files or a specific date by which they will be deleted.
 - To place a hold on previous versions, enable **Non-current versions**. You can then set a number of days to hold these files.
 - To place a hold on incomplete multi-part uploads, enable **Incomplete multi-part uploads**. You can then set a number of days to hold these partially uploaded files until they are deleted.
 - Optionally, you can enable **Expired delete markers** to automatically remove expired objects.
6. Once selected, configure your action.
7. Click **Done**.
The action is added as a rule to your policy.

Editing a Expiration Lifecycle policy

To edit an Expiration Lifecycle policy:

Procedure

1. From the **Buckets** page, navigate to your respective bucket and click its name in the **Bucket name** column to select it.
2. On the bucket's page, click the **Properties** tab.
3. Edit the **Expiration Lifecycle** policy by clicking **Configure**.
4. Click **Update**.

Removing an Expiration Lifecycle policy

To remove an Expiration Lifecycle policy from a bucket:

Procedure

1. From the **Buckets** page, navigate to your respective bucket and click its name in the **Bucket name** column to select it.
2. On the bucket page, click the **Properties** tab.
3. Remove the **Expiration Lifecycle** policy by clicking its selection toggle.
The policy is greyed out.
4. Click **Update**.
The policy is removed from the bucket.

Sync-from Replication policy

The *Sync-from Replication* policy provides information about replicated objects, their remote buckets, and information from the remote queue.

A set of rules that define asynchronous replication *from* remote buckets is applied. Each rule defines the objects to be replicated, the remote bucket these objects are replicated from, and the corresponding AWS SQS queue. The queue is used for notifications about the changes in the remote bucket.

Each policy can contain up to 1,000 rules and each rule contains filters (such as prefixes and tags). If a filter is not applied to a Sync-from Replication policy on bucket, then the policy applies to all objects.



Important: When adding a rule to this policy, the All objects filter is selected by default. To add tags, a prefix, or both, click Filter objects.

Adding a Sync-from Replication policy to a new bucket

To add a Sync-from Replication policy to a new bucket:

Procedure

1. From the **Buckets** page, click **Create bucket**.
2. Enable the **Sync-from Replication** policy by clicking its selection toggle.
3. Edit your policy by clicking **Configure**.
 - a. Add your S3 access information to the **Remote bucket configuration** section.
 - b. *Optional:* To test your S3 connection, click the **Test bucket connection** button.
 - c. Add your AWS SQS credentials to the **AWS SQS queue** section.



Important: The **Queue** and **Region** fields are the *from* of the sync-from replication policy.

- d. *Optional:* To test your AWS SQS queue connection, click the **Test queue connection** button.
4. Click **Create**.

Adding a Sync-from Replication policy to a pre-existing bucket

To add a replication Sync-from Replication policy to a pre-existing bucket:

Procedure

1. From the **Buckets** page, navigate to your respective bucket and click its name in the **Bucket name** column to select it.
2. Enable the **Sync-from Replication** policy by clicking its selection toggle.
3. Edit your policy by clicking **Configure**.
4. Click **Update**.

Editing a Sync-from Replication policy

To edit a bucket's Sync-from Replication policy:

Procedure

1. From the **Buckets** page, navigate to your respective bucket and click its name in the **Bucket name** column to select it.
2. On the bucket's page, click the **Properties** tab.
3. Edit the **Sync-from Replication** policy by clicking **Configure**.
4. Click **Update**.

Removing a Sync-from Replication policy

To remove a Sync-from Replication policy from a bucket:

Procedure

1. From the **Buckets** page, navigate to your respective bucket and click its name in the **Bucket name** column to select it.
2. On the bucket page, click the **Properties** tab.
3. Remove the **Sync-from Replication** policy by clicking its selection toggle.
The policy is greyed out.
4. Click **Update**.
The policy is removed from the bucket.

Sync-to Replication policy

The *Sync-to Replication* policy provides information about replicated objects and their remote buckets.

A set of rules that define asynchronous replication to remote buckets is applied. Each rule defines the objects to be replicated and the remote bucket these objects are to be replicated in.

Each policy can contain up to 1,000 rules and each rule contains filters (such as prefixes and tags). If a filter is not applied to an Sync-to Replication policy on bucket, then the policy applies to all objects.



Important: When adding a rule to this policy, the All objects filter is selected by default. To add tags, a prefix, or both, click Filter objects.

Adding a Sync-to Replication policy to a new bucket

To add a Sync-to Replication policy to a new bucket:

Procedure

1. From the **Buckets** page, click **Create bucket**.
2. Enable the **Sync-to Replication** policy by clicking its selection toggle.
3. Edit your policy by clicking **Configure**.

- a. Add your S3 access information to the **Remote bucket configuration** section.



Important: The **S3 hostname** field is the *to* of the sync-to replication policy.

- b. *Optional:* To test your S3 connection, click the **Test bucket connection** button.

4. Click **Create**.

Adding a Sync-to Replication policy to a pre-existing bucket

To add a Sync-to Replication policy to a pre-existing bucket:

Procedure

1. From the **Buckets** page, navigate to your respective bucket and click its name in the **Bucket name** column to select it.
2. Enable the **Sync-to Replication** policy by clicking its selection toggle.
3. Configure your policy by clicking **Configure**.
4. Click **Update**.

Editing a Sync-to Replication policy

To edit a bucket's Sync-to Replication policy:

Procedure

1. From the **Buckets** page, navigate to your respective bucket and click its name in the **Bucket name** column to select it.
2. On the bucket's page, click the **Properties** tab.
3. Edit the **Sync-to Replication** policy by clicking **Configure**.
4. Click **Update**.

Removing a Sync-to Replication policy

To remove a Sync-to Replication policy from a bucket:

Procedure

1. From the **Buckets** page, navigate to your respective bucket and click its name in the **Bucket name** column to select it.
2. On the bucket page, click the **Properties** tab.
3. Remove the **Sync-to Replication** policy by clicking its selection toggle.
The policy is greyed out.
4. Click **Update**.
The policy is removed from the bucket.

