Elastic Cloud Storage (ECS)

Version 3.1

Administration Guide

302-003-863
02

DELL EMC

Published September 2017

Dell EMC
Hopkinton, Massachusetts 01748-9103
1-508-435-1000 In North America 1-866-464-7381
www.DellEMC.com

CONTENTS

FIGURES

TABLES

CHAPTER 1

Overview

Introduction

Dell EMC Elastic Cloud Storage (ECS) provides a complete software-defined cloud storage platform that supports the storage, manipulation, and analysis of unstructured data on a massive scale on commodity hardware. You can deploy ECS as a turnkey storage appliance or as a software product that is installed on a set of qualified commodity servers and disks. ECS offers the cost advantages of a commodity infrastructure and the enterprise reliability, availability, and serviceability of traditional arrays.

ECS uses a scalable architecture that includes multiple nodes and attached storage devices. The nodes and storage devices are commodity components, similar to devices that are generally available, and are housed in one or more racks.

A rack and its components that are supplied by Dell EMC and that have preinstalled software, is referred to as an ECS *appliance*. A rack and commodity nodes that are not supplied by Dell EMC, is referred to as a Dell EMC ECS *software only solution*. Multiple racks are referred to as a *cluster*.

A rack, or multiple joined racks, with processing and storage that is handled as a coherent unit by the ECS infrastructure software is referred to as a *site*, and at the ECS software level as a *Virtual Data Center* (VDC).

Management users can access the ECS UI, which is referred to as the ECS Portal, to perform administration tasks. Management users include the System Administrator, Namespace Administrator, and System Monitor roles. Management tasks that can be performed in the ECS Portal can also be performed by using the ECS Management REST API.

ECS administrators can perform the following tasks in the ECS Portal:

- Configure and manage the object store infrastructure (compute and storage resources) for object users.
- Manage users, roles, and buckets within namespaces. Namespaces are equivalent to tenants.

Object users cannot access the ECS Portal, but can access the object store to read and write objects and buckets by using clients that support the following data access protocols:

- Amazon Simple Storage Service (Amazon S3)
- EMC Atmos
- OpenStack Swift
- ECS CAS (content-addressable storage)

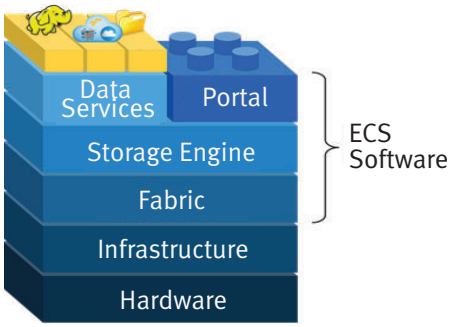
For more information about object user tasks, see the *ECS Data Access Guide*, available from the ECS Product Documentation page.

For more information about System Monitor tasks, see the *ECS Monitoring Guide*, available from the ECS Product Documentation page.

ECS platform

The ECS platform is composed of the data services, portal, storage engine, fabric, infrastructure, and hardware component layers.

Figure 1 ECS component layers



Data services

The data services component layer provides support for access to the ECS object store through object, HDFS, and NFS v3 protocols. In general, ECS provides multi-protocol access; data that is ingested through one protocol can be accessed through another. For example, data that is ingested through S3 can be modified through Swift, NFS v3, or HDFS. This multi-protocol access has some exceptions due to protocol semantics and representations of how the protocol was designed.

The following table shows the object APIs and the protocols that are supported and that interoperate.

Table 1 ECS supported data services

Protocols		Support	Interoperability
Object	S3	Additional capabilities such as byte range updates and rich ACLS	File systems (HDFS and NFS), Swift
	Atmos	Version 2.0	NFS (only path-based objects, not object ID style objects)
	Swift	V2 APIs, Swift and Keystone v3 authentication	File systems (HDFS and NFS), S3
	CAS	SDK v3.1.544 and later	Not applicable
HDFS		Hadoop 2.7 compatibility	S3, Swift*
NFS		NFSv3	S3, Swift, Atmos (only path-based objects)
* When a bucket is enabled for file system access, permissions set using HDFS are in effect when you access the bucket as an NFS file system, and vice-versa..			

Portal

The ECS Portal component layer provides a Web-based GUI that allows you to manage, license, and provision ECS nodes. The portal has the following comprehensive reporting capabilities:

- Capacity utilization for each site, storage pool, node and disk
- Performance monitoring on latency, throughput, transactions per second, and replication progress and rate
- Diagnostic information, such as node and disk recovery status and statistics on hardware and process health for each node, which helps identify performance and system bottlenecks

Storage engine

The storage engine component layer provides an unstructured storage engine that is responsible for storing and retrieving data, managing transactions, and protecting and replicating data. The storage engine provides access to objects ingested using multiple object storage protocols and the NFS and HDFS file protocols.

Fabric

The fabric component layer provides cluster health management, software management, configuration management, upgrade capabilities, and alerting. The fabric layer is responsible for keeping the services running and managing resources such as the disks, containers, firewall, and network. It tracks and reacts to environment changes such as failure detection and provides alerts related to system health.

Infrastructure

The infrastructure component layer uses SUSE Linux Enterprise Server 12 as the base operating system for the ECS appliance, or qualified Linux operating systems for commodity hardware configurations. Docker is installed on the infrastructure to deploy the other ECS component layers. The Java Virtual Machine (JVM) is installed as part of the infrastructure because ECS software is written in Java.

Hardware

The hardware component layer is an ECS appliance or qualified industry standard hardware. For more information about ECS hardware, see the *ECS Hardware and Cabling Guide*, available from the ECS Product Documentation page.

ECS data protection

ECS protects data within a site by mirroring the data onto multiple nodes, and by using erasure coding to break down data chunks into multiple fragments and distribute the fragments across nodes. Erasure coding (EC) reduces the storage overhead and ensures data durability and resilience against disk and node failures.

By default, the storage engine implements the Reed-Solomon 12 + 4 erasure coding scheme in which an object is broken into 12 data fragments and 4 coding fragments. The resulting 16 fragments are dispersed across the nodes in the local site. When an object is erasure-coded, ECS can read the object directly from the 12 data fragments without any decoding or reconstruction. The code fragments are used only for object reconstruction when a hardware failure occurs. ECS also supports a 10 + 2 scheme for use with cold storage archives to store objects that do not change frequently and do not require the more robust default EC scheme.

The following table shows the requirements for the supported erasure coding schemes.

Table 2 Erasure encoding requirements for regular and cold archives

Use case	Minimum required nodes	Minimum required disks	Recommended disks	EC efficiency	EC scheme
Regular archive	4	16	32	1.33	12 + 4
Cold archive	6	12	24	1.2	10 + 2

Sites can be federated, so that data is replicated to another site to increase availability and data durability, and to ensure that ECS is resilient against site failure. For three or more sites, in addition to the erasure coding of chunks at a site, chunks that are replicated to other sites are combined using a technique called XOR to provide increased storage efficiency.

The following table shows the storage efficiency that can be achieved by ECS where multiple sites are used.

Table 3 Storage overhead

Number of sites in replication group	Storage overhead	
	Default (Erasure Code: 12+4)	Cold archive (Erasure Code: 10+2)
1	1.33	1.2
2	2.67	2.4
3	2.00	1.8
4	1.77	1.6
5	1.67	1.5
6	1.60	1.44
7	1.55	1.40
8	1.52	1.37

If you have one site, with erasure coding the object data chunks use more space (1.33 or 1.2 times storage overhead) than the raw data bytes require. If you have two sites, the storage overhead is doubled (2.67 or 2.4 times storage overhead) because both sites store a replica of the data, and the data is erasure coded at both sites. If you have three or more sites, ECS combines the replicated chunks so that, counterintuitively, the storage overhead reduces.

For a detailed description of the mechanism used by ECS to provide data durability, resilience, and availability, see the *Elastic Cloud Storage High Availability Design White Paper*.

Configurations for availability, durability, and resilience

Depending on the number of sites in the ECS system, different data protection schemes can increase availability and balance the data protection requirements against performance. ECS uses the replication group to configure the data protection schemes (see Introduction to storage pools, VDCs, and replication groups on page 28). The following table shows the data protection schemes that are available.

Table 4 ECS data protection schemes

Number of sites	Data protection scheme			
	Local Protection	Full Copy Protection*	Geo-Active	Geo-Passive
1	Yes	Not applicable	Not applicable	Not applicable
2	Yes	Always	Not applicable	Not applicable

Table 4 ECS data protection schemes (continued)

Number of sites	Data protection scheme			
	Local Protection	**Full Copy Protection***	**Geo-Active**	**Geo-Passive**
3 or more	Yes	Optional	Normal	Optional
* Full Copy Protection can be selected with Geo Active. Full Copy Protection is not available if Geo-Passive is selected.				

Local Protection

Data is protected locally by using triple mirroring and erasure coding which provides resilience against disk and node failures, but not against site failure.

Full Copy Protection

When a replication group is enabled with the **Replicate to All Sites** property, the replication group makes a full readable copy of all objects to all sites within the replication group. Having full readable copies of objects on all VDCs in the replication group provides data durability and improves local performance at all sites at the cost of storage efficiency.

Geo-Active

Geo-Active is the default ECS configuration. When a replication group is configured as Geo-Active, data is replicated to federated sites and can be accessed from all sites with strong consistency. If you have two sites, full copies of data chunks are copied to the other site. If you have three or more sites, the replicated chunks are combined (XOR'ed) to provide increased storage efficiency.

When data is accessed from a site that is not the owner of the data, until that data is cached at the non-owner site, the access time increases. Similarly, if the owner site that contains the primary copy of the data fails, and if you have a global load balancer that directs requests to a non-owner site, the non-owner site must recreate the data from XOR'ed chunks, and the access time increases.

Geo-Passive

The Geo-Passive configuration always includes exactly three sites and is available where you have at least three sites. In this configuration, two sites are active. The third site is passive and is a replication target. You can designate a specific site as the backup site and achieve the storage efficiency (2.0 times storage overhead) of three sites. The storage efficiency is the same as the Geo-Active three-site configuration. In the Geo-Passive configuration, all replication data chunks are sent to the passive site and XOR operations occur only at the passive site. In a Geo-Active configuration, the XOR operations occur at all sites.

If all sites are on-premise, any of the three sites can be the replication target. An important use case for the Geo-Passive configuration is when the passive site is hosted by a third party and the hosted site is selected as the replication target. This configuration can be modified by using the ECS Management REST API to select an on-premise site as the replication target.

ECS network

ECS network infrastructure consists of top of rack switches allowing for the following types of network connections:

- Public network – connects ECS nodes to your organization's network, providing data.
- Internal private network – manages nodes and switches within the rack and across racks.

For more information about ECS networking, see the *ECS Networking and Best Practices White Paper*.

Load balancing considerations

It is recommended that a load balancer is used in front of ECS.

In addition to distributing the load across ECS cluster nodes, a load balancer provides High Availability (HA) for the ECS cluster by routing traffic to healthy nodes. Where network separation is implemented, and data and management traffic are separated, the load balancer must be configured so that user requests, using the supported data access protocols, are balanced across the IP addresses of the data network. ECS Management REST API requests can be made directly to a node IP on the management network or can be load balanced across the management network for HA.

The load balancer configuration is dependent on the load balancer type. For information about tested configurations and best practice, contact your customer support representative.

CHAPTER 2

Getting Started with ECS

Initial configuration

The initial configuration steps that are required to get started with ECS include logging in to the ECS Portal for the first time, using the ECS Portal Getting Started Task Checklist and Dashboard, uploading a license, and setting up an ECS virtual data center (VDC).

To initially configure ECS, the root user or System Administrator must at a minimum:

Procedure

1. Upload an ECS license.

   See Licensing on page 161.

2. Select a set of nodes to create at least one storage pool.

   See Create a storage pool on page 30.

3. Create a VDC.

   See Create a VDC for a single site on page 33.

4. Create at least one replication group.

   See Create a replication group on page 38.

   a. Optional: Set authentication.

      You can add Active Directory (AD), LDAP, or Keystone authentication providers to ECS to enable users to be authenticated by systems external to ECS. See Introduction to authentication providers on page 44.

5. Create at least one namespace. A namespace is the equivalent of a tenant.

   See Create a namespace on page 57.

   a. Optional: Create object and/or management users.

      See Working with users in the ECS Portal on page 73.

6. Create at least one bucket.

   See Create a bucket on page 88.

   After you configure the initial VDC, if you want to create an additional VDC and federate it with the first VDC, see Add a VDC to a federation on page 34.

Log in to the ECS Portal

You must log in to the ECS Portal to set up the initial configuration of a VDC. Log in to the ECS Portal from the browser by specifying the IP address or fully qualified domain name (FQDN) of any node, or the load balancer that acts as the front end to ECS. The login procedure is described below.

Before you begin

Logging in to the ECS Portal requires the System Administrator, System Monitor, Lock Administrator (`emcsecurity` user), or Namespace Administrator role.

Note

When you log in to the ECS Portal for the first time, you log in as the default `root` user with the System Administrator role.

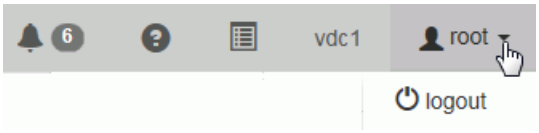
Procedure

1. Type the public IP address of the first node in the system, or the address of the load balancer that is configured as the front-end, in your browser's address bar: `https://<node1_public_ip>`.
2. Log in with the default root credentials:
   - **User Name:** `root`
   - **Password:** `ChangeMe`

   You are prompted to change the password for the `root` user immediately.
3. After you change the password at first login, click **Save**.

   You are logged out and the ECS login screen appears.
4. Type the **User Name** and **Password**.
5. To log out of the ECS Portal, in the upper-right menu bar, click the arrow beside your user name, and then click **logout**.

Figure 2 Logging out of the portal



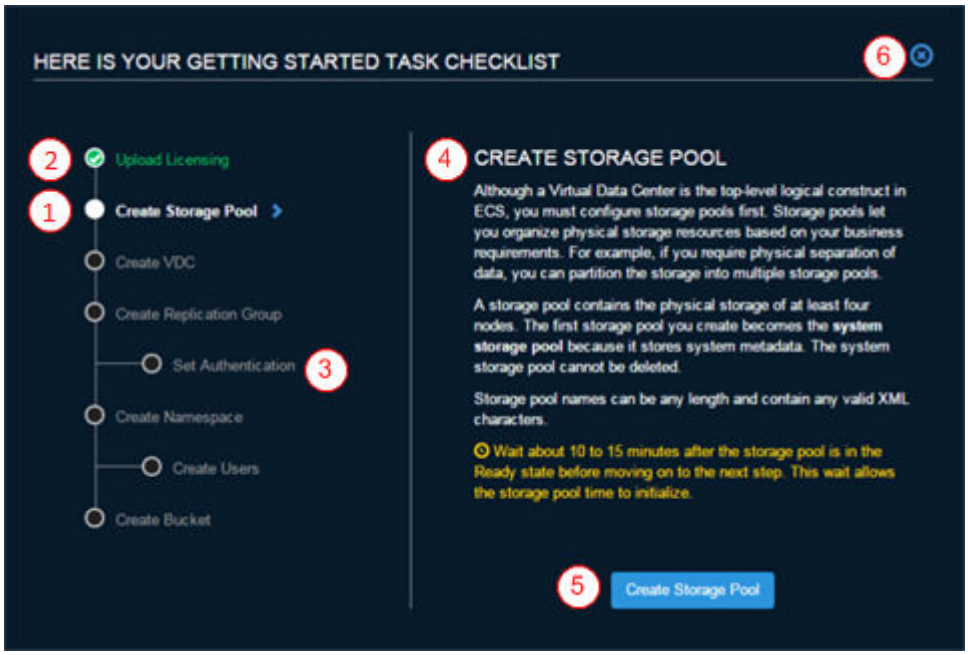
View the Getting Started Task Checklist

The Getting Started Task Checklist in the ECS Portal guides you through the initial ECS configuration. The checklist appears when you first log in and when the portal detects that the initial configuration is not complete. The checklist automatically appears until you dismiss it. On any ECS Portal page, in the upper-right menu bar, click the Guide icon to open the checklist.

Figure 3 Guide icon

The Getting Started Task Checklist displays in the portal.

Figure 4 Getting Started Task Checklist



1. The current step in the checklist.
2. A completed step.
3. An optional step. This step does not display a check mark even if you have completed the step.
4. Information about the current step.
5. Available actions.
6. Dismiss the checklist.

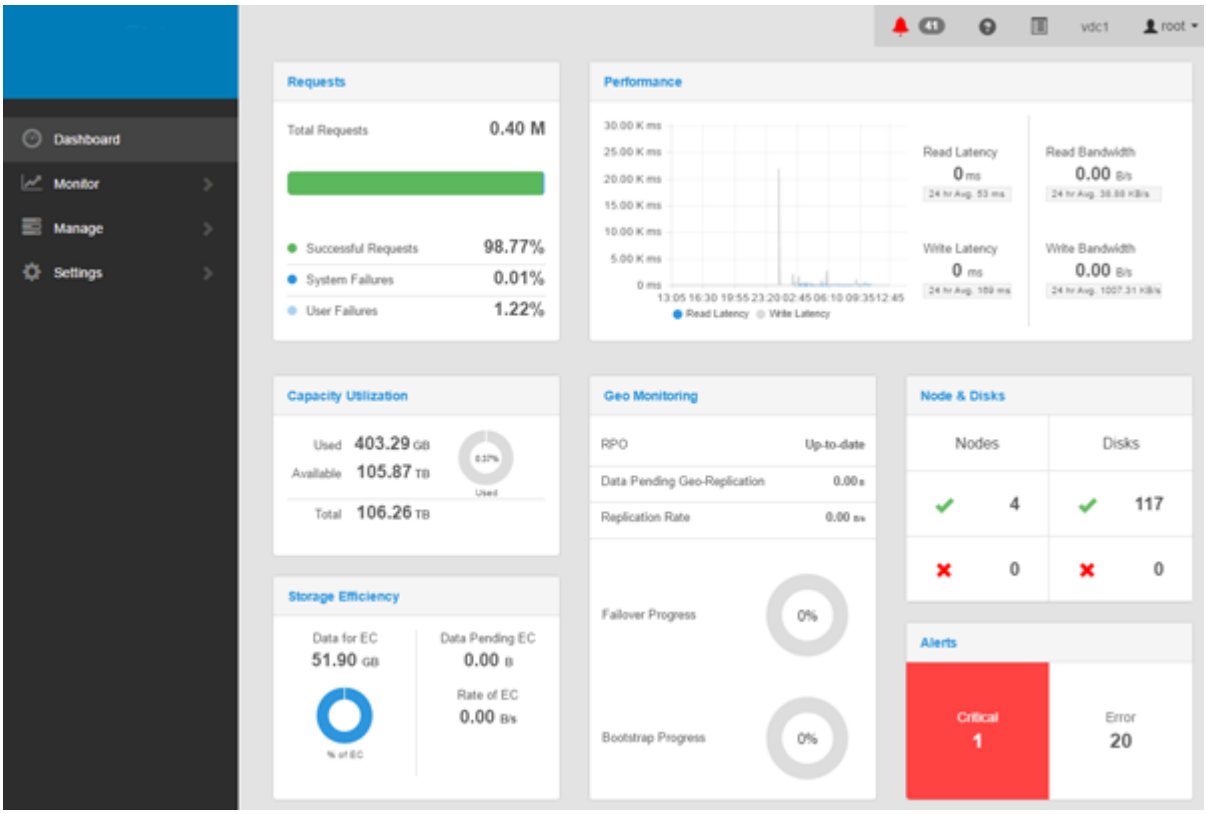
A completed checklist gives you the option to browse the list again or recheck your configuration.

View the ECS Portal Dashboard

The ECS Portal Dashboard provides critical information about the ECS processes on the VDC you are currently logged in to.

The Dashboard is the first page you see after you log in. The title of each panel (box) links to the portal monitoring page that shows more detail for the monitoring area.

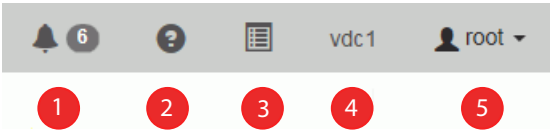
Figure 5 ECS Dashboard



Upper-right menu bar

The upper-right menu bar appears on each ECS Portal page.

Figure 6 Upper-right menu bar



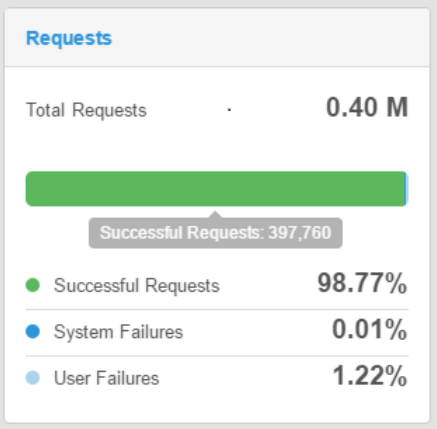
Menu items include the following icons and menus:

1. The Alert icon displays a number that indicates how many unacknowledged alerts are pending for the current VDC. The number displays `99+` if there are more than 99 alerts. You can click the Alert icon to see the Alert menu, which shows the five most recent alerts for the current VDC.
2. The Help icon brings up the online documentation for the current portal page.
3. The Guide icon brings up the Getting Started Task Checklist.
4. The VDC menu displays the name of the current VDC. If your AD or LDAP credentials allow you to access more than one VDC, you can switch the portal view to the other VDCs without entering your credentials.
5. The User menu displays the current user and allows you to log out.

View requests

The **Requests** panel displays the total requests, successful requests, and failed requests.

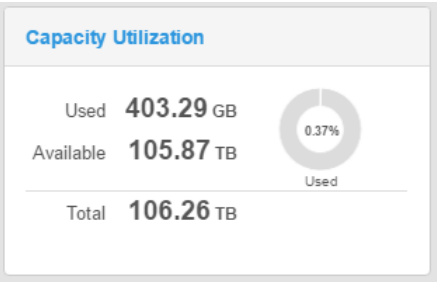
Failed requests are organized by system error and user error. User failures are typically HTTP 400 errors. System failures are typically HTTP 500 errors. Click **Requests** to see more request metrics.



View capacity utilization

The **Capacity Utilization** panel displays the total, used, and available capacity.

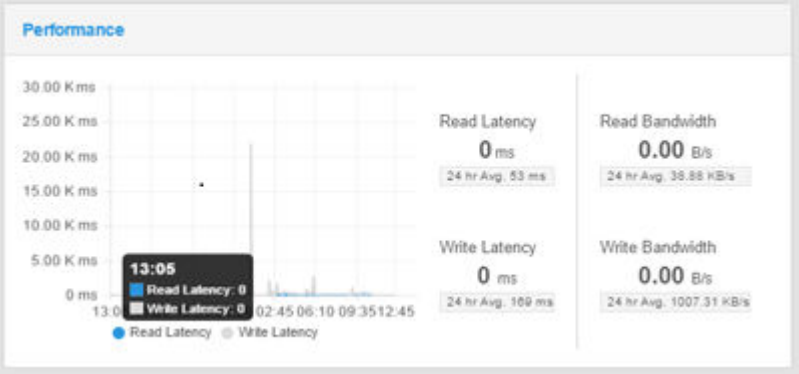
The percentage displayed in the center of the graph indicates the amount of usable capacity that is currently in use. Capacity takes into account ingested data, replicas, and system data. Click **Capacity Utilization** to see more capacity metrics.



View performance

The **Performance** panel displays how network read and write operations are currently performing, and the average read/write performance statistics over the last 24 hours.

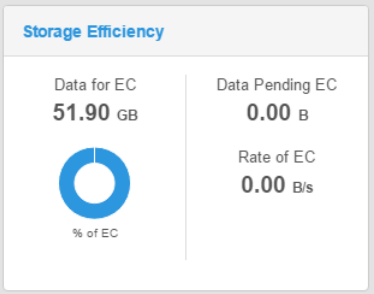
Click **Performance** to see more comprehensive performance metrics.



View storage efficiency

The **Storage Efficiency** panel displays the efficiency of the erasure coding (EC) process.

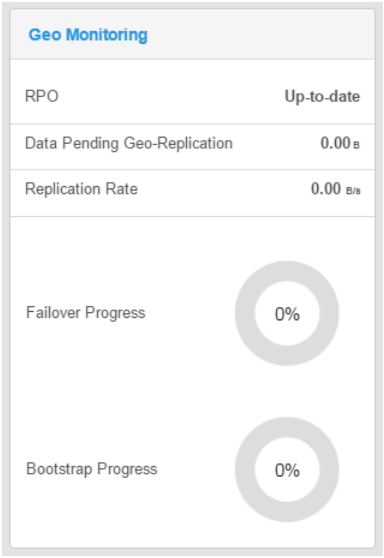
The chart shows the progress of the current EC process, and the other values show the amount of EC data waiting for the EC process, and the current rate of the EC process. Click **Storage Efficiency** to see more storage efficiency metrics.



View geo monitoring

The **Geo Monitoring** panel displays how much data from the local VDC is waiting for geo-replication, and the rate of the replication.

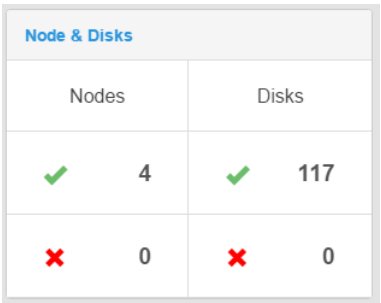
Recovery Point Objective (**RPO**) refers to the point in time in the past to which you can recover. The value is the oldest data at risk of being lost if a local VDC fails before replication is complete. **Failover Progress** shows the progress of any active failover that is occurring in the federation involving the local VDC. **Bootstrap Progress** shows the progress of any active process to add a new VDC to the federation. Click **Geo Monitoring** to see more geo-replication metrics.



View node and disk health

The **Node & Disks** panel displays the health status of disks and nodes.

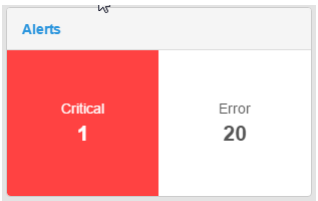
A green check mark beside the node or disk number indicates the number of nodes or disks in good health. A red x indicates bad health. Click **Node & Disks** to see more hardware health metrics. If the number of bad disks or nodes is a number other than zero, clicking on the count takes you to the corresponding **Hardware Health** tab (**Offline Disks** or **Offline Nodes**) on the **System Health** page.



View alerts

The **Alerts** panel displays a count of critical alerts and errors.

Click **Alerts** to see the full list of current alerts. Any Critical or Error alerts are linked to the **Alerts** tab on the **Events** page where only the alerts with a severity of Critical or Error are filtered and displayed.



CHAPTER 3

Storage Pools, VDCs, and Replication Groups

Introduction to storage pools, VDCs, and replication groups

This topic provides conceptual information on storage pools, virtual data centers (VDCs), and replication groups and the following topics describe the operations required to configure them:

- Working with storage pools at the ECS Portal
- Working with VDCs at the ECS Portal
- Working with replication groups at the ECS Portal

The storage that is associated with a VDC must be assigned to a storage pool and the storage pool must be assigned to one or more replication groups to allow the creation of buckets and objects.

A storage pool can be associated with more than one replication group. A best practice is to have a single storage pool for a site. However, you can have as many storage pools as required, with a minimum of four nodes (and 16 disks) in each pool.

You might need to create more than one storage pool at a site for the following reasons:

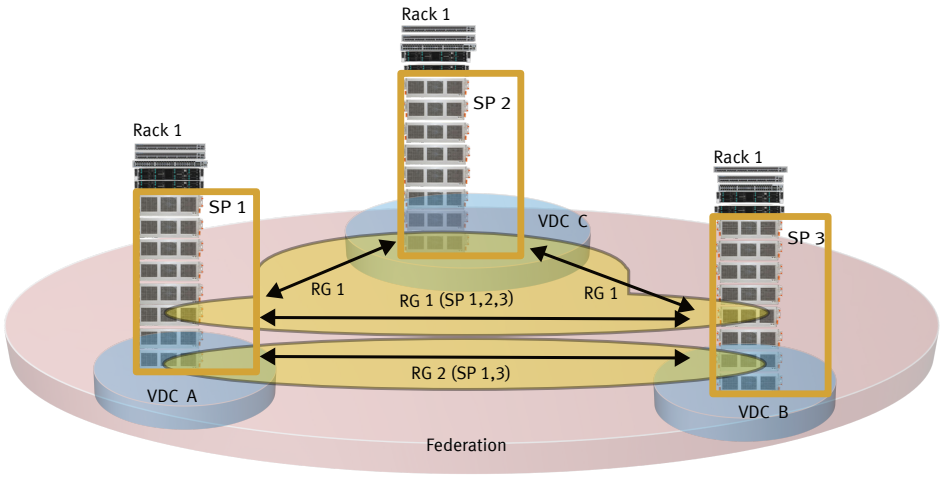
- The storage pool is used for Cold Archive. The erasure coding scheme used for cold archive uses 10+2 coding rather than the default ECS 12+4 scheme.
- A tenant requires the data to be stored on separate physical media.

A storage pool must have a minimum of four nodes and must have three or more nodes with more than 10% free capacity in order to allow writes. This reserved space is required to ensure that ECS does not run out of space while persisting system metadata. If this criteria is not met, the write will fail. The ability of a storage pool to accept writes does not affect the ability of other pools to accept writes. For example, if you have a load balancer that detects a failed write, the load balancer can redirect the write to another VDC.

The replication group is used by ECS for replicating data to other sites so that the data is protected and can be accessed from other, active sites. When you create a bucket, you specify the replication group it is in. ECS ensures that the bucket and the objects in the bucket are replicated to all the sites in the replication group.

ECS can be configured to use more than one replication scheme, depending on the requirements to access and protect the data. The following figure shows a replication group (RG 1) that spans all three sites. RG 1 takes advantage of the XOR storage efficiency provided by ECS when using three or more sites. In the figure, the replication group that spans two sites (RG 2), contains full copies of the object data chunks and does not use XOR'ing to improve storage efficiency.

Figure 7 Replication group spanning three sites and replication group spanning two sites



The physical storage that the replication group uses at each site is determined by the storage pool that is included in the replication group. The storage pool aggregates the disk storage of each of the minimum of four nodes to ensure that it can handle the placement of erasure coding fragments. A node cannot exist in more than one storage pool. The storage pool can span racks, but it is always within a site.

Working with storage pools in the ECS Portal

You can use storage pools to organize storage resources based on business requirements. For example, if you require physical separation of data, you can partition the storage into multiple storage pools.

You can use the **Storage Pool Management** page available from **Manage** > **Storage Pools** to view the details of existing storage pools, to create new storage pools, and to modify existing storage pools. You cannot delete storage pools in this release.

Figure 8 Storage Pool Management page

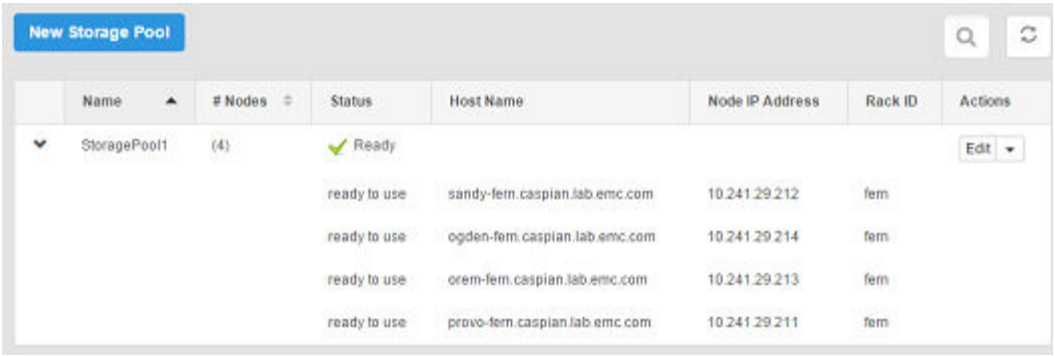


Table 5 Storage pool properties

Field	Description
Name	The name of the storage pool.

Table 5 Storage pool properties (continued)

Field	Description
# Nodes	The number of nodes assigned to the storage pool.
Status	The state of the storage pool and of the nodes. • Ready: At least four nodes are installed and all nodes are in the `ready to use` state. • Not Ready: A node in the storage pool is not in the `ready to use` state. • Partially Ready: Less than four nodes, and all nodes are in the `ready to use` state.
Host Name	The fully qualified host name assigned to the node.
Node IP address	The public IP address assigned to the node.
Rack ID	The name assigned to the rack that contains the nodes.
Actions	The actions that can be completed for the storage pool. • **Edit**: Change the storage pool name or modify the set of nodes included in the storage pool. • **Delete**: Used by Customer Support to delete the storage pool. System Administrators or root users should not attempt to delete the storage pool. If you attempt this operation in the ECS Portal, you receive an error message that states this operation is not supported. If you must delete a storage pool, contact your customer support representative.
Cold Storage	A storage pool that is specified as Cold Storage. Cold Storage pools use an erasure coding (EC) scheme that is more efficient for infrequently accessed objects. Cold Storage is also known as a Cold Archive. After a storage pool is created, this setting cannot be changed.

Create a storage pool

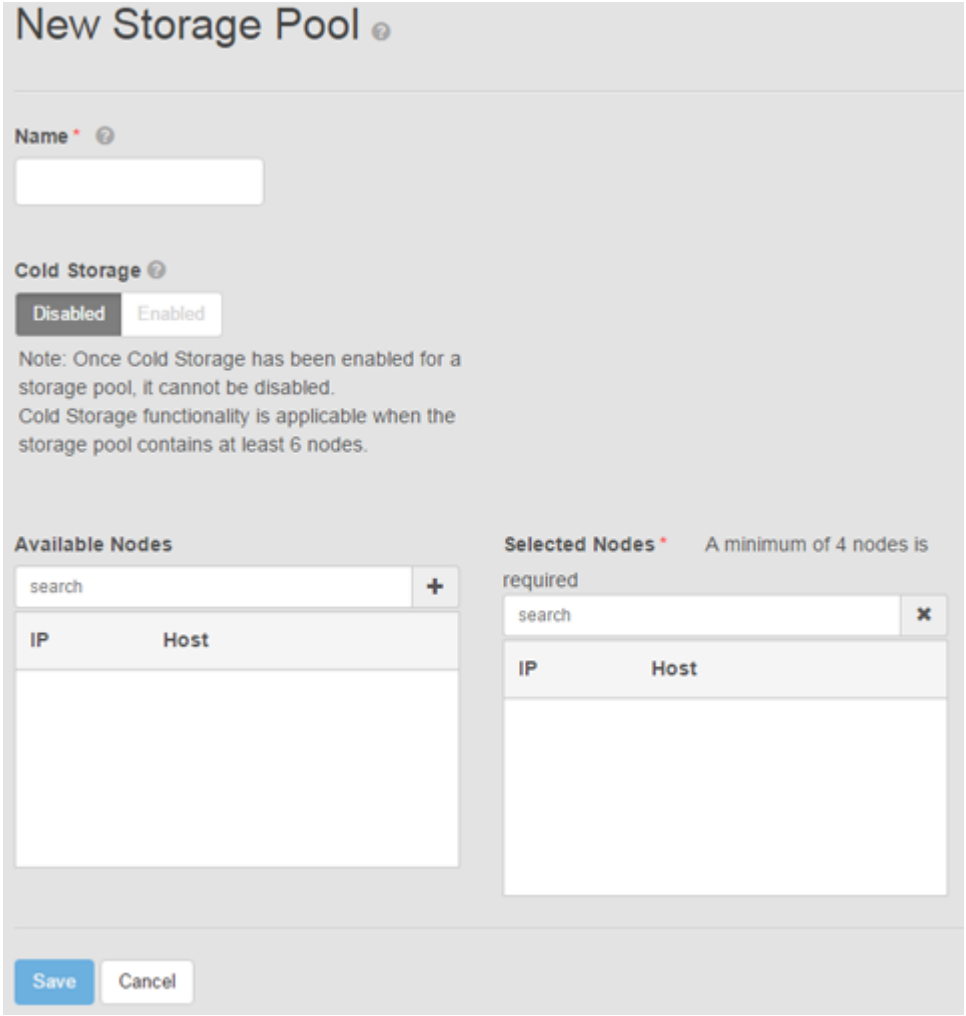
Storage pools must contain a minimum of four nodes. The first storage pool that is created is known as the system storage pool because it stores system metadata.

Before you begin

This operation requires the System Administrator role in ECS.

Procedure

1. In the ECS Portal, select **Manage** > **Storage Pools**.
2. On the **Storage Pool Management** page, click **New Storage Pool**.



3. On the **New Storage Pool** page, in the **Name** field, type the storage pool name (for example,`StoragePool1`).
4. In the **Cold Storage** field, specify if this storage pool is Cold Storage. Cold storage contains infrequently accessed data. The ECS data protection scheme for cold storage is optimized to increase storage efficiency. After a storage pool is created, this setting cannot be changed.