Chapter 7: Bucket synchronization

Hitachi Content Platform for cloud scale (HCP for cloud scale) lets you configure and manage bucket synchronization.

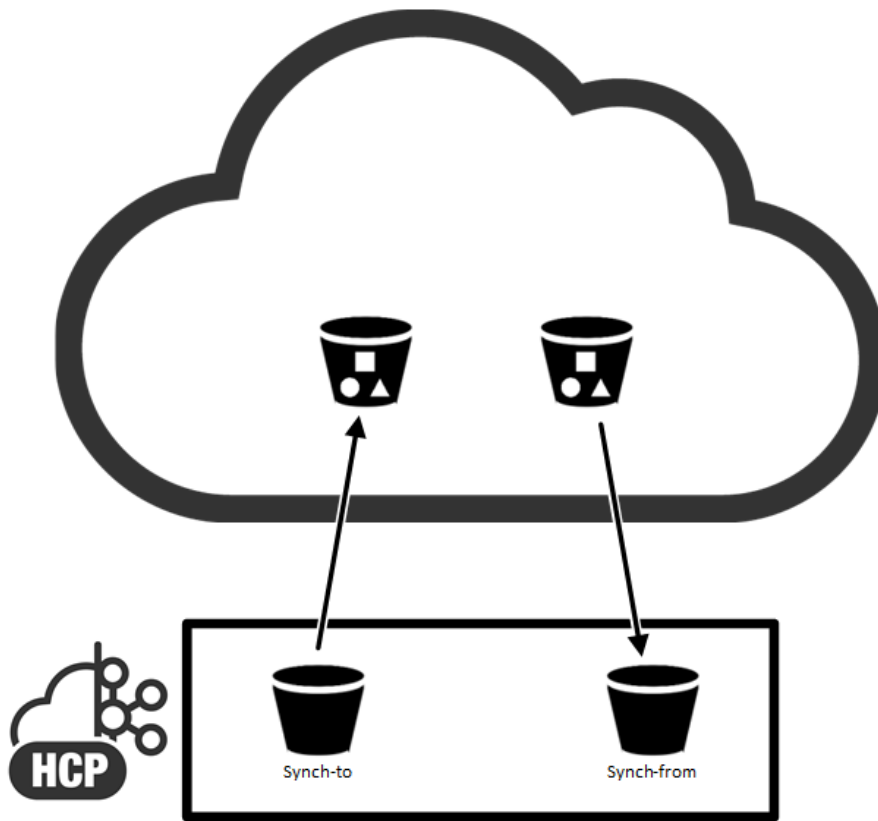
To configure bucket synchronization, use S3 `put bucket replication` API requests. Scripts are available to simplify the process.

About bucket synchronization

HCP for cloud scale can synchronize the following kinds of data in buckets:

- Object data
- All user metadata (that is, anything that can be returned in the header `x-amz-meta-*`)
- Tags
- `Content-Type` system metadata
- Objects that the owner of the source bucket doesn't have permission to read

This diagram illustrates the concept of bucket synchronization.



Limitations on bucket synchronization

Objects that existed before synchronization functions are configured are not synchronized.

HCP for cloud scale verifies the rules that are valid at the time an object is synchronized, not at the time the object is ingested.

Objects that are marked as deleted are not synchronized.

Most system metadata is not synchronized, specifically:

- Owner ID and Name
- Timestamps (when last modified)
- Metadata returned in `x-amz-grant-*`
- Metadata returned in `x-amz-acl`
- Metadata returned in `x-amz-grant-*`
- Metadata returned in `x-amz-acl`
- Metadata returned in `x-amz-storage-class`
- Metadata returned in `x-amz-replication-status`
- Metadata returned in `x-amz-server-side-encryption-*`
- Metadata returned in `x-amz-restore-*`
- Metadata returned in `x-amz-version-id-*`

- Metadata returned in `x-amz-website-redirect-location`
- Metadata returned in `x-amz-object-lock-*`

The bucket sync-from function only supports one rule for the same external SQS queue and external bucket. If a bucket has multiple sync-from rules for the same external queue, objects might not be synchronized. To use multiple rules for an external bucket, use one SQS queue for each rule.

Comparing synchronization to replication

Unlike AWS replication, HCP for cloud scale can synchronize with buckets on storage systems outside of AWS.

AWS determines the destination bucket using rules, but only applies one rule to each new object. In contrast, HCP for cloud scale can apply multiple rules to each new object so long as the destination buckets are different. This is how one-to-many synchronization is implemented.

AWS does not replicate, but HCP for cloud scale synchronizes, objects that the owner of the source bucket doesn't have permission to read.

In contrast with AWS replication, HCP for cloud scale does not synchronize the following:

- Access control lists (ACLs)
- Lock retention information
- Objects that are encrypted using Amazon S3 managed keys (SSE-S3) and AWS KMS managed keys (SSE-KMS)

If an object being synchronized has the same name as an object in the target bucket, the result depends on whether the target bucket uses versioning:

- If versioning is used, the old object is kept as an old version.
- If versioning is not used, the old object is replaced by the new object.

HCP for cloud scale buckets always use versioning. The best practice is to use versioning in all target buckets.

Best-effort ordering

HCP for cloud scale guarantees that operations are applied in the order of their arrival (*strong consistency*). However, synchronizing multiple operations applied in a short period of time to the same object presents the following difficulties:

- In a distributed system, especially when many systems are involved, synchronizing all operations in correct order is complex.
- Even if HCP for cloud scale synchronizes all operations in correct order to an external storage component, that component might not guarantee that the operations are applied with strong consistency. In particular, AWS guarantees only "eventual consistency."
- For bucket sync-from, the external queue service might not guarantee that messages are provided in correct order. In particular, AWS Simple Queue Service (SQS) does not support first-in, first-out (FIFO) queues for S3 notifications.