   Note

   Cold storage requires a minimum hardware configuration of six nodes. For more information, see ECS data protection on page 14.

5. From the **Available Nodes** list, select the nodes to add to the storage pool.
   a. To select nodes one-by-one, click the **+** icon beside each node.
   b. To select all available nodes, click the **+** icon at the top of the **Available Nodes** list.
   c. To narrow the list of available nodes, in the **search** field, type the public IP address for the node or the host name.
6. When you finish the node selection, click **Save**.
7. Wait 10 minutes after the storage pool is in the **Ready** state before you perform other configuration tasks, to allow the storage pool time to initialize.

If you receive the following error, wait a few more minutes before you attempt any further configuration. `Error 7000 (http: 500): An error occurred in the API Service. An error occurred in the API service.Cause: error insertVdcInfo. Virtual Data Center creation failure may occur when Data Services has not completed initialization.`

Edit a storage pool

You can change the name of a storage pool or change the set of nodes included in the storage pool.

Before you begin

This operation requires the System Administrator role in ECS.

Procedure

1. In the ECS Portal, select **Manage** > **Storage Pools**.
2. On the **Storage Pool Management** page, locate the storage pool you want to edit in the table. Click **Edit** in the **Actions** column beside the storage pool you want to edit.
3. On the **Edit Storage Pool** page:
   - To modify the storage pool name, in the **Name** field, type the new name.
   - To modify the nodes included in the storage pool:
     - In the **Selected Nodes** list, remove an existing node in the storage pool by clicking the **-** icon beside the node.
     - In the **Available Nodes** list, add a node to the storage pool by clicking the **+** icon beside the node.
4. Click **Save**.

Working with VDCs in the ECS Portal

An ECS virtual data center (VDC) is the top-level resource that represents the collection of ECS infrastructure components to manage as a unit.

You can use the **Virtual Data Center Management** page available from **Manage** > **Virtual Data Center** to view VDC details, to create a new VDC, to modify existing VDCs, to delete VDCs, and to federate multiple VDCs for a multi-site deployment. The following example shows the **Virtual Data Center Management** page for a multi-site, federated deployment. It is configured with two sites named vdc1 and vdc2.

Figure 9 Virtual Data Center Management page

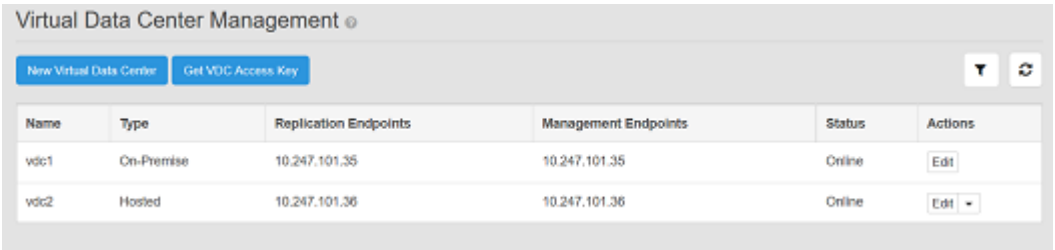


Table 6 VDC properties

Field	Description
Name	The name of the VDC.
Type	The type of VDC, which can be Hosted or On-Premise.
Replication Endpoints	Endpoints for communication of replication data between sites. If a separate replication network is configured, each node's replication IP address is displayed. If no separate replication network is configured, each node's public IP address is displayed. If the replication network and management network are not separated, the replication endpoints and management endpoints are the same. If a load balancer is configured to distribute the load between the replication IP addresses of the nodes, the address configured on the load balancer is displayed.
Management Endpoints	Endpoints for communication of management commands between sites. If a separate management network is configured, each node's management IP address is displayed. If no separate management network is configured, each node's public IP address is displayed. If the management network and replication network are not separated, the management endpoints and replication endpoints are the same.
Status	The state of the VDC. • Online • Permanently Failed: The VDC was deleted.
Actions	The actions that can be completed for the VDC. • **Edit**: Change the name of a VDC, the VDC access key, and the VDC replication and management endpoints. • **Delete**: Delete the VDC. The delete operation triggers permanent failover of the VDC, You cannot add the VDC again by using the same name. You cannot delete a VDC that is part of a replication group until you first remove it from the replication group. You cannot delete a VDC when you are logged in to the VDC you are trying to delete.

Create a VDC for a single site

You can create a VDC for a single-site deployment, or when you create the first VDC in a multi-site federation.

Before you begin

This operation requires the System Administrator role in ECS.

Ensure that one or more storage pools are available and in the `Ready` state.

Procedure

1. In the ECS Portal, select **Manage** > **Virtual Data Center**.
2. On the **Virtual Data Center Management** page, click **New Virtual Data Center**.
3. On the **New Virtual Data Center** page, in the **Name** field, type the VDC name (for example: `VDC1`).

   VDC names can be from 1 to 255 characters. Valid characters include a to z, A to Z, 0 to 9, dash (-), and underscore (_).

4. To create an access key for the VDC, either:
   - Type the VDC access key value in the **Key** field, or
   - Click **Generate** to generate a VDC access key.

   The VDC Access Key is used as a symmetric key for encrypting replication traffic between VDCs in a multi-site federation.
5. In the **Replication Endpoints** field, type the replication IP address of each node in the storage pools that are assigned to the VDC. Type them as a comma-separated list.

   If network separation was configured at installation, each node's replication IP address is displayed. If the replication network was not separated at installation, each node's public IP address is displayed.
   If a load balancer is configured to distribute the load between the replication IP addresses of the nodes, the address configured on the load balancer is displayed.
6. In the **Management Endpoints** field, type the management IP address of each node in the storage pools that are assigned to the VDC. Type them as a comma-separated list.

   If network separation was configured at installation, each node's management IP address is displayed. If the management network was not separated at installation, each node's public IP address is displayed.
7. Click **Save**.

   When the VDC is created, ECS automatically sets the VDC's **Type** to either **On-Premise** or **Hosted**.

Add a VDC to a federation

You can add a VDC to an existing VDC (for example, vdc1) to create a federation.

Before you begin

Obtain the **ECS Portal** credentials for the root user, or for a user with System Administrator credentials, to log in to both sites.

Ensure that you have the public IP addresses for the nodes from the site you are adding, or, if you separated the management network and the replication network, the management IP addresses and the replication IP addresses. If a load balancer is configured to distribute the load between the replication IP addresses of the nodes, you must have the IP address configured on the load balancer.

Ensure that the site you are adding has a valid ECS license uploaded and has at least one storage pool in the `Ready` state.

⚠ CAUTION

Remember that you are not creating a VDC on the site you are adding to the federation. Instead, retrieve the VDC Access Key from the site you want to add, and create the new VDC from the first node in the existing federation.

Procedure

1. On the site that you want to add, log in to the ECS Portal (for example: `vdc2`).

   The default credentials are `root/ChangeMe`.

2. In the ECS Portal, select **Manage** > **Virtual Data Center**.
3. On the **Virtual Data Center Management** page, click **Get VDC Access Key**.
4. Select the access key, and press `Ctrl-c` to copy it.
5. Log out of the ECS Portal on the site you are adding.
6. On the first VDC in the federation, log in to the ECS Portal (for example, `vdc1`).
7. Select **Manage** > **Virtual Data Center**.
8. On the **Virtual Data Center Management** page, click **New Virtual Data Center**.
9. On the **New Virtual Data Center** page, in the **Name** field, type the VDC name (for example, `vdc2`).
10. Click the **Key** field, and then press `Ctrl-v` to paste the access key you copied from the site you are adding (`vdc2`).
11. In the **Replication Endpoints** and **Management Endpoints** fields, type the replication and management IP addresses of each node in the storage pools that are assigned to the site you are adding. Type them as comma-separated lists.

    If the replication network and management network are not separated, both of these fields contain the same public IP addresses for the nodes in the VDC. If the management and replication networks are separated, you must list the IP addresses for the appropriate network.
    If a load balancer is configured to distribute the load between the replication IP addresses of the nodes, the replication endpoint is the IP address configured on the load balancer.

12. Click **Save**.

Results

When you add the VDC to the federation, ECS automatically sets the type of the VDC to either **On-Premise** or **Hosted**.

Edit a VDC

You can change the name of the VDC, the VDC access key, or the replication and management endpoints.

Before you begin

This operation requires the System Administrator role in ECS.

Procedure

1. In the ECS Portal, select **Manage** > **Virtual Data Center**.
2. On the **Virtual Data Center Management** page, locate the VDC you want to edit in the table. Click **Edit** in the **Actions** column beside the VDC you want to edit.
3. On the **Edit Virtual Data Center** page:
   - To modify the VDC name, in the **Name** field, type the new name.
   - To modify the VDC access key for the node you are logged into, in the **Key** field, type the new key value, or click **Generate** to generate a new VDC access key.
   - To modify the replication and management IP addresses of the nodes in the VDC, in the **Replication Endpoints** and **Management Endpoints** fields, type the new IP addresses in a comma-separated list. If network separation

is not configured, the replication and management IP addresses are the same.

You cannot edit the VDC **Type** field. When you initially create a VDC, ECS automatically determines its type to be **On-Premise** or **Hosted**.

4. Click **Save**.

Delete a VDC and fail over a site

You can delete a VDC. Deleting a VDC initiates site failover when the VDC you delete is part of a multi-site federation.

Before you begin

This operation requires the System Administrator role in ECS.

If a disaster occurs, an entire VDC can become unrecoverable. ECS initially treats the unrecoverable VDC as a temporary site failure. If the failure is permanent, you must remove the VDC from the federation to initiate failover processing which reconstructs and reprotects the objects that are stored on the failed VDC. The recovery tasks run as a background process.

To delete a VDC from the federation, you must delete the VDC from all of the replication groups to which it belongs.

Important: Before you complete the following steps, contact your customer support representative to perform required internal configuration procedures before removing the VDC from your system.

Procedure

1. Log in to one of the operational VDCs in the federation.
2. In the ECS Portal, select **Manage** > **Replication Group**.
3. On the **Replication Group Management** page, beside the replication group that contains the VDC that you want to delete, click **Edit**.
4. On the **Edit Replication Group** page, in the row that contains the VDC and storage pool that you want to remove, click **Delete**.
5. Click **Save**.
6. Repeat steps 3 through 5 for all replication groups that contain the VDC you want to delete
7. In the ECS Portal, select **Manage** > **VDC**.

   On the **Virtual Data Center Management** page, the status for the permanently removed VDC changes to `Permanently failed`.
8. On the **Virtual Data Center Management** page, in the row of the VDC with the `Permanently failed` status, select **Delete** in the **Actions** column.
9. Click **Save**.

Results

You can view the progress of the recovery process on the **Monitor** > **Geo Replication** > **Failover Processing** page.

Delete a VDC that has a replication group belonging only to that VDC

You can delete a VDC if it contains a replication group that belongs only to that VDC, but you must add another VDC to the replication group before you delete the VDC.

Before you begin

This operation requires the System Administrator role in ECS.

In the scenario where you have a replication group that belongs only to the VDC you want to delete, you must add another VDC to the replication group before you delete the VDC. This is required because ECS does not support deleting replication groups. For example, consider a geo-federated setup where VDC1 contains replication group RG1. RG1 belongs only to VDC1. VDC2 contains RG2. RG2 belongs only to VDC2. To delete VDC1, you must do the following:

Procedure

1. On the **Edit Replication Group RG1** page, click **Add VDC** and add VDC2 to RG1.
2. Click **Save**.

   An error message displays: `Error 30026 (http: 503): Service is busy. The Data services system is busy. Please try again.` Disregard this error message.

   When you click **Save**, the **Name** field of the replication group becomes blank.
3. Click the **Delete** button next to VDC1.

   The **You must confirm that you understand the consequence of this delete** dialog displays. This dialog informs you that removing VDC1 from RG1 means that you are permanently removing VDC1 from the geo-federated system.
4. In the blank field in the dialog, type **`Please remove VDC1 (Permanently Failed) from the system.`**
5. Click **OK**.

   VDC1 no longer belongs to RG1. Now RG1 belongs only to VDC2. VDC1 is permanently removed from the ECS geo-federated system.
6. On the **Edit Replication Group** page, in the blank **Name** field, type **`RG1`**.
7. Click **Save**.
8. On the **Virtual Data Center Management** page, next to VDC1 that has the `Permanently failed` status, select **Delete** in the **Actions** column.
9. Click **Save**.

Working with replication groups in the ECS Portal

You can use replication groups to define where storage pool content is protected. Replication groups can be local or global. Local replication groups do not replicate data to other VDCs, but protect objects within the same VDC against disk or node failures using mirroring and erasure coding techniques. Global replication groups protect objects by replicating them to another site within an ECS federation and, by doing so, protect against site failures.

You can use the **Replication Group Management** page to view replication group details, to create new replication groups, and to modify existing replication groups. You cannot delete replication groups in this release.

Figure 10 Replication Group Management page

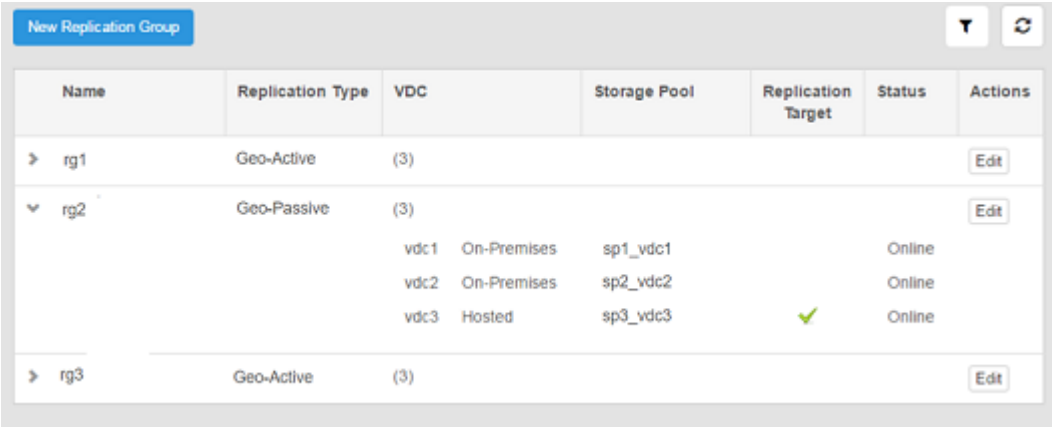


Table 7 Replication Group properties

Field	Description
Name	The name of the replication group.
Replication Type	The replication type can be Geo Active or Geo Passive. Geo Passive means that a site is designated as the target for replication data and is only available where there are exactly three sites. Geo Passive cannot be selected if Replicate to All Sites is enabled. If you have a Hosted site, it will automatically be selected as the target for replication in a Geo Passive configuration.
VDC	The number of VDCs in the replication group and the names of the VDCs where the storage pools are located.
Storage Pool	The names of the storage pools and their associated VDCs. A replication group can contain a storage pool from each VDC in a federation.
Replication Target	The storage pool in the replication group that is the replication target in a Geo-Passive configuration.
Status	The state of the replication group. • Online • Temp Unavailable: Replication traffic to this VDC has failed. If all replication traffic to the same VDC is in the Temp Unavailable state, further investigation about the cause of the failure is recommended.
Actions	The actions that can be completed for the replication group. **Edit**: Modify the replication group name and the set of VDCs and storage pools in the replication group.

Create a replication group

Replication groups can be local to a VDC or can protect data by replicating data across sites.

Before you begin

This operation requires the System Administrator role in ECS.

If you want the replication group to span multiple VDCs, you must ensure that the sites are installed, initialized with a VDC identity and a storage pool that can be part of the replication group, and federated to the primary VDC.

Procedure

1. In the ECS Portal, select **Manage** > **Replication Group**.
2. On the **Replication Group Management** page, click **New Replication Group**.
3. On the **New Replication Group** page, in the **Name** field, type a name (for example, `ReplicationGroup1`).
4. Optionally, enable **Replicate to All Sites** for this replication group. This option can be enabled only at the time of creation and cannot be disabled later.

   For a Geo-Passive configuration, select **Disabled**.

Option	Description
Replicate to All Sites Disabled	The replication group uses default replication. With default replication, data is stored at the primary site and a full copy is stored at a secondary site chosen from the sites within the replication group. The secondary copy is protected by triple-mirroring and erasure coding. This process provides data durability with storage efficiency.
Replicate to All Sites Enabled	The replication group makes a full readable copy of all objects to all sites (VDCs) within the replication group. Having full readable copies of objects on all VDCs in the replication group provides data durability and improves local performance at all sites at the cost of storage efficiency.

5. Select the **Replication Type**: **Geo-Active** or **Geo-Passive**.

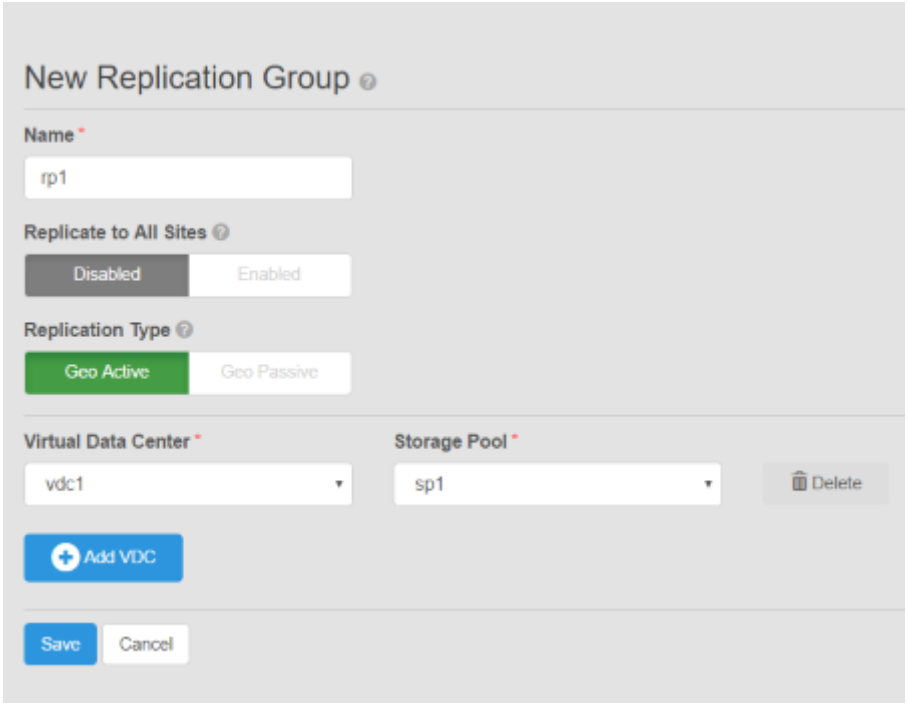
   Geo-Passive is available only when you have three or more sites.

6. Click **Add VDC** to add storage pools from sites to the replication group.

   The steps to add storage pools to a replication group depends on whether you have a single site, a Geo-Active, or Geo-Passive environment.

7. To add storage pools to a Geo-Active (or to a single site) configuration, use the steps below.

   a. From the **Virtual Data Center** list, select the VDC that will provide a storage pool for the replication group.



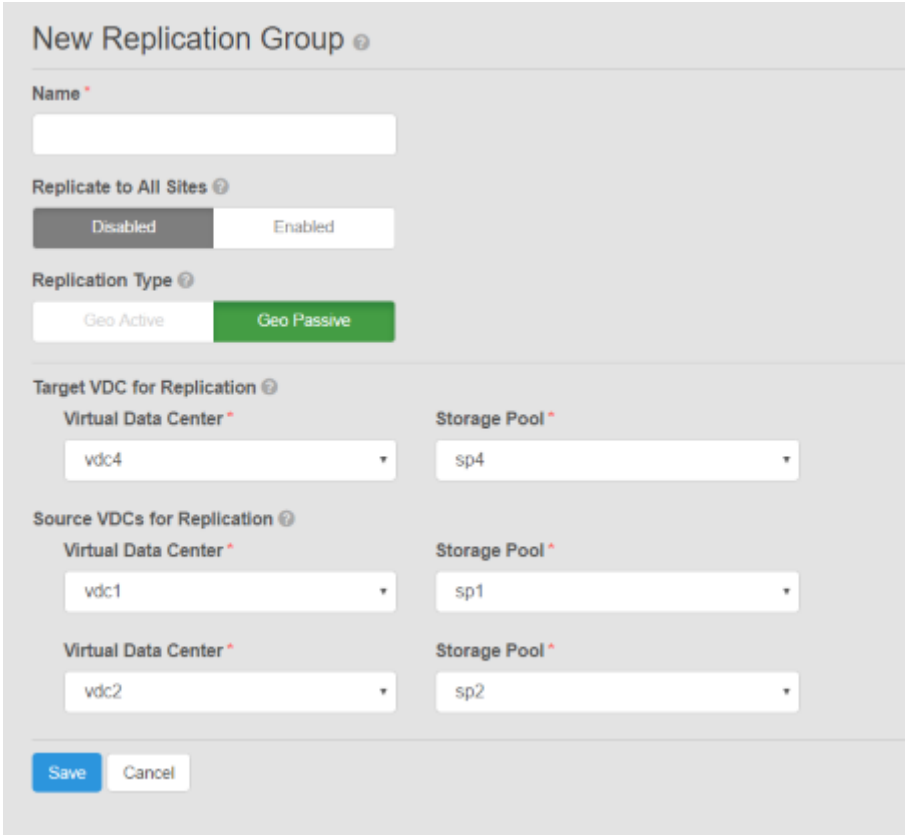
b. From the **Storage Pool** list, select the storage pool that belongs to the selected VDC.

c. To include other sites in the replication group, click **Add VDC**.

d. Repeat these steps for each storage pool that you want to add to the replication group.

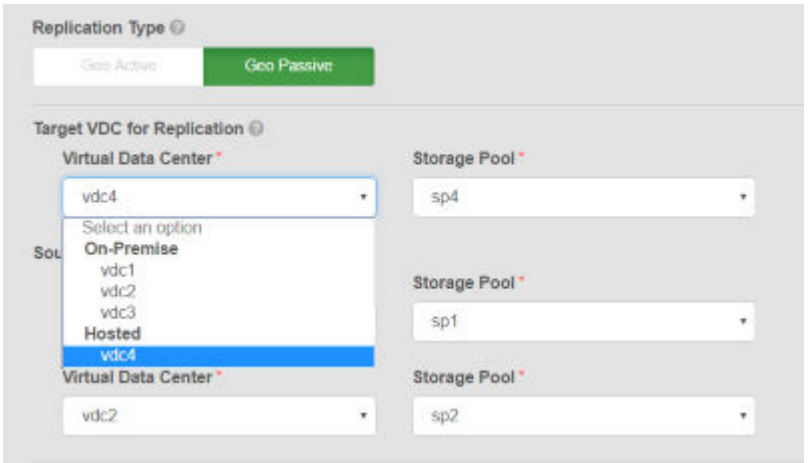
8. To add storage pools for a Geo-Passive configuration, complete the following steps.

   For a **Geo-Passive** configuration, ECS assumes that you want to add one site as a passive replication target and two active sites.



a. In the **Target VDC for Replication Virtual Data Center** list, select the site that you want to add as the replication target.

   If you have a hosted site, it is automatically selected as the replication target.



b. In the **Target VDC for Replication Storage Pool** list, select the storage pool that belongs to the selected VDC.

c. In each of the two **Source VDC for Replication Virtual Data Center** lists, select the site that you want to add as the active site.

d. In each of the two **Source VDC for Replication Storage Pool** lists, select the storage pool that belongs to each selected VDC. These storage pools will provide storage at the two active sites.

9. Click **Save**.

Edit a replication group

You can change the name of the replication group or change the set of VDCs and storage pools in the replication group.

Before you begin

This operation requires the System Administrator role in ECS.

⚠ CAUTION

In a multi-site federation, when you edit a replication group and choose to delete a VDC from the replication group(s) to which it belongs, you are permanently removing that VDC from the federation. Permanently removing the VDC from the federation triggers the failover process where objects owned by that VDC (and its storage pool) will be owned by one of the remaining VDCs. In ECS, each object has a primary, or owning VDC. The time taken to complete this failover process depends on the amount of data that must be moved. If you created the replication group with the Replicate to All Sites option enabled, the time taken to move all data to the remaining sites is short, as a copy exists at all sites.

You cannot edit the **Replicate to All Sites** or **Replication Type** fields. After you set these options when you first create the replication group, they cannot be changed.

Procedure

1. In the ECS Portal, select **Manage** > **Replication Group**.
2. On the **Replication Group Management** page, beside the replication group you want to edit, click **Edit**.
3. On the **Edit Replication Group** page,
   - To modify the replication group name, in the **Name** field, type the new name.
   - To add a VDC to the replication group, click **Add VDC** and select the VDC and storage pool from the list.
   - To delete a VDC from the replication group, click the **Delete** button beside the VDC (and its storage pool).

     ⚠ CAUTION

     Deleting a VDC from the replication group(s) to which it belongs means you are removing this VDC permanently from the federation. See Delete a VDC and fail over a site on page 36.
4. Click **Save**.

CHAPTER 4

Authentication Providers

Introduction to authentication providers

You can add authentication providers to ECS if you want users to be authenticated by systems external to ECS.

An authentication provider is a system that is external to ECS that can authenticate users on behalf of ECS. ECS stores the information that allows it to connect to the authentication provider so that ECS can request authentication of a user.

In ECS, the following types of authentication provider are available:

- Active Directory (AD) authentication or Lightweight Directory Access Protocol (LDAP) authentication: Used to authenticate domain users that are assigned to management roles in ECS.
- Keystone: Used to authenticate OpenStack Swift object users.

Authentication providers can be created from the ECS Portal (see Working with authentication providers in the ECS Portal on page 44) or by using the ECS Management REST API or CLI. You can use the following procedures to create AD/LDAP or Keystone authentication providers.

- Add an AD or LDAP authentication provider on page 46
- Add a Keystone authentication provider on page 49

Working with authentication providers in the ECS Portal

You can use the **Authentication Provider Management** page available from **Manage** > **Authentication** to view the details of existing authentication providers, to add authentication providers, to modify existing authentication providers, and to delete authentication providers.

Figure 11 Authentication Provider Management page

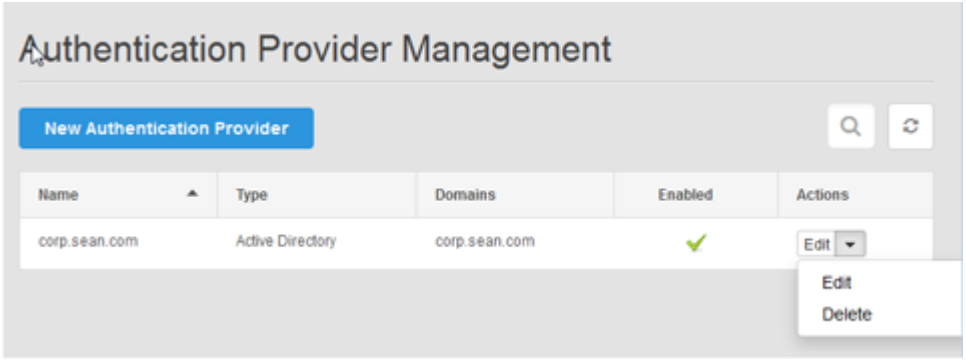


Table 8 Authentication provider properties

Field	Description
Name	The name for the authentication provider.
Type	The type of authentication provider. The authentication provider is an Active Directory (AD), Lightweight Directory Access Protocol (LDAP), or Keystone V3 server.
Domains	The domains that the authentication provider provides access to.

Table 8 Authentication provider properties (continued)

Field	Description
Enabled	Indicates whether the authentication provider is Enabled or Disabled.
Actions	The actions that can be completed for the authentication provider. • **Edit**: Change the AD or LDAP authentication provider settings listed in Table 9 on page 46 or change the Keystone authentication provider settings listed in Table 10 on page 50. • **Delete**: Delete the authentication provider. • **New Authentication Provider** button: Add an authentication provider.

Considerations when adding Active Directory authentication providers

When you configure ECS to work with Active Directory (AD), you must decide whether to add a single AD authentication provider to manage multiple domains, or to add separate AD authentication providers for each domain.

The decision to add a single AD authentication provider, or multiple, depends on the number of domains in the environment, and the location on the tree from which the manager user is able to search. Authentication providers have a single search base from which the search begins, and a single manager account that has read access at the search base level and below.

You can add a single authentication provider for multiple domains in the following conditions:

- You manage an AD forest
- The manager account has privileges to search all user entries in the tree
- The search is conducted throughout the whole forest from a single search base, not just the domains listed in the provider

Otherwise, add separate authentication providers for each domain.

Note

If you manage an AD forest and you have the necessary manager account privileges, there are scenarios in which you might still want to add an authentication provider for each domain. For example, if you want tight control on each domain and more granularity on setting the search base starting point for the search.

The search base must be high enough in the directory structure of the forest for the search to correctly find all the users in the targeted domains. The following search examples describe the best options for adding either single or multiple authentication providers.

- In the scenario where the forest in the configuration contains ten domains but you want to target only three, you would not want to add a single authentication provider to manage multiple domains, because the search would unnecessarily span the whole forest. This might adversely affect performance. In this case, you should add three separate authentication providers for each domain.

- In the scenario where the forest in the configuration contains ten domains and you want to target ten domains, adding a single authentication provider to manage multiple domains is a good choice, because there is less overhead to set up.

Add an AD or LDAP authentication provider

You can add one or more authentication providers to ECS to perform user authentication for ECS domain users.

Before you begin

- This operation requires the System Administrator role in ECS.
- You need access to the authentication provider information listed in AD/LDAP authentication provider settings. Note especially the requirements for the Manager DN user.

Procedure

1. In the ECS Portal, select **Manage** > **Authentication**.
2. On the **Authentication Provider Management** page, click **New Authentication Provider**.
3. On the **New Authentication Provider** page, type values in the fields. For more information about these fields, see AD/LDAP authentication provider settings.
4. Click **Save**.
5. To verify the configuration, add a user from the authentication provider at **Manage** > **Users** > **Management Users**, then try to log in as the new user.

After you finish

You must add (assign) the domain users into a namespace if you want these users to perform ECS object user operations. For more information, see Add domain users into a namespace on page 78.

AD or LDAP authentication provider settings

You must provide authentication provider information when you add or edit an AD or LDAP authentication provider.

Table 9 AD or LDAP authentication provider settings

Field	Description and requirements
Name	The name of the authentication provider. You can have multiple providers for different domains.
Description	Free text description of the authentication provider.
Type	The type of authentication provider. Active Directory or LDAP.
Domains	The collection of administratively defined objects that share a common directory database, security policies, and trust relationships. A domain can span multiple physical locations or sites and can contain millions of objects. Example: `mycompany.com` If an alternate UPN suffix is configured in the Active Directory, the Domains field should also contain the alternate UPN configured for the domain. For example, if `myco` is added as an alternate UPN suffix for `mycompany.com`, then the Domains field should contain both `myco` and `mycompany.com`.
Server URLs	The LDAP or LDAPS (secure LDAP) with the domain controller IP address. The default port for LDAP is 389. The default port for LDAPS is 636.

Table 9 AD or LDAP authentication provider settings (continued)

Field	Description and requirements
	You can specify one or more LDAP or LDAPS authentication provider. Example: `ldap://<Domain controller IP>:<port>` (if not default port) or `ldaps://<Domain controller IP>:<port>`(if not default port) If the authentication provider supports a multidomain forest, use the global catalog server IP and always specify the port number. The default port for LDAP is 3268. The default port for LDAPS is 3269. Example: `ldap(s)://<Global catalog server IP>:<port>`
Manager DN	The Active Directory Bind user account that ECS uses to connect to the Active Directory or LDAP server. This account is used to search Active Directory when a ECS administrator specifies a user for role assignment. This user account must have `Read all inetOrgPerson information` in Active Directory. The `InetOrgPerson` object class is used in several non-Microsoft, LDAP and X.500 directory services to represent people in an organization. To set this privilege in Active Directory: 1. Open **Active Directory Users and Computers**. 2. Right-click the domain, select **Delegate Control**, and then click **Next**. 3. In the **Delegation of Control** wizard, click **Next**, and then click **Add**. 4. In the **Select Users, Computers, or Groups** dialog box, select the user that you are using for `managerdn`, and then click **Next**. 5. In the **Tasks to Delegate** page, in **Delegate the following common tasks**, check the `Read all inetOrgPerson information` task, and then click **Next**. 6. Click **Finish**. In this example: `CN=Manager,CN=Users,DC=mydomaincontroller,DC=com`, the Active Directory Bind user is `Manager`, in the `Users` tree of the `mydomaincontroller.com` domain. Usually `managerdn` is a user who has fewer privileges than Administrator, but has sufficient privileges to query Active Directory for users attributes and group information. **Important:** You must update this user account in ECS if the `managerdn` credentials change in Active Directory.
Manager Password	The password of the `managerdn` user. **Important:** You must update this password in ECS if the `managerdn` credentials change in Active Directory.
Providers	This setting is Enabled by default when adding an authentication provider. ECS validates the connectivity of the enabled authentication provider and that the name and domain of the enabled authentication provider are unique. Select Disabled only if you want to add the authentication provider to ECS, but you do not immediately want to use it for authentication. ECS does not validate the connectivity of a disabled authentication provider, but it does validate that the authentication provider name and domain are unique.
Group Attribute	This attribute applies only to Active Directory; it does not apply to other types of authentication providers. The AD attribute that is used to identify a group. Used for searching the directory by groups.

Table 9 AD or LDAP authentication provider settings (continued)

Field	Description and requirements
	Example: `CN` **Note** After you set this attribute for an AD authentication provider, you cannot change it, because the tenants using this provider might already have role assignments and permissions configured with group names in a format that uses this attribute.
Group Whitelist	This setting applies only to Active Directory; it does not apply to other types of authentication providers. Optional. One or more group names as defined by the authentication provider. This setting filters the group membership information that ECS retrieves about a user. • When a group or groups are included in the whitelist, ECS is aware only of a user's membership in the specified groups. Multiple values (one value on each line in the ECS Portal, and values comma-separated in CLI and API) and wildcards (for example MyGroup*,TopAdminUsers*) are allowed. • The default setting is blank. ECS is aware of all groups that a user belongs to. Asterisk (*) is the same as blank. Example: UserA belongs to `Group1` and `Group2`. If the whitelist is blank, ECS knows that UserA is a member of `Group1` and `Group2`. If the whitelist is `Group1`, ECS knows that UserA is a member of `Group1`, but does not know that UserA is a member of `Group2` (or of any other group). Use care when adding a whitelist value. For example, if you map a user to a namespace that is based on group membership, then ECS must be aware of the user's membership in the group. To restrict access to a namespace to only users of certain groups, complete the following tasks. • Add the groups to the namespace user mapping. The namespace is configured to accept only users of these groups. • Add the groups to the whitelist. ECS is authorized to receive information about them. By default, if no groups are added to the namespace user mapping, users from any groups are accepted, regardless of the whitelist configuration.
Search Scope	The levels to search. Possible values are: • One Level (search for users one level under the search base) • Subtree (search the entire subtree under the search base)
Search Base	The Base Distinguished Name that ECS uses to search for users at login time and when assigning roles or setting ACLs. The following example searches for all users in the `Users` container. `CN=Users,DC=mydomaincontroller,DC=com` The following example searches for all users in the `Users` container in the `myGroup` organization unit. Note that the structure of the search base value begins with the leaf level and goes up to the domain controller level, which is the reverse of the structure seen in the **Active Directory Users and Computers** snap-in.

Table 9 AD or LDAP authentication provider settings (continued)

Field	Description and requirements
	`CN=Users,OU=myGroup,DC=mydomaincontroller,DC=com`
Search Filter	The string used to select subsets of users. Example: `userPrincipalName=%u` **Note** ECS does not validate this value when you add the authentication provider. If an alternate UPN suffix is configured in the Active Directory, the Search Filter value must be of the format `sAMAccountName=%U` where `%U` is the username, and does not contain the domain name.

Add a Keystone authentication provider

You can add a Keystone authentication provider to authenticate OpenStack Swift users.

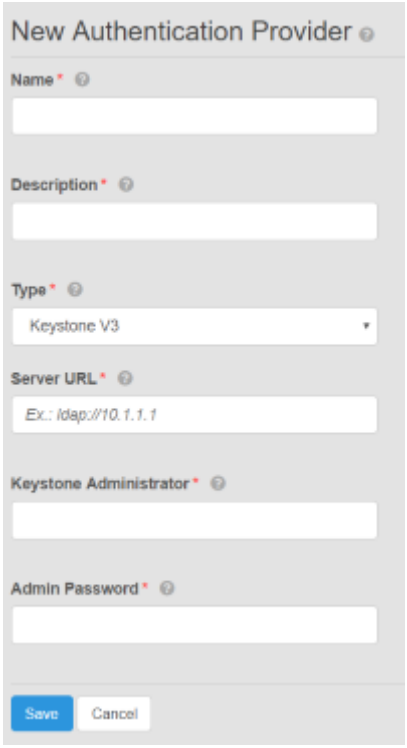
Before you begin

- This operation requires the System Administrator role in ECS.
- You can add only one Keystone authentication provider.
- Obtain the authentication provider information listed in Keystone authentication provider settings on page 50.

Procedure

1. In the ECS Portal, select **Manage** > **Authentication**.
2. On the **Authentication Provider Management** page, click **New Authentication Provider**.
3. On the **New Authentication Provider** page, in the **Type** field, select **Keystone V3**.

   The required fields are displayed.



4. Type values in the **Name**, **Description**, **Server URL**, **Keystone Administrator**, and **Admin Password** fields. For more information about these fields, see Keystone authentication provider settings on page 50.
5. Click **Save**.

Keystone authentication provider settings

You must provide authentication provider information when you add or edit a Keystone authentication provider.

Table 10 Keystone authentication provider settings

Field	Description
Name	The name of the Keystone authentication provider. This name is used to identify the provider in ECS.
Description	Free text description of the authentication provider.
Type	Keystone V3.
Server URL	URI of the Keystone system that ECS connects to in order to validate Swift users.
Keystone Administrator	Username for an administrator of the Keystone system. ECS connects to the Keystone system using this username.
Administrator Password	Password of the specified Keystone administrator.

CHAPTER 5

Namespaces

Introduction to namespaces

You can use namespaces to provide multiple tenants with access to the ECS object store and to ensure that the objects and buckets written by users of each tenant are segregated from the other tenants.

The topics in this section introduce some concepts around tenants and namespace settings:

- Namespace settings on page 53
- Working with namespaces in the ECS Portal on page 57

and describe the operations required to configure a namespace using the ECS Portal:

- Create a namespace on page 57

The namespace configuration tasks that can be performed in the ECS Portal can also be performed using the ECS Management REST API.

ECS supports access by multiple tenants, where each tenant is defined by a namespace and the namespace has a set of configured users who can store and access objects within the namespace. Users from one namespace cannot access the objects that belong to another namespace.

Namespaces are global resources in ECS. A System Administrator or Namespace Administrator can access ECS from any federated VDC and can configure the namespace settings. The object users that you assign to a namespace are global and can access the object store from any federated VDC.

You configure a namespace with attributes that define which users can access the namespace and what characteristics the namespace has. Users with the appropriate privileges can create buckets, and can create objects within buckets, in the namespace.

An object in one namespace can have the same name as an object in another namespace. ECS can identify objects by the namespace qualifier.

You can configure namespaces to monitor and meter their usage, and you can grant management rights to the tenant so that it can perform configuration, monitoring, and metering operations.

You can use buckets to create subtenants. The bucket owner is the subtenant administrator and can assign users to the subtenant by using access control lists (ACLs). However, subtenants do not provide the same level of segregation as tenants. Any user assigned to the tenant could be assigned privileges on a subtenant, so care must be taken when assigning users.

Namespace tenancy

A System Administrator can set up namespaces in the following tenant scenarios:

Enterprise single tenant

All users access buckets and objects in the same namespace. Buckets can be created for subtenants, to allow a subset of namespace users to access the same set of objects. For example, a subtenant might be a department within the organization.

Enterprise multitenant

Departments within an organization are assigned to different namespaces and department users are assigned to each namespace.

Cloud Service Provider single tenant

A single namespace is configured and the Service Provider provides access to the object store for users within the organization or outside the organization.

Cloud Service Provider multitenant

The Service Provider assigns namespaces to different companies and assigns an administrator for the namespace. The Namespace Administrator for the tenant can then add users and can monitor and meter the use of buckets and objects.

Namespace settings

The following table describes the settings you can specify when you create or edit a ECS namespace.

The way in which namespace and bucket names are used when addressing objects in ECS is described in Object base URL on page 142.

Table 11 Namespace settings

Field	Description	Can be edited
Name	The name of the namespace, in lowercase characters.	No
User Admin	The user ID of one or more users assigned to the Namespace Administrator role; a list of users is comma separated. Namespace Administrators can be local or domain users. If the Namespace Administrator is a domain user, ensure that an authentication provider is added to ECS. See Introduction to users and roles on page 64 for details.	Yes
Domain Group Admin	The domain group assigned to the Namespace Administrator role. Any member, when authenticated, is assigned the Namespace Administrator role for the namespace. The domain group must be assigned to the namespace by setting the Domain User Mappings for the namespace. To use this feature you must ensure that an authentication provider is added to ECS. See Introduction to users and roles on page 64 for details.	Yes
Replication Group	The default replication group for the namespace.	Yes
Namespace Quota	The storage space limit that is specified for the namespace. You can specify a storage limit for the namespace and define notification and access behavior when the quota is reached. The quota setting for a namespace cannot be less than 1 GB. You can select one of the following quota behavior options: **Notification Only at <*quota*>** Quota setting at which you are notified. **Block Access Only at <*quota*>** Hard quota which, when reached, prevents write/update access to the bucket.	Yes

Table 11 Namespace settings (continued)

Field	Description	Can be edited
	Block Access at <*quota*> and Send Notification at <*% of quota*> Hard quota which, when reached, prevents write/update access to the bucket and the percentage of the quota setting at which you are notified.	
Default Bucket Quota	The default storage limit that is specified for buckets created in this namespace. This is a hard quota which, when reached, prevents write/update access to the bucket. Changing the default bucket quota does not change the bucket quota for buckets that are already created.	Yes
Server-side Encryption	The default value for server-side encryption for buckets created in this namespace. Server-side encryption, also known as Data At Rest Encryption or D@RE, encrypts data inline before storing it on ECS disks or drives. This encryption helps prevent sensitive data from being acquired from discarded or stolen media. If the namespace enables encryption, then all its buckets are encrypted and this setting cannot be changed when a bucket is created. If you want the buckets in the namespace to be unencrypted, then the namespace must have encryption disabled. When encryption is disabled for the namespace, individual buckets can be set as encrypted when created. For a complete description of the feature, see the *ECS Security Guide*, available from the ECS Product Documentation page.	No
Access During Outage	The default behavior when accessing data in the buckets created in this namespace during a temporary site outage in a geo-federated setup. If you set this flag to Enabled, and a temporary site outage occurs, if you cannot access a bucket at the failed site where the bucket was created (owner site), you will be able to access a copy of the bucket at another site. Note that objects that you access in the buckets in the namespace might have been updated at the failed site, but changes might not have been propagated to the site from which you are accessing the object. If you set the flag to Disabled, data in the site which has the temporary outage is not available for access from other sites, and object reads for data that is owned by the failed site will fail. For more information, see TSO behavior with the ADO bucket property enabled on page 168.	Yes
Compliance	The rules that limit changes that can be made to retention settings on objects under retention. ECS has object retention features enabled or defined at the object level, bucket level, and namespace level. Compliance strengthens these features by limiting changes that can be made to retention settings on objects under retention. Compliance can be enabled only at the time the namespace is created, and it cannot be disabled after it has been enabled. Compliance is supported by S3 and CAS systems. For details about the rules enforced by compliance, see the *ECS Data Access Guide*, available from the ECS Product Documentation page.	No
Retention Policies	Enables one or more retention policies to be added and configured.	Yes

Table 11 Namespace settings (continued)

Field	Description	Can be edited
	A namespace can have one or more associated retention polices, where each policy defines a retention period. When you apply a retention policy to a number of objects, rather than to an individual object, a change to the retention policy changes the retention period for all the objects to which the policy is applied. A request to modify an object before the expiration of the retention period is disallowed. In addition to specifying a retention policy for a number of objects, you can specify retention policies and a quota for the entire namespace. For more information on retention, see Retention periods and policies on page 55.	
Domain	Enables Active Directory (AD) or Lightweight Directory Access Protocol (LDAP) domains to be specified and the rules for including users from the domain to be configured. Domain users can be assigned to ECS management roles and can use the ECS self-service capability to register as object users. The mapping of domain users into a namespace is described in Domain users require an assigned namespace to perform object user operations on page 66.	Yes

The following attribute can be set using the ECS Management REST API, not from the ECS Portal.

Allowed (and Disallowed) Replication Groups

Enables a client to specify the replication groups that the namespace can use.

Retention periods and policies

ECS provides the ability to prevent data from being modified or deleted within a specified retention period.

You can specify retention by using retention periods and retention policies that are defined in the metadata that is associated with objects and buckets. The retention periods and retention policies are checked each time a request to modify an object is made. Retention periods are supported on all ECS object protocols (S3, Swift, Atmos, and CAS).

Note

For detailed information about setting retention on object interfaces, including CAS retention and CAS advanced retention, see the *ECS Data Access Guide*, available from the ECS Product Documentation page.

Retention Periods

You can assign retention periods at the object level or the bucket level. Each time a user requests to modify or delete an object, an expiration time is calculated, where the object expiration time equals the object creation time plus the retention period. When you assign a retention period for a bucket, the object expiration time is calculated based on the retention period set on the object and the retention period set on the bucket, whichever is the longest.

When you apply a retention period to a bucket, the retention period for all objects in a bucket can be changed at any time, and can override the value written to the object by an object client by setting it to a longer period.

You can specify that an object is retained indefinitely.

Retention Policies

Retention polices are associated with a namespace. Any policy that is associated with the namespace can be assigned to an object belonging to the namespace. A retention policy has an associated retention period.

When you change the retention period that is associated with a policy, the retention period automatically changes for objects that have that policy assigned.

You can apply a retention policy to an object. When a user attempts to modify or delete an object, the retention policy is retrieved. The retention period in the retention policy is used with object retention period and bucket retention period to verify whether the request is allowed.

For example, you could define a retention policy for each of the following document types, and each policy could have an appropriate retention period. When a user requests to modify or delete the legal document four years after it was created, the larger of the bucket retention period or the object retention period is used to verify whether the operation can be performed. In this case, the request is not allowed, and the document cannot be modified or deleted for one more year.

- Email - six months
- Financial - three years
- Legal - five years

ECS Management REST API retention policy methods

The retention policy creation and configuration tasks that can be performed in the ECS Portal can also be performed using the ECS Management REST API. The following table describes the ECS Management REST API methods that relate to retention policies.

ECS Management REST API method	Description
PUT /object/bucket/{*bucketName*}/retention	The retention value for a bucket that defines a mandatory retention period which is applied to every object within a bucket. If the retention value is one year, an object from the bucket can not be modified or deleted for one year.
GET /object/bucket/{*bucketName*}/retention	Returns the retention period that is currently set for a specified bucket.
POST /object/namespaces/namespace/{*namespace*}/retention	The retention setting for namespaces that acts like a policy, where each policy is a <*name*>: <*retention period*> pair. You can define a number of retention policies for a namespace and you can assign a policy, by name, to an object within the namespace. This allows you to change the retention period for a set of objects that have the same policy assigned, by changing the corresponding policy.
PUT /object/namespaces/namespace/{*namespace*}/retention/{*class*}	Updates the period for a retention class that is associated with a namespace.
GET /object/namespaces/namespace/{*namespace*}/retention	Returns the retention classes that are defined for a namespace.

For information on how to access the ECS Management REST API, see the *ECS Data Access Guide*, available from the ECS Product Documentation page.

Working with namespaces in the ECS Portal

You can use the **Namespace Management** page available from **Manage** > **Namespace** to view the details of existing namespaces, to create new namespaces, to modify existing namespaces, and to delete namespaces.

Figure 12 Namespace management page

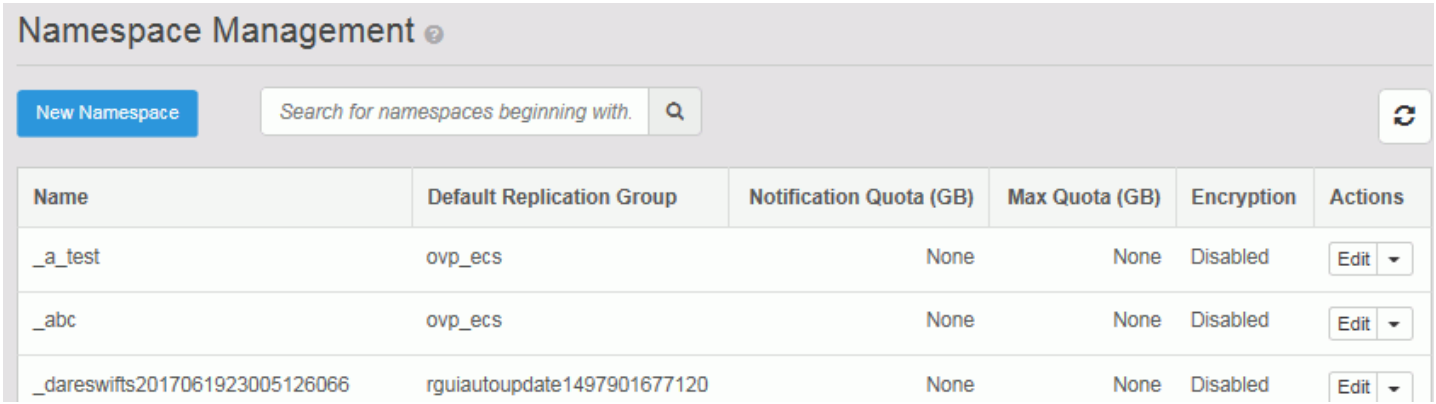


Table 12 Namespace properties

Field	Description
Name	The name of the namespace.
Default Replication Group	The default replication group for the namespace.
Notification Quota	The quota limit at which notification is generated.
Max Quota	The quota limit at which writes to the namespace are blocked.
Encryption	Specifies if D@RE server-side encryption is enabled for the namespace.
Actions	The actions that can be completed for the namespace. • **Edit**: Change the namespace name, the Namespace Administrator, the default replication group, namespace quota, bucket quota, server-side encryption, access during outage, and compliance settings for the namespace. • **Delete**: Delete the namespace.

Create a namespace

You can create a namespace.

Before you begin

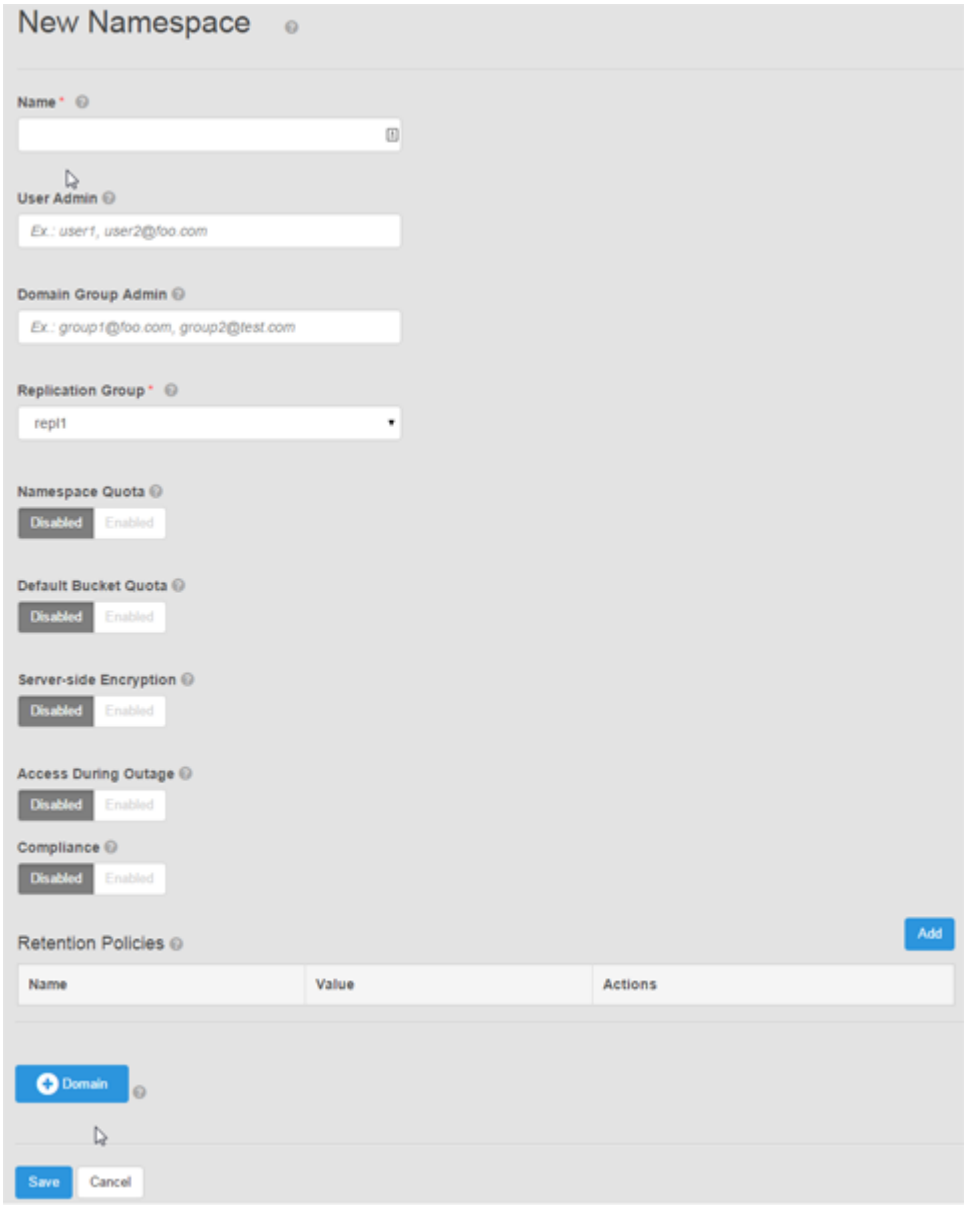
- This operation requires the System Administrator role in ECS.
- A replication group must exist. The replication group provides access to storage pools in which object data is stored.
- If you want to allow domain users to access the namespace, an authentication provider must be added to ECS. To configure domain object users or a domain group, you must plan how you want to map users into the namespace. For more

information on mapping users, see Domain users require an assigned namespace to perform object user operations on page 66.

For more information about namespaces, see Namespace settings on page 53.

Procedure

1. In the ECS Portal, select **Manage** > **Namespace**.
2. On the **Namespace Management** page, click **New Namespace**.



3. On the **New Namespace** page, in the **Name** field, type the name of the namespace.

   The name must contain only lowercase characters.

4. In the **User Admin** field, specify the user ID of one or more domain or local users to whom you want to assign the Namespace Administrator role. In the **Domain Group Admin** field, you can also add one or more domain groups to whom you want to assign the Namespace Administrator role.

   You can add multiple users or groups as comma separated lists.

5. In the **Replication Group** field, select the default replication group for this namespace.
6. In the **Namespace Quota** field, specify whether you want to enable a storage space limit for this namespace. If you enable a namespace quota, select one of the following quota behavior options:
   a. **Notification Only at <*quota*>**

      Select this option if you want to be notified when the quota setting is reached.

   b. **Block Access Only at <*quota*>**

      Select this option is you want write/update access to the buckets in this namespace blocked when the quota is reached.

   c. **Block Access at <*quota*> and Send Notification at <*% of quota*>**

      Select this option if you want write/update access to the buckets in this namespace blocked and you want to be notified when the quota reaches a specified percentage of the total quota.
7. In the **Default Bucket Quota** field, specify whether you want to enable a storage space limit for all buckets created in this namespace.
8. In the **Server-side Encryption** field, specify whether this namespace requires server-side encryption. When enabled, all buckets created n the namespace will have server-side encryption enabled and every object in the buckets will be encrypted. If you leave as **Disabled**, you can apply server-side encryption to individual buckets in the namespace at the time of creation.
9. In the **Access During Outage** field, specify the default behavior when accessing data in the buckets created in this namespace during a temporary site outage in a geo-federated setup.

   When enabled, if a temporary site outage occurs in a geo-federated system and you cannot access a bucket at the failed site where it was created (owner site), you will be able to access a copy of the bucket at another site.

   When disabled, data in the site which has the temporary outage is not available for access from other sites, and object reads for data that is owned by the failed site will fail.
10. In the **Compliance** field, specify whether you want to enable compliance features for objects in this namespace.

    Once this setting is enabled, it cannot be disabled.

    This setting can only be enabled during namespace creation.

    If you enable the Compliance setting, you can add a retention policy by completing the following steps:

    a. In the **Retention Policies** area, click **Add** to add a new policy.

    b. In the **Name** field, type the name of the policy.

    c. In the **Value** field, type the retention period for this retention policy and select the unit of measure (seconds, minutes, hours, days, months, years).

       Instead of specifying a specific retention period, you can select the **Infinite** checkbox to ensure that buckets assigned to this retention policy are never deleted.

11. To specify an Active Directory (AD) or Lightweight Directory Access Protocol (LDAP) domain that contains the users who can log in to ECS and perform administration tasks for the namespace, click **Domain**.
    a. In the **Domain** field, type the name of the domain.
    b. Specify the groups and attributes for the domain users that are allowed to access ECS in this namespace by typing the values in the **Groups**, **Attribute**, and **Values** fields.

    For information on how to perform complex mappings using groups and attributes, see Domain users require an assigned namespace to perform object user operations on page 66.
12. Click **Save**.

Edit a namespace

You can change the configuration of an existing namespace.

Before you begin

This operation requires the System Administrator role in ECS.

The Namespace Administrator role can modify the AD or LDAP domain that contains the users in the namespace that are object users or management users that can be assigned the Namespace Administrator role for the namespace.

You cannot edit the **Name**, **Server-side Encryption**, or **Compliance** fields after the namespace has been created.

Procedure

1. In the ECS Portal, select **Manage** > **Namespace**.
2. On the **Namespace Management** page, locate the namespace you want to edit in the table. Click **Edit** in the **Actions** column beside the namespace you want to edit.
3. On the **Edit Namespace** page:
    - To modify the domain or local users to whom you want to assign the Namespace Administrator role, in the **User Admin** or **Domain Group Admin** fields, change the user IDs.
    - To modify the default replication group for this namespace, in the **Replication Group** field, select a different replication group.
    - To modify which of the following features are enabled, click the appropriate **Enabled** or **Disabled** options.
        - **Namespace Quota**
        - **Default Bucket Quota**
        - **Access During Outage**
4. To modify an existing retention policy, in the **Retention Policies** area:
    a. Click **Edit** in the **Actions** column beside the retention policy you want to edit.
    b. To modify the policy name, in the **Name** field, type the new retention policy name.
    c. To modify the retention period, in the **Value** field, type the new retention period for this retention policy.

5. To modify the AD or LDAP domain that contains the object users in the namespace and management users that can be assigned the Namespace Administrator role for the namespace, click **Domain.**
   a. To modify the domain name, in the **Domain** field, type the new domain name.
   b. To modify the groups and attributes for the domain users that are allowed to access ECS in this namespace, type the new values in the **Groups**, **Attribute**, and **Values** fields.
6. Click **Save**.

Delete a namespace

You can delete a namespace, but you must delete the buckets in the namespace first.

Before you begin

This operation requires the System Administrator role in ECS.

Procedure

1. In the ECS Portal, select **Manage** > **Namespace**.
2. On the **Namespace Management** page, locate the namespace you want to delete in the table. Click **Delete** in the **Actions** column beside the namespace you want to delete.

   An alert displays informing you of the number of buckets in the namespace and instructs you to delete the buckets in the namespace before removing the namespace. Click **OK**.
3. Delete the buckets in the namespace.
   a. Select **Manage** > **Buckets**.
   b. On the **Bucket Management** page, locate the bucket you want to delete in the table. Click **Delete** in the **Actions** column beside the bucket you want to delete.
   c. Repeat step 4b. for all the buckets in the namespace.
4. On the **Namespace Management** page, locate the namespace you want to delete in the table. Click **Delete** in the **Actions** column beside the namespace you want to delete.

   Since there are no longer any buckets in this namespace, a message displays to confirm that you want to delete this namespace. Click **OK**.
5. Click **Save**.

CHAPTER 6

Users and Roles

Introduction to users and roles

In ECS you can configure users and roles to control access to the ECS management tasks and to the object store. Management users can perform administration tasks in the ECS Portal. Object users cannot access the ECS Portal but can access the object store using clients that support the ECS data access protocols.

Roles in ECS determine the operations that a management user can perform in the ECS Portal or by using the ECS Management REST API.

Management users and object users are stored in different tables and their credentials are different. Management users require a local username and password, or a link to a domain user account. Object users require a username and a secret key. You can create a management user and an object user with the same name, but they are effectively different users as their credentials are different.

Management user and object user names can be unique across the ECS system or can be unique within a namespace, which is referred to as user scope.

Local and domain users can be assigned as management users or object users. Local user credentials are stored by ECS. Domain users are users defined in an Active Directory AD/LDAP database and ECS must talk to the AD or LDAP server to authenticate user login request.

The following topics describe the types of users in ECS, the management roles that can be assigned to management users, and the user-related tasks that can be performed in the ECS Portal.

- Users in ECS on page 64
- Management roles in ECS on page 69
- Working with users in the ECS Portal on page 73

Users in ECS

ECS requires two types of user: management users, who can perform administration of ECS, and object users, who access the object store to read and write objects and buckets.

The following topics describe ECS user types and concepts.

- Management users on page 64
- Default management users on page 65
- Object users on page 65
- Domain and local users on page 66
- User scope on page 68
- User tags on page 68

Management users

Management users can perform the configuration and administration of the ECS system and of namespaces (tenants) configured in ECS.

The roles that can be assigned to management users are System Administrator, System Monitor, Namespace Administrator, and Lock Administrator as described in Management roles in ECS on page 69.

Management users can be local users or domain users. Management users that are local users are authenticated by ECS against the locally held credentials. Management

users that are domain users are authenticated in Active Directory (AD) or Lightweight Directory Access Protocol (LDAP) systems. For more information about domain and local users, see Domain and local users on page 66.

Management users are not replicated across geo-federated VDCs.

Default management users

On installation, ECS creates two default local management users, root and emcsecurity, to allow for the initial and ongoing configuration of ECS. The root and emcsecurity management users can access the ECS system by using the ECS Portal or the ECS Management REST API. These default users cannot be removed from the ECS system and they are not replicated across sites in a geo-federation. The following table describes the default management users.

Table 13 Default management users

Default user	Default password	User description	Role assigned to user	Role permissions	Can more users be created?
root	ChangeMe	This user performs the initial configuration of the ECS system, and creates users with System Administrator roles. The first time the root user accesses ECS, the user is prompted to change the password and immediately log in again with the new password. From an audit perspective, it is important to know which user carried out changes to the system, so the root user should not be used after system initialization. After system initialization, each System Administrator user should log into the system using their own credentials, not the root user credentials.	System Administrator	Create and delete management and object users Change user passwords Grant permissions to object users Create, delete, modify storage pools, namespaces, buckets, and replication groups View monitoring metrics	Yes
emcsecurity	ChangeMe	This user can prevent remote SSH access to nodes by locking them. The password for this user should be changed after system installation and securely recorded.	Lock Administrator	Lock and unlock nodes Change its own password	No

Object users

Object users are end-users of the ECS object store, and they access ECS through object clients that are using the object protocols that ECS supports (S3, EMC Atmos, Openstack Swift, and CAS). Object users can also be assigned Unix-style permissions to access buckets exported as file systems for HDFS.

A management user (System or Namespace Administrator) can create an object user. The management user defines a username and assigns a secret key to the object user when the user is created or at any time thereafter. A username can be a local name or a domain-style username that includes @ in the name. The object user uses the secret

key to access the ECS object store. The object user's secret key is distributed by email or other means.

Users that are added to ECS as domain users can subsequently add themselves as object users by creating their own secret key using the ECS self-service capability through a client that communicates with the ECS Management REST API. The object username that they are given is the same as their domain name. Object users do not have access to the ECS Portal. For more information about domain users, see the Domain and local users on page 66. For information on creating a secret key, see the *ECS Data Access Guide*, available from the ECS Product Documentation page.

Object users are global resources. An object user can have privileges to read and write buckets, and objects within the namespace to which they are assigned, from any VDC.

Domain and local users

ECS provides support for local user and domain users. Local and domain users can be assigned as management users or object users.

The ECS self-service capability authenticates domain users and allows domain users to create a secret key for themselves. When a domain user creates their own secret key, they become an object user in the ECS system. You can use AD and LDAP to give a large number of users from an existing user database access to the ECS object store (as object users), without creating each user individually.

Note

Domain users that are object users must be added (mapped) into a namespace. For more information, see Add domain users into a namespace on page 78

Local user credentials are stored by ECS. The credentials for object users are global resources and are available from all VDCs in ECS.

Domain users are defined in an Active Directory AD or LDAP database. Domain usernames are defined by using the `user@domain.com` format. Usernames without @ are authenticated against the local user database. ECS uses an authentication provider to supply the credentials to communicate with the AD or LDAP server to authenticate a domain user login request. Domain users assigned to management roles can be authenticated against their AD or LDAP credentials to allow them to access ECS and perform ECS administration operations.

Domain users require an assigned namespace to perform object user operations

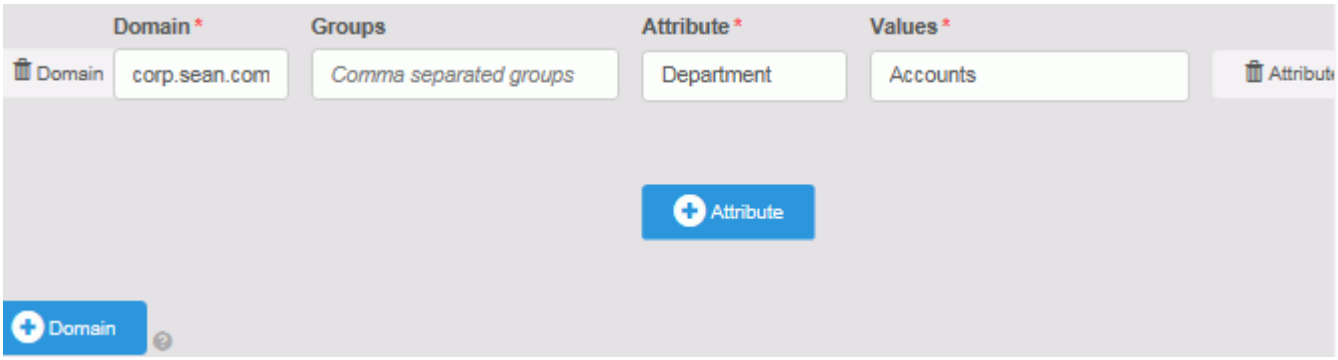
You must add (assign) domain users into a namespace if you want these users to perform ECS object user operations. To access the ECS object store, object users and Namespace Administrators must be assigned to a namespace. You can add an entire domain of users into a namespace, or you can add a subset of the domain users into a namespace by specifying a particular group or attribute associated with the domain.

A domain can provide users for multiple namespaces. For example, you might decide to add a set of users such as the Accounts department in the `corp.sean.com` domain into Namespace1, and a set of users such as the Finance department in the `corp.sean.com` domain into Namespace2. In this case, the `corp.sean.com` domain is providing users for two namespaces.

An entire domain, a particular set of users, or a particular user cannot be added into more than one namespace. For example, the `corp.sean.com` domain can be added into Namespace1, but the domain cannot also be added into Namespace2.

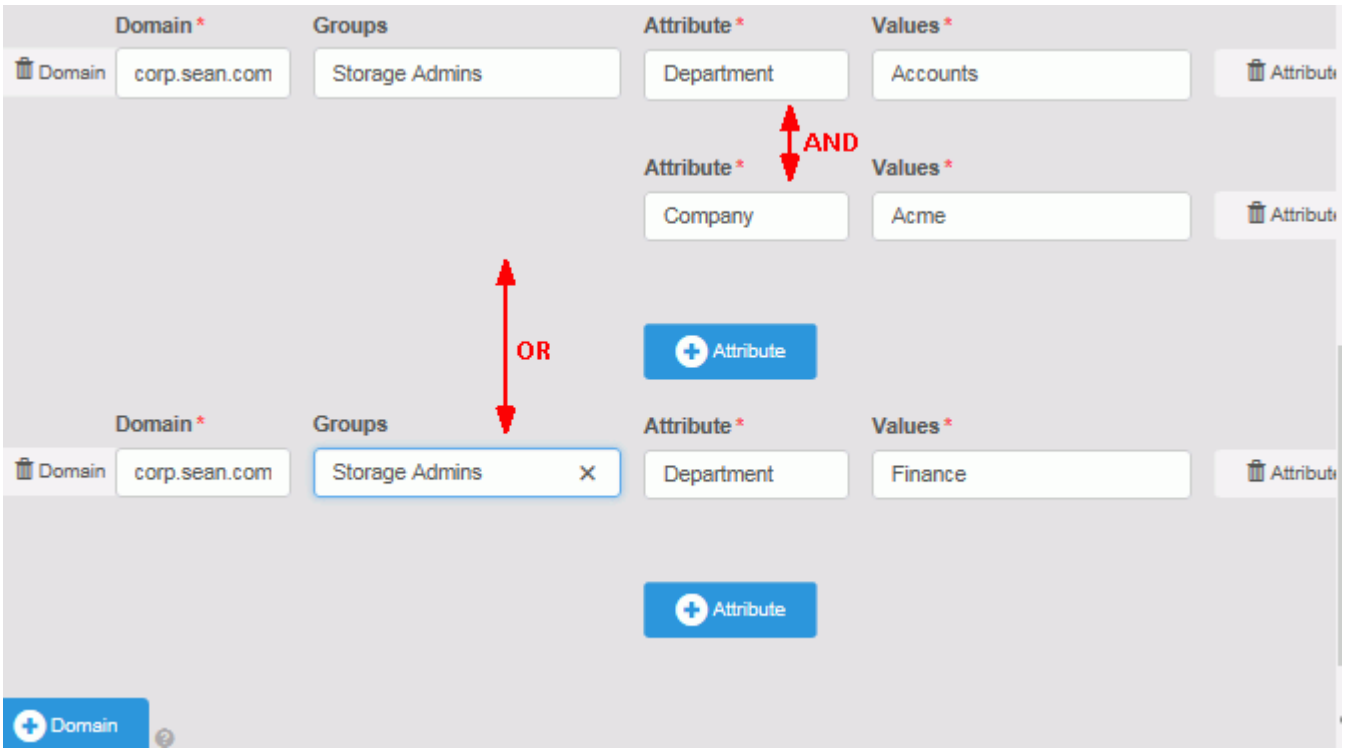
The following example shows that a System or Namespace Administrator has added into a namespace a subset of users in the `corp.sean.com` domain; the users that have their Department attribute = Accounts in Active Directory. The System or Namespace Administrator has added the users in the Accounts department from this domain into a namespace by using the **Edit Namespace** page in the ECS Portal.

Figure 13 Adding a subset of domain users into a namespace using one AD attribute



The following example shows a different example where the System or Namespace Administrator is using more granularity in adding users into a namespace. In this case, the System or Namespace Administrator has added the members in the `corp.sean.com` domain who belong to the Storage Admins group with the Department attribute = Accounts AND Company attribute = Acme, OR belong to the Storage Admins group with the Department attribute = Finance.

Figure 14 Adding a subset of domain users into a namespace using multiple AD attributes



For more information about adding domain users into namespaces using the ECS Portal, see Add domain users into a namespace on page 78.

User scope

The user scope setting affects all object users, in all namespaces across all federated VDCs.

The user scope can be `GLOBAL` or `NAMESPACE`. If the scope is set to `GLOBAL`, object user names are unique across all VDCs in the ECS system. If the scope is set to `NAMESPACE`, object user names are unique within a namespace, so the same object user names can exist in different namespaces.

The default setting is `GLOBAL`. If you intend to use ECS in a multitenant configuration and you want to ensure that namespaces can use names that are in use in another namespace, you must change this setting to `NAMESPACE`.

Note

You must set the user scope before you create the first object user.

Set the user scope

You can set the user scope using the ECS Management REST API.

Before you begin

This operation requires the System Administrator role in ECS.

If you are going to change the default user scope setting from `GLOBAL` to `NAMESPACE`, you must do so before you create the first object user in ECS.

The user scope setting affects all object users in ECS.

Procedure

1. In the ECS Management REST API, use the `PUT /config/object/properties` API call and pass the user scope in the payload.

   The following example shows a payload that sets the `user_scope` to `NAMESPACE`.

```
PUT /config/object/properties/

<property_update>
    <properties>
        <properties>
            <entry>
            <key>user_scope</key>
            <value>NAMESPACE</value>
            </entry>
    </properties>
</property_update>
```

User tags

A tag in the form of name=value pairs can be associated with the user ID for an object user, and retrieved by an application. For example, an object user can be associated with a project or cost-center. Tags cannot be associated with management users.

This functionality is not available from the ECS Portal. Tags can be set on an object user, and the tags associated with the object user can be retrieved by using the ECS Management REST API. You can add a maximum of 20 tags.