Therefore, HCP for cloud scale makes its best effort to synchronize only the latest state of an object, not each version or operation for the object. For example:

- Assume that a client sends three operations to an object and that they are all committed: (1) PUT, (2) PUT, (3) DEL. The latest state of the object is (3) DEL. HCP for cloud scale only synchronizes DEL.
- Assume that a client sends three operations to an object and that they are all committed: (1) PUT, (2) DEL, (3) PUT. The latest state of the object is (3) PUT. HCP for cloud scale only synchronizes (3) PUT.

This approach does not guarantee that the latest state of an object will be in the external storage for all situations. Partly because of the "eventual consistency" offered by AWS S3 API, corner cases still exist.

Synchronization to an external bucket: high-level tasks

Synchronization to an external bucket involves assigning roles and permissions to users, creating and synchronizing the buckets, and then reading from and writing to the buckets.

This description of high-level tasks assumes three classes of user:

1. An HCP for cloud scale system administrator to create roles and assign them to users using an IdP
2. An HCP for cloud scale bucket administrator, who could be a tenant administrator, to create and configure an HCP for cloud scale bucket
3. An Amazon Web Services (AWS) user, who could be a customer, to create a remote bucket using AWS S3 and then read and write data



Note: The default HCP for cloud scale account has full permissions and can perform the tasks assigned to the first two user classes.

Procedure

1. The system administrator assigns permissions to the bucket administrator to configure bucket synchronization.
 - a. In the System Management application, create a role with the permission group **bucket_sync**.
 - b. In the IdP server, set up two groups: bucket administrators and bucket users.
 - c. In the IdP server, register users in these groups.
 - d. In the System Management application, assign the role to the bucket administrator group.
2. The bucket administrator creates local and remote buckets.
 - a. In the S3 User Credentials application, generate S3 credentials.



Tip: Use the base64 utility to encode S3 credentials.

- b. Using the S3 credentials, use an S3 API to create an HCP for cloud scale (local) bucket.

- c. Use an AWS S3 API to create an S3 (remote) bucket.
3. The bucket administrator configures bucket synchronization between the HCP for cloud scale bucket and the S3 bucket using an S3 PUT Bucket Replication method, replacing the bucket ARN with configuration settings. By using multiple rules and filters, the bucket administrator can specify what objects are synchronized to the S3 bucket.
4. The bucket administrator sets access control lists to let the bucket user write data to the HCP for cloud scale bucket.
 - a. Using a management API, get the user ID of the bucket user.
 - b. Using an S3 API, assign write permission to the bucket user for the HCP for cloud scale bucket.
5. The AWS user is now free to write objects to the HCP for cloud scale bucket, which is now synchronized with the remote bucket.

Synchronization from an external bucket: high-level tasks

Synchronization from an external bucket involves assigning roles and permissions to users, creating and synchronizing buckets, and then reading from and writing to the buckets.

This description of high-level tasks assumes three classes of user:

1. An HCP for cloud scale system administrator to create roles and assign them to users using an IdP
2. An HCP for cloud scale bucket administrator, who could be a tenant administrator, to create and configure an HCP for cloud scale bucket
3. An AWS user, who could be a customer, to create a remote bucket using AWS S3, create an AWS SQS queue, and then configure S3 notifications to SQS



Note: The default HCP for cloud scale account has full permissions and can perform the tasks assigned to the first two user classes.

Procedure

1. The system administrator assigns permissions to the bucket administrator to configure bucket synchronization.
 - a. In the System Management application, create a role with the permission group **bucket_sync**.
 - b. In the IdP server, set up two groups: bucket administrators and bucket users.
 - c. In the IdP server, register users in these groups.
 - d. In the System Management application, assign the role to the bucket administrator group.
2. The bucket administrator creates local and remote buckets.
 - a. In the S3 User Credentials application, generate S3 credentials.



Tip: Use the base64 utility to encode S3 credentials.

- b. Using the S3 credentials, use an S3 API to create an HCP for cloud scale (local) bucket.
 - c. Use an AWS S3 API to create an S3 (remote) bucket.
3. The AWS user creates a standard queue in SQS.
 - a. Using an AWS account, create a queue of the type **Standard Queue**.
 - b. Create a policy document.
4. The AWS user configures the remote bucket to send S3 notifications to the AWS SQS queue.
 - a. Add a notification for all object creation events to the remote bucket.
5. The bucket administrator configures bucket synchronization between the S3 bucket and the HCP for cloud scale bucket using an S3 PUT Bucket Replication method, replacing the bucket ARN with configuration settings. By using multiple rules and filters, the bucket administrator can specify what objects are synchronized to the local bucket.
6. The bucket administrator sets access control lists to let the bucket user read data from the HCP for cloud scale bucket.
 - a. Using a management API, get the user ID of the bucket user.
 - b. Using an S3 API, assign write permission to the bucket user for the HCP for cloud scale bucket.
7. The AWS user is now free to read objects from the HCP for cloud scale bucket, which is now synchronized with the remote bucket.

Bucket synchronization configuration

Bucket synchronization is configured using `PUT bucket replication` API requests that define rules. Each bucket can have up to 1,000 rules, but all rules must be sync-to or sync-from rules. Each rule defines the following:

- External bucket settings
- A set of one or more prefixes; an object with one of the prefixes is mirrored
- A set of one or more tags; an object with all, or any, of the tags is mirrored
- For sync-from, external queue settings

Because you can configure multiple rules with multiple tags, you have flexibility in selecting objects to mirror. For example:

- To mirror all objects that contain Tag₁ and Tag₂, you can configure one rule that includes both tags.
- To mirror all objects that contain Tag₁ or Tag₂, you can configure two rules, one for each tag.

For information on `PUT bucket replication` see [Configure bucket synchronization \(PUT bucket replication\) \(on page 37\)](#).