Management roles in ECS

ECS defines roles to determine the operations that a user can perform in the ECS Portal or when accessing ECS using the ECS Management REST API. Management users and groups can be assigned to administration roles in ECS and can be either local users or domain users. Roles can also be assigned to Active Directory group names.

The following management roles are defined:

- System Administrator on page 69
- System Monitor on page 69
- Namespace Administrator on page 69
- Lock Administrator on page 70

System Administrator

The System Administrator role allows a user to configure ECS and specify the storage used for the object store, how the store is replicated, how tenant access to the object store is configured (by defining namespaces), and which users have permissions within an assigned namespace. The System Administrator can also configure namespaces and perform namespace administration, or can assign a user who belongs to the namespace as the Namespace Administrator.

The System Administrator has access to the ECS Portal and system administration operations can also be performed from programmatic clients using the ECS Management REST API.

After initial installation of ECS, the System Administrator is a pre-provisioned local management user called `root`. The default root user is described in Default management users on page 65.

Because management users are not replicated across sites, a System Administrator must be created at each VDC that requires one.

System Monitor

The System Monitor role allows a user to have read-only access to the ECS Portal. The System Monitor can view all ECS Portal pages and all information on the pages, except user detail information such as passwords and secret key data. The System Monitor cannot provision or configure the ECS system. For example, the monitor cannot create or update storage pools, replication groups, namespaces, buckets, users and so on via the portal or ECS management API. Monitors cannot modify any other portal setting except their own passwords.

Because management users are not replicated across sites, a System Monitor must be created at each VDC that requires one.

Namespace Administrator

The Namespace Administrator is a management user who can access the ECS Portal and can assign local users as object users for the namespace and create and manage buckets within the namespace. Namespace Administrator operations can also be performed using the ECS Management REST API. A Namespace Administrator can only be the administrator of a single namespace.

Because authentication providers and namespaces are replicated across sites (they are ECS global resources), a domain user who is a Namespace Administrator can log in at any site and perform namespace administration from that site.

Note

If a domain user is to be assigned to the Namespace Administrator role, the user must be mapped into the namespace.

Local management users are not replicated across sites, so a local user who is a Namespace Administrator can only log in at the VDC at which the management user was created. If you want the same username to exist at another VDC, the user must be created at the other VDC. As they are different users, changes to a same-named user at one VDC, such as a password change, will not be propagated to the user with the same name at the other VDC.

Lock Administrator

The Lock Administrator is the only management user who can lock and unlock nodes from the ECS Portal or the ECS Management API. *Locking* a node is the ability to disable remote SSH access to the node. The Lock Administrator is a default local user called `emcsecurity`. The `emcsecurity` user is described in Default management users on page 65.

The Lock Administrator can only change their passwords and lock and unlock nodes. The Lock Administrator role cannot be assigned to another user. System Administrators and System Monitors can view the lock status of nodes. For instructions on locking and unlocking nodes, see Lock and unlock nodes using the ECS Portal on page 160.

Tasks performed by role

The tasks that can be performed in the ECS Portal or ECS Management REST API by each role are described in the following table.

Table 14 Tasks performed by ECS management user role

Task	System Admin	System Monitor	Namespace Admin	Lock Admin
Tenancy				
Create namespaces (tenants)	Yes	No	No	No
Delete namespaces	Yes	No	No	No
User management (management and object users unless otherwise noted)				
Create local object users and assign them to namespaces	Yes (in all namespaces)	No	Yes (in one namespace)	No
Create local management users and assign them to namespaces	Yes (in all namespaces)	No	No	No
Delete local object users	Yes (in all namespaces)	No	Yes (in one namespace)	No
Delete local management users	Yes (in all namespaces)	No	No	No

Table 14 Tasks performed by ECS management user role (continued)

Task	System Admin	System Monitor	Namespace Admin	Lock Admin
Set user scope (global or namespace) for all object users	Yes	No	No	No
Add an AD, LDAP, or Keystone authentication provider	Yes (in all namespaces)	No	No	No
Delete an AD, LDAP, or Keystone authentication provider	Yes (in all namespaces)	No	No	No
Add AD and LDAP domain users or AD groups into a namespace	Yes (in all namespaces)	No	Yes (in one namespace)	No
Create an AD group (LDAP and Keystone groups are not supported)	Yes (in all namespaces)	No	No	No
Delete domain users or AD groups	Yes (in all namespaces)	No	No	No
Role management				
Assign administration roles to local and domain management users and AD groups	Yes (in all namespaces)	No	No	No
Revoke roles from local and domain users and AD groups	Yes (in all namespaces)	No	No	No
Storage configuration				
Create, modify, storage pools	Yes (in the VDC where the System Admin was created)	No	No	No
Create, modify, delete VDCs	Yes (in the VDC where the System Admin was created)	No	No	No
Create, modify replication groups	Yes (in the VDC where the System Admin was created)	No	No	No
Create, modify, delete buckets	Yes (in all namespaces)	No	Yes (in one namespace)	No
Set the bucket ACL permissions for a user	Yes (buckets in all namespaces)	No	Yes (buckets in one namespace)	No
Create, modify, delete NFS exports	Yes (buckets in all namespaces)	No	Yes (buckets in one namespace)	No
Create, modify, delete mapping of users and groups to files and objects in buckets	Yes (buckets in all namespaces)	No	Yes (buckets in one namespace)	No
Add, modify, delete the object Base URL to use ECS object storage for Amazon S3 applications	Yes (in all namespaces)	No	No	No
Monitoring and reports				

Table 14 Tasks performed by ECS management user role (continued)

Task	System Admin	System Monitor	Namespace Admin	Lock Admin
Get metering information for each namespace and bucket	Yes (in all namespaces)	Yes (in all namespaces)	Yes (in one namespace)	No
Get audit information (list of all activity of users using the ECS Portal and ECS Management API)	Yes (in all namespaces)	Yes (in all namespaces)	No	No
View alerts and perform alert actions (such as acknowledging or assigning alerts)	Yes (in all namespaces)	Yes (in all namespaces)	No	No
Configure alerts	Yes (in all namespaces)	No	No	No
Monitor capacity utilization of storage pools, nodes, disks, and the entire VDC	Yes (in all namespaces)	Yes (in all namespaces)	No	No
Monitor the health and utilization of the infrastructure environment (nodes, disks, NIC bandwidth, CPU, and memory utilization)	Yes (in all namespaces)	Yes (in all namespaces)	No	No
Monitor requests and network performance for VDCs and nodes	Yes (in all namespaces)	Yes (in all namespaces)	No	No
Monitor status of data erasure encoding for each storage pool	Yes (in all namespaces)	Yes (in all namespaces)	No	No
Monitor recovery status of storage pools after an outage or failure (data rebuilding process)	Yes (in all namespaces)	Yes (in all namespaces)	No	No
Monitor disk use metrics at the VDC or individual node level	Yes (in all namespaces)	Yes (in all namespaces)	No	No
Monitor geo-replication metrics including network traffic, data pending replication and XOR, failover and bootstrapping processing status	Yes (in all namespaces)	Yes (in all namespaces)	No	No
Monitor information on cloud-hosted VDCs and cloud replication traffic	Yes (in all namespaces)	Yes (in all namespaces)	No	No
Licensing, ESRS, security, and event configuration				
View license and subscription information for all components	Yes	Yes	No	No
Procure and apply new licenses	Yes	No	No	No
Add, modify, delete EMC Secure Remote Services (ESRS) server	Yes	No	No	No
Change password	Yes	Yes	Yes	Yes

Table 14 Tasks performed by ECS management user role (continued)

Task	System Admin	System Monitor	Namespace Admin	Lock Admin
Lock nodes to prevent remote access through SSH	No	No	No	Yes
Add or delete an SNMP trap recipient to forward ECS events	Yes	No	No	No
Add or delete a syslog server to remotely store ECS logging messages	Yes	No	No	No

Working with users in the ECS Portal

You can use the **User Management** page available from **Manage** > **Users** to create local users assigned as object users for a namespace. You can also create management users, which can be new local users to whom you assign management roles, or domain users to whom you assign management roles.

The **User Management** page has two tabs: the **Object Users** tab and the **Management Users** tab.

Object Users tab

You can use the **Object Users** tab to view the details of object users, to edit object users, and to delete object users. The object users listed on this page include:

- The local object users created by a System or Namespace Administrator in the ECS Portal.
- The domain users that have become object users by way of obtaining a secret key using a client that communicates with the ECS Management REST API.

A System Administrator sees the object users for all namespaces. A Namespace Administrator sees only the object users in their namespace.

Figure 15 User Management page

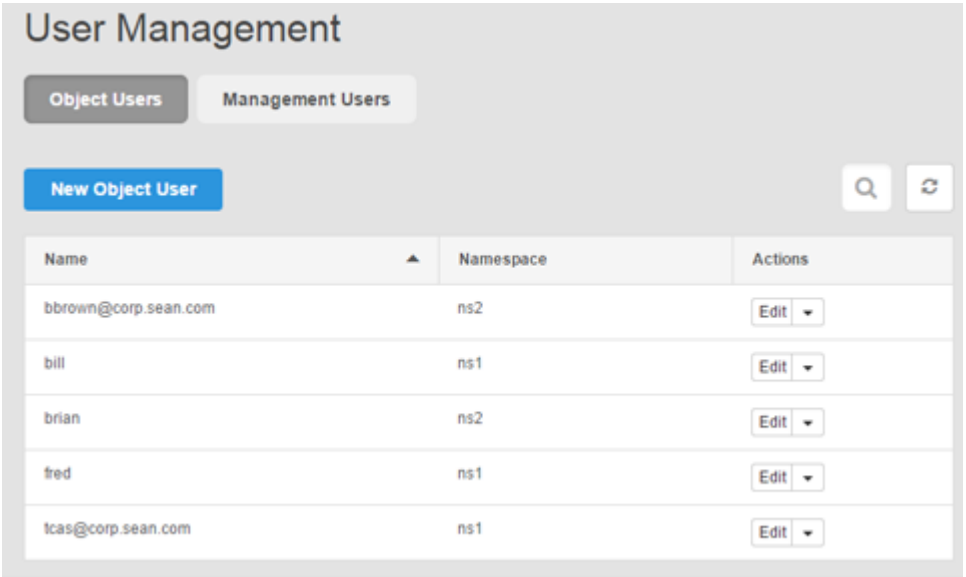


Table 15 Object user properties

Attribute	Description
Name	The name of the object user.
Namespace	The namespace to which the object user is assigned.
Actions	The actions that can be completed for the object user. • **Edit**: Change the object user's name, the namespace to which the user is assigned, or the S3, Swift, or CAS object access passwords for the user. • **Delete**: Delete the object user. • **New Object User** button: Add a new object user.

Management Users tab

You can use the **Management Users** tab to view the details of local and domain management users, to edit management users, and to delete management users. This tab is only visible to System Administrators.

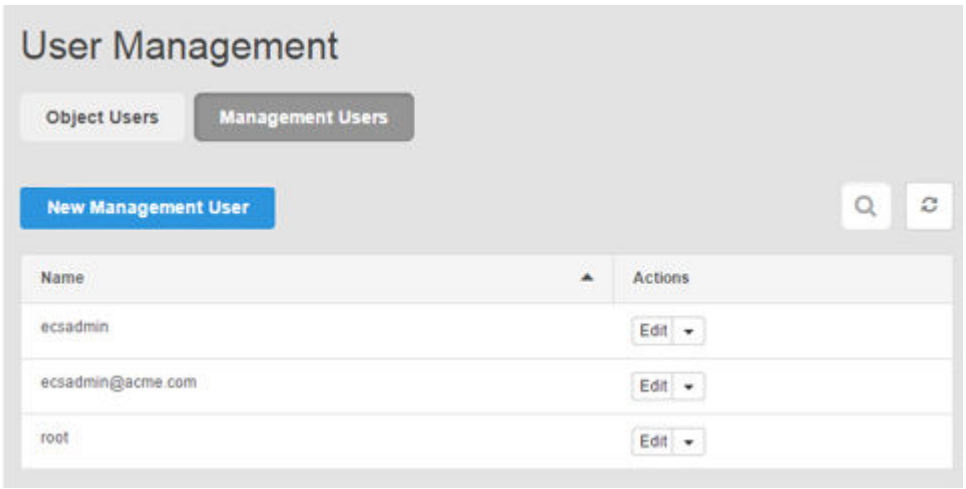


Table 16 Management user properties

Column	Description
Name	The name of the management user.
Actions	The actions that can be completed for the management user. • **Edit**: For a local management user: Change the user's name, password, and System Administrator or System Monitor role assignment. For a domain management user: Change the AD or LDAP username or AD group name, and System Administrator or System Monitor role assignment. • **Delete**: Delete the management user. • **New Management User** button: Add a new management user that can be assigned the System Administrator role or the System Monitor role.

Add an object user

You can create object users and configure them to use the supported object access protocols. You can edit an object user configuration by adding or removing access to an object protocol, or by creating a new secret key for the object user.

Before you begin

- This operation requires the System Administrator or Namespace Administrator role in ECS.
- A System Administrator can assign new object users into any namespace.
- A Namespace Administrator can assign new object users into the namespace in which they are the administrator.
- If you create an object user who will access the ECS object store through the OpenStack Swift object protocol, the Swift user must belong to an OpenStack group. A group is a collection of Swift users that have been assigned a role by an OpenStack administrator. Swift users that belong to the `admin` group can perform all operations on Swift buckets (containers) in the namespace to which they belong. You should not add ordinary Swift users to the `admin` group. For Swift users that belong to any group other than the `admin` group, authorization depends on the permissions that are set on the Swift bucket. You can assign permissions on the bucket from the OpenStack Dashboard UI or in the ECS Portal using the Custom Group ACL for the bucket. For more information, see Set custom group bucket ACLs on page 93.

Procedure

1. In the ECS Portal, select **Manage** > **Users**.
2. On the **User Management** page, click **New Object User**.
3. On the **New Object User** sub-page, in the **Name** field, type a name for the local object user.

   You can type domain-style names that include @ (for example, `user@domain.com`). You might want to do this to keep names unique and consistent with AD names. However, note that local object users are authenticated using a secret key assigned to the username, not through AD or LDAP.

   Note

   User names can include uppercase letters, lowercase letters, numbers, and any of the following characters: ! # $ & ' ( ) * + , - . / : ; = ? @ _ ~

4. In the **Namespace** field, select the namespace that you want to assign the object user to, and then complete one of the following steps.
   - To add the object user, and return later to specify passwords or secret keys to access the ECS object protocols, click **Save**.
   - To specify passwords or secret keys to access the ECS object protocols, click **Next to Add Passwords**.
5. On the **Update Passwords for User <*username*>** sub-page, in the **Object Access** area, for each of the protocols that you want the user to use to access the ECS object store, type or generate a key for use in accessing the S3, Swift, or CAS interfaces.

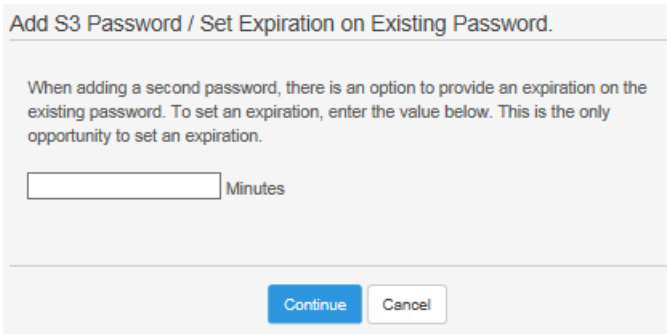
a. For S3 access, in the **S3** box, click **Generate & Add Password**.

The password (secret key) is generated.

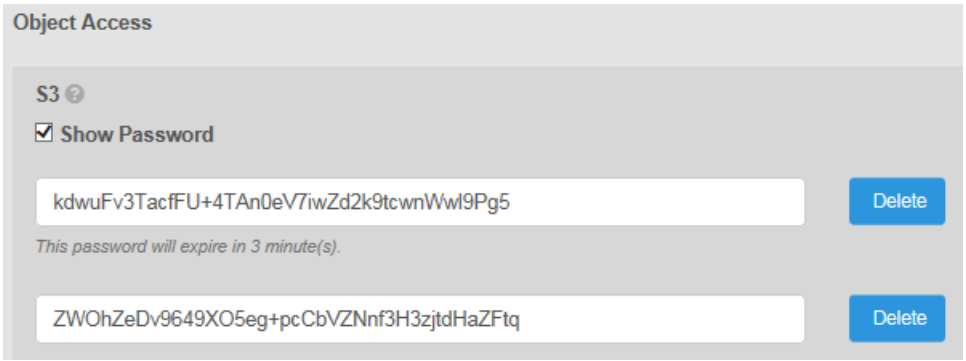
To view the password in plain text, select the **Show Password** checkbox.

To create a second password to replace first password for security reasons, click **Generate & Add Password**.

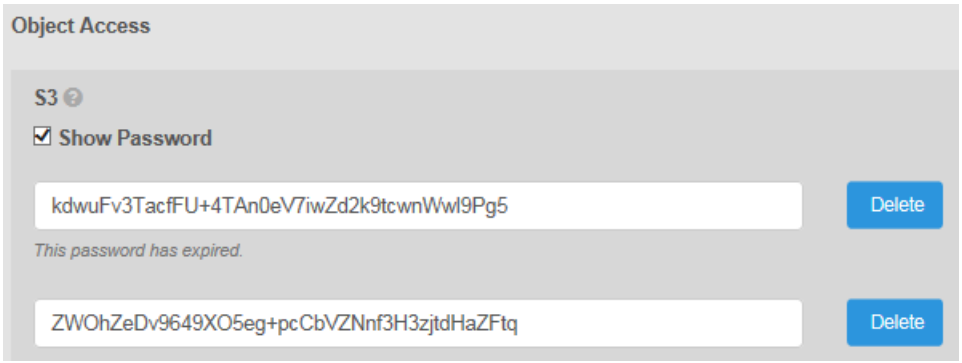
The **Add S3 Password/Set Expiration on Existing Password** dialog appears. When adding a second password, you can specify for how long to retain the first password. Once this time has expired, the first password will expire.



In the **Minutes** field, type the number of minutes for which you want to retain the first password before it expires. For example, if you typed 3 minutes, you would see the following in the portal:



After 3 minutes, you would see that the first password displays as expired and you could then delete it.



b. For Swift access:

- In the **Groups** field, type the OpenStack group to which the user belongs.
- In the **Swift password** field, type the OpenStack Swift password for the user.
- Click **Set Password & Groups**.

If you want an S3 user to be able to access Swift buckets, you must add a Swift password and group for the user. The S3 user is authenticated by using the S3 secret key, and the Swift group membership enables access to Swift buckets.

c. For CAS access:

- In the **CAS** box, type the password and click **Set Password** or click **Generate** to automatically generate the password and click **Set Password**.
- Click **Generate PEA file** to generate a Pool Entry Authorization (PEA) file. The file output displays in the **PEA file** box and the output is similar to the following example. The PEA file provides authentication information to CAS before CAS grants access to ECS; this information includes the username and secret. The secret is the base64-encoded password used to authenticate the ECS application.

```
<pea version="1.0.0">
<defaultkey name="s3user4">
<credential id="csp1.secret" enc="base64">WlFOOTlTZUFSaUl3Mlg3VnZaQ0k=</
credential>
</defaultkey>
<key type="cluster" id="93b8729a-3610-33e2-9a38-8206a58f6514" name="s3user4">
<credential id="csp1.secret" enc="base64">WlFOOTlTZUFSaUl3Mlg3VnZaQ0k=</
credential>
</key>
</pea>
```

- In the **Default Bucket** field, select a bucket, and click **Set Bucket**.
- Optional. Click **Add Attribute** and type values in the **Attribute** and **Group** fields.
- Click **Save Metadata**.

6. Click **Close**.

The passwords/secret keys are saved automatically.

Add a domain user as an object user

You can configure domain users so that they can access the ECS object store and generate secret keys for themselves. By doing so, they add themselves as object users to ECS.

Before you begin

- AD or LDAP domain users must have been added to ECS through an AD or LDAP authentication provider. Adding an authentication provider must be performed by a System Administrator and is described in Add an AD or LDAP authentication provider on page 46.
- Domain users must have been added into a namespace by a System or Namespace Administrator as described in Add domain users into a namespace on page 78.

Procedure

1. Domain users can create secret keys for themselves by using the instructions in the *ECS Data Access Guide*, available from the ECS Product Documentation page.

   When a domain user creates their own secret key, they become an object user in the ECS system.

Add domain users into a namespace

In the ECS Portal, you can add domain users into a namespace based on the AD or LDAP domain, groups, and attributes associated with the users. Domain users must be added (mapped) into a namespace in order to perform ECS object user operations.

Before you begin

- This operation requires the System Administrator or Namespace Administrator role in ECS.
- An authentication provider must exist in the ECS system that provides access to the domain that includes the users you want to add into the namespace.

Procedure

1. In the ECS Portal, select **Manage** > **Namespace**.
2. On the **Namespace Management** page, beside the namespace, click **Edit**.
3. On the **Edit Namespace** page, click **Domain** and type the name of the domain in the **Domain** field.
4. In the **Groups** field, type the names of the groups that you want to use to add users into the namespace.

   The groups that you specify must exist in AD.

5. In the **Attribute** and **Values** fields, type the name of the attribute and the values for the attribute.

   The specified attribute values for the users must match the attribute values specified in AD or LDAP.

   If you do not want to use attributes to add users into the namespace, click the **Attribute** button with the trash can icon to remove the attribute fields.

6. Click **Save**.

Create a local management user or assign a domain user or AD group to a management role

You can create a local management user, and you can assign a management role to a local user, a domain user, or an AD group. Management users can perform system-level administration (VDC administration) and namespace administration. You can also remove the management role assignment.

Before you begin

- This operation requires the System Administrator or Namespace Administrator role in ECS.
- By default, the ECS root user is assigned the System Administrator role and can perform the initial assignment of a user to the System Administrator role.
- To assign a domain user or an AD group to a management role, the domain users or AD group must have been added to ECS through an authentication provider. Adding an authentication provider must be performed by a System Administrator and is described in Add an AD or LDAP authentication provider on page 46.
- To assign the Namespace Administrator role to a management user, you must create a management user using the following procedure and perform the role assignment on the **Edit Namespace** page in the ECS Portal (see Assign the Namespace Administrator role to a user or AD group on page 80). The user cannot log in until the Namespace Administrator role is assigned.

Procedure

1. In the ECS Portal, select **Manage** > **Users**.
2. On the **User Management** page, click the **Management Users** tab.
3. Click **New Management User**.
4. Click **AD/LDAP User or AD Group** or **Local User**.
   - For a domain user, in the **Username** field, type the name of the user. The username and password that ECS uses to authenticate a user are held in AD or LDAP, so you do not need to define a password.
   - For an AD group, in the **Group Name** field, type the name of the group. The username and password that ECS uses to authenticate the AD group are held in AD, so you do not need to define a password.
   - For a local user, in the **Name** field, type the name of the user and in the **Password** field, type the password for the user.

   Note

   User names can include uppercase letters, lowercase letters, numbers and any of the following characters: ! # $ & ' ( ) * + , - . / : ; = ? @ _ ~
5. To assign the System Administrator role to the user or AD group, in the **System Administrator** box, click **Yes**.

   If you select **Yes**, but at a later date you want to remove System Administrator privileges from the user, you can edit this setting and select **No**.
6. To assign the System Monitor role to the user or AD group, in the **System Monitor** box, click **Yes**.
7. Click **Save**.

Assign the Namespace Administrator role to a user or AD group

You can assign the Namespace Administrator role to a local management user, a domain user, or AD group that exists in the ECS system.

Before you begin

- This operation requires the System Administrator role in ECS.

Procedure

1. In the ECS Portal, select **Manage** > **Namespace**.
2. On the **Namespace Management** page, beside the namespace into which you want to assign the Namespace Administrator, click **Edit**.
3. On the **Edit Namespace** page:
   a. For a local management user or a domain user, in the **User Admin** field, type the name of the user to whom you want to assign the Namespace Administrator role.

      To add more than one Namespace Administrator, separate the names with commas.

      A user can be assigned as the Namespace Administrator only for a single namespace.

   b. For an AD group, in the **Domain Group Admin** field, type the name of the AD group to which you want to assign the Namespace Administrator role.

      When the AD group is assigned the Namespace Administrator role, all users in the group are assigned this role.

      An AD group can be the Namespace Administrator only for one namespace,
4. Click **Save**.

CHAPTER 7

Buckets

Introduction to buckets

Containers are used to control access to objects and to set properties that define attributes for all contained objects, such as retention periods and quotas.

The main concepts relating to buckets are described in the following topics:

- Bucket ownership
- Bucket access
- Bucket settings
- Bucket and key naming conventions

This section also describes how to create and edit buckets, as well as assign Access Control List (ACL) permissions to buckets using the ECS Portal:

- Create a bucket
- Edit a bucket
- Bucket policy editor
- Set ACLs

In addition, this section discusses how to Create a bucket using the S3 API (with s3curl) on page 99.

In S3, object containers are called *buckets* and this term has been adopted as a general term in ECS. In Atmos, the equivalent of a bucket is a *subtenant*. In Swift, the equivalent of a bucket is a *container*. In CAS, a bucket is a *CAS pool*.

In ECS, buckets are assigned a type, which can be S3, Swift, Atmos, or CAS. S3, Atmos, or Swift buckets can be configured to support file system access (for NFS and HDFS). A bucket that is configured for file system access can be read and written by using its object protocol and by using the NFS or HDFS protocol. S3 and Swift buckets can alse be accessed using each other's protocol. Accessing a bucket using more than one protocol is often referred to as *cross-head support*.

You can create buckets for each object protocol using its API, usually using a client that supports the appropriate protocol. You can also create S3, File system-enabled (NFS or HDFS), and CAS buckets using the ECS Portal and the ECS Management API.

Bucket ownership

A bucket is assigned to a namespace and object users are also assigned to a namespace. An object user can create buckets only in the namespace to which the object user is assigned. An ECS System or Namespace Administrator can assign the object user as the owner of a bucket, or a grantee in a bucket ACL, even if the user does not belong to the same namespace as the bucket, so that buckets can be shared between users in different namespaces. For example, in an organization where a namespace is a department, a bucket can be shared between users in different departments.

Bucket access

Access to a bucket depends on the replication group that the bucket is associated with.

Objects in a bucket that belongs to a replication group which spans multiple VDCs can be accessed from all of the VDCs in the replication group. Objects in a bucket that

belongs to a replication group that is associated with only one VDC can be accessed from only that VDC. Buckets cannot be accessed or listed from other VDCs that are not in the replication group. However, because the identity of a bucket and its metadata, such as its ACL, are global management information in ECS, and the global management information is replicated across the system storage pools, the existence of the bucket can be seen from all VDCs in the federation.

For information on how objects in buckets can be accessed during site outages, see TSO behavior with the ADO bucket property enabled on page 168.

Bucket settings

The attributes associated with a bucket are described in the following table.

Table 17 Bucket attributes

Attribute	Description	Can be Edited
Name	Name of the bucket. For information on bucket naming, see Bucket, object, and namespace naming conventions on page 102.	No
Namespace	Namespace with which the bucket is associated.	No
Replication Group	Replication group in which the bucket is created.	No
Bucket Owner	Bucket owner.	Yes
Bucket Tagging	Name-value pairs that are defined for a bucket and enable buckets to be classified. For more information on bucket tagging, see Bucket tagging on page 87.	Yes
Quota	The storage space limit that is specified for the bucket. You can specify a storage limit for the bucket and define notification and access behavior when the quota is reached. The quota setting for a bucket cannot be less than 1 GB. You can select one of the following quota behavior options: **Notification Only at <*quota*>** Quota setting at which you are notified. **Block Access Only at <*quota*>** Hard quota which, when reached, prevents write/update access to the bucket. **Block Access at <*quota*> and Send Notification at <*% of quota*>** Hard quota which, when reached, prevents write/update access to the bucket and the percentage of the quota setting at which you are notified. **Note** Quota enforcement depends on the usage reported by ECS metering. Metering is a background process designed so that it does not impact foreground traffic and so the metered value can lag the actual usage. Because of the metering lag, there can be a delay in the enforcement of quotas.	Yes
Server-side Encryption	Indicates whether server-side encryption is enabled or disabled. Server-side encryption, also known as Data At Rest Encryption or D@RE, encrypts data inline before storing it on ECS disks or drives. This encryption helps prevent	No

Table 17 Bucket attributes (continued)

Attribute	Description	Can be Edited
	sensitive data from being acquired from discarded or stolen media. If you enable encryption when the bucket is created, this feature cannot be disabled later. If the bucket's namespace is encrypted, then every bucket is encrypted. If the namespace is not encrypted, you can select encryption for individual buckets. For a complete description of the feature, see the *ECS Security Guide*, available from the ECS Product Documentation page.	
File System	Indicates whether the bucket can be used as a file system (NFS export or HDFS). To simplify access to the file system, a default group, and default permissions associated with the group, can be defined. For more information, see Default group on page 85. **Note** File system enabled buckets only support the / delimiter when listing objects.	No
CAS	Indicates whether the bucket is enabled or disabled for CAS data.	No
Metadata Search	Indicates that metadata search indexes is created for the bucket based on specified key values. If Enabled, metadata keys that are used as the basis for indexing objects in the bucket can be defined. These keys must be specified at bucket create time. After the bucket is created, search can be disabled altogether, but the configured index keys cannot be modified. The way the attribute is defined is described in Metadata index keys on page 85. **Note** From ECS 3.1, metadata used for indexing is not encrypted, so metadata search can still be used on a bucket when Server-side Encryption (D@RE) is enabled.	No
Access During Outage	The default behavior when accessing data in the bucket during a temporary site outage in a geo-federated setup. If you set this flag to Enabled, and a temporary site outage occurs, if you cannot access a bucket at the failed site where the bucket was created (owner site), you will be able to access a copy of the bucket at another site. Note that objects that you access in the buckets in the namespace might have been updated at the failed site, but changes might not have been propagated to the site from which you are accessing the object. If you set the flag to Disabled, data in the site which has the temporary outage is not available for access from other sites, and object reads for data that is owned by the failed site will fail. For more information, see TSO behavior with the ADO bucket property enabled on page 168.	Yes
Read-Only Access During Outage	Specifies whether a bucket that is enabled for Access During Outage is accessible as read-only or read-write. If you enable Read-Only Access During Outage, the bucket is only accessible in read-only mode during the outage.	No
Bucket Retention	Retention period for a bucket.	Yes

Table 17 Bucket attributes (continued)

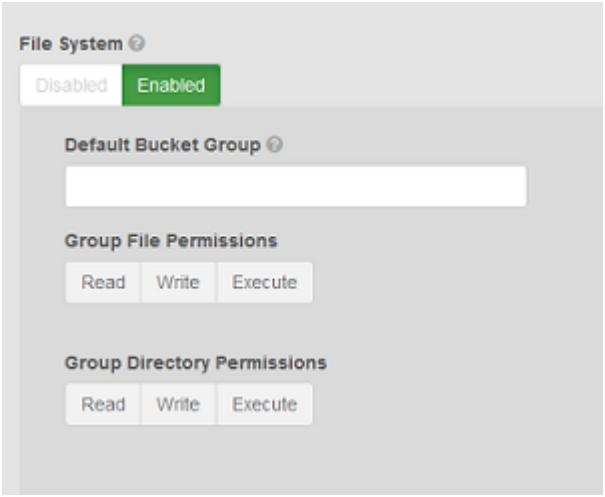
Attribute	Description	Can be Edited
	The expiration of a retention period on an object within a bucket is calculated when a request to modify an object is made and is based on the value set on the bucket and the objects themselves. The retention period can be changed during the lifetime of the bucket. You can find more information on retention and applying retention periods and policies in Retention periods and policies on page 55.	

Default group

Where you enable a bucket for file system access, you can assign a default group for the bucket. The default group is a UNIX group, the members of which have permissions on the bucket when it is accessed as a file system. Without this assignment, only the bucket owner can access the file system.

You can also specify UNIX permissions that are applied to files and directories created using object protocols so that they are accessible when the bucket is accessed as a file system.

The following screenshot shows the File System Enabled dialog that enables the default group and file and directory permissions to be set.



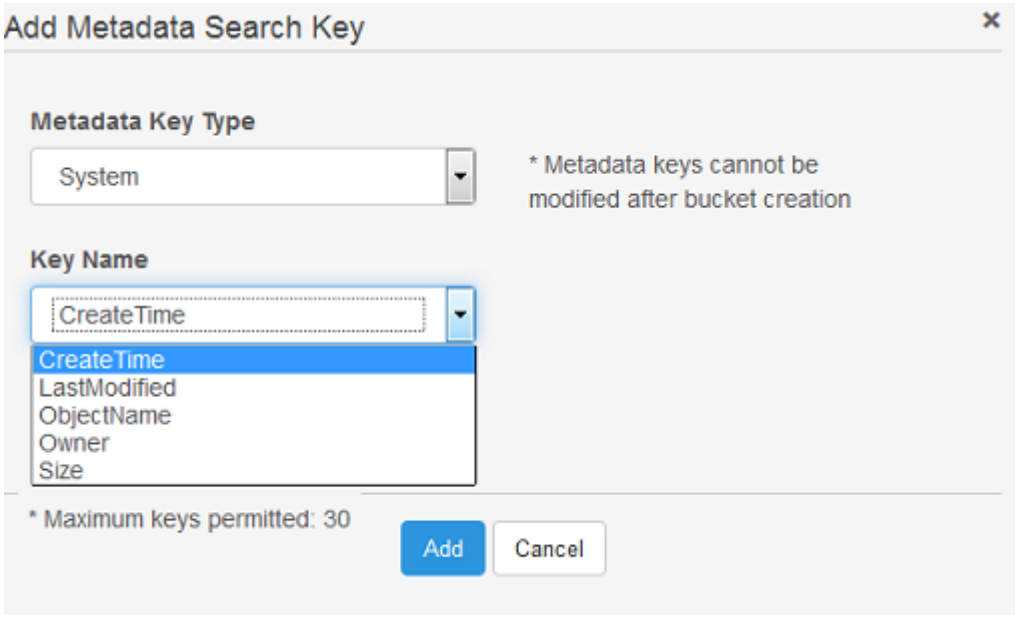
Metadata index keys

When Metadata Search is enabled for a bucket, objects in the bucket can be indexed based on their metadata fields. S3 object clients can search for objects based on the indexed metadata using a rich query language.

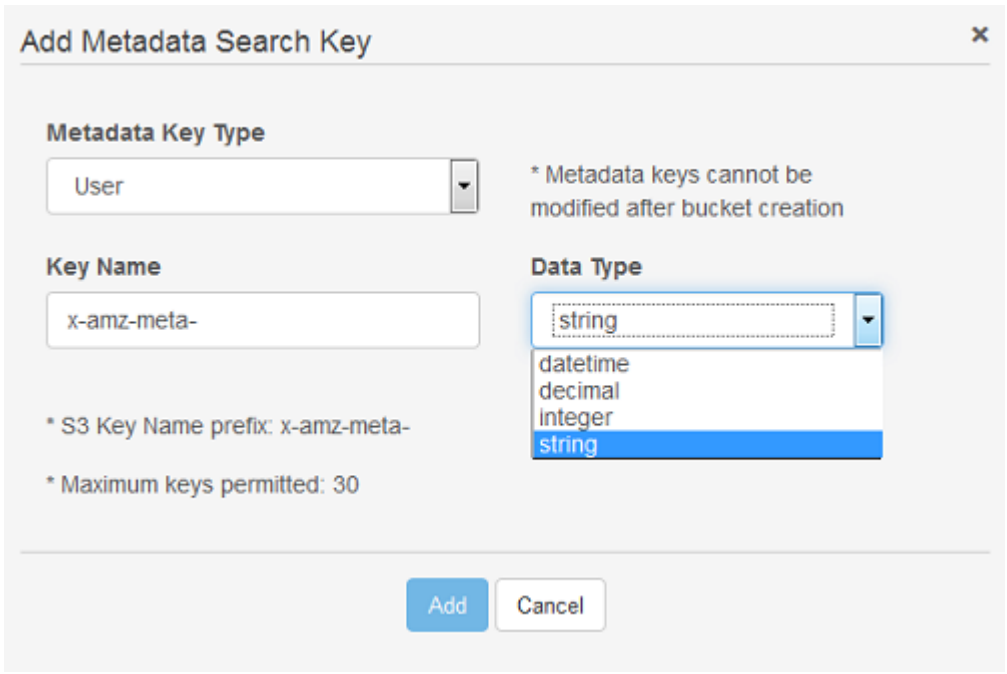
Each object has a set of system metadata that is automatically assigned, and can also have user assigned metadata. Both system and user metadata can be included in the index and used as the subject of metadata searches.

The **Add Metadata Search Key** dialog enables the metadata search key to be selected as either System or User. When you select a Metadata Key Type of System, metadata that is automatically assigned to objects in a bucket is listed for selection in the Key

Names menu. The dialog with the System key type selected is shown in the following screenshot.