Visibility of new buckets and objects

After they are created, buckets and objects are not immediately visible. Some client applications (such as Cloudberry Explorer) immediately retrieve the list of buckets to display the new bucket or object, which is not visible. If you create a new bucket or object and it's not immediately visible, update the list manually.

Rule collisions

HCP for cloud scale can apply multiple bucket synchronization rules to each new object so long as the destination buckets are different. This is how one-to-many synchronization is implemented.

A rule collision is when two or more rules that apply to an object have the same destination (that is, the same external host, port, and bucket). HCP for cloud scale does not allow rule collisions, so `PUT bucket replication` requests are rejected if they contain rule collisions. To avoid rule collisions, you can define as many tags in a rule as necessary, so that multiple rules with the same destination are not needed.

Effect of configuration changes

After an object operation is performed, the policy engine asynchronously checks if that object needs to be copied according to the sync-to rules. When bucket synchronization rules are created, updated, or deleted, the changes only apply to new objects, object operations, and to objects that have not been yet processed by the policy engine. Objects that existed before the rules were configured are not synchronized. If an object exists in the `PENDING` state when a rule is created, updated, or deleted, the rule change might not be applied.

Synchronizing to the same source and destination

You cannot set up bucket synchronization with the same bucket as both the source and the destination.

Configure bucket synchronization (PUT bucket replication)

You can configure S3 bucket sync-to and sync-from settings.



Notes:

- If you use the AWS command-line interface to configure bucket synchronization, use at least `aws-cli v1.16.211` and `aws-sdk 1.11.610`.
- Configuration rules should be provided to AWS CLI from a file, rather than inline. This is to avoid problems with double quote characters in some terminals.

HTTP request syntax (URI)

```
aws --endpoint-url https://10.08.1019 s3api put-bucket-replication --
bucket "hcpcs_bucket" --replication-configuration file://rules.json
```

Request structure

A rule consists of up to 1000 prefixes and tag-value pairs. You can configure up to 1000 rules per bucket. Separate tag-value pairs in the rule using the keywords "And" : or "Or" :.

The content of the configuration JSON file is:

```
{
  "Role": "",
  "Rules": [{
    "ID": "string",
    "Filter": {
      "Prefix": "string",
      "Tag": {
        "Key": "string",
        "Value": "string"
      }
    },
    "Status": "boolean",
    "Destination": {
      "Bucket": "json"
    }
  },
  .
  .
  .
  }]
```



Note: S3 parameters not shown are not required, not supported, and if specified should be left empty.

Account Parameter	Required	Type	Description
Role	Yes	N/A	Not supported; leave empty.
ID	No	String	Unique identifier for rule, up to 255 characters. All rules must specify the same bucket.
Priority	Yes	Integer	Not supported; ignored.
DeleteMarkerReplication.Status	No	String	Not supported; if provided, leave as <code>Disabled</code> .
Prefix	No	String	Prefix (one per rule). Up to 1024 characters.

Account Parameter	Required	Type	Description
Key	No	String	Tag key (up to 1000 per rule). Up to 128 characters.
Value	No	String	Tag value. Up to 256 characters.
Rules.Status	Yes	Boolean	Enabled or Disabled. If Disabled, rule is ignored.
Rules.Destination.Bucket	Yes	Base64-encoded JSON	<p>External S3 bucket access settings.</p> <ul style="list-style-type: none"> For bucket sync-to, the settings to access the external bucket. For bucket sync-from, the settings to access the external bucket and the SQS queue settings. <p>You can't specify the same bucket name and host as both source and destination.</p>
Rules.Destination.Account	No	N/A	Not supported; leave empty.

Bucket sync-to structure

Bucket sync-to settings are defined by a set of parameters and passed in the value of `Rules.Destination.Bucket` as a Base64-encoded JSON structure.

The syntax inside the bucket parameter for the sync-to setting is:

```
{
  'version': 'version',
  'action': 'sync-from',
  'externalBucket': {
    'host': 'host',
    'type': 'type',
    'region': 'region',
    'remoteBucketName': 'bucket_name',
    'accessKey': 'B64_key',
    'secretKey': 'B64_key',
    'port': 'port',
    'authVersion': 'auth_version',
    'usePathStyleAlways': '[true|false]'
  },
  'notifications': {
    'type': 'type',
```

```

    'region': 'region',
    'queue': 'queue',
    'accessKey': 'B64_key',
    'secretKey': 'B64_key'
  }
}

```

Parameter	Required	Type	Description
version	Yes	String	1.0.
host	Yes	IP address	Host IP address.
type	Yes	String	Destination storage class: <code>AMAZON_S3</code> or <code>GENERIC_S3</code> .
region	Yes	String	The S3 region.
remoteBucketName	Yes	String	The name of the bucket, from 3 to 63 characters long, containing only lowercase characters (a-z), numbers (0-9), periods (.), or hyphens (-). The bucket must already exist.
accessKey	Yes	Base64 encoded string	The S3 access key credentials to the external S3 bucket.
secretKey	Yes	Base64 encoded string	The S3 secret key credentials to the external S3 bucket.
port	Yes	integer	Host port.
authVersion	Yes	String	AWS Signature version: <code>v2</code> or <code>v4</code> .
usePathStyleAlways	Yes	Boolean	Path-style URLs for bucket access: <code>true</code> or <code>false</code> .

Bucket sync-from structure

Bucket sync-from settings include both a bucket address and a notification queue. The settings are defined by a set of parameters and passed in the value of `Rules.Destination.Bucket` as a Base64-encoded string.

The syntax inside the bucket parameter for sync-from setting is:

```

"{
  'version': 'version',
  'action': 'sync-from',
  'externalBucket': {
    'host': 'host',

```



```

    'type': 'type',
    'region': 'region',
    'remoteBucketName': 'bucket_name',
    'accessKey': 'B64_key',
    'secretKey': 'B64_key',
    'port': 'port',
    'authVersion': 'auth_version',
    'usePathStyleAlways': '[true|false]'
  }
}"

```

Parameter	Required	Type	Description
version	Yes	String	Enter 1.0.
host	Yes	IP address	Host IP address.
type	Yes	String	Destination storage class: AMAZON_S3 or GENERIC_S3.
region	Yes	String	The S3 region.
remoteBucketName	Yes	String	The name of the bucket, from 3 to 63 characters long, containing only lowercase characters (a-z), numbers (0-9), periods (.), or hyphens (-). The bucket must already exist.
accessKey	Yes	Base64 encoded string	The S3 access key credentials to the external S3 bucket.
secretKey	Yes	Base64 encoded string	The S3 secret key credentials to the external S3 bucket.
port	Yes	integer	Host port.
authVersion	Yes	String	AWS Signature version: v2 or v4.
usePathStyleAlways	Yes	Boolean	Path-style URLs for bucket access: true or false.
Destination.type	Yes	String	Always set as AWS_SQS.
Destination.region	Yes	String	Region of your AWS_SQS queue.
Destination.queue	Yes	String	Name of your AWS_SQS queue.
Destination.accessKey	Yes	Base64 encoded string	accessKey for permissions to read from your AWS_SQS queue.

Parameter	Required	Type	Description
Destination.secretKey	Yes	Base64 encoded string	secretKey for permissions to read from your AWS_SQS queue.

Response structure

None.

Example

Request example:

```
aws --endpoint-url https://10.08.1019 s3api put-bucket-replication --
bucket "hcpcs_bucket" --replication-configuration file://rules.json
```

Configuration rules.json:

```
{
  "ID": "sync_rule2_for_music",
  "Filter": {
    "Prefix": "/music/october/",
    "Tag": {
      "Key": "target",
      "Value": "cloud"
    }
  },
  "Status": "Enabled",
  "Destination": {
    "Bucket": "{
      'version' : '1.0',
      'action' : 'sync_from',
      'externalBucket' : {
        'type' : 'AMAZON_S3',
        'region' : 'us-east-1',
        'remoteBucketName' : 'bluebucket',
        'authVersion' : 'V4',
        'usePathStyleAlways' : 'true',
        'accessKey' : 'access_key',
        'secretKey' : 'secret_key'
      },
      'notifications' : {
        'type' : "AMAZON_SQS",
        'region' : "us-east-1",
        'queue' : "testQueue",
        'accessKey' : "access_key",
        'secretKey' : "secret_key"
      }
    }
  }
}
```

```

    }
  },
}
}
}]
}

```

Script to generate bucket sync-to JSON

HCP for cloud scale includes a script to generate the JSON needed to configure bucket synchronization to an external bucket (sync-to).

The script is written in Python and located in the folder *install_path/product/bin* (for example, */opt/hcps/bin*).

The script generates the JSON string that you can use for the field `destination.bucket` in the AWS S3 command `put-bucket-replication`. Optionally, the script verifies whether the destination bucket exists. If you omit the secret key, the script prompts you for it, which lets you create a script that calls this script without storing the secret key. You can mix the short and full form of arguments.



Note: The script produces JSON using single quotes.

Syntax

```

SyncToBucketJsonGenerator.py
  [--help]
  --s3host host
  --region region
  --bucket bucket
  --accessKey access_key
  [--secretKey secret_key]
  [--s3type { GENERIC_S3 | AMAZON_S3 }]
  [--port port]
  [--authVersion { v2 | v4 }]
  [--usePathStyleAlways {true | false}]
  [--jsonSample output_file.json]
  [--verifyTarget]
  [--http]
  [--quietMode]

```

Options and parameters

-h, --help

Optional. Displays a help message and exits.

--s3host *host*, -s3 *host*

Host name of the remote S3 storage component.

--region *region*, -r *region*
Region of the remote bucket.

--bucket *bucket*, -b *bucket*
Name of the remote bucket.

--accessKey *access_key*, -ak *access_key*
Access key for the remote bucket.

--secretKey *secret_key*, -sk *secret_key*
Secret key for the remote bucket. The script prompts for the key if you don't specify it.

--s3type { *GENERIC_S3* | *AMAZON_S3* }, -s3t { *GENERIC_S3* | *AMAZON_S3* }
Optional. The remote bucket type:

- *GENERIC_S3*: An S3-compatible node
- *AMAZON_S3*: An Amazon Web Services S3-compatible node

If not specified, the default bucket type is *AMAZON_S3*.

--port *port*, -p *port*
Optional. Port of the remote bucket. If not specified, the default port is 443.

--authVersion { *v2* | *v4* }, -av { *v2* | *v4* }
Optional. The Auth Version of the remote bucket. If not specified, the default version is *v4*.

--usePathStyleAlways {*true* | *false*}, -upsa {*true* | *false*}
Optional. Sets the Use Path Style Always flag for the remote bucket. If not specified, the default is *true*.

--jsonSample *output_file.json*, -json *output_file.json*
Optional. Creates a file named *output_file.json* with a sample JSON structure for bucket replication configuration. If not specified, no sample file is created.

--verifyTarget, -verify
Optional. Verifies that the remote bucket exists. SSL certificates are not validated. This option requires *python3* and *boto3*. If not specified, the bucket's existence isn't verified.

--http, -http
Optional. Use HTTP when verifying the remote bucket. If not specified, the default is to use HTTPS.

--quietMode, -qm
Optional. Displays only the Destination.Bucket element.



Note: You can't specify both `--quietMode` and `--verifyTarget` together.