When you select a **Metadata Key Type** of User, you must specify the name of the user metadata to create an index for. You must also specify its data type so that ECS knows how to interpret the metadata values provided in search queries. The dialog with the User key type selected is shown in the following screenshot.



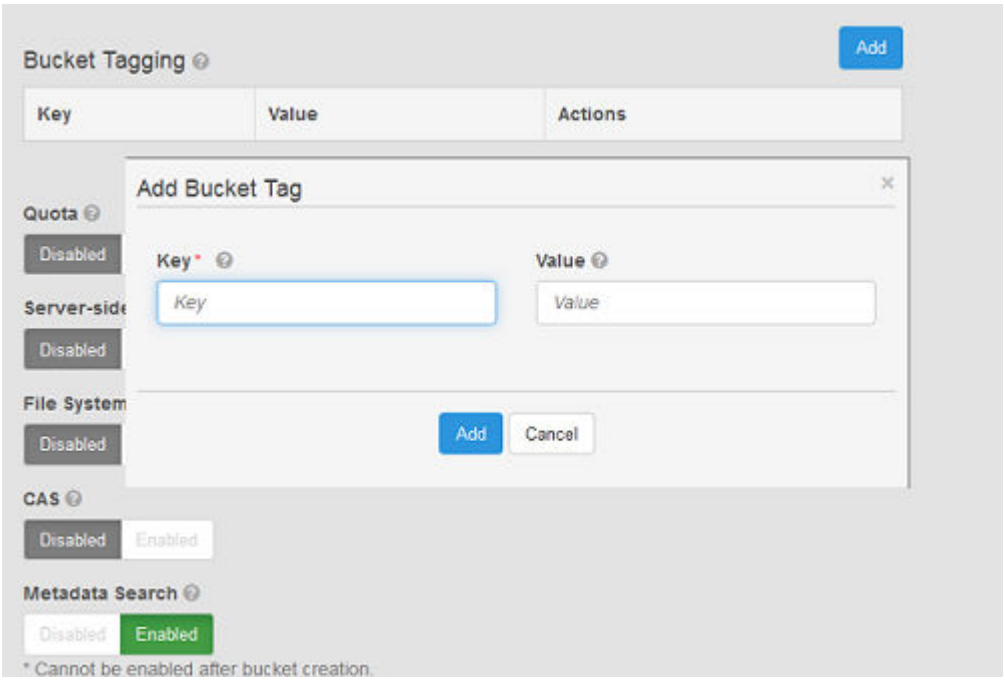
You can read more about the metadata search feature in the *ECS Data Access Guide*, available from the ECS Product Documentation page.

Bucket tagging

You can assign tags to a bucket to enable the object data in the bucket to be categorized. Using tags, you can associate a bucket with a cost-center or project, for example. Tags are in the form of name-value pairs.

You can assign bucket tags and values using the ECS Portal or using a custom client through the ECS Management REST API. Bucket tags are included in the metering data reports displayed in the ECS Portal or retrieved using the ECS Management REST API.

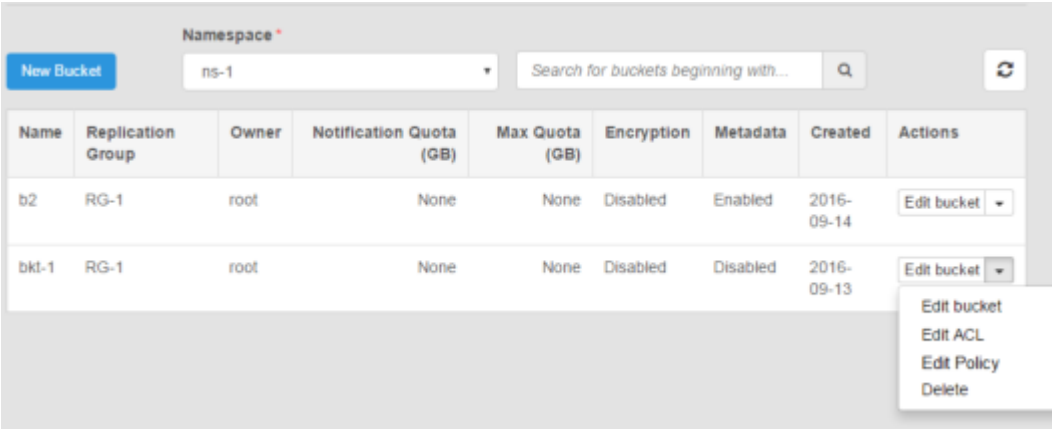
The following screenshot shows the **Bucket Tagging** dialog.



Working with buckets in the ECS Portal

You can use the **Bucket Management** page available from **Manage** > **Buckets** to view the details of existing buckets in a selected namespace, to modify the Access Control List (ACL) for buckets, to modify the policy for buckets, and to delete buckets.

Figure 16 Bucket Management page



Create a bucket

You can create and configure S3, S3+FS, or CAS buckets in the ECS Portal.

Before you begin

- This operation requires the System Administrator or Namespace Administrator role in ECS.
- A System Administrator can create buckets in any namespace.
- A Namespace Administrator can create buckets in the namespace in which they are the administrator.

For CAS-specific instructions on setting up a CAS bucket for a CAS object user, see the *ECS Data Access Guide*, available from the ECS Product Documentation page.

Procedure

1. In the ECS Portal, select **Manage** > **Buckets**.
2. On the **Manage Buckets** page, select **New Bucket**.
3. On the **New Bucket** page, in the **Name** field, type the bucket name.
4. In the **Namespace** field, select the namespace that you want the bucket and its objects to belong to.
5. In the **Replication Group** field, select the replication group that you want to associate the bucket with.
6. In the **Bucket Owner** field, type the bucket owner.

   The bucket owner must be an ECS object user for the namespace. If you do not specify a user, you are assigned as the owner, however, you can not access the bucket unless your username is also assigned as an object user.

   The user that you specify is given Full Control.
7. In the **Bucket Tagging** field, click **Add** to add tags, and type name-value pairs.

   For more information, see Bucket tagging on page 87.
8. In the **Quota** field, click **Enabled** to specify a quota for the bucket and select the quota setting you require.

   The settings you can specify are described in Bucket settings on page 83.

9. In the **Server-side Encryption** field, click **Enabled** to specify that the bucket is encrypted.
10. In the **File System** field, click **Enabled** to specify that the bucket supports operation as a file system (for NFS or HDFS access).

    The bucket will be an S3 bucket that supports file systems.

    You can set a default Unix group for access to the bucket and for objects created in the bucket. More details are provided in Default group on page 85.
11. In the **CAS** field, click **Enabled** to set the bucket as a CAS bucket.

    By default, CAS is disabled and the bucket is marked as an S3 bucket.
12. In the **Metadata Search** field, click **Enabled** to specify that the bucket supports searches based on object metadata.

    If you enable Metadata Search, you can add User and System metadata keys that are used to create object indexes. For more information on entering metadata search keys, see Metadata index keys on page 85.

    Note

    If the bucket supports CAS, metadata search is automatically enabled and a CreateTime key is automatically created. The metadata can be searched using the S3 metadata search capability or using the Centera API.
13. In the **Access During Outage** field, click **Enabled** if you want the bucket to be available during a temporary site outage. For more information on this option, see TSO behavior with the ADO bucket property enabled on page 168.
14. If you enabled **Access During Outage** on the bucket, you can enable **Read-Only Access During Outage** if you want to restrict create, update, or delete operations on the objects in the bucket during a temporary site outage. Note that once the **Read-Only Access During Outage** option is enabled on the bucket, you cannot change it after the bucket is created. For more information on this option, see TSO behavior with the ADO bucket property enabled on page 168.
15. In the **Bucket Retention Period** field, type a time period to set a bucket retention period for the bucket, or click **Infinite** if you want objects in the bucket to be retained forever.

    For more information on retention periods, see Retention periods and policies on page 55.
16. Click **Save** to create the bucket.

Results

To assign permissions on the bucket for users or groups, see the tasks later in this section.

Edit a bucket

You can edit some bucket settings after the bucket has been created and after it has had objects written to it.

Before you begin

- This operation requires the System Administrator or Namespace Administrator role in ECS.

- A System Administrator can edit the settings for a bucket in any namespace.
- A Namespace Administrator can edit the settings for a bucket in the namespace in which they are the administrator.

To edit a bucket, you must be assigned to the Namespace Administrator or System Administrator role.

You can edit the following bucket settings:

- Quota
- Bucket Owner
- Bucket Tagging
- Access During Outage
- Bucket Retention

You cannot change the following bucket settings:

- Replication Group
- Server-side Encryption
- File System Enabled
- CAS Enabled
- Metadata Search

Procedure

1. In the ECS Portal, select **Manage** > **Buckets**.
2. On the **Bucket Management** page, in the Buckets table, select the **Edit** action for the bucket for which you want to change the settings.
3. Edit the settings that you want to change.

   You can find out more information about the bucket settings in Bucket settings on page 83.
4. Click **Save**.

Set ACLs

The privileges a user has when accessing a bucket are set using an Access Control List (ACL). You can assign ACLs for a user, for a set of pre-defined groups, such as all users, and for a custom group.

When you create a bucket and assign an owner to it, an ACL is created that assigns a default set of permissions to the bucket owner - the owner is, by default, assigned full control.

You can modify the permissions assigned to the owner or you can add new permissions for a user by selecting the Edit ACL operation for the bucket.

In the ECS Portal, the **Bucket ACLs Management** page has **User ACLs**, **Group ACLs**, and **Custom Group ACLs** tabs to manage the ACLs associated with individual users and pre-defined groups, and to allow groups to be defined that can be used when accessing the bucket as a file system.

Note

For information about ACLs with CAS buckets, see the *ECS Data Access Guide*, available from the ECS Product Documentation page.

Bucket ACLs reference

The ACL permissions that can be assigned are provided in the following table. The permissions that are applicable depend on the type of bucket.

Table 18 Bucket ACLs

ACL	Permission
Read	Allows user to list the objects in the bucket.
Read ACL	Allows user to read the bucket ACL.
Write	Allows user to create or update any object in the bucket.
Write ACL	Allows user to write the ACL for the bucket.
Execute	Sets the execute permission when accessed as a file system. This permission has no effect when the object is accessed using the ECS object protocols.
Full Control	Allows user to Read, Write, Read ACL, and Write ACL.
Privileged Write	Allows user to perform writes to a bucket or object when the user does not have normal write permission. Required for CAS buckets.
Delete	Allows user to delete buckets and objects. Required for CAS buckets.
None	User has no privileges on the bucket.

Set the bucket ACL permissions for a user

The ECS Portal enables you to set the bucket ACL for a user or for a pre-defined group.

Before you begin

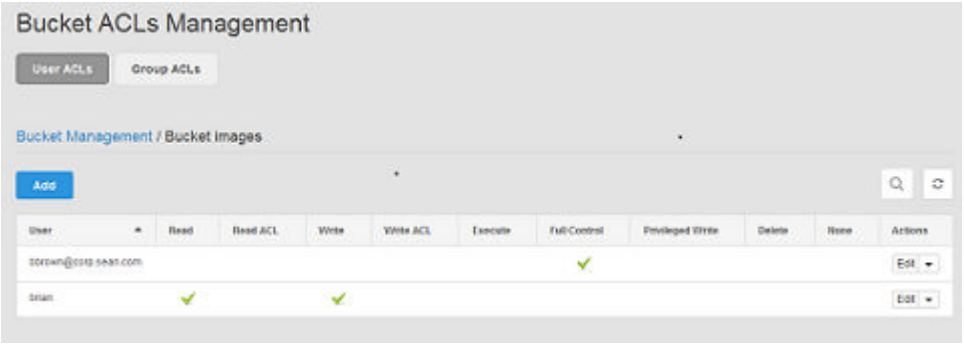
- This operation requires the System Administrator or Namespace Administrator role in ECS.
- A System Administrator can edit the ACL settings for a bucket in any namespace.
- A Namespace Administrator can edit the ACL settings for a bucket n the namespace in which they are the administrator.

Procedure

1. In the ECS Portal, select **Manage** > **Buckets**.
2. On the **Bucket Management** page, in the Buckets table, select the **Edit ACL** action for the bucket for which you want to change the settings.
3. On the **Bucket ACLs Management** page, click **User ACLs** to set the ACL permissions for a user.

   The **User ACLs** tab shows the ACLs that have been applied to users and enables ACLs to be assigned to a user using the **Add** operation.

Figure 17 User ACLs tab in the **Bucket ACLs Management** page



Note

Because the ECS Portal supports S3, S3 + File system (HDFS or NFS), and CAS buckets, the range of permissions that can be set are not applicable to all bucket types.

4. To edit the permissions for a user that already has permissions assigned, or to add a user that you want to assign permissions for, perform one of the following steps.
   - To set (or remove) the ACL permissions for a user that already has permissions, in the ACL table, in the Action column, click **Edit** or **Remove**.
   - To add a user and assign ACL permissions, click **Add**.

   The bucket owner has default permissions assigned.
5. If you have added an ACL, enter the username of the user that the permissions will apply to.
6. Specify the permissions that you want to apply to the user.

   More information on ACL privileges is provided in Bucket ACLs reference on page 91.
7. Click **Save**.

Set the bucket ACL permissions for a pre-defined group

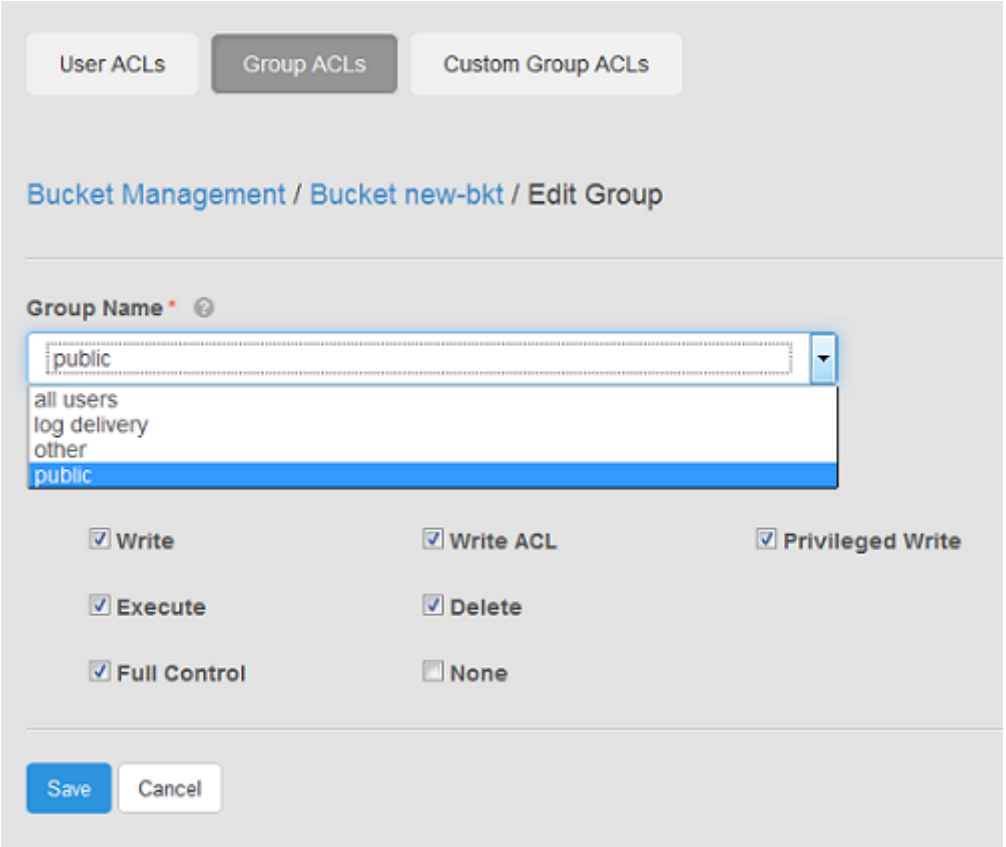
You can set the ACL for a bucket for a pre-defined group from the ECS Portal.

Before you begin

- This operation requires the System Administrator or Namespace Administrator role in ECS.
- A System Administrator can edit the group ACL settings for a bucket in any namespace.
- A Namespace Administrator can edit the group ACL settings for a bucket in the namespace in which they are the administrator.

Procedure

1. In the ECS Portal, select **Manage** > **Buckets**.
2. On the **Bucket Management** page, in the Buckets table, select the **Edit ACL** action for the bucket for which you want to change the settings.
3. Click the **Group ACLs** tab to set the ACL permissions for a pre-defined group, as shown in the following screenshot.



The group names are described in the following table.

Group	Description
public	All users, authenticated or not.
all users	All authenticated users
other	Authenticated users but not the bucket owner.
log delivery	Not supported.

4. Select the privileges that you want to assign to the group.
5. Click **Save**.

Set custom group bucket ACLs

You can set a group ACL for a bucket in the ECS Portal and you can set bucket ACLs for a group of users (Custom Group ACL), for individual users, or a combination of both. For example, you can grant full bucket access to a group of users, but you can also restrict (or even deny) bucket access to individual users in that group.

Before you begin

- This operation requires the System Administrator or Namespace Administrator role in ECS.
- A System Administrator can edit the group ACL settings for a bucket in any namespace.
- A Namespace Administrator can edit the group ACL settings for a bucket in the namespace in which they are the administrator.

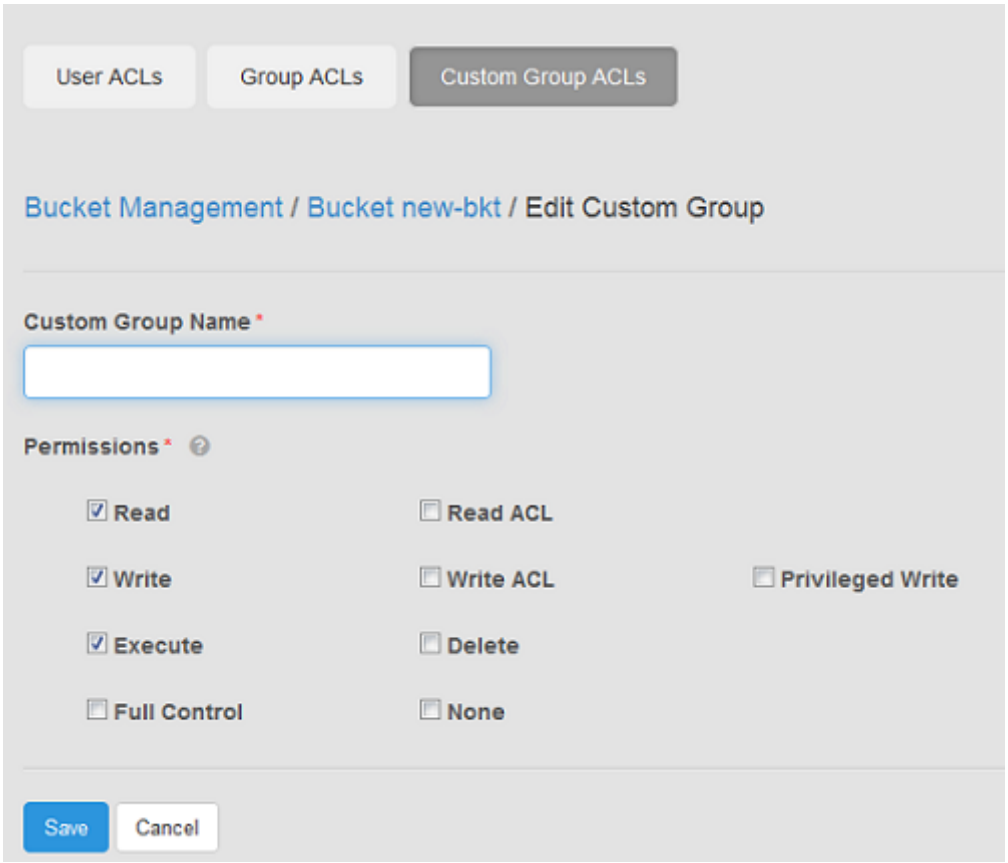
Custom group ACLs enable groups to be defined and for permissions to be assigned to the group. The main use case for assigning groups to a bucket is to support access to the bucket as a file system, for example, when making the bucket available for NFS or HDFS.

Members of the UNIX group can access the bucket when it is accessed as a file system (using NFS or HDFS).

Procedure

1. In the ECS Portal, select **Manage** > **Buckets**.
2. On the **Bucket Management** page, locate the bucket you want to edit in the table and select the **Edit ACL** action.
3. Click **Custom Group User ACLs** to set the ACL for a custom group.
4. Click **Add**.

   The **Edit Custom Group** page displays, as shown in the following screenshot.



5. On the **Edit Custom Group** page, in the **Custom Group Name** field, type the name for the group.

   This name can be a UNIX/Linux group, or an Active Directory group.
6. Set the permissions for the group.

   At a minimum you should assign **Read**, **Write**, **Execute** and **Read ACL**.
7. Click **Save**.

Using the Bucket Policy Editor

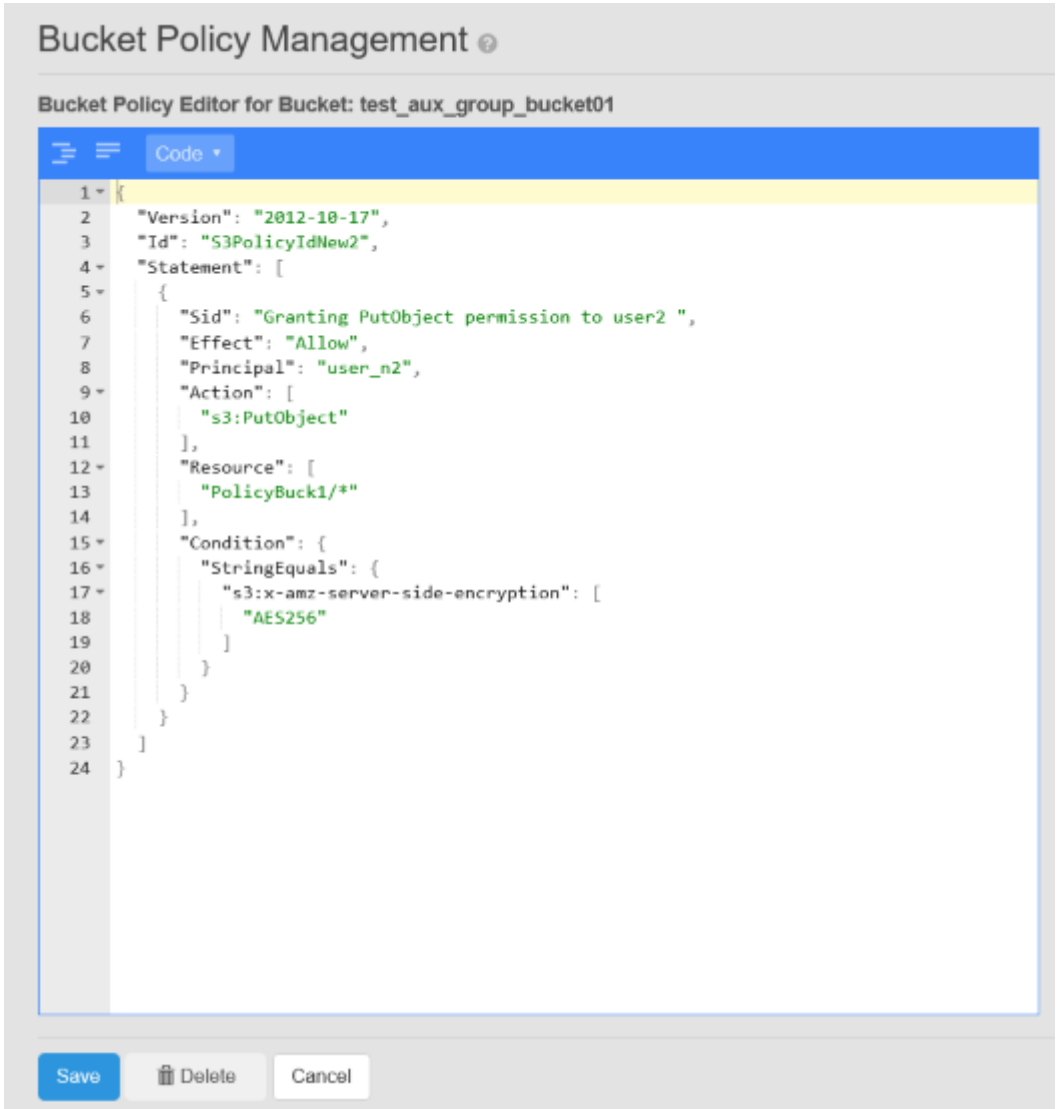
The ECS Portal provides a Bucket Policy Editor to enable you to associate a bucket policy with an existing bucket.

For each bucket, you can define ACLs for an object user. Bucket policies provide greater flexibility than ACLs and allow fine grained control over permissions for bucket operations and for operations on objects within the bucket. Policy conditions are used to assign permissions for a range of objects that match the condition and are used to automatically assign permissions to newly uploaded objects.

Policies are defined in JSON format and the syntax used for policies is the same as that used for Amazon AWS. The operations for which permissions can be assigned are limited to those operations supported by ECS. For more information, see the *ECS Data Access Guide*, available from the ECS Product Documentation page.

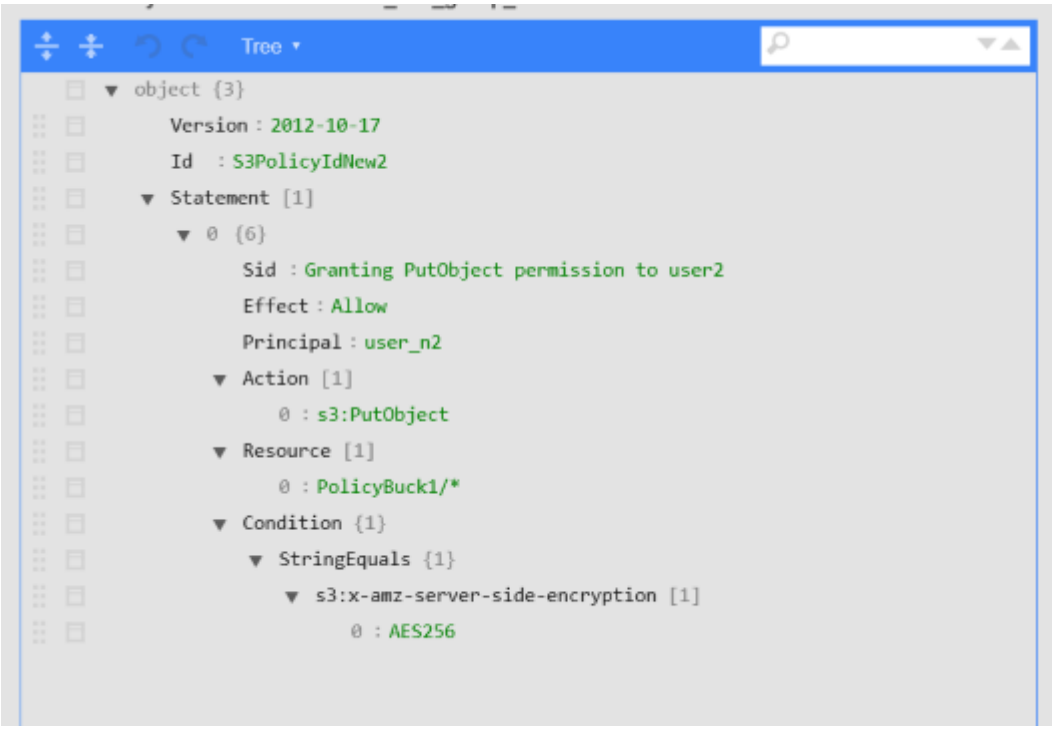
The bucket policy editor has a code view and a tree view. The code view, shown in the following screenshot, enables you to enter JSON policies from scratch or to paste existing policies into the editor and modified. For example, if you have existing policies in JSON format, you can paste them into the code view and modify them.

Figure 18 Bucket Policy Editor code view



The tree view, shown in the following screenshot, provides a mechanism for navigating a policy and is useful where you have a large number of statements in a policy. You can expand and contract the statements and search them.

Figure 19 Bucket Policy Editor tree view



Bucket policy scenarios

In general, the bucket owner has full control on a bucket and can grant permissions to other users and can set S3 bucket policies using an S3 client. In ECS, it is also possible for an ECS System or Namespace Administrator to set bucket policies using the Bucket Policy Editor from the ECS Portal.

You can use bucket policies in the following typical scenarios:

- Grant bucket permissions to a user on page 97
- Grant bucket permissions to all users on page 98
- Automatically assign permissions to created objects on page 98

Grant bucket permissions to a user

Where you want a user other than the bucket owner to be granted permissions on a bucket, you can specify the resource that you want to change the permissions for, set the principal attribute to the name of the user, and specify one or more actions that you want to allow.

The following example shows a policy that grants a user named `user1` the permission to update and read objects in the bucket named `mybucket`.

```
{
    "Version": "2012-10-17",
    "Id": "S3PolicyId1",
    "Statement": [
        {
            "Sid": "Grant permission to user1",
            "Effect": "Allow",
            "Principal": ["user1"],
            "Action": [ "s3:PutObject","s3:GetObject" ],
            "Resource":[ "mybucket/*" ]
```

```
        }
    ]
}
```

You can also add conditions. For example, if you only want the user to be able to read and write object when accessing the bucket from a specific IP address, you can add an `IpAddress` condition as shown in the following policy.

```
{
    "Version": "2012-10-17",
    "Id": "S3PolicyId1",
    "Statement": [
        {
            "Sid": "Grant permission ",
            "Effect": "Allow",
            "Principal": ["user1"],
            "Action": [ "s3:PutObject","s3:GetObject" ],
            "Resource":[ "mybucket/*" ]
            "Condition": {"IpAddress": {"aws:SourceIp": "<Ip address>"}
            }
     ]
}
```

Grant bucket permissions to all users

Where you want a user other than the bucket owner to be granted permissions on a bucket, you can specify the resource that you want to change the permissions for, set the principal attribute as anybody (`*`), and specify one or more actions that you want to allow.

The following example shows a policy that grants anyone permission to read objects in the bucket named `mybucket`.

```
{
    "Version": "2012-10-17",
    "Id": "S3PolicyId2",
    "Statement": [
        {
            "Sid": "statement2",
            "Effect": "Allow",
            "Principal": ["*"],
            "Action": [ "s3:GetObject" ],
            "Resource":[ "mybucket/*" ]
        }
    ]
}
```

Automatically assign permissions to created objects

You can use bucket policies to automatically enable access to ingested object data. In the following example bucket policy, `user1` and `user2` can create sub-resources (that is, objects) in the bucket named `mybucket` and can set object ACLs. With the ability to set ACLs, the users can then set permissions for other users. If you set the ACL in the same operation, a condition can be set such that a canned ACL public-read must be specified when the object is created. This ensures that all of the created objects can be read by anybody.

```
{
    "Version": "2012-10-17",
    "Id": "S3PolicyId3",
```

```
    "Statement": [
        {
            "Sid": "statement3",
            "Effect": "Allow",
            "Principal": ["user1", "user2"],
            "Action": [ "s3:PutObject, s3:PutObjectAcl" ],
            "Resource":[ "mybucket/*" ]
            "Condition":{"StringEquals":{"s3:x-amz-acl":["public-read"]}}
        }
    ]
}
```

Create a bucket policy

You can create a bucket policy for a selected bucket using the Bucket Policy Editor in the ECS Portal.

Before you begin

This operation requires the System Administrator or Namespace Administrator role (for the namespace to which the bucket belongs).

You can also create a bucket policy in a text editor and deploy it using the ECS Management REST API or the S3 API.

Procedure

1. In the ECS Portal, select **Manage** > **Buckets**
2. From the **Namespace** drop-down, select the namespace to which the bucket belongs.
3. In the **Actions** column for the bucket, select **Edit Policy** from the drop-down menu.
4. Provided your policy is valid, you can switch to the tree view of the Bucket Policy Editor. The tree view makes it easier to view your policy and to expand and contract statements.
5. In the Bucket Policy Editor, type the policy or copy and paste a policy that you have previously created.

   Some examples are provided in Bucket policy scenarios on page 97 and full details of the supported operations and conditions are provided in *ECS Data Access Guide*, available from the ECS Product Documentation page.
6. **Save**.

   The policy is validated and, if valid, the Bucket Policy Editor exits and the portal displays the **Bucket Management** page. If the policy is invalid, the error message provides information on the reason the policy is invalid.

Create a bucket using the S3 API (with s3curl)

You can use the S3 API to create a bucket in an replication group. Because ECS uses custom headers (x-emc), the string to sign must be constructed to include these headers. In this procedure the s3curl tool is used; there are also a number of programmatic clients you can use, for example, the S3 Java client.

Before you begin

- To create a bucket, ECS must have at least one replication group configured.
- Ensure that Perl is installed on the Linux machine on which you will run s3curl.

- Ensure that curl tool and the s3curl tool are installed. The s3curl tool acts as a wrapper around curl.
- To use `s3curl` with x-emc headers, minor modifications must be made to the `s3curl` script. You can obtain the modified, ECS-specific version of s3curl from the EMCECS Git Repository.
- Ensure that you have obtained a secret key for the user who will create the bucket. For more information, see the *ECS Data Access Guide*, available from the ECS Product Documentation page.

The EMC headers that can be used with buckets are described in Bucket HTTP headers on page 102.

Procedure

1. Obtain the identity of the replication group in which you want the bucket to be created, by typing the following command.

```
GET https://<ECS IP Address>:4443/vdc/data-service/vpools
```

The response provides the name and identity of all data services virtual pools. In the following example, the ID is urn:storageos:ReplicationGroupInfo:8fc8e19bedf0-4e81-bee8-79accc867f64:global.

```
<data_service_vpools>
<data_service_vpool>
    <creation_time>1403519186936</creation_time>
    <id>urn:storageos:ReplicationGroupInfo:8fc8e19b-edf0-4e81-
bee8-79accc867f64:global</id>
    <inactive>false</inactive>
    <tags/>
    <description>IsilonVPool1</description>
    <name>IsilonVPool1</name>
    <varrayMappings>
        <name>urn:storageos:VirtualDataCenter:1de0bbc2-907c-4ede-b133-
f5331e03e6fa:vdc1</name>
        <value>urn:storageos:VirtualArray:793757ab-ad51-4038-b80a-682e124eb25e:vdc1</
value>
    </varrayMappings>
</data_service_vpool>
</data_service_vpools>
```

2. Set up `s3curl` by creating a `.s3curl` file in which to enter the user credentials.

   The `.s3curl` file must have permissions 0600 (rw-/---/---) when `s3curl.pl` is run.

   In the following example, the profile `my_profile` references the user credentials for the `user@yourco.com` account, and `root_profile` references the credentials for the root account.

```
%awsSecretAccessKeys = (
    my_profile => {
        id  => 'user@yourco.com',
        key => 'sZRCTZyk93IWukHEGQ3evPJEvPUq4ASL8Nre0awN'
    },
```

```
    root_profile => {
         id  => 'root',
         key => 'sZRCTZyk93IWukHEGQ3evPJEvPUq4ASL8Nre0awN'
     },
);
```

3. Add the endpoint that you want to use `s3curl` against to the `.s3curl` file.

   The endpoint is the address of your data node or the load balancer that sits in front of your data nodes.

```
push @endpoints , (
    '203.0.113.10',  'lglw3183.lss.emc.com',
);
```

4. Create the bucket using `s3curl.pl` and specify the following parameters:
   - Profile of the user
   - Identity of the replication group in which to create the bucket (<vpool_id>, which is set using the `x-emc-dataservice-vpool` header
   - Any custom x-emc headers
   - Name of the bucket (<BucketName>).

   The following example shows a fully specified command.

```
./s3curl.pl --debug --id=my_profile --acl public-read-write
--createBucket -- -H 'x-emc-file-system-access-enabled:true'
-H 'x-emc-dataservice-vpool:<vpool_id>' http://<DataNodeIP>:9020/<BucketName>
```

The example uses the`x-emc-dataservice-vpool` header to specify the replication group in which the bucket is created and the `x-emc-file-system-access-enabled` header to enable the bucket for file system access, such as for NFS or HDFS.

Note

T2he `-acl public-read-write` argument is optional, but can be used to set permissions to enable access to the bucket (for example, if you intend to access to bucket as NFS from an environment that is not secured using Kerberos).