Example

```
$ SyncToBucketJsonGenerator.py -s3 s3.us-east-2.amazonaws.com -b hcpcs-
bucket-5 -r us-east-2 -ak A1234567890 -sk S1234567890 -verify -json
testto.json
```

This example can produce the following output:

```
Verifying that a remote bucket "hcpcs-bucket-5" exists...
Verification successfully completed: remote bucket "hcpcs-bucket-5" is
FOUND

Generated a JSON string for the Destination->Bucket element for bucket
replication sync-to configuration:

{'action': 'sync-to', 'version': '1.0', 'externalBucket': {'host': 's3.us-
east-2.amazonaws.com', 'type': 'AMAZON_S3', 'region': 'us-east-2',
'remoteBucketName': 'hcpcs-bucket-5', 'accessKey': 'A1234567890=',
'secretKey': 'S1234567890==', 'port': 443,
'authVersion': 'v4', 'usePathStyleAlways': 'true'}}
```

Saved sample JSON file for bucket replication sync-to configuration in
'testto.json'

You can use 'testto.json' sample JSON file as an input to put-bucket-
replication S3 API. For example, using aws s3api command:
aws s3api put-bucket-replication --no-verify-ssl --endpoint-url https://
cloudscale-hostname --bucket cloudscale-bucket --replication-configuration
file://testto.json

Script to generate bucket sync-from JSON

A script is included to generate the JSON needed to configure bucket synchronization from an external bucket (sync-from).

The script is written in Python and located in the folder *install_path/product/bin* (for example, */opt/hcpcs/bin*).

The script generates the JSON string that you can use for the field `destination.bucket` in the AWS S3 command `put-bucket-replication`. Optionally, the script verifies whether the destination bucket or the target AWS SQS queue exist. If you omit the secret key, the script prompts you for it, which lets you create a script that calls this script without storing the secret key. If you omit the access key for a queue, the script uses the access key and secret key for the bucket. You can mix the short and full form of arguments.



Note: The script produces JSON using single quotes.

Syntax

```
SyncFromBucketJsonGenerator.py
  [--help]
  --s3host host
  --region region
  --bucket bucket
  --accessKey access_key
  [--secretKey secret_key]
  [--s3type { GENERIC_S3 | AMAZON_S3 }]
  [--port port]
  [--authVersion { v2 | v4 }]
  [--usePathStyleAlways {true | false}]
  [--jsonSample output_file.json]
  [--verifyTarget]
  [--http]
  --notificationsQueue queue
  [--notificationsRegion region]
  [--notificationsAccessKey access_key]
  [--notificationsSecretKey secret_key]
  [--quietMode]
```

Options and parameters

-h, --help

Optional. Displays a help message and exits.

--s3host *host*, -s3 *host*

Host name of the remote S3 storage component.

--region *region*, -r *region*

Region of the remote bucket.

--bucket *bucket*, -b *bucket*

Name of the remote bucket.

--accessKey *access_key*, -ak *access_key*

Access key for the remote bucket.

--secretKey *secret_key*, -sk *secret_key*

Secret key for the remote bucket. The script prompts for the key if you don't specify it.

--s3type { GENERIC_S3 | AMAZON_S3 }, -s3t { GENERIC_S3 | AMAZON_S3 }
Optional. The remote bucket type:

- **GENERIC_S3**: An S3-compatible node
- **AMAZON_S3**: An Amazon Web Services S3-compatible node

If not specified, the default bucket type is AMAZON_S3.

--port *port*, -p *port*

Optional. Port of the remote bucket. If not specified, the default port is 443.

--authVersion { *v2* | *v4* }, -av { *v2* | *v4* }

Optional. The Auth Version of the remote bucket. If not specified, the default version is v4.

--usePathStyleAlways {*true* | *false*}, -upsa {*true* | *false*}

Optional. Sets the Use Path Style Always flag for the remote bucket. If not specified, the default is true.

--jsonSample *output_file.json*, -json *output_file.json*

Optional. Creates a file named *output_file.json* with a sample JSON structure for bucket replication configuration. If not specified, no sample file is created.

--verifyTarget, -verify

Optional. Verifies that the remote bucket exists. SSL certificates are not validated. This option requires python3 and boto3. If not specified, the bucket's existence isn't verified.

--http, -http

Optional. Use HTTP when verifying the remote bucket. If not specified, the default is to use HTTPS.

--notificationsQueue *queue*, -nq *queue*

Name of the notifications queue.

--notificationsRegion *region*, -nq *region*

Optional. Name of the notifications region. If not specified, the default is the region of the remote bucket.

--notificationsAccessKey *access_key*, -nak *access_key*

Optional. The notifications access key. If not specified, the default is the access key of the remote bucket.

--notificationsSecretKey *secret_key*, -nsk *secret_key*

Optional. The notifications secret key. If not specified, the default is the secret key of the remote bucket.

--quietMode, -qm

Optional. Displays only the JSON for QueueArn.



Note: You can't specify both `--quietMode` and `--verifyTarget` together.

Example

```
$ SyncFromBucketJsonGenerator.py -s3 s3.us-east-2.amazonaws.com -b hcpcs-
bucket-5 -r us-east-2 -ak A1234567890 -sk S1234567890 -nq 'bucketevents2' -
verify -json testfrom.json
```

This example can produce the following output:

```
Verifying that a remote bucket "hcpcs-bucket-5" exists...
Verification successfully completed: remote bucket "hcpcs-bucket-5" is
found

Verifying that a remote notification queue with a prefix "bucketevents2"
exists...
Verification successfully completed: found "bucketevents2" queue.

Generated a JSON string for the Destination->Bucket element for bucket
replication sync-from configuration:

{'action': 'sync-from', 'version': '1.0', 'externalBucket': {'host':
's3.us-east-2.amazonaws.com', 'type': 'AMAZON_S3', 'region': 'us-east-2',
'remoteBucketName': 'hcpcs-bucket-5', 'accessKey': 'A1234567890=',
'secretKey': 'S1234567890==', 'port': 443,
'authVersion': 'v4', 'usePathStyleAlways': 'true'}, 'notifications':
{'type': 'AWS_SQS', 'queue': 'bucketevents2', 'region': 'us-east-2',
'accessKey': 'A1234567890=', 'secretKey': 'S1234567890=='}}

Saved sample JSON file for bucket replication sync-from configuration in
'testfrom.json'

You can use 'testfrom.json' sample JSON file as an input to put-bucket-
replication S3 API. For example, using aws s3api command:
aws s3api put-bucket-replication --no-verify-ssl --endpoint-url https://
cloudscale-hostname --bucket cloudscale-bucket --replication-configuration
file://testfrom.json
```

Get bucket synchronization rules (GET bucket replication)

You can retrieve the synchronization rules for a bucket.