If successful (with --debug on) output similar to the following appears:

```
s3curl: Found the url: host=203.0.113.10; port=9020; uri=/S3B4; query=;
s3curl: ordinary endpoint signing case
s3curl: StringToSign='PUT\n\n\nThu, 12 Dec 2013 07:58:39 +0000\nx-amz-acl:public-read-
write
\nx-emc-file-system-access-enabled:true\nx-emc-dataservice-vpool:
urn:storageos:ReplicationGroupInfo:8fc8e19b-edf0-4e81-bee8-79accc867f64:global:\n/
S3B4'
s3curl: exec curl -H Date: Thu, 12 Dec 2013 07:58:39 +0000 -H Authorization: AWS
```

```
root:AiTcfMDhsi6iSq2rIbHEZon0WNo= -H x-amz-acl: public-read-write -L -H content-
type:
--data-binary  -X PUT -H x-emc-file-system-access-enabled:true
-H x-emc-dataservice-vpool:urn:storageos:ObjectStore:e0506a04-340b-4e78-
a694-4c389ce14dc8: http://203.0.113.10:9020/S3B4
```

After you finish

You can list the buckets using the S3 interface, using:

```
./s3curl.pl --debug --id=my_profile http://<DataNodeIP>:9020/
```

Bucket HTTP headers

There are a number of headers that determine the behavior of ECS when creating buckets using the objects APIs.

The following `x-emc` headers are provided.

Table 19 Bucket headers

Header	Description
x-emc-dataservice-vpool	Determines the replication group is used to store the objects associated with this bucket. If you do not specify a replication group using the `x-emc-dataservice-vpool` header, ECS selects the default replication group associated with the namespace.
x-emc-file-system-access-enabled	Configures the bucket for NFS or HDFS access. The header must not conflict with the interface that is used. That is, a create bucket request from NFS or HDFS cannot specify `x-emc-file-system-access-enabled=false`.
x-emc-namespace	Specifies the namespace used for this bucket. If the namespace is not specified using the S3 convention of host-style or path-style request, then it is specified using the `x-emc-namespace` header. If the namespace is not specified in this header, the namespace associated with the user is used.
x-emc-retention-period	Specifies the retention period that is applied to objects in a bucket. Each time a request is made to modify an object in a bucket, the expiration of the retention period for the object is calculated based on the retention period associated with the bucket.
x-emc-is-stale-allowed	Specifies whether the bucket is accessible during a temporary VDC outage in a federated configuration
x-emc-server-side-encryption-enabled	Specifies whether objects written to a bucket are encrypted.
x-emc-metadata-search	Specifies one or more user or system metadata values that are used to create indexes of objects for the bucket. The indexes are used to perform object searches that are filtered based on the indexed metadata.

Bucket, object, and namespace naming conventions

Bucket and object (also referred to as key) names must conform to the specification presented here.

- S3 bucket and object naming in ECS on page 103

- OpenStack Swift container and object naming in ECS on page 103
- Atmos bucket and object naming in ECS on page 104
- CAS pool and object naming in ECS on page 104

Note

To use a bucket for HDFS, you must not use underscores in the bucket name as they are not supported by the URI Java class. For example, `viprfs://my_bucket.ns.site/` is an invalid URI and is thus not understood by Hadoop.

Namespace name
The following rules apply to the naming of ECS namespaces:

- Cannot be null or an empty string
- Length range is 1..255 (Unicode char)
- Valid characters are defined by regex /[a-zA-Z0-9-_]+/. That is, alphanumeric characters and hyphen (-) and underscore (_) special characters.

S3 bucket and object naming in ECS

Bucket and object names must conform to the ECS naming specification when using the ECS S3 Object API.

Bucket name
The following rules apply to the naming of S3 buckets in ECS:

- Must be between one and 255 characters in length. (S3 requires bucket names to be from 1 to 255 characters long)
- Can include dot (.), hyphen (-), and underscore (_) characters and alphanumeric characters ([a-zA-Z0-9])
- Can start with a hyphen (-) or alphanumeric character.
- Cannot start with a dot (.)
- Cannot contain a double dot (..)
- Cannot end with a dot (.)
- Must not be formatted as IPv4 address.

You can compare this with naming restriction specified by the S3 specification: http://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html.

Object name
The following rules apply to the naming of S3 objects in ECS.

- Cannot be null or an empty string
- Length range is 1..255 (Unicode char)
- No validation on characters.

OpenStack Swift container and object naming in ECS

Container and object names must conform to the ECS naming specification when using the ECS OpenStack Swift Object API.

Container name
The following rules apply to the naming of Swift containers:

- Cannot be null or an empty string
- Length range is 1..255 (Unicode char)
- Can include dot (.), hyphen (-), and underscore (_) characters and alphanumeric characters ([a-zA-Z0-9])

Object name
The following rules apply to the naming of Swift objects:

- Cannot be null or an empty string
- Length range is 1..255 (Unicode char)
- No validation on characters.

Atmos bucket and object naming in ECS

Subtenant and object names must conform to the ECS naming specification when using the ECS Atmos Object API.

Subtenant (bucket)
The subtenant is created by the server, so the client does not need to know the naming scheme.

Object name
The following rules apply to the naming of Atmos objects:

- Cannot be null or an empty string
- Length range is 1..255 (Unicode characters)
- No validation on characters
- Name should be percent-encoded UTF-8.

CAS pool and object naming in ECS

CAS pools and objects (*clips* in CAS terminology) names must conform to the ECS naming specification when using the CAS API.

CAS pool naming
The following rules apply to the naming of CAS pools in ECS:

- Can contain a maximum of 255 characters
- Cannot contain: ' " / & ? * < > <tab> <newline> or <space>

Clip naming
The CAS API does not support user-defined keys. When an application using CAS API creates a clip, it opens a pool, creates a new clip, and adds tags, attributes, streams and so on. After a clip is complete it is written to a device.

A corresponding clip ID is returned by the CAS engine and can be referred to using <pool name>/<clip id>.

CHAPTER 8

File Access

Introduction to file access

The main concepts relating to file access are described in the following topics:

- Multi-protocol access to directories and files
- ECS Portal support for NFS configuration
- Best practice when using ECS NFS
- Permissions for multi-protocol (cross-head) access
- File API summary

This section also describes these NFS configuration tasks:

- Create a bucket for NFS using the ECS Portal
- Add an NFS export using the ECS Portal
- Add a user or group mapping using the ECS Portal
- Configure NFS security with Kerberos
- Mounting an NFS export: example

ECS enables you to configure object buckets for access as NFS file systems using NFSv3.

To enable Unix users to access the file system, ECS provides a mechanism for mapping ECS object users to Unix users. An ECS bucket always has an owner, and mapping the bucket owner to a Unix ID gives that Unix user permissions on the file system. In addition, ECS enables the assignment of a default custom group to the bucket so that members of a Unix group mapped to the ECS default custom group can access the bucket.

ECS supports multi-protocol access, so that files written using NFS can also be accessed using Amazon Simple Storage Service (Amazon S3), OpenStack Swift, and EMC Atmos object protocols. Similarly, objects written using S3 and OpenStack Swift object protocols can be made available through NFS. For Atmos, objects created using the namespace interface can be listed using NFS, however, objects created using an object ID cannot. Objects and directories created using object protocols can be accessed by Unix users and by Unix group members by mapping the object users and groups.

ECS NFS provides advisory locking and supports the following locks:

- Lock over multiple sites
- Shared and exclusive locks

ECS NFS supports Kerberos security.

Multi-protocol access to directories and files

ECS supports writing objects using the S3 protocol and accessing them as files using NFS and, conversely, writing files using NFS and accessing the files as objects using the S3 protocol. You must understand how directories are managed when you use multi-protocol access.

The S3 protocol does not make provision for the creation of folders or directories.

To enable multi-protocol operation, ECS support for the S3 protocol formalizes the use of `/` and creates *directory* objects for all intermediate paths in an object name. An object named `/a/b/c.txt` results in the creation of a file object named `c.txt` and directory objects for `a` and `b`. The directory objects are not exposed to users through

the S3 protocol, and are maintained only to provide multi-protocol access and compatibility with file system-based APIs. This means that ECS can display files within a directory structure when the bucket is viewed as an NFS or HDFS file system.

Limitations

- An issue can arise where both a directory object and a file object are created with the same name. This can occur in the following ways:
  - A file `path1/path2` is created from NFS, then an object `path1/path2/path3` is created from S3. Because S3 allows creation of objects that have another object's name as the prefix, this operation is valid and is supported. A file and a directory called `path2` will exist.
  - A directory `path1/path2` is created from NFS, then an object `path1/path2` is created from S3. This operation is a valid operation from S3 because directory `path1/path2` is not visible through the S3 API. A file and a directory called `path2` will exist.

  To resolve this issue, requests from S3 always return the file, and requests from NFS always return the directory. However, this means that in the first case the file created by NFS is hidden by the object created by S3.
- NFS does not support filenames with a trailing `/` in them, but the S3 protocol does. NFS does not show these files.

ECS Portal support for NFS configuration

You can use the **File** page available from **Manage** > **File** to create NFS exports and to map ECS users and groups so that they can access the NFS export.

The **File** page has an **Exports** tab and a **User/Group Mapping** tab. By default the **Exports** tab displays. You can also configure NFS exports and set up user mappings by using the ECS Management REST API and the CLI.

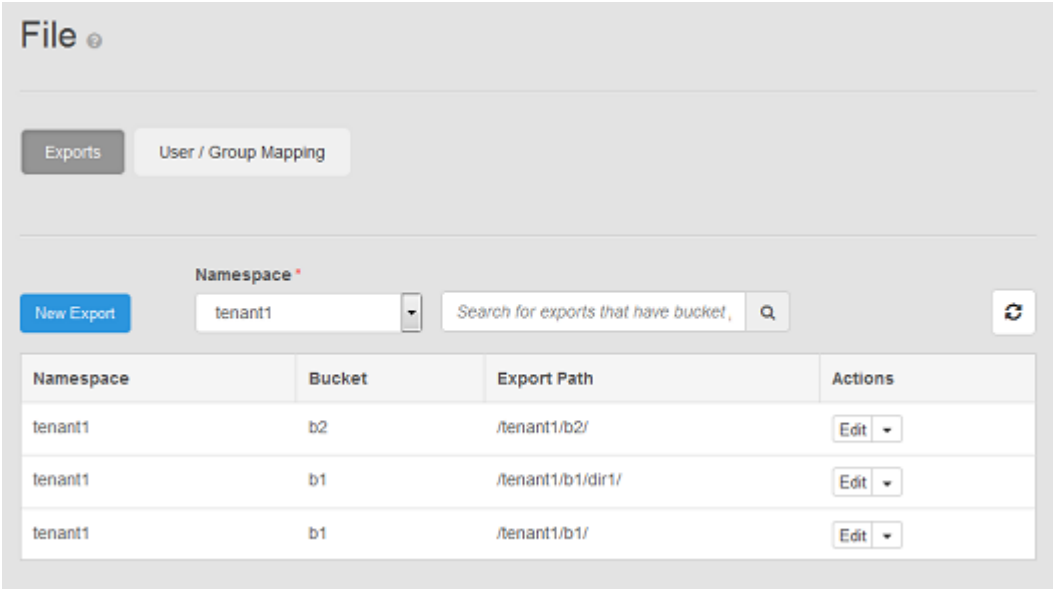
The **Exports** and **User/Group Mapping** tabs are described in the following topics:

- Working with exports in the ECS Portal on page 108
- Working with user/group mappings in the ECS Portal on page 108

Working with exports in the ECS Portal

The **Exports** tab shows the NFS exports that have been created and enables you to create new NFS exports and edit existing exports.

Figure 20 Exports tab on the File page



The **Exports** tab has a **Namespace** field that allows you to select the namespace for which you want to see the currently defined exports. The following fields are displayed in the **Exports** table.

Field	Description
Namespace	The tenant/namespace that the underlying storage belongs to.
Bucket	The bucket that provides the underlying storage for the NFS export.
Export Path	The mount point associated with the export, in the form: /<*namespace_name*>/<*bucket_name*>/<*export_name*> . The export name is only be specified if you are exporting a directory that exists within the bucket.
Actions	The actions that can be completed on the NFS export. • **Edit**: Edit the bucket quota and export host options. • **Delete**: Delete the NFS export.

You can add a new NFS export by clicking **New Export**.

Working with user/group mappings in the ECS Portal

ECS stores the owner and group for the bucket, and the owner and group for files and directories within the bucket, as ECS object username and custom group names, respectively. The names must be mapped to Unix IDs in order that NFS users can be given access with the appropriate privileges.

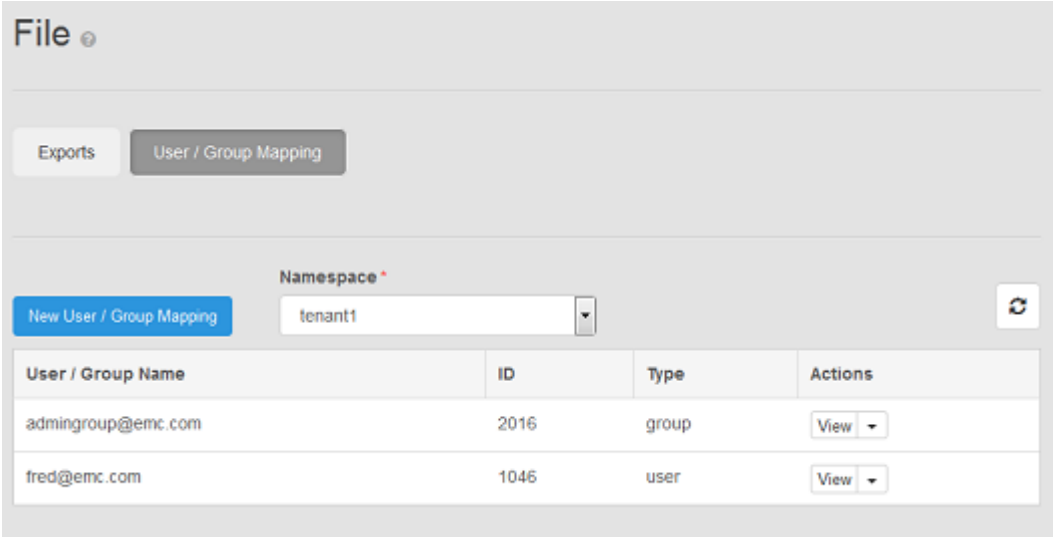
The mapping enables ECS to treat an ECS object user and an NFS user as the same user but with two sets of credentials, one to access ECS using NFS, and one to access

the ECS using the object protocols. Because the accounts are mapped, files written by an NFS user will be accessible as objects by the mapped object user and objects written by the object users will be accessible as files by the NFS user.

The permissions associated with the file or object will be based on a mapping between POSIX and object protocol ACL privileges. The mapping is described in detail in Permissions for multi-protocol (cross-head) access on page 124.

The **User/Group Mapping** tab on the **File** page shows a User/Group Mapping table.

Figure 21 User/Group Mapping tab on the File page



The **User/Group Mapping** tab has a **Namespace** field that enables the table to show the users and groups configured for the selected namespace. The following fields are displayed in th User/Group Mapping table:

Field	Description
User/Group Name	The object username of the user.
ID	The Unix User ID or Group ID that has been mapped to the object user.
Type	Indicates whether the ID is for a User or Group.
Actions	The actions that can be completed on the user or group mapping. They are: **View** and **Delete**

You can add new mappings by clicking **New User/Group Mapping**.

ECS NFS configuration tasks

To configure ECS NFS, the following tasks must be performed.

Procedure

1. Create a bucket for NFS using the ECS Portal on page 110
2. Add an NFS export using the ECS Portal on page 111
3. Add a user or group mapping using the ECS Portal on page 114
4. Configure NFS with Kerberos security on page 115

Create a bucket for NFS using the ECS Portal

You can use the ECS Portal to create a bucket configured for use with NFS.

Before you begin

- This operation requires the Namespace Administrator or System Administrator role in ECS.
- If you are a Namespace Administrator you can create buckets in your namespace.
- If you are System Administrator you can create a bucket belonging to any namespace.

The steps provided here focus on the configuration you need to perform to make a bucket suitable for use by NFS. The bucket you create is an S3 bucket enabled for file system use.

Procedure

1. In the ECS Portal, select **Manage** > **Buckets** > **New Bucket**.
2. On the **New Bucket** page, in the **Name** field, type a name for the bucket.
3. In the **Namespace** field, select the namespace that the bucket will belong to.
4. In the **Replication Group** field, select a replication group or leave blank to use the default replication group for the namespace.
5. In the **Bucket Owner** field, type the name of the bucket owner.
6. In the **CAS** field, do not enable CAS.

   Note

   A bucket that is intended for use as NFS cannot be used for CAS. The **CAS** field is disabled when the **File System** field is enabled.

7. Enable any other bucket features that you require.

   You can enable any of the following features on a NFS bucket:

   - Quota
   - Server-side Encryption
   - Metadata Search
   - Access During Outage
   - Read-Only Access During Outage
   - Compliance (see note)
   - Bucket Retention Period

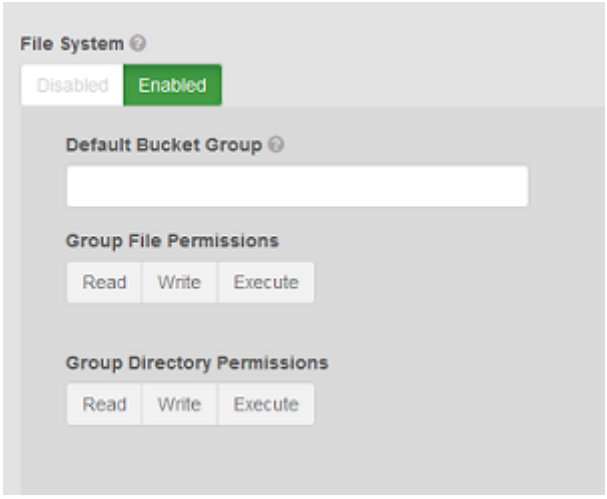
   For information on these settings, see Bucket settings on page 83.

   Note

   A bucket that is compliance-enabled cannot be written to using the NFS protocol. However, data written using object protocols can be read from NFS.

8. In the **File System** field, click **Enabled**.

   Once enabled, the fields for setting a default group for the file system/bucket and for assigning group permissions for files and directories created in the bucket are available.



9. In the **Default Bucket Group** field, type a name for the default bucket group.

   This group is the group associated with the NFS root file system and with any files or directories created in the NFS export. It enables users who are members of the group to access the NFS export and to access files and directories.

   This group must be specified at bucket creation. If it is not, the group would have to be assigned later from the NFS client.

10. Set the default permissions for files and directories created in the bucket using the object protocol.

    These settings are used to apply Unix group permissions to objects created using object protocols.

    The S3 protocol does not have the concept of groups so there is no opportunity for setting group permissions in S3 and mapping them to Unix permissions. Hence, this provides a one-off opportunity for a file or directory created using the S3 protocol to be assigned to the specified default group with the permissions specified here.

    a. Set the **Group File Permissions** by clicking the appropriate permission buttons.

       You normally set Read and Execute permissions.

    b. Set the **Group Directory Permissions** by clicking the appropriate permission buttons.

       You normally set Read and Execute permissions.

11. Click **Save** to create the bucket.

Add an NFS export using the ECS Portal

You can use the ECS Portal to create an NFS export and set the options that control access to the export.

Before you begin

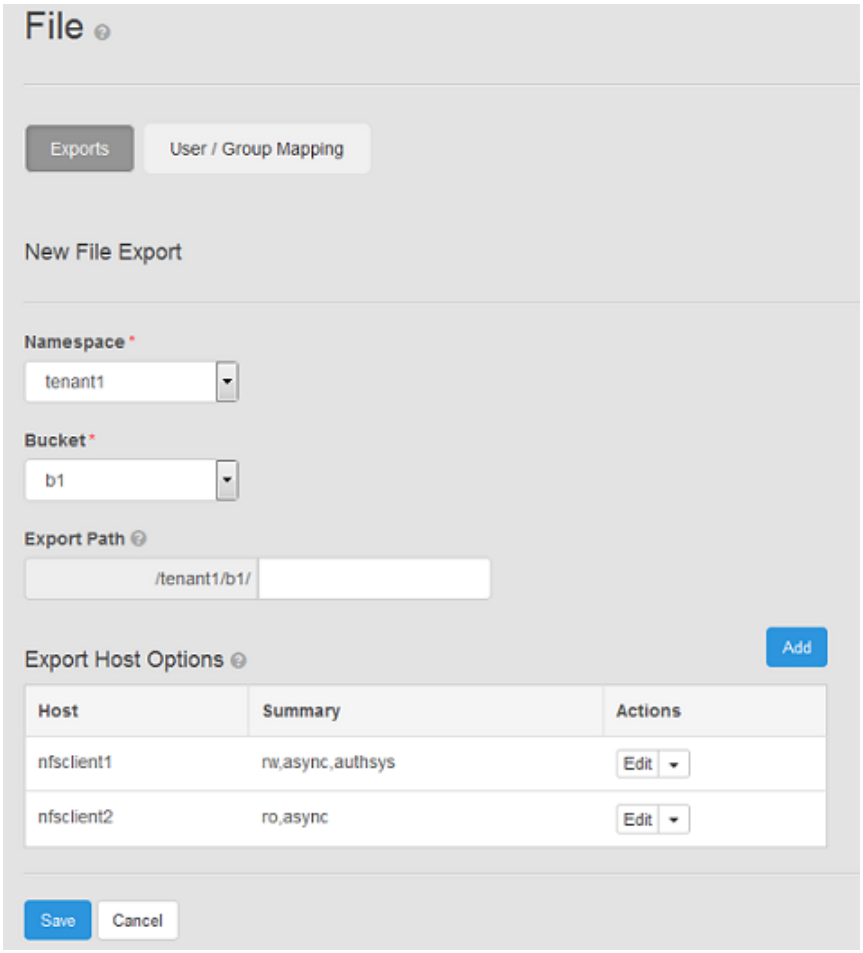
- This operation requires the Namespace Administrator or System Administrator role in ECS.
- If you are a Namespace Administrator you can add NFS exports into your namespace.

- If you are System Administrator you can add NFS exports into any namespace.
- You must have created a bucket to provide the underlying storage for the export.

Procedure

1. In the ECS Portal, select the **File** > **Exports** > **New Export** page.

   The **New File Export** sub-page appears. The following figure shows an export configured for access by NFS client hosts.



=

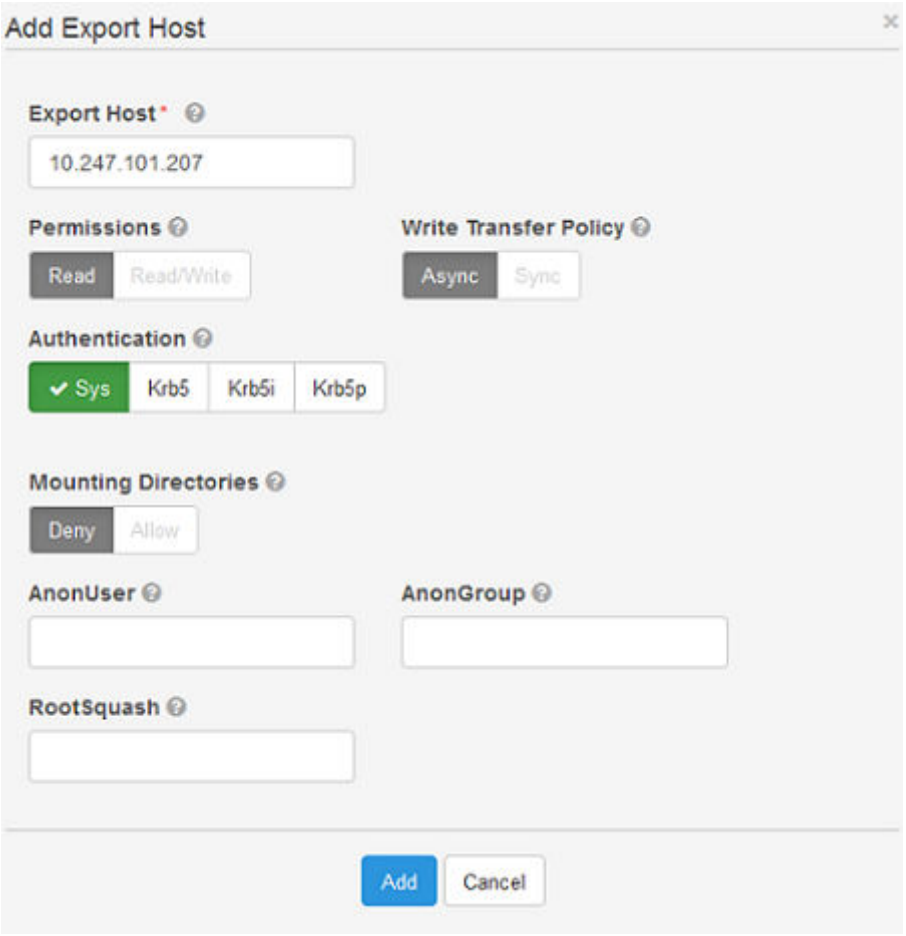
2. On the **New File Export** sub-page, in the **Namespace** field, select the namespace that owns the bucket that you want to export.
3. In the **Bucket** field, select the bucket.
4. In the **Export Path** field, specify the path.

   ECS automatically generates the export path based on the namespace and bucket. You only need to enter a name if you are exporting a directory that already exists within the bucket. So if you enter `/namespace1/bucket1/dir1`, for example, you should ensure that `dir1` exists. If it does not, mounting the export will fail.

5. To add the hosts that you want to be able to access the export, complete the following steps.

   a. In the **Export Host Options** area, click **Add**.

      The **Add Export Host** dialog appears.



b. In the **Add Export Host** dialog, specify one or more hosts that you want to be able to access the export and configure the access options.

   You must choose an **Authentication** option. This is normally `sys` unless you are intending to configure Kerberos. Default values for Permissions (`ro`) and Write Transfer Policy (`async`) are already set in the **Add Export Host** dialog and are passed to the NFS sever. The remaining options are the same as the NFS server defaults and so are only passed by ECS if you change them.

   The following table describes the parameters that you can specify when you add a host.

Setting	Description
Export Host	Sets the IP address of the host or hosts that can access the export. Use a comma separated list to specify more than one host.
Permissions	Enables access to the export to be set as Read/Write or Read only. This is the same as setting `rw` or `ro` in `/etc/exports`.
Write Transfer Policy	Sets the write transfer policy as synchronous or asynchronous. The default is asynchronous. This parameter is the same as setting `sync` or `async` for an export in `/etc/exports`.
Authentication	Sets the authentication types supported by the export.
Mounting Directories Inside Export	Specifies whether subdirectories of the export path are allowed as mount points. This parameter is the same as the `alldir` setting in `/etc/exports`.

Setting	Description
	With the `alldir` option, if you exported `/namespace1/bucket1`, for example, you can also mount subdirectories, such as`/namespace1/bucket1/dir1`, provided the directory exists.
AnonUser	Sets the effective user ID for anonymous user access to an export and for root access where `root_squash` is set. This is the same as setting `anonuid` in `/etc/exports`.
AnonGroup	Sets the effective group ID for anonymous group access to an export and for root access where `root_squash` has been set. This is the same as setting `anongid` in `/etc/exports`.
RootSquash	Specifies whether root is allowed on the export. If root is disallowed, the UID of the root user (`UID=0`) is translated to the UID of the user `nobody`, or to the UID you specify in `AnonUser`. This parameter is the same as using `root_squash` in `/etc/exports`.

c. Click **Add** to finish defining the host options.

6. If you want to add more hosts that can access the export, but with different options, repeat the previous step.
7. Click **Save** to save the NFS export definition.

Add a user or group mapping using the ECS Portal

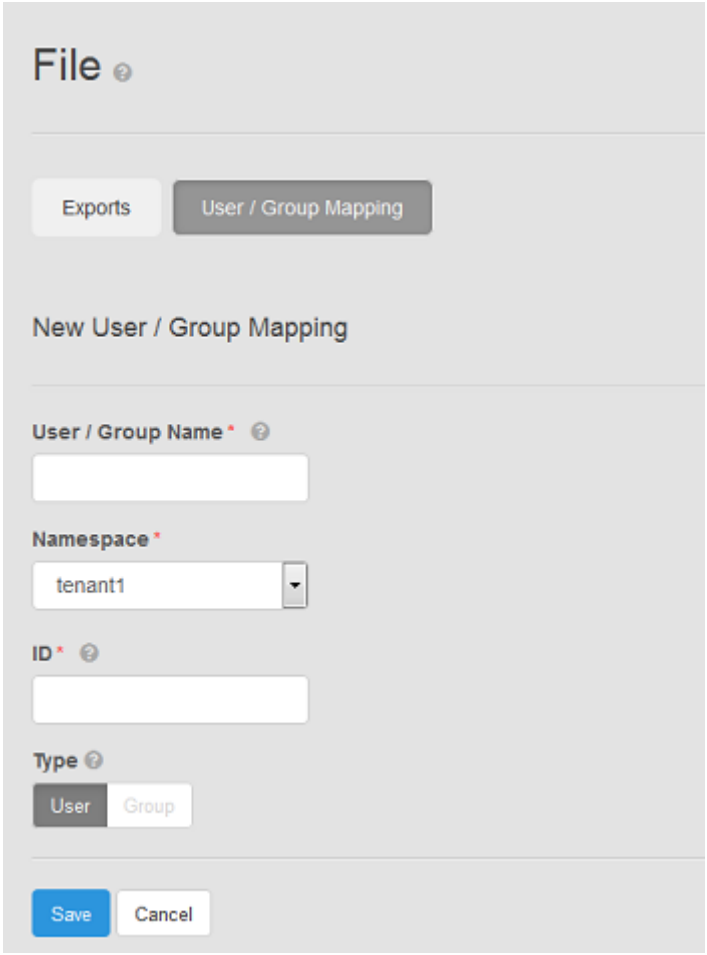
To provide NFS access to the file system (the bucket), you must map an object user who has permissions on the bucket to a Unix User Id (UID) so that the Unix user acquires the same permissions as the object user. Alternatively, you can map an ECS custom group that has permissions on the bucket to a Unix Group Id (GID) to provide access for members of the Unix group.

Before you begin

- This operation requires the Namespace Administrator or System Administrator role in ECS.
- If you are a Namespace Administrator you can add user or group mappings into your namespace.
- If you are System Administrator you can add user or group mappings into any namespace.
- Ensure that the UID exists on the NFS client and the username is an ECS object username.
- Ensure that a default custom group has been assigned to the bucket in order that group members have access to the file system.
- Ensure that default object and directory permissions have been assigned to the bucket in order that group members have access to objects and directories created using object protocols.

Procedure

1. In the ECS Portal, on the **Manage** > **File** page, click the **User/Group Mapping** tab.
2. On the **User/Group Mapping** tab, click **New Mapping** to display the **New User/Group Mapping** sub-page.



3. In the **User/Group Name** field, type the name of the ECS object user or ECS custom group that you want to map to a Unix UID or GID.
4. In the **Namespace** field, select the namespace that the ECS object user or custom group belongs.
5. In the **ID** field, enter the Unix UID or GID that you want the ECS user or group to map to.
6. In the **Type** field, click the type of mapping: **User** or **Group** so that ECS knows that the ID you have entered is a UID or a GID.
7. Click **Save**.

Configure NFS with Kerberos security

You can configure NFS with Kerberos security.

The following scenarios are supported:

- ECS client to single ECS node. The keytab on each ECS that you want to use as the NFS server must be specific to that node.
- ECS client to load balancer. The keytab on all ECS nodes is the same, and uses the hostname of the load balancer.

See Configure ECS NFS with Kerberos security on page 116.

Depending on your internal IT setup, you can use a KDC or you can Active Directory (AD) as your KDC.

To use AD, follow the steps in the following tasks: Register an ECS node with Active Directory on page 119 and Register a Linux NFS client with Active Directory on page 121.

Configure ECS NFS with Kerberos security

To configure Kerberos authentication with ECS NFS, you must configure both the ECS nodes and the NFS client, and create keytabs for the NFS server principal and for the NFS client principal.

Procedure

1. Ensure that the hostname of the ECS node can be resolved.

   You can use the `hostname` command to ensure that the FQDN of the ECS node is added to `/etc/HOSTNAME`.

```
dataservice-10-247-142-112:~ # hostname ecsnode1.yourco.com
dataservice-10-247-142-112:~ # hostname -i
10.247.142.112
dataservice-10-247-142-112:~ # hostname -f
ecsnode1.yourco.com
dataservice-10-247-142-112:~ #
```

2. Create the Kerberos configuration file (`krb5.conf`) on the ECS node as `/opt/emc/caspian/fabric/agent/services/object/data/hdfs/krb5.conf`. Unless HDFS has already been configured, you must create the `hdfs` directory with 755 (drwxr-xr-x) permissions (chmod 755 hdfs) and make user with uid 444 and group with gid 444 as the owner (chown 444:444 hdfs).

   Change the file permissions to 644 and make the user with id 444(storageos) the owner of the file.

   In the example below, the following values are used and must be replaced with your own settings.

   Kerberos REALM

   Set to NFS-REALM in this example.

   KDC

   Set to `kdcname.yourco.com` in this example.

   KDC Admin Server

   In this example, the KDC acts as the admin server.

```
[libdefaults]
        default_realm = NFS-REALM.LOCAL
[realms]
        NFS-REALM.LOCAL = {
                kdc = kdcname.yourco.com
                admin_server = kdcname.yourco.com
        }
[logging]
    kdc = FILE:/var/log/krb5/krb5kdc.log
    admin_server = FILE:/var/log/krb5/kadmind.log
    default = SYSLOG:NOTICE:DAEMON
```

Note

If HDFS for Kerberos is already configured, instead of replacing `/opt/emc/caspian/fabric/agent/services/object/data/hdfs/krb5.conf`, merge the REALM information, if it is different, into the existing `krb5.conf` file. Usually there is no change to this file as REALM has been configured by HDFS. In addition, the default permissions and owner should have already been configured by HDFS and should not require any change.

3. Add a host principal for the ECS node and create a keytab for the principal.

   In this example, the FQDN of the ECS node is `ecsnode1.yourco.com`

```
$ kadmin
kadmin> addprinc -randkey nfs/ecsnode1.yourco.com
kadmin> ktadd -k /datanode.keytab nfs/ecsnode1.yourco.com
kadmin> exit
```

4. Copy the keytab (`datanode.keytab`) to `/opt/emc/caspian/fabric/agent/services/object/data/hdfs/krb5.keytab`. Unless HDFS has already been configured, you need to create the `hdfs` directory with 755 (drwxr-xr-x) permissions (chmod 755 hdfs)and make user with uid 444 and group with gid 444 as the owner (chown 444:444 hdfs).

   Change its file permissions to 644 and make the user with id 444(storageos) the owner of the file.

   If HDFS is already configured, instead of replacing `/opt/emc/caspian/fabric/agent/services/object/data/hdfs/krb5.keytab`, merge the `datanode.keytab` file into the existing keytab file using `ktutil`. Default permissions and owner should already be configured by HDFS and should not require any change.

5. Download the *unlimited* JCE policy archive from `oracle.com` and extract it to the `/opt/emc/caspian/fabric/agent/services/object/data/jce/unlimited` directory.

   Kerberos may be configured to use a strong encryption type, such as AES-256. In that situation, the JRE within the ECS nodes must be reconfigured to use the 'unlimited' policy.

   Note

   This step should be performed only if you are using a strong encryption type.

   If HDFS is already configured, this step would have been completed by HDFS Kerberos configuration.

6. Run the following command from inside the object container.

```
service storageos-dataservice restarthdfs
```

7. To set up the client, begin by making sure that the hostname of the client can be resolved.

You can use the `hostname` command to ensure that the FQDN of the ECS node is added to `/etc/HOSTNAME`.

```
dataservice-10-247-142-112:~ # hostname ecsnode1.yourco.com
dataservice-10-247-142-112:~ # hostname -i
10.247.142.112
dataservice-10-247-142-112:~ # hostname -f
ecsnode1.yourco.com
dataservice-10-247-142-112:~ #
```

8. If your client is running SUSE Linux make sure that line `NFS_SECURITY_GSS="yes"` is uncommented in `/etc/sysconfig/nfs`.
9. If you are on Ubuntu make sure to have line `NEED_GSSD=yes` in `/etc/default/nfs-common`.
10. Install rpcbind and nfs-common.

    Use `apt-get` or `zypper`. On SUSE Linux, for nfs-common, use:

    ```
    zypper install yast2-nfs-common
    ```

    By default these are turned off in Ubuntu client.
11. Set up your Kerberos configuration file.

    In the example below, the following values are used and you must replace them with your own settings.

    Kerberos REALM

    Set to `NFS-REALM` in this example.

    KDC

    Set to `kdcname.yourco.com` in this example.

    KDC Admin Server

    In this example, the KDC acts as the admin server.

    ```
    [libdefaults]
            default_realm = NFS-REALM.LOCAL
    [realms]
            NFS-REALM.LOCAL = {
                    kdc = kdcname.yourco.com
                    admin_server = kdcname.yourco.com
            }
    [logging]
        kdc = FILE:/var/log/krb5/krb5kdc.log
        admin_server = FILE:/var/log/krb5/kadmind.log
        default = SYSLOG:NOTICE:DAEMON
    ```
12. Add a host principal for the NFS client and create a keytab for the principal.