HTTP request syntax (URI)

```
aws --endpoint -url https://host_ip s3api get-bucket-replication --bucket
"bucket"
```

Request structure

Not applicable.

Response structure

The response body is shown below:

```
{
  "ReplicationConfiguration": {
```



```

"Role": "",
"Rules": [
  {
    "Filter": {
      "And": {
        "Prefix": "string",
        "Tags": [
          {
            "Key": "string",
            "Value": "string"
          }
        ]
      }
    },
    "Status": "boolean",
    "Destination": {
      "Bucket": "access_settings",
    },
    "ID": "string",
  }
],
}

```

Parameter	Required	Type	Description
Role	Yes	N/A	Not supported; empty.
Prefix	No	String	Prefix.
Key	No	String	Tag key.
Value	No	String	Tag value. Sets of prefixes and key-value pairs.
Status	Yes	Boolean	If <code>false</code> , rule is ignored.
Bucket	Yes	Base64-encoded JSON	Bucket access settings. S3 access and secret keys are masked.
ID	No	String	Unique identifier for rule, up to 255 characters.

HTTP status codes

Status code	HTTP name	Description
200	OK	The request was executed successfully.
401	Unauthorized	Access was denied due to invalid credentials.

Example

Request example:

```
aws --endpoint-url https://10.08.1019 s3api get-bucket-replication --
bucket "hcpcs_bucket"
```

JSON response:

```
{
  "ReplicationConfiguration": {
    "Role": "",
    "Rules": [
      {
        "Filter": {
          "And": {
            "Prefix": "SQS",
            "Tags": [
              {
                "Value": "cloud",
                "Key": "target"
              }
            ]
          }
        },
        "Status": "Enabled",
        "Destination": {
          "Bucket": {
            'version': 'version',
            'action': 'sync-from',
            'externalBucket': {
              'host': 'host',
              'type': 'type',
              'region': 'region',
              'remoteBucketName': 'bucket_name',
              'port': 'port',
              'authVersion': 'auth_version',
              'usePathStyleAlways': '[true|false]'
            }
          }
        }
      }
    ]
  }
}
```

```

    },
    "ID": "mirrorBack_rule_for_images"
  }
]
}
}

```

Get object synchronization status

The synchronization status of an object is returned in metadata as part of the response to a GET object or HEAD object request.

For a GET object or HEAD object request, the synchronization functions return a replication status header in addition to the standard response metadata. This information is useful before deletion from a source bucket to verify synchronization.

When an object is created, HCP for cloud scale evaluates the sync-to rules for the bucket. If the object matches the rules, it sets the object's sync state as `PENDING`. Most of the time, this sync state is accurate. However, it is never definitive because users may change the sync-to rules for the bucket before the policy engine starts processing the object, which happens asynchronously. The policy engine evaluates the sync-to rules again when processing an object to act according to the latest sync rules.

For example:

- An object was ingested that matches the sync-to rules, so its sync state is set as `PENDING`. Then, a user changes the sync-to rules. The object does not match the rules anymore so the object is actually not synced and that sync state is removed.
- An object was ingested that does not match the sync-to rules, so its sync state is not set. Then, a user changes the sync rules. The object now matches the rules so the object is actually synced and the sync state is set to `COMPLETED`.

Response header	Description
x-amz-replication-status	<p>Status of synchronization:</p> <ul style="list-style-type: none"> ▪ COMPLETED: For sync-to, all rules were successfully executed and the object was successfully synchronized. <p>Note: This status is also returned for objects that match a sync-to rule but were skipped because they are not the most recent version.</p> <ul style="list-style-type: none"> ▪ PENDING: For sync-to, one of the following: (1) a check is pending to see if the object needs synchronization; (2) the object needs synchronization, but the process is not complete.

Response header	Description
	<ul style="list-style-type: none"> ▪ FAILED: For sync-to, the process has failed multiple times. To be synchronized, the object must be reloaded. ▪ REPLICA: For sync-from, the object is a replica created by Amazon S3.
(Header not in response)	The object did not match any rules.

Delete bucket synchronization rules (DELETE bucket replication)

You can delete S3 synchronization settings for buckets. This function is the same as in AWS S3.

HTTP request syntax (URI)

```
aws --endpoint -url https://host_ip s3api delete-bucket-replication --
bucket "bucket"
```

Request structure

None.

Response structure

None.

Example

Request example:

```
aws --endpoint-url https://10.08.1019 s3api delete-bucket-replication --
bucket "hcpcs_bucket"
```



Note: If a sync-from action fails it is retried and the SQS message about the failure is retained. To avoid a possible accumulation of SQS failure messages, the best practice is to define a suitable retention policy for SQS and to delete the sync-from rule once the desired results are obtained.

Chapter 8: S3 event notification

Hitachi Content Platform for cloud scale (HCP for cloud scale) lets you configure and manage S3 event notification.

A script is available to simplify the process of configuring S3 event notification.

About S3 event notification

HCP for cloud scale supports the AWS S3 methods `PUT Bucket Notification Configuration` and `GET Bucket Notification Configuration`. To enable notifications, an S3 user adds a notification configuration that identifies the events to be published and the destinations (notification target systems) where notifications are sent.

HCP for cloud scale supports overlapping notification rules. Unlike AWS, the same event can be sent to multiple queues.

Access to the event notification functions is controlled by role-based permissions to write or read (set and get) bucket configurations.

Supported events

The list of supported events is:

- `s3:ObjectCreated:*`
- `s3:ObjectCreated:Put`
- `s3:ObjectCreated:Post`
- `s3:ObjectCreated:Copy`
- `s3:ObjectCreated:CompleteMultipartUpload`
- `s3:ObjectRemoved:*`
- `s3:ObjectRemoved:Delete`
- `s3:ObjectRemoved:DeleteMarkerCreated`

AWS S3 methods such as PUT, POST, and COPY can create an object. Using the ObjectCreated event types, you can enable notification when an object is created using a specific method, or you can use the `s3:ObjectCreated:*` event type to request notification regardless of the method used to create an object. You do not receive an event notifications from failed operations.

Using the `ObjectRemoved` event types, you can enable notification when an object or a batch of objects is removed from a bucket. You can request notification when an object is deleted or a versioned object is permanently deleted by using the `s3:ObjectRemoved:Delete` event type. This event is also sent when a delete marker is created. You can request notification when a delete marker is created for a versioned object by using the `s3:ObjectRemoved:DeleteMarkerCreated` event. You can also use a wildcard (`s3:ObjectRemoved:*`) to request notification any time an object is deleted. You do not receive an event notification from automatic deletions from lifecycle policies or from failed operations.

Configuration

HCP for cloud scale fully supports notification configuration.

Configuration can include up to 100 rules. Each rule consists of:

- One or more event types (such as PUT, COPY, or DELETE)
- (Optional) A filter with zero or one prefix and zero or one suffix (tags are not supported)
- A notification target (an SQS queue)

Rules can overlap. That is, an HCP for cloud scale event notification can be sent to multiple targets. However, multiple rules can't send notification of the same event to the same target. A configuration containing rules that overlap in this way is blocked.

Script to generate S3 event notification configuration JSON

HCP for cloud scale includes a script to generate the JSON needed to configure S3 event notification.

The script is written in Python and located in the folder `install_path/product/bin` (for example, `/opt/hcps/bin`).

The script generates the JSON string that you can use for the element `QueueArn` in the AWS S3 command `put-bucket-notification-configuration` to configure the queue's Amazon Resource Name (ARN). Optionally, the script verifies whether the target AWS SQS queue exists, and if more than one matching SQS queue is found returns them all. If you omit the secret key, the script prompts you for it, which lets you create a script that calls this script without storing the secret key. You can mix the short and full form of arguments.



Note: The script produces JSON using single quotes.

Syntax

```
EventNotificationsJsonGenerator.py
  [--help]
  --queue queue
  --region region
  --accessKey access_key
```

```
[--secretKey secret_key]
[--jsonSample output_file.json]
[--verifyTarget]
[--insecure]
[--eventType event_type[,...]]
[--prefix prefix]
[--suffix suffix]
[--id queue_id]
[--quietMode]
```

Options and parameters

-h, --help

Optional. Displays a help message and exits.

--queue *queue*, -q *queue*

Name of the event notification queue.

--region *region*, -r *region*

Region of the event notification queue.

--accessKey *access_key*, -ak *access_key*

Access key for the event notification queue.

--secretKey *secret_key*, -sk *secret_key*

Secret key for the event notification queue. The script prompts for the key if you don't specify it.

--jsonSample *output_file.json*, -json *output_file.json*

Optional. Creates a file named *output_file.json* with a sample JSON structure for event notification configuration. If not specified, no sample file is created.

--verifyTarget, -verify

Optional. Verifies that the remote queue exists. SSL certificates aren't validated. This option requires python3 and boto3. If not specified, the queue's existence isn't verified.



Note: You can't specify both `--quietMode` and `--verifyTarget` together.

--insecure, -i

Optional. Suppresses Python warning messages.

--eventType *event_type[,...]*, -et *event_type[,...]*

Optional. Event notification types. One or more comma-separated types.

- s3:ObjectCreated:*
- s3:ObjectCreated:Put
- s3:ObjectCreated:Post
- s3:ObjectCreated:Copy

- s3:ObjectCreated:CompleteMultipartUpload
- s3:ObjectRemoved:*
- s3:ObjectRemoved:Delete
- s3:ObjectRemoved:DeleteMarkerCreated

The default is s3:ObjectCreated:*, s3:ObjectRemoved:.*.

--prefix, -px

Optional. Filter prefix. If not specified, no prefix is used.

--suffix, -sx

Optional. Filter suffix. If not specified, no suffix is used.

--id *queue_id*, -id *queue_id*

Optional. The queue configuration ID. The default is SampleEvenId.

--quietMode, -qm

Optional. Displays only JSON for the element QueueArn.



Note: You can't specify both --quietMode and --verifyTarget together.

Example

```
$ EventNotificationJsonGenerator.py -q queue1 -r us-east-2 -ak A1234567890
-sk S1234567890 -verify -json testqueue.json
```

This example can produce the following output:

```
Verifying that a remote notification queue with a prefix "queue1" exists...
Verification successfully completed: found "queue1" queue.
```

```
Generated a JSON string for QueueArn element for S3 Event Notifications
configuration:
```

```
"{'type': 'AWS_SQS', 'queue': 'queue1', 'region': 'us-east-2',
  'accessKey': 'QUtJQVNPS1cyRUkzQVlKSVZMTkY=', 'secretKey':
  'bUtOQnUydUZaaFZqQTQ0eGs3M1NaRzZoMUdnVkt2MHpLOEFhOFdmUQ=='}"
```

```
Saved sample JSON file for event notification configuration in
'testqueue.json'
```

```
You can use 'testqueue.json' sample JSON file as an input to put-bucket-
notification-configuration S3 API. For example, using aws s3api command:
aws s3api put-bucket-notification-configuration --no-verify-ssl --bucket
cloudscale-bucket --notification-configuration file://testqueue.json
```


Hitachi Vantara



Corporate Headquarters

2535 Augustine Drive

Santa Clara, CA 95054 USA

HitachiVantara.com | community.HitachiVantara.com

Contact Information

USA: 1-800-446-0744

Global: 1-858-547-4526

HitachiVantara.com/contact