    In this example, the FQDN of the NFS client is `nfsclient.yourco.com`

    ```
    $kadmin
    kadmin> addprinc -randkey host/nfsclient.yourco.com
    ```

```
kadmin> ktadd -k /nkclient.keytab host/nfsclient.yourco.com
kadmin> exit
```

13. Copy the keytab file (`nfsclient.keytab`) from the KDC machine to `/etc/krb5.keytab` on the NFS client machine.

```
scp /nkclient.keytab root@nfsclient.yourco.com:/etc/krb5.keytab
ssh root@nfsclient.yourco.com 'chmod 644 /etc/krb5.keytab'
```

14. Create a principal for a user to access the NFS export.

```
$kadmin
kadmin> addprinc yourusername@NFS-REALM.LOCAL
kadmin> exit
```

15. Log in as root and add the following entry to your `/etc/fstab` file.

```
HOSTNAME:MOUNTPOINT    LOCALMOUNTPOINT        nfs
rw,user,nolock,noauto,vers=3,sec=krb5 0       0
```

For example:

```
ecsnode1.yourco.com:/s3/b1     /home/kothan3/1b1       nfs
rw,user,nolock,noauto,vers=3,sec=krb5   0 0
```

16. Log in as non root user and `kinit` as the non-root user that you created.

```
kinit yourusername@NFS-REALM.LOCAL
```

17. You can now mount the NFS export.

Note

Mounting as the root user does not require you to use `kinit`. However, when using root, authentication is done using the client machine's host principal rather than your Kerberos principal. Depending upon your operating system, you can configure the authentication module to fetch the Kerberos ticket when you login, so that there is no need to fetch the ticket manually using kinit and you can mount the NFS share directly.

Register an ECS node with Active Directory

To use Active Directory (AD) as the KDC for your NFS Kerberos configuration, you must create accounts for the client and server in AD and map the account to a

principal. For the NFS server, the principal represents the NFS service accounts, for the NFS client, the principal represents the client host machine.

Before you begin

You must have administrator credentials for the AD domain controller.

Procedure

1. Log in to AD.
2. In Server Manager, go to **Tools** > **Active Directory Users and Computers**.
3. Create a user account for the NFS principal using the format "nfs-<host>" , for example, "nfs-ecsnode1". Set a password and set the password to never expire.
4. Create an account for yourself (optional and one time).
5. Execute the following command to create a keytab file for the NFS service account.

   ```
   ktpass -princ nfs/<fqdn>REALM.LOCAL +rndPass -mapUser nfs-
   <host>@REALM.LOCAL -mapOp set -crypto All -ptype
   KRB5_NT_PRINCIPAL -out filename.keytab
   ```

   For example, to associate the nfs-ecsnode1 account with the principle nfs/ecsnode1.yourco.com@NFS-REALM.LOCAL, you can generate a keytab using:

   ```
   ktpass -princ nfs/ecsnode1.yourco.com@NFS-REALM.LOCAL
   +rndPass -mapUser nfs-ecsnode1@NFS-REALM.LOCAL -mapOp set -
   crypto All -ptype KRB5_NT_PRINCIPAL -out nfs-ecsnode1.keytab
   ```

6. Import the keytab to the ECS node.

   ```
   ktutil
   ktutil> rkt <keytab to import>
   ktutil> wkt /etc/krb5.keytab
   ```

7. Test registration by running.

   ```
   kinit -k nfs/<fqdn>@NFS-REALM.LOCAL
   ```

8. See the cached credentials by running the `klist` command.
9. Delete the cached credentials by running the `kdestroy` command.
10. View the entries in the keytab file by running the `klist` command.

    Example:

    ```
    klist -kte /etc/krb5.keytab
    ```

11. Follow steps 2 on page 116, 4 on page 117, and 5 on page 117 from Configure ECS NFS with Kerberos security on page 116 to place the Kerberos configuration files (`krb5.conf`, `krb5.keytab` and `jce/unlimited`) on the ECS node.

Register a Linux NFS client with Active Directory

To use Active Directory (AD) as the KDC for your NFS Kerberos configuration, you need to create accounts for the client and server in AD and map the account to a principal. For the NFS server, the principal represents the NFS service accounts, for the NFS client, the principal represents the client host machine.

Before you begin

You must have administrator credentials for the AD domain controller.

Procedure

1. Log in to AD.
2. In Server Manager, go to **Tools** > **Active Directory Users and Computers**.
3. On the **Active Directory Users and Computers** page, create a computer account for the client machine. For example: `nfsclient`. Set a password and set the password to never expire.
4. Create an account for a user (optional and one time)
5. Execute the following command to create a keytab file for the NFS service account.

```
ktpass -princ host/<fqdn>@REALM.LOCAL +rndPass -mapUser <host>@REALM.LOCAL -mapOp set
-crypto All -ptype KRB5_NT_PRINCIPAL -out filename.keytab
```

For example, to associate the `nfs-ecsnode1` account with the principle `host/nfsclient.yourco.com@NFS-REALM.LOCAL`, you can generate a keytab using:

```
ktpass -princ host/nfsclient.yourco.com@NFS-REALM.LOCAL +rndPass -mapUser nfsclient
$@NFS-REALM.LOCAL -mapOp set -crypto All -ptype KRB5_NT_PRINCIPAL -out
nfsclient.keytab
```

6. Import the keytab to the client node.

```
ktutil
ktutil> rkt <keytab to import>
ktutil> wkt /etc/krb5.keytab
```

7. Test registration by running.

```
kinit -k host/<fqdn>@NFS-REALM.LOCAL
```

8. See the cached credentials by running the `klist` command.
9. Delete the cached credentials by running the `kdestroy` command.
10. View the entries in the keytab file by running the `klist` command.

    For example:

```
 klist -kte /etc/krb5.keytab
```

11. Follow steps 2 on page 116, 4 on page 117, and 5 on page 117 from Configure ECS NFS with Kerberos security on page 116 to place the Kerberos configuration files (`krb5.conf`, `krb5.keytab` and `jce/unlimited`) on the ECS node.

Mount an NFS export example

When you mount an export, you must ensure that the following prerequisites steps are carried out:

- The bucket owner name is mapped to a Unix UID.
- A default group is assigned to the bucket and the name of the default group is mapped to a Linux GID. This ensures that the default group shows as the associated Linux group when the export is mounted.

The following steps provide and an example of how to mount an ECS NFS export file system.

1. Create a directory on which to mount the export. The directory should belong to the same owner as the bucket.
   In this example, the user `fred` creates a directory `/home/fred/nfsdir` on which to mount an export.

```
su - fred
mkdir /home/fred/nfsdir
```

2. As the root user, mount the export in the directory mount point that you created.

```
mount -t nfs -o "vers=3,nolock" 10.247.179.162:/s3/tc-nfs6 /home/fred/nfsdir
```

   When mounting an NFS export, you can specify the name or IP address of any of the nodes in the VDC or the address of the load balancer.

   It is important that you specify `-o "vers=3"`.

3. Check that you can access the file system as user `fred`.

   a. Change to user `fred`.

```
$ su - fred
```

   b. Check you are in the directory in which you created the mount point directory.

```
$ pwd
/home/fred
```

   c. List the directory.

```
fred@lrmh229:~$ ls -al
total
drwxr-xr-x  7 fred  fredsgroup    4096 May 31 05:38 .
drwxr-xr-x 18 root  root         4096 May 30 04:03 ..
-rw-------  1 fred  fred           16 May 31
05:31 .bash_history
drwxrwxrwx  3 fred  anothergroup   96 Nov 24  2015 nfsdir
```

   In this example, the bucket owner is `fred` and a default group, `anothergroup`, was associated with the bucket.

If no group mapping had been created, or no default group has been associated with the bucket, you will not see a group name but a large numeric value, as shown below.

```
fred@lrmh229:~$ ls -al
total
drwxr-xr-x  7 fred  fredssgroup  4096 May 31 05:38 .
drwxr-xr-x 18 root  root         4096 May 30 04:03 ..
-rw-------  1 fred  fred           16 May 31 05:31 .bash_history
drwxrwxrwx  3 fred  2147483647     96 Nov 24  2015 nfsdir
```

If you have forgotten the group mapping, you can create appropriate mapping in the ECS Portal.

You can find the group ID by looking in `/etc/group`.

```
fred@lrmh229:~$ cat /etc/group | grep anothergroup
anothergroup:x:1005:
```

And adding a mapping between the name and GID (in this case: `anothergroup => GID 1005`).

If you try and access the mounted file system as the root user, or another user that does not have permissions on the file system, you will see `?`, as below.

```
root@lrmh229:~# cd /home/fred
root@lrmh229:/home/fred# ls -al
total
drwxr-xr-x  8 fred  fredsgroup 4096 May 31 07:00 .
drwxr-xr-x 18 root  root       4096 May 30 04:03 ..
-rw-------  1 fred  fred       1388 May 31 07:31 .bash_history
d?????????  ? ?     ?            ?            ?   nfsdir
```

Best practice when using ECS NFS

The following recommendations apply when mounting ECS NFS exports.

Use async
Whenever possible you should use the "async" mount option. Using this option dramatically reduces latency and improves throughput and reduces the number of connections from the client.

Set wsize and rsize to reduce round trips from the client
Where you are expecting to read and/or write large files, you should ensure that the read or write size of files is set appropriately using the rsize and wsize mount options. It is generally recommended that you set the wsize and rsize to the highest possible value to reduce the number of round trips from the client. The is typically 512KB (524288 B).

For example, to write a 10MB file, if the wsize is set to 524288 (512KB) the client would make 20 separate calls, whereas, if the write size had been set as 32KB this would result in 16 times as many calls.

When using the mount command, you can supply the read and write size using the options (-o) switch. For example:

```
# mount 10.247.97.129:/home /home -o
"vers=3,nolock,rsize=524288,wsize=524288"
```

Permissions for multi-protocol (cross-head) access

Objects can be accessed using NFS and using the object service. Each access method has a way of storing permissions: Object Access Control List (ACL) permissions and File System permissions.

When an object is created or modified using the object protocol, the permissions associated with the object owner are mapped to NFS permissions and the corresponding permissions are stored. Similarly, when an object is created or modified using NFS, ECS maps the NFS permissions of the owner to object permissions and stores them.

The S3 object protocol does not have the concept of groups. Changes to group ownership or permissions from NFS do not need to be mapped to corresponding object permissions. When you create a bucket or an object within a bucket (the equivalent of a directory and a file), ECS can assign Unix group permissions, and they can be accessed by NFS users.

For NFS, the following ACL attributes are stored:

- Owner
- Group
- Other

For object access, the following ACLs are stored:

- Users
- Custom Groups
- Groups (Pre-defined)
- Owner (a specific user from Users)
- Primary Group (a specific group from Custom Groups)

For more information on ACLs, see Set ACLs on page 90.

The following table shows the mapping between NFS ACL attributes and object ACL attributes.

NFS ACL attribute	Object ACL attribute
Owner	User who is also Owner
Group	Custom Group that is also Primary Group
Others	Pre-Defined Group

Examples of this mapping are discussed later in this topic.

The following Access Control Entries (ACE) can be assigned to each ACL attribute.

NFS ACEs:

- Read (R)
- Write (W)

- Execute (X)

Object ACEs:

- Read (R)
- Write (W)
- Execute (X)
- ReadAcl (RA)
- WriteAcl (WA)
- Full Control (FC)

Creating and modifying an object using NFS and accessing using the object service

When an NFS user creates an object using the NFS protocol, the owner permissions are mirrored to the ACL of the object user who is designated as the owner of the bucket. If the NFS user has RWX permissions, Full Control is assigned to the object owner through the object ACL.

The permissions that are assigned to the group that the NFS file or directory belongs to are reflected onto a custom group of the same name, if it exists. ECS reflects the permissions associated with Others onto pre-defined groups permissions.

The following example illustrates the mapping of NFS permissions to object permissions.

```
NFS ACL  Setting               Object ACL      Setting

Owner    John : RWX            Users           John : Full Control
Group    ecsgroup : R-X --->   Custom Groups   ecsgroup : R-X
Other    RWX                   Groups          All_Users : R, RA
                               Owner           John
                               Primary Group   ecsgroup
```

When a user accesses ECS using NFS and changes the ownership of an object, the new owner inherits the owner ACL permissions and is given Read_ACL and Write_ACL. The previous owner permissions are kept in the object user's ACL.

When a `chmod` operation is performed, the ECS reflects the permissions in the same way as when creating an object. Write_ACL is preserved in Group and Other permissions if it already exists in the object user's ACL.

Creating and modifying objects using the object service and accessing using NFS

When an object user creates an object using the object service, the user is the object owner and is automatically granted Full Control of the object. The file owner is granted RWX permissions. If the owner permissions are set to other than Full Control, ECS reflects the object RWX permissions onto the file RWX permissions. An object owner with RX permissions results in an NFS file owner with RX permissions. The object primary group, which is set using the Default Group on the bucket, becomes the Custom Group that the object belongs to and the object permissions are set based on the default permissions that have been set. These permissions are reflected onto the NFS.group permissions. If the object Custom Group has Full Control, these permissions become the RWX permissions for the NFS group. If pre-defined groups are specified on the bucket, these are applied to the object and are reflected as Others permissions for the NFS ACLs.

The following example illustrates the mapping of object permissions onto NFS permissions.

```
Object ACL      Setting                    NFS ACL
Setting

Users           John : Full Control        Owner    John : RWX
Custom Groups   ecsgroup : R-X      ----> Group    ecsgroup : R-X
Groups          All_Users : R, RA          Other    RWX
Owner           John
Primary Group   ecsgroup
```

If the object owner is changed, the permissions associated with the new owner applied to the object and reflected onto the file RWX permissions .

File API summary

NFS access can be configured and managed using the ECS Management REST API.

The following table provides a summary of the available APIs.

Method	Description
POST /object/nfs/exports	Creates an export. The payload specifies the export path, the hosts that can access the export, and a string that defines the security settings for the export.
PUT/GET/DELETE /object/nfs/exports/{id}	Performs the selected operation on the specified export.
GET /object/nfs/exports	Retrieves all user exports that are defined for the current namespace.
POST /object/nfs/users	Creates a mapping between an ECS object user name or group name and a Unix user or group ID.
PUT/GET/DELETE /object/nfs/users/{mappingid}	Performs the selected operation on the specified user or group mapping.
GET /object/nfs/users	Retrieves all user mappings that are defined for the current namespace.

The API documentation provides full details of the API and the documentation for the NFS export methods can be accessed in the ECS API Reference.

CHAPTER 9

Certificates

Introduction to certificates

ECS ships with an SSL certificate installed in the keystore for each node. This certificate is not trusted by applications that talk to ECS, or by the browser when users access ECS through the ECS Portal.

To prevent users from seeing an untrusted certificate error, or to allow applications to communicate with ECS, you should install a certificate signed by a trusted Certificate Authority (CA). You can generate a self-signed certificate to use until you have a CA signed certificate. The self-signed certificate is installed into the certificate store of any machines that will access ECS.

ECS uses the following types of SSL certificates:

Management certificates

Used for management requests using the ECS Management REST API. These HTTPS requests use port 4443.

Object certificates

Used for requests using the supported object protocols. These HTTPS requests use ports 9021 (S3), 9023 (Atmos), 9025 (Swift).

You can upload a self-signed certificate, a certificate signed by a CA authority, or, for an object certificate, you can request ECS to generate a certificate or you. The key/certificate pairs can be uploaded to ECS by using the ECS Management REST API on port 4443.

The following topics explain how to create, upload, and verify certificates:

- Generate certificates on page 128
- Upload a certificate on page 134
- Verify installed certificates on page 137

Generate certificates

You can generate a self-signed certificate, or you can purchase a certificate from a certificate authority (CA). The CA-signed certificate is strongly recommended for production purposes because it can be validated by any client machine without any extra steps.

Certificates must be in PEM-encoded x509 format.

When you generate a certificate, you typically specify the hostname where the certificate is used. Because ECS has multiple nodes, and each node has its own hostname, installing a certificate created for a specific hostname could cause a common name mismatch error on the nodes that do not have that hostname. You can create certificates with alternative IPs or hostnames called Subject Alternative Names (SANs).

For maximum compatibility with object protocols, the Common Name (CN) on your certificate must point to the wildcard DNS entry used by S3, because S3 is the only protocol that utilizes virtually-hosted buckets (and injects the bucket name into the hostname). You can specify only one wildcard entry on an SSL certificate and it must be under the CN. The other DNS entries for your load balancer for the Atmos and Swift protocols must be registered as a Subject Alternative Names (SANs) on the certificate.

The topics in this section show how to generate a certificate or certificate request using `openssl`, however, your IT organization may have different requirements or procedures for generating certificates.

Create a private key

You must create a private key to sign self-signed certificates and to create signing requests.

SSL uses public-key cryptography which requires a private and a public key. The first step in configuring it is to create a private key. The public key is created automatically, using the private key, when you create a certificate signing request or a certificate. The following steps describe how to use the openssl tool to create a private key.

Procedure

1. Log in to an ECS node or to a node that you can connect to the ECS cluster.
2. Use the `openssl` tool to generate a private key.

   For example, to create a key called `server.key`, use:

   ```
   openssl genrsa -des3 -out server.key 2048
   ```

3. When prompted, enter a passphrase for the private key and reenter it to verify. You will need to provide this passphrase when creating a self-signed certificate or a certificate signing request using the key.

   You must create a copy of the key with the passphrase removed before uploading the key to ECS. For more information, see Upload a certificate on page 134.

4. Set the permissions on the key file.

   ```
   chmod 0400 server.key
   ```

Generate a SAN configuration

If you want your certificates to support Subject Alternative Names (SANs), you must define the alternative names in a configuration file.

OpenSSL does not allow you to pass Subject Alternative Names (SANs) through the command line, so you must add them to a configuration file first. To do this, you must locate your default OpenSSL configuration file. On Ubuntu, it is located at `/usr/lib/ssl/openssl.cnf`.

Procedure

1. Create the configuration file.

   ```
   cp /usr/lib/ssl/openssl.cnf request.conf
   ```

2. Edit the configuration file with a text editor and make the following changes.

   a. Add the [alternate_names] .

   For example:

   ```
   [ alternate_names ]
   DNS.1 = os.example.com
   ```

```
DNS.2 = atmos.example.com
DNS.3 = swift.example.com
```

Note

There is a space between the bracket and the name of the section.

If you are uploading the certificates to ECS nodes rather than to a load balancer, the format is:

```
[ alternate_names ]
IP.1 = <IP node 1>
IP.2 = <IP node 2>
IP.3 = <IP node 3>
...
```

b. In the section [v3_ca], add the following lines:

```
subjectAltName    = @alternate_names
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
```

The following line is likely to already exist in this [v3_ca] section. If you create a certificate signing request, you must comment it out as shown:

```
#authorityKeyIdentifier=keyid:always,issuer
```

c. In the [req] section, add the following lines:

```
x509_extensions = v3_ca    #for self signed cert
req_extensions = v3_ca     #for cert signing req
```

d. In the section [CA_default], uncomment or add the line:

```
copy_extension=copy
```

Create a self-signed certificate

You can create a self-signed certificate.

Before you begin

- You must create a private key using the procedure in Create a private key on page 129.
- To create certificates that use SAN, you must create a SAN configuration file using the procedure in Generate a SAN configuration on page 129.

Procedure

1. Use the private key to create a self-signed certificate.

   Two ways of creating the signing request are shown. One for use if you have already prepared a SAN configuration file to specify the alternative server name, another if you have not.

   If you are using SAN:

```
openssl req -x509 -new -key server.key -config request.conf -out server.crt
```

   If you are not, use:

```
openssl req -x509 -new -key server.key -out server.crt
```

   Example output.

```
Signature ok
subject=/C=US/ST=GA/
```

2. Enter the pass phrase for your private key.
3. At the prompts, enter the fields for the DN for the certificate.

   Most fields are optional. You must enter a Common Name (CN).

   Note

   The CN should be a FQDN. Even if you install the certificate on the ECS nodes, you must use an FQDN and all of the IP addresses must be in the alternate names section.

   You will see the following prompts:

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Acme
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:*.acme.com
Email Address []:
```

4. Enter the Distinguished Name (DN) details when prompted. More information on the DN fields are provided in Distinguished Name (DN) fields on page 132.

5. View the certificate.

```
openssl x509 -in server.crt -noout -text
```

Distinguished Name (DN) fields

The following table describes the fields that comprise the Distinguished Name (DN).

Name	Description	Example
Common Name (CN)	The fully qualified domain name (FQDN) of your server. This is the name that you specified when you installed the ECS appliance.	*.yourco.com ecs1.yourco.com
Organization	The legal name of your organization. This must not be abbreviated and should include suffixes such as Inc, Corp, or LLC.	Yourco Inc.
Organizational Unit	The division of your organization handling the certificate.	IT Department
Locality/City	The state/region where your organization is located. This must not be abbreviated.	Mountain View
State/Province	The city where your organization is located.	California
Country	The two-letter ISO code for the country where your organization is location.	US
Email address	An email address to contact your organization.	contact@yourco.com

Create a certificate signing request

You can create a certificate signing request to submit to a CA to obtain a signed certificate.

Before you begin

- You must create a private key using the procedure in Create a private key on page 129.
- To create certificates that use SAN, you must create a SAN configuration file using the procedure in Generate a SAN configuration on page 129.

Procedure

1. Use the private key to create a certificate signing request.

   Two ways of creating the signing request are shown. One for if you have already prepared a SAN configuration file to specify the alternative server name, another if you have not.

   If you are using SAN:

```
openssl req -new -key server.key -config request.conf -out server.csr
```

   If you are not, use:

```
openssl req -new -key server.key -out server.csr
```

When creating a signing request, you are asked to supply the Distinguished Name (DN) which comprises a number of fields. Only the Common Name is required and you can accept the defaults for the other parameters.

2. Enter the pass phrase for your private key.
3. At the prompts, enter the fields for the DN for the certificate.

   Most fields are optional. However, you must enter a Common Name (CN).

   Note

   The CN should be a FQDN. Even if you install the certificate on the ECS nodes, you must use an FQDN and all of the IP addresses must be in the alternate names section.

   You will see the following prompts:

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Acme
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:*.acme.com
Email Address []:
```

   More information on the DN fields are provided in Distinguished Name (DN) fields on page 132.

4. You are prompted to enter an optional challenge password and a company name.

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

5. View the certificate.

```
openssl req -in server.csr -text -noout
```

Results

You can submit the certificate signing request it to your CA who will return a signed certificate file.

Upload a certificate

You can upload management or data certificates to ECS. Whichever type of certificate you upload, you must authenticate with the API.

- Authenticate with the ECS Management REST API on page 134
- Upload a management certificate on page 134
- Upload a data certificate for data access endpoints on page 136

Authenticate with the ECS Management REST API

To run ECS Management REST API commands, you must first authenticate with the API service and obtain an authentication token.

Procedure

1. Authenticate with the ECS Management REST API and obtain an authentication token that can be used when using the API to upload or verify certificates.

   a. Run the following command:

```
export TOKEN=`curl -s -k -v -u <user>:<password> https://$(hostname -i):4443/login
2>&1 | grep X-SDS-AUTH-TOKEN | awk '{print $2, $3}'`
```

   The username and password are those used to access the ECS Portal. The public_ip is the public IP address of the node.

   b. Verify the token exported correctly.

```
echo $TOKEN
```

   Example output:

```
X-SDS-AUTH-TOKEN:
BAAcTGZjUjJ2Zm1iYURSUFZzKzhBSVVPQVFDRUUwPQMAjAQASHVybjpzdG9yYWdlb3M6VmlydHVhbERhdGF
DZW50ZXJEYXRhOjcxYjA1ZTgwLTNkNzktND
dmMC04OThhLWI2OTU4NDk1YmVmYgIADTE0NjQ3NTM2MjgzMTIDAC51cm46VG9rZW46YWMwN2Y0NGYtMjE5O
S00ZjA4LTgyM2EtZTAwNTc3ZWI0NDAyAgAC
0A8=
```

Upload a management certificate

You can upload a management certificate which is used to authenticate access to management endpoints, such as the ECS Portal and the ECS Management REST API.

Before you begin

- Ensure that you have authenticated with the ECS Management REST API and stored the token in a variable (`$TOKEN`) as described in Authenticate with the ECS Management REST API on page 134.
- Ensure that the machine that you use has a suitable REST client (such as curl) and can access the ECS nodes using the ECS Management REST API.
- Ensure your private key and certificate are available on the machine from which you intend to perform the upload.

Procedure

1. Ensure that your private key does not have a passphrase.

   If it does, you can create a copy with the passphrase stripped, by typing the following command:

   ```
   openssl rsa -in server.key -out server_nopass.key
   ```

2. Upload the keystore for the data path using your private key and signed certificate.

   Using curl:

```
curl -svk -H "$TOKEN" -H "Content-type: application/xml" -H "X-EMC-REST-CLIENT: TRUE"
-X PUT -d "<rotate_keycertchain>
<key_and_certificate><private_key>`cat privateKeyFile` <private_key>`</
private_key><certificate_chain>`cat
certificateFile`</certificate_chain></key_and_certificate></rotate_keycertchain>"
https://<ecs_node_address>:4443/vdc/keystore
```

   Using the ECS command line interface (`ecscli.py`):

```
python ecscli.py vdc_keystore update -hostname <ecs_host_ip> -port 4443
-cf <cookiefile> -privateKey privateKeyFile -certificateChain certificateFile
```

   The *privateKeyFile*, for example `<path>/server_nopass.key`, and *certificateFile*, for example `<path>/server.crt`, must be replaced with the path to the key and certificate files.

3. Log in to one of the ECS nodes as the admin user.
4. Verify that the `MACHINES` file has all nodes in it.

   The `MACHINES` file is used by ECS wrapper scripts that execute commands on all nodes, such as `viprexec`.

   The `MACHINES` file is in `/home/admin`.

   a. Display the contents of the `MACHINES` file.

   ```
   cat /home/admin/MACHINES
   ```

   b. If the `MACHINES` file does not contain all nodes, recreate it.

   ```
   getrackinfo -c MACHINES
   ```

   Verify that the `MACHINES` file now contains all nodes.

5. Restart the `objcontrolsvc` and `nginx` services once the management certificates are applied.

   a. Restart the object service.

```
viprexec -f ~/MACHINES -i 'pidof objcontrolsvc;
kill `pidof objcontrolsvc`; sleep 60; pidof objcontrolsvc'
```

b. Restart the `nginx` service.

```
sudo -i viprexec -i -c "/etc/init.d/nginx restart;sleep 60;/etc/init.d/nginx
status"
```

After you finish

You can verify the certificate has uploaded correctly using the following procedure: Verify the management certificate on page 137.

Upload a data certificate for data access endpoints

You can upload a data certificate which is used to authenticate access for the S3, EMC Atmos, or OpenStack Swift protocols.

Before you begin

- Ensure that you have authenticated with the ECS Management REST API and stored the token in a variable (`$TOKEN`). See Authenticate with the ECS Management REST API on page 134.
- Ensure that the machine that you use has a suitable REST client (such as curl) and can access the ECS nodes using the ECS Management REST API.
- Ensure your private key and certificate are available on the machine from which you intend to perform the upload.

Procedure

1. Ensure that your private key does not have a pass phrase.

   If it does, you can create a copy with the pass phrase stripped, using:

   ```
   openssl rsa -in server.key -out server_nopass.key
   ```

2. Upload the keystore for the data path using your private key and signed certificate.

   ```
   curl -svk -H "$TOKEN" -H "Content-type: application/xml" -H
   "X-EMC-REST-CLIENT: TRUE" -X PUT -d "<rotate_keycertchain>
   <key_and_certificate><private_key>`cat privateKeyFile`</
   private_key><certificate_chain>`cat
   certificateFile`</certificate_chain></key_and_certificate></
   rotate_keycertchain>"
   https://<ecs_node_address>:4443/object-cert/keystore
   ```

   Using the ECS command line interface (`ecscli.py`):

   ```
   python ecscli.py keystore update -hostname <ecs_host_ip> -
   port 4443 -cf <cookiefile>
   -pkvf privateKeyFile -cvf certificateFile -ss false
   ```

   The *privateKeyFile*, for example <*path*>`/server_nopass.key`, and *certificateFile*, for example <*path*>`/server.crt`, must be replaced with the path to the key and certificate files.

3. The certificate is distributed when the `dataheadsvc` service is restarted. You can do this with the commands below.

Note

You do not need to restart the services when changing data certificate, `dataheadsvc` is restarted automatically on each node two hours from certificate update.

```
ssh admin@<ecs_ip_where_cert_uploaded>
```

```
sudo kill `pidof dataheadsvc`
```

After you finish

You can verify that the certificate has correctly uploaded using the following procedure: Verify the object certificate on page 138.

Verify installed certificates

The object certificate and management certificate each has an ECS Management REST API GET request to retrieve the installed certificate.

- Verify the management certificate on page 137
- Verify the object certificate on page 138

Verify the management certificate

You can retrieve the installed management certificate using the ECS Management REST API.

Before you begin

- Ensure that you have authenticated with the ECS Management REST API and stored the token in a variable (`$TOKEN`). See Authenticate with the ECS Management REST API on page 134.
- If you have restarted services, the certificate is available immediately. Otherwise, you must wait two hours to be sure that the certificate is propagated to all nodes.

Procedure

1. Use the `GET /vdc/keystore` method to return the certificate.

   Using the curl tool, the method can be run by typing the following:

```
curl -svk -H "X-SDS-AUTH-TOKEN: $TOKEN" https://x.x.x.x:4443/vdc/keystore
```

   Using the ECS command line interface (`ecscli.py`):

```
python ecscli.py vdc_keystore get -hostname <ecs_host_ip> -port 4443 -cf <cookiefile>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><certificate_chain><chain>
-----BEGIN CERTIFICATE-----
MIIDgjCCAmoCCQCEDeNwcGsttTANBgkqhkiG9w0BAQUFADCBgjELMAkGA1UEBhMC&#xD;
VVMxCzAJBgNVBAgMAkdBMQwwCgYDVQQHDANBVEwxDDAKBgNVBAoMA0VNQzEMMAoG&#xD;
```

```
A1UECwwDRU5HMQ4wDAYDVQQDDAVjaHJpczEsMCoGCSqGSIb3DQEJARYdY2hyaXN0&#xD;
b3BoZXIuZ2hva2FzaWFuQGVtYy5jb20wHhcNMTYwNjAxMTg0MTIyWhcNMTcwNjAy&#xD;
MTg0MTIyWjCBgjELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAkdBMQwwCgYDVQQHDANB&#xD;
VEwxDDAKBgNVBAoMA0VNQzEMMAoGA1UECwwDRU5HMQ4wDAYDVQQDDAVjaHJpczEs&#xD;
MCoGCSqGSIb3DQEJARYdY2hyaXN0b3BoZXIuZ2hva2FzaWFuQGVtYy5jb20wggEi&#xD;
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDb9WtdcW5HJpIDOuTB7o7ic0RK&#xD;
dwA4dY/nJXrk6Ikae5zDWO8XH4noQNhAu8FnEwS5kjtBK1hGI2GEFBtLkIH49AUp&#xD;
c4KrMmotDmbCeHvOhNCqBLZ5JM6DACfO/elHpb2hgBENTd6zyp7mz/7MUf52s9Lb&#xD;
x5pRRCp1iLDw3s15iodZ5GL8pRT62puJVK1do9mPfMoL22woR3YB2++AkSdAgEFH&#xD;
1XLIsFGkBsEJObbDBoEMEjEIivnTRPiyocyWki6gfLh50u9Y9B2GRzLAzIlgNiEs&#xD;
L/vyyrHcwOs4up9QqhAlvMn3Al01VF+OH0omQECSchBdsc/R/Bc35FAEVdmTAgMB&#xD;
AAEwDQYJKoZIhvcNAQEFBQADggEBAAyYcvJtEhOq+n87wukjPMgC7l9n7rgvaTmo&#xD;
tzpQhtt6kFoSBO7p//76DNzXRXhBDADwpUGG9S4tgHChAFu9DpHFzvnjNGGw83ht&#xD;
qcJ6JYgB2M3lOQAssgW4fU6VD2bfQbGRWKy9G1rPYGVsmKQ59Xeuvf/cWvplkwW2&#xD;
bKnZmAbWEfE1cEOqt+5m20qGPcf45B7DPp2J+wVdDD7N8198Jj5HJBJt3T3aUEwj&#xD;
kvnPx1PtFM9YORKXFX2InF3UOdMs0zJUkhBZT9cJ0gASi1w0vEnx850secu1CPLF&#xD;
WB9G7R5qHWOXlkbAVPuFN0lTav+yrr8RgTawAcsV9LhkTTOUcqI=&#xD;
-----END CERTIFICATE-----</chain></certificate_chain>
```

2. You can verify the certificate using `openssl` on all nodes.

```
openssl s_client -showcerts -connect <node_ip>:<port>
```

Note

The management port is 4443.

For example:

```
openssl s_client -showcerts -connect 10.1.2.3:4443
```

Verify the object certificate

You can retrieve the installed object certificate using the ECS Management REST API.

Before you begin

- Ensure that you have authenticated with the ECS Management REST API and stored the token in a variable (`$TOKEN`). See Authenticate with the ECS Management REST API on page 134.
- If you have restarted services, the certificate will be available immediately. Otherwise, you need to wait two hours to be sure that the certificate has propagated to all nodes.

Procedure

1. Use the `GET /object-cert/keystore` method to return the certificate.

   Using the curl tool, the method can be run by typing the following:

```
curl -svk -H "X-SDS-AUTH-TOKEN: $TOKEN" https://x.x.x.x:4443/object-cert/keystore
```

   Using the ECS command line interface (`ecscli.py`):

```
python ecscli.py keystore show -hostname <ecs_host_ip> -port 4443 -cf <cookiefile>
```

2. You can verify the certificate using `openssl` on all nodes.

```
openssl s_client -showcerts -connect <node_ip>:<port>
```

Note

Ports are: s3: 9021, Atmos: 9023, Swift: 9025

Example:

```
openssl s_client -showcerts -connect 10.1.2.3:9021
```

CHAPTER 10

ECS Settings

Introduction to ECS settings

This section describes the settings that the System Administrator can view and configure in the **Settings** section of the ECS Portal. These settings include:

- Object base URL
- Password
- ESRS
- Event notification
- Platform locking
- Licensing
- About this VDC

Object base URL

ECS supports Amazon S3 compatible applications that use virtual host style and path style addressing schemes. In multitenant configurations, ECS allows the namespace to be provided in the URL.

The base URL is used as part of the object address where virtual host style addressing is used and enables ECS to know which part of the address refers to the bucket and, optionally, namespace.

For example, if you are using an addressing scheme that includes the namespace so that you have addresses of the form `mybucket.mynamespace.mydomain.com`, you must tell ECS that `mydomain.com` is the base URL so that ECS identifies `mybucket.mynamespace` as the bucket and namespace.

By default, the base URL is set to `s3.amazonaws.com`.

An ECS System Administrator can add a base URL by using the ECS Portal or by using the ECS Management REST API.

The following topics describe the addressing schemes supported by ECS, how ECS processes API requests from S3 applications, how the addressing scheme affects DNS resolution, and how to add a base URL in the ECS Portal.

- Bucket and namespace addressing on page 142
- DNS configuration on page 144
- Add a Base URL on page 145

Bucket and namespace addressing

When an S3 compatible application makes an API request to perform an operation on an ECS bucket, ECS can identify the bucket in several ways.

For authenticated API requests, ECS infers the namespace by using the namespace that the authenticated user is a member of. To support anonymous, unauthenticated requests that require CORS support or anonymous access to objects, you must include the namespace in the address so that ECS can identify the namespace for the request.

When the user scope is `NAMESPACE`, the same user ID can exist in multiple namespaces (for example, namespace1/user1 and namespace2/user1). Therefore, you

must include the namespace in the address. ECS cannot infer the namespace from the user ID.

Namespace addresses require wildcard DNS entries (for example, `*.ecs1.yourco.com`) and also wildcard SSL certificates to match if you want to use HTTPS. Non-namespace addresses and path style addresses do not require wildcards since there is only one hostname for all traffic. If you use non-namespace addresses with virtual host style buckets, you will still need wildcard DNS entries and wildcard SSL certificates.

You can specify the namespace in the `x-emc-namespace` header of an HTTP request. ECS also supports extraction of the location from the host header.

Virtual host style addressing

In the virtual host style addressing scheme, the bucket name is in the hostname. For example, you can access the bucket named `mybucket` on host `ecs1.yourco.com` using the following address:

`http://mybucket.ecs1.yourco.com`

You can also include a namespace in the address.

Example: `mybucket.mynamespace.ecs1.yourco.com`

To use virtual host style addressing, you must configure the base URL in ECS so that ECS can identify which part of the URL is the bucket name. You must also ensure that the DNS system is configured to resolve the address. For more information on DNS configuration, see DNS configuration on page 144

Path style addressing

In the path style addressing scheme, the bucket name is added to the end of the path.

Example: `ecs1.yourco.com/mybucket`

You can specify a namespace by using the `x-emc-namespace` header or by including the namespace in the path style address.

Example: `mynamespace.ecs1.yourco.com/mybucket`

ECS address processing

When ECS processes a request from an S3 compatible application to access ECS storage, ECS performs the following actions:

1. Try to extract the namespace from the `x-emc-namespace` header. If found, skip the steps below and process the request.
2. Get the hostname of the URL from the host header and check if the last part of the address matches any of the configured base URLs.
3. Where there is a BaseURL match, use the prefix part of the hostname (the part left when the base URL is removed), to obtain the bucket location.

The following examples demonstrate how ECS handles incoming HTTP requests with different structures.

Note

When you add a base URL to ECS, you can specify whether or not your URLs contain a namespace in the **Use base URL with Namespace** field on the **New Base URL** page in the ECS Portal. This tells ECS how to treat the bucket location prefix. For more information, see Add a Base URL on page 145

Example 1: Virtual Host Style Addressing, Use base URL with Namespace is enabled

```
Host:          baseball.image.yourco.finance.com
BaseURL:       finance.com
Use BaseURL with namespace enabled

Namespace:     yourco
Bucket Name:   baseball.image
```

Example 2: Virtual Host Style Addressing, Use base URL with Namespace is disabled

```
Host:          baseball.image.yourco.finance.com
BaseURL:       finance.com
Use BaseURL without namespace enabled

Namespace:     null (Use other methods to determine namespace)
Bucket Name:   baseball.image.yourco
```

Example 3: ECS treats this request as a path style request

```
Host:          baseball.image.yourco.finance.com
BaseURL:       not configured

Namespace:     null (Use other methods to determine namespace.)
Bucket Name:   null (Use other methods to determine the bucket
name.)
```

DNS configuration

In order for an S3 compatible application to access ECS storage, you must ensure that the URL resolves to the address of the ECS data node, or the data node load balancer.

If your application uses path style addressing, you must ensure that your DNS system can resolve the address. For example, if your application issues requests in the form `ecs1.yourco.com/bucket`, you must have a DNS entry that resolves `ecs1.yourco.com` to the IP address of the load balancer that is used for access to the ECS nodes. If your application is configured to talk to Amazon S3, the URI is in the form `s3-eu-west-1.amazonaws.com`.

If your application uses virtual host style addressing, the URL includes the bucket name and can include a namespace. Under these circumstances, you must have a DNS entry that resolves the virtual host style address by using a wildcard in the DNS entry. This also applies where you are using path style addresses that include the namespace in the URL.

For example, if the application issues requests in the form `mybucket.s3.yourco.com`, you must have the following DNS entries.

- `ecs1.yourco.com`
- `*.ecs1.yourco.com`

If the application previously connected to the Amazon S3 service using `mybucket.s3.amazonaws.com`, you must have the following DNS entries.

- `s3.amazonaws.com`
- `*.s3.amazonaws.com`

These entries resolve the virtual host style bucket address and the base name when you issue service-level commands (for example, list buckets).

If you create an SSL certificate for the ECS S3 service, it must have the wildcard entry on the name of the certificate and the non-wildcard version as a Subject Alternate Name.

Add a Base URL

You must add a base URL if you use object clients that encode the location of an object, its namespace (optional), and bucket in a URL.

Before you begin

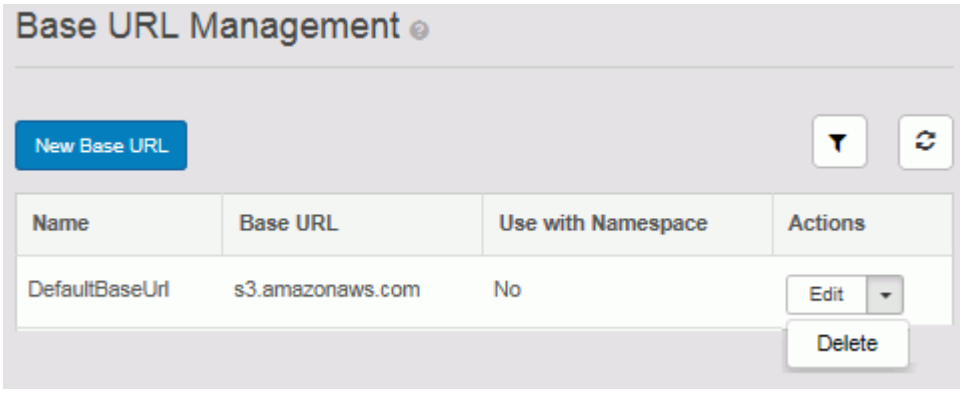
This operation requires the System Administrator role in ECS.

You must ensure that the domain specified in a request that uses a URL to specify an object location resolves to the location of the ECS data node or a load balancer that sits in front of the data nodes.

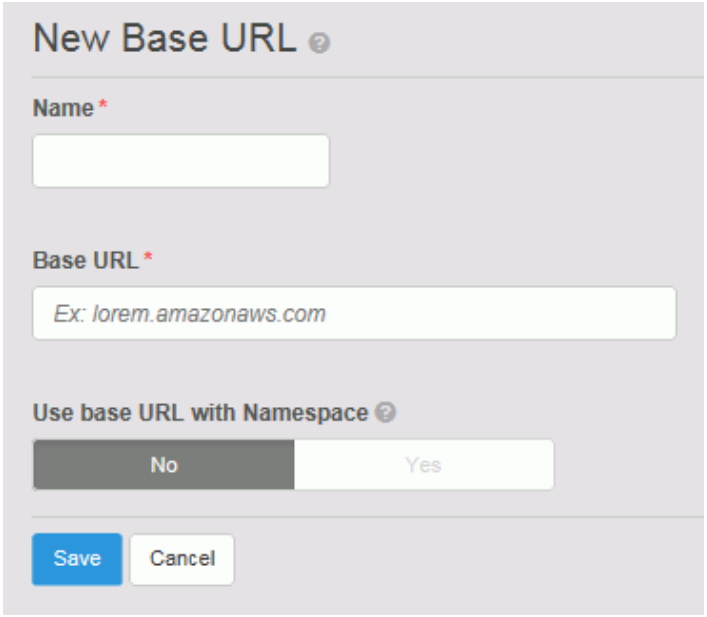
Procedure

1. In the ECS Portal, select **Settings** > **Object Base URL**.

   The **Base URL Management** page appears with a list of base URLs. The **Use with Namespace** property indicates whether your URLs include a namespace.



2. On the **Base URL Management** page, click **New Base URL**.



3. On the **New Base URL** page, in the **Name** field, type the name of the base URL.

   The name provides a label for this base URL in the list of base URLs on the **Base URL Management** page.

4. In the **Base URL** field, type the base URL.

   If your object URLs are in the form `https://mybucket.mynamespace.acme.com` (that is, `bucket.namespace.baseurl`) or `https://mybucket.acme.com` (that is, `bucket.baseurl`), the base URL would be `acme.com`.

5. In the **Use base URL with Namespace** field, click **Yes** if your URLs include a namespace.
6. Click **Save**.

Change password

You can use the ECS Portal to change your password.

Before you begin

This operation requires the System Administrator, Namespace Administrator, Lock Administrator (emcsecurity), or System Monitor role in ECS.

Procedure

1. In the ECS Portal, select **Settings** > **Change Password**
2. On the **Change Password** page, in the **Password** field, enter a new password, and then in the **Confirm Password** field, enter it again.
3. Click **Save**.

EMC Secure Remote Services (ESRS)

You can use the ECS Portal to add or update an ESRS server. EMC Secure Remote Services (ESRS) is designed to deliver immediate, secure response to ECS event

reports such as error alerts which can help increase the availability of your information infrastructure.

Before you begin

- The initial ESRS setup and configuration must have been completed. To perform the initial steps to configure ESRS on ECS, contact your customer support representative.
- If you already have an ESRS server enabled, you must delete it, then add the new server.
- This operation requires the System Administrator role in ECS.

Procedure

1. In the ECS Portal, select **Settings** > **ESRS** > **New Server**.
2. On the **New ESRS Server** page:
   a. In the **FQDN/IP** field, type the ESRS Gateway FQDN or IP address.
   b. In the **PORT** field, type the ESRS Gateway port (9443 by default).
   c. In the **Username** field, type the login user name used to interface with ECS support. This is the same login user name used to log in to `support.emc.com`.
   d. In the **Password** field, type the password set up with the login user name.
3. Click **Save**.

   Server connectivity may take a few minutes to complete. To monitor the process, click the refresh button. Possible states of transition are **Processing**, **Connected**, or **Failed**.
4. If ESRS is configured with more than one gateway for high availability, repeat steps 1 to 3 to add additional ESRS gateway servers.

Verify ESRS setup

You can verify that ESRS is set up and can successfully contact the customer support center.

You can test whether ESRS is working by generating a test alert and then checking whether the alert is received.

Procedure

1. To generate a test alert, complete the following steps.
   a. Authenticate with the ECS Management REST API.

      For example, using the curl command-line tool, type the following command.

```
curl -L --location-trusted -k https://192.0.2.49:4443/login -u "root:ChangeMe" -v
```

The `X-SDS-AUTH-TOKEN:` is used to authenticate with ECS to run ECS Management REST API commands.

```
X-SDS-AUTH-TOKEN:
BAAcaGtkTzRZU2k0SE5acVYxdFNCYUlUby9mOVM4PQMAjAQASHVybjpzdG9yYWdlb3M6VmlydHVhbERhdGF
DZW50ZXJEYXRhOjMwOTFjMDY1LTgzMDAtNGNlNi1iNDY3LTU5NDFiM2MyYTBmZAIADTE0NzM3NzkxMzA4OT
MDAC51cm46VG9rZW46M2Y4NTMzMzctN2ZiZi00NWRiLTk0M2YtY2NkYTc2OWQ0ZTc4AgAC0A8=
```

b. You can store the authentication token (`X-SDS-AUTH-TOKEN`) in an environment variable.

```
export AUTH_TOKEN="X-SDS-AUTH-TOKEN:
BAAcaGtkTzRZU2k0SE5acVYxdFNCYUlUby9mOVM4PQMAjAQASHVybjpzdG9yYWdlb3M6VmlydHVhbERhdGF
DZW50ZXJEYXRhOjMwOTFjMDY1LTgzMDAtNGNlNi1iNDY3LTU5NDFiM2MyYTBmZAIADTE0NzM3NzkxMzA4OT
MDAC51cm46VG9rZW46M2Y4NTMzMzctN2ZiZi00NWRiLTk0M2YtY2NkYTc2OWQ0ZTc4AgAC0A8="
```

c. Generate a test alert.

For example, using the `$AUTH_TOKEN` environment variable, type the following command.

The `user_str` parameter enables you to specify a test message, and the `contact` parameter enables you to supply an email address.

```
curl -ks -H "$AUTH_TOKEN" -H "Content-Type: application/json" -d '{"user_str":
"test alert > for ESRS", "contact": "test_user@yourco.com"}' https://
10.241.207.57:4443/vdc/callhome/alert | xmllint -format -
```

2. In the ECS Portal, check that the ESRS notification has been received.
3. Check that the latest test alert is present.
   a. SSH into the ESRS server.
   b. Go to the location of the RSC file.

```
cd /opt/connectemc/archive/
```

   c. Check for the latest RSC file, using:

```
ls -lrt RSC_<SERIAL NUMBER>*”
```

   d. Open the file and check whether the latest test alert is present in the description.

Event notification servers

You can add SNMPv2 servers, SNMPv3 servers, and Syslog servers to ECS to route SNMP and Syslog event notifications to external systems.

In ECS, you can add the following types of event notification servers:

- Simple Network Management Protocol (SNMP) servers, also known as SNMP agents, provide data about network-managed device status and statistics to SNMP Network Management Station clients. For more information, see SNMP servers.
- Syslog servers provide a method for centralized storage and retrieval of system log messages. ECS supports forwarding of alerts and audit messages to remote syslog servers, and supports operations using the BSD Syslog and Structured Syslog application protocols. For more information, see Syslog servers.

You can add event notification servers from the ECS Portal or by using the ECS Management REST API or CLI.

- Add an SNMPv2 trap recipient on page 149
- Add an SNMPv3 trap recipient on page 151
- Add a Syslog server on page 156

SNMP servers

Simple Network Management Protocol (SNMP) servers, also known as SNMP agents, provide data about network managed device status and statistics to SNMP Network Management Station clients.

To allow communication between SNMP agents and SNMP Network Management Station clients, you must configure both sides to use the same credentials. For SNMPv2, both sides must use the same Community name. For SNMPv3, both sides must use the same Engine ID, username, authentication protocol and authentication passphrase, and privacy protocol and privacy passphrase.

To authenticate traffic between SNMPv3 servers and SNMP Network Management Station clients, and to verify message integrity between hosts, ECS supports the SNMPv3 standard use of the following cryptographic hash functions:

- Message Digest 5 (MD5)
- Secure Hash Algorithm 1 (SHA-1)

To encrypt all traffic between SNMPv3 servers and SNMP Network Management Station clients, ECS supports encryption of SNMPv3 traffic by using the following cryptographic protocols:

- Digital Encryption Standard (using 56-bit keys)
- Advanced Encryption Standard (using 128-bit, 192-bit or 256-bit keys)

Note

Support for advanced security modes (AES192/256) provided by the ECS SNMP trap feature might be incompatible with certain SNMP targets (for example, iReasoning).

Add an SNMPv2 trap recipient

You can configure Network Management Station clients as SNMPv2 trap recipients for the SNMP traps that are generated by the ECS Fabric using SNMPv2 standard messaging.

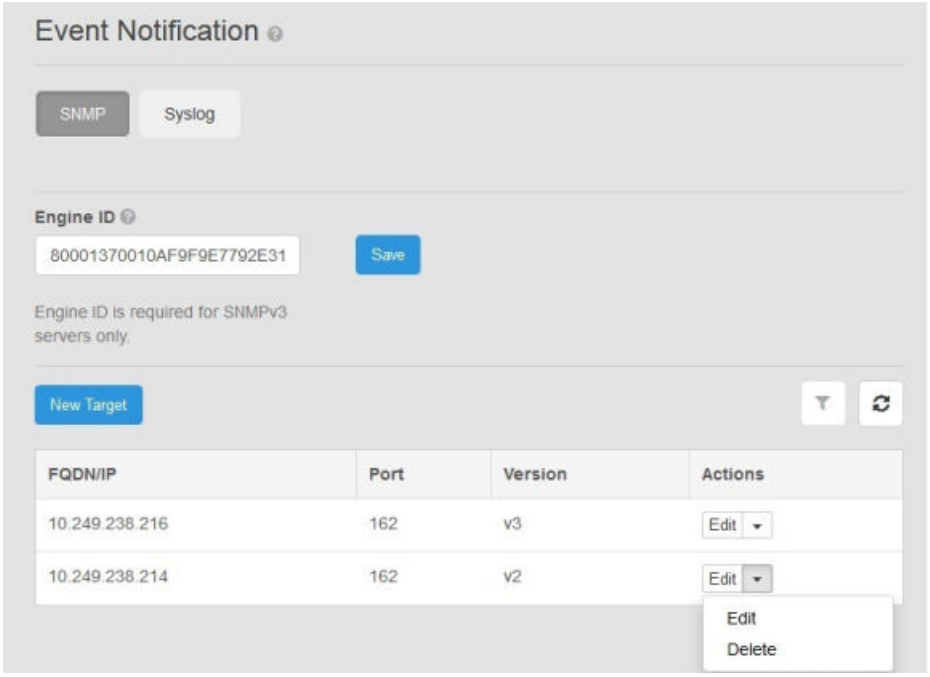
Before you begin

This operation requires the System Administrator role in ECS.

Procedure

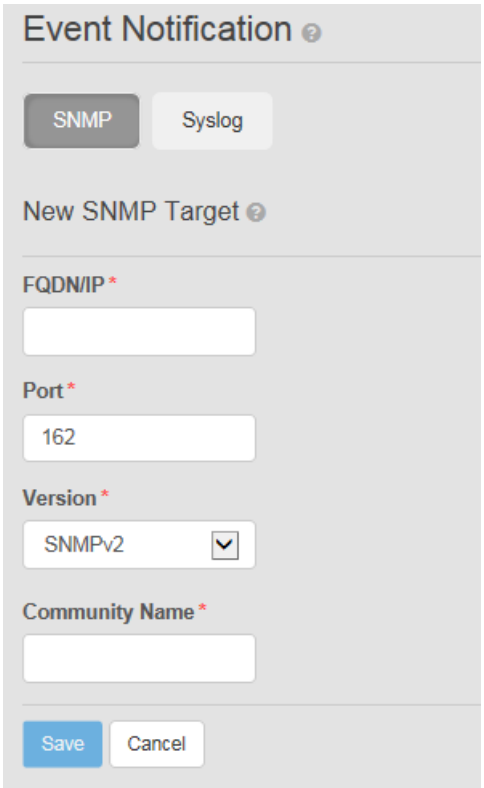
1. In the ECS Portal, select **Settings** > **Event Notification**.

   The **Event Notification** page appears with the **SNMP** tab open. This page lists the SNMP servers that have been added to ECS and allows you to configure SNMP server targets.



2. On the **Event Notification** page, click **New Target**.

   The **New SNMP Target** sub-page appears.



3. On the **New SNMP Target** sub-page, complete the following steps.

   a. In the **FQDN/IP** field, type the Fully Qualified Domain Name or IP address for the SNMP v2c trap recipient node that runs the `snmptrapd` server.

   b. In the **Port** field, type the port number of the SNMP v2c `snmptrapd` running on the Network Management Station clients.

The default port number is `162`.

c. In the **Version** field, select **SNMPv2**.

d. In the **Community Name** field, type the SNMP community name.

Both the SNMP server and any Network Management Station clients that access it must use the same community name in order to ensure authentic SNMP message traffic, as defined by the standards in RFC 1157 and RFC 3584.

The default community name is `public`.

4. Click **Save**.

Add an SNMPv3 trap recipient

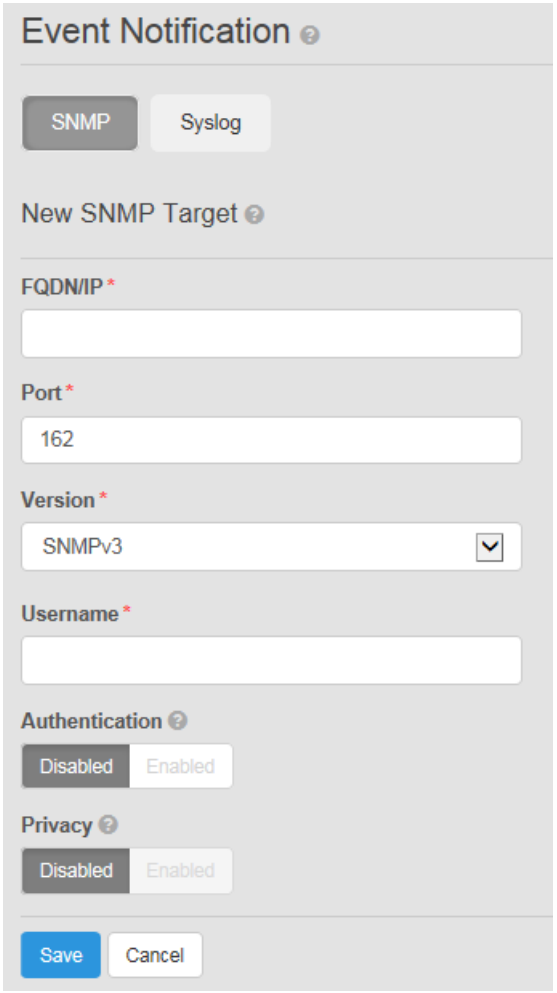
You can configure Network Management Station clients as SNMPv3 trap recipients for the SNMP traps that are generated by the ECS Fabric using SNMPv3 standard messaging.

Before you begin

This operation requires the System Administrator role in ECS.

Procedure

1. In the ECS Portal, select **Settings** > **Event Notification**.
2. On the **Event Notification** page, click **New Target**



3. On the **New SNMP Target** sub-page, complete the following steps.
   a. In the **FQDN/IP** field, type the Fully Qualified Domain Name or IP address for the SNMPv3 trap recipient node that runs the `snmptrapd` server.
   b. In the **Port** field, type the port number of the SNMP 3c `snmptrapd` running on the Network Management Station client.

      The default port number is `162`.
   c. In the **Version** field, select **SNMPv3**.
   d. In the **Username** field, type in the username that will be used in authentication and message traffic as per the User-based Security Model (USM) defined by RFC 3414.

      Both the SNMP server and any Network Management Station clients that access it must specify the same username in order to ensure communication. This is an octet string of up to 32 characters in length.
   e. In the **Authentication** box, click **Enabled** if you want to enable Message Digest 5 (MD5) (128-bit) or Secure Hash Algorithm 1 (SHA-1) (160-bit) authentication for all SNMPv3 data transmissions, and do the following:
      - In the **Authentication Protocol** field, select **MD5** or **SHA**.
        This is the cryptographic hash function to use to verify message integrity between hosts. The default is **MD5**.
      - In the **Authentication Passphrase** field, type the string to use as a secret key for authentication between SNMPv3 USM standard hosts, when calculating a message digest.
        The passphrase can be 16 octets long for MD5 and 20 octets long for SHA-1.

   f. In the **Privacy** box, click **Enabled** if you want to enable Digital Encryption Standard (DES) (56-bit) or Advanced Encryption Standard (AES) (128-bit, 192-bit or 256-bit) encryption for all SNMPv3 data transmissions, and do the following:
      - In the **Privacy Protocol** field, select **DES**, **AES128**, **AES192**, or **AES256**.
        This is the cryptographic protocol to use in encrypting all traffic between SNMP servers and SNMP Network Management Station clients. The default is **DES**.
      - In the **Privacy Passphrase** field, type the string to use in the encryption algorithm as a secret key for encryption between SNMPv3 USM standard hosts.
        The length of this key must be 16 octets for DES and longer for the AES protocols.
4. Click **Save**.

Results

When you create the first SNMPv3 configuration, the ECS system creates an SNMP Engine ID to use for SNMPv3 traffic. The **Event Notification** page displays that SNMP Engine ID in the **Engine ID** field. You could instead obtain an Engine ID from a Network Monitoring tool and specify that Engine ID in the **Engine ID** field. The important issue is that the SNMP server and any SNMP Network Management Station clients that need to communicate with it using SNMPv3 traffic must use the same SNMP Engine ID in that traffic.

Support for SNMP data collection, queries, and MIBs in ECS

ECS provides support for Simple Network Management Protocol (SNMP) data collection, queries, and MIBs in the following ways:

- During the ECS installation process, your customer support representative can configure and start an `snmpd` server to support specific monitoring of ECS node-level metrics. A Network Management Station client can query these kernel-level `snmpd` servers to gather information about memory and CPU usage from the ECS nodes, as defined by standard Management Information Bases (MIBs). For the list of MIBs for which ECS supports SNMP queries, see SNMP MIBs supported for querying in ECS on page 153.
- The ECS Fabric lifecycle layer includes an `snmp4j` library which acts as an SNMP server to generate SNMPv2 traps and SNMPv3 traps and send them to as many as ten SNMP trap recipient Network Management Station clients. For details of the MIBs for which ECS supports as SNMP traps, see ECS-MIB SNMP Object ID hierarchy and MIB definition on page 153. You can add the SNMP trap recipient servers by using the **Event Notification** page in the ECS Portal. For more information, see Add an SNMPv2 trap recipient on page 149 and Add an SNMPv3 trap recipient on page 151.

SNMP MIBs supported for querying in ECS

You can query the `snmpd` servers that can run on each ECS node from Network Management Station clients for the following SNMP MIBs:

- MIB-2
- DISMAN-EVENT-MIB
- HOST-RESOURCES-MIB
- UCD-SNMP-MIB

You can query ECS nodes for the following basic information by using an SNMP Management Station or equivalent software:

- CPU usage
- Memory usage
- Number of processes running

ECS-MIB SNMP Object ID hierarchy and MIB definition

This topic describes the SNMP OID hierarchy and provides the full SNMP MIB-II definition for the enterprise MIB known as ECS-MIB.

The SNMP enterprise MIB named ECS-MIB defines the objects `trapAlarmNotification`, `notifyTimestamp`, `notifySeverity`, `notifyType`, and `notifyDescription`. The SNMP enterprise includes supported SNMP traps that are associated with managing ECS appliance hardware. ECS sends traps from the Fabric lifecycle container, using services provided by the `snmp4j` Java library. The objects contained in the ECS-MIB have the following hierarchy:

```
emc.............................1.3.6.1.4.1.1139
    ecs.........................1.3.6.1.4.1.1139.102
        trapAlarmNotification...1.3.6.1.4.1.1139.102.1.1
            notifyTimestamp.....1.3.6.1.4.1.1139.102.0.1.1
            notifySeverity......1.3.6.1.4.1.1139.102.0.1.2
```

```
                notifyType.........1.3.6.1.4.1.1139.102.0.1.3
                notifyDescription...1.3.6.1.4.1.1139.102.0.1.4
```

You can download the ECS-MIB definition (as the file `ECS-MIB-v2.mib`) from the Support Site in the Downloads section under Add-Ons. The following Management Information Base syntax defines the SNMP enterprise MIB named ECS-MIB:

```
ECS-MIB DEFINITIONS ::= BEGIN
    IMPORTS enterprises, Counter32, OBJECT-TYPE,
    MODULE-IDENTITY, NOTIFICATION-TYPE
    FROM SNMPv2-SMI;

    ecs MODULE-IDENTITY
        LAST-UPDATED "201605161234Z"
        ORGANIZATION "EMC ECS"
        CONTACT-INFO "EMC Corporation 176 South Street Hopkinton, MA 01748"
        DESCRIPTION "The EMC ECS Manager MIB module"
    ::= { emc 102 }

    emc OBJECT IDENTIFIER ::= { enterprises 1139 }

     -- Top level groups

    notificationData OBJECT IDENTIFIER ::= { ecs 0 }
    notificationTrap OBJECT IDENTIFIER ::= { ecs 1 }

    -- The notificationData group
    -- The members of this group are the OIDs for VarBinds
    -- that contain notification data.

    genericNotify OBJECT IDENTIFIER ::= { notificationData 1 }

    notifyTimestamp OBJECT-TYPE
        SYNTAX OCTET STRING
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION "The timestamp of the notification"
    ::= { genericNotify 1 }

    notifySeverity OBJECT-TYPE
        SYNTAX OCTET STRING
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION "The severity level of the event"
    ::= { genericNotify 2 }

    notifyType OBJECT-TYPE
        SYNTAX OCTET STRING
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION "A type of the event"
    ::= { genericNotify 3 }

    notifyDescription OBJECT-TYPE
        SYNTAX OCTET STRING
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION "A complete description of the event"
    ::= { genericNotify 4 }

    -- The SNMP trap
    -- The definition of these objects mimics the SNMPv2 convention for
    -- sending traps. The enterprise OID gets appended with a 0
    -- and then with the specific trap code.
```

```
    trapAlarmNotification NOTIFICATION-TYPE
        OBJECTS {
            notifyTimestamp,
            notifySeverity,
            notifyType,
            notifyDescription,
        }
        STATUS current
        DESCRIPTION "This trap identifies a problem on the ECS. The description can be
used to describe the nature of the change"
    ::= { notificationTrap 1 }
END
```

Trap messages that are formulated in response to a `Disk Failure` alert are sent to the ECS Portal **Monitor** > **Events** > **Alerts** page in the format `Disk {diskSerialNumber} on node {fqdn} has failed`:

```
2016-08-12 01:33:22 lviprbig248141.lss.emc.com [UDP: [10.249.248.141]:39116-
>[10.249.238.216]]:
iso.3.6.1.6.3.18.1.3.0 = IpAddress: 10.249.238.216      iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.
3.6.1.4.1.1139.102.1.1       iso.3.6.1.4.1.1139.102.0.1.1 = STRING: "Fri Aug 12 13:48:03
GMT 2016"   iso.3.6.1.4.1.1139.102.0.1.2 = STRING: "Critical"       iso.
3.6.1.4.1.1139.102.0.1.3 = STRING: "2002"   iso.3.6.1.4.1.1139.102.0.1.4 = STRING: "Disk
1EGAGMRB on node provo-mustard.ecs.lab.emc.com has failed"
```

Trap messages that are formulated in response to a `Disk Back Up` alert are sent to the ECS Portal **Monitor** > **Events** > **Alerts** page in the format `Disk {diskSerialNumber} on node {fqdn} was revived`:

```
2016-08-12 04:08:42 lviprbig249231.lss.emc.com [UDP: [10.249.249.231]:52469-
>[10.249.238.216]]:
iso.3.6.1.6.3.18.1.3.0 = IpAddress: 10.249.238.216      iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.
3.6.1.4.1.1139.102.1.1       iso.3.6.1.4.1.1139.102.0.1.1 = STRING: "Fri Aug 12 16:23:23
GMT 2016"   iso.3.6.1.4.1.1139.102.0.1.2 = STRING: "Info"   iso.3.6.1.4.1.1139.102.0.1.3 =
STRING: "2025"   iso.3.6.1.4.1.1139.102.0.1.4 = STRING: "Disk 1EV1H2WB on node provo-
copper.ecs.lab.emc.com was revived"
```

Syslog servers

Syslog servers provide a method for centralized storage and retrieval of system log messages. ECS supports forwarding of alerts and audit messages to remote syslog servers, and supports operations using the following application protocols:

- BSD Syslog
- Structured Syslog

Alerts and audit messages that are sent to Syslog servers are also displayed on the ECS Portal, with the exception of OS level Syslog messages (such as node SSH login messages), which are sent only to Syslog servers and not displayed in the ECS Portal.

Once you add a Syslog server, ECS initiates a syslog container on each node. The message traffic occurs over either TCP or the default UDP.

ECS sends Audit log messages to Syslog servers, including the severity level, using the following format:

`${serviceType} ${eventType} ${namespace} ${userId} ${message}`

ECS sends Alert logs to Syslog servers using the same severity as appears in the ECS Portal, using the following format:

```
${alertType} ${symptomCode} ${namespace} ${message}
```

ECS sends Fabric alerts using the following format:

```
Fabric {symptomCode} "{description}"
```

Starting with ECS 3.1, ECS forwards only the following OS logs to Syslog servers:

- External SSH messages
- All `sudo` messages with Info severity and higher
- All messages from the auth facility with Warning severity and higher, which are security-related and authorization-related messages

Add a Syslog server

You can configure a Syslog server to remotely store ECS logging messages.

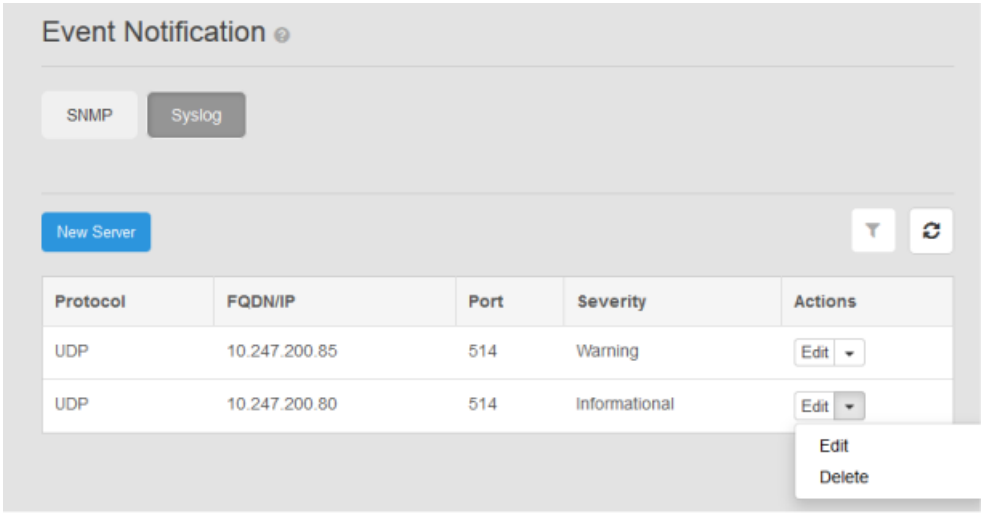
Before you begin

- This operation requires the System Administrator role in ECS.

Procedure

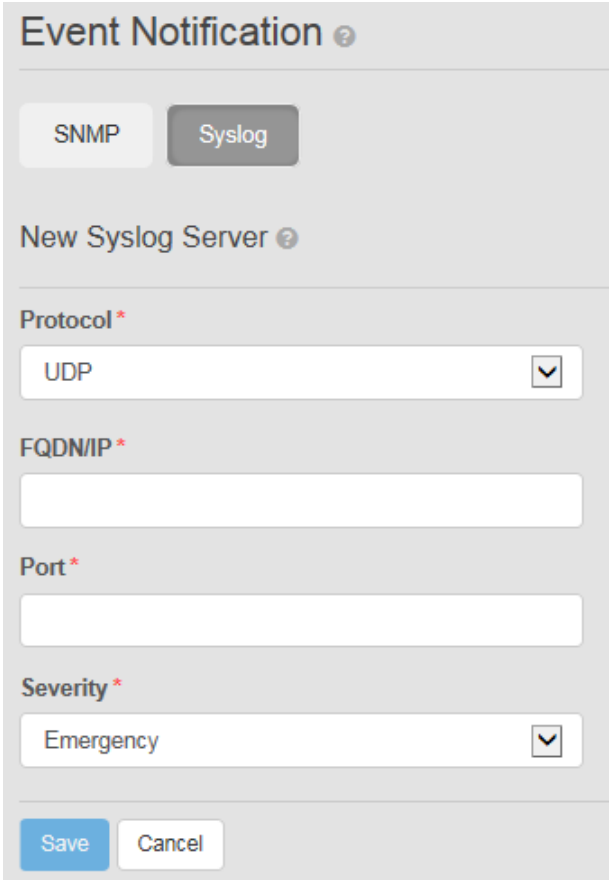
1. In the ECS Portal, select **Settings** > **Event Notification**.
2. On the **Event Notification** page, click **Syslog**.

   This page lists the Syslog servers that have been added to ECS and allows you to configure new Syslog servers.



3. On the **Event Notification** page, click **New Server**.

   The **New Syslog Server** sub-page appears.



4. On the **New Syslog Server** sub-page, complete the following steps.
   a. In the **Protocol** field, select **UDP** or **TCP**.

      UDP is the default protocol.

   b. In the **FQDN/IP** field, type the Fully Qualified Domain Name or IP address for the node that runs the Syslog server.

   c. In the **Port** field, type the port number for the Syslog server on which you want to store log messages.

      The default port number is 514.

   d. In the **Severity** field, select the severity of threshold for messages to send to the log. The drop-down options are **Emergency**, **Alert**, **Critical**, **Error**, **Warning**, **Notice**, **Informational**, or **Debug**.
5. Click **Save**.

Server-side filtering of Syslog messages

This topic describes how an ECS Syslog message can be further filtered with server-side configuration.

You can configure Syslog servers in the ECS Portal (or by using the ECS Management REST API) to specify the messages that are delivered to the servers. You can then use server-side filtering techniques to reduce the number of messages that are saved to the logs. Filtering is done at the facility level. A facility segments messages by type. ECS directs messages to facilities as described in the following table.

Table 20 Syslog facilities used by ECS

Facility	Keyword	Defined use	ECS use
1	user	User-level messages	Fabric alerts
3	daemon	System daemons	OS messages
4	auth	Security and authorization messages	`ssh` and `sudo` success and failure messages
16	local0	Local use 0	Object alerts, object audits
All facilities	*		

For each facility, you can filter by severity level by using the following format:

facility-keyword.severity-keyword

Severity keywords are described in the following table.

Table 21 Syslog severity keywords

Severity level number	Severity level	Keyword
0	Emergency	emerg
1	Alert	alert
2	Critical	crit
3	Error	err
4	Warning	warn
5	Notice	notice
6	Informational	info
7	Debug	debug
All severities	All severities	*

Modify the Syslog server configuration using the /etc/rsyslog.conf file

You can modify your existing configuration by editing the `/etc/rsyslog.conf` file on the Syslog server.

Procedure

1. You might configure the `/etc/rsyslog.conf` file in the following ways:
   a. To receive incoming ECS messages from all facilities and all severity levels, use this configuration and specify the complete path and name of your target log file:

```
*.* /var/log/ecs-messages.all
```

b. To receive all fabric alerts, object alerts and object audits, use this configuration with the full path and name of your target log file:

```
user.*,local0.* /var/log/ecs-fabric-object.all
```

c. To receive all fabric alerts, object alerts and object audits, and limit auth facility messages to warning severity and above, use this configuration with the full path and name of your target log file:

```
user.*,local0.*/var/log/ecs-fabric-
object.allauth.warn /var/log/ecs-auth-messages.warn
```

d. To segment the traffic to a facility into multiple files log files:

```
auth.info /var/log/ecs-auth-info.log
auth.warn /var/log/ecs-auth-warn.log
auth.err /var/log/ecs-auth-error.log
```

2. After any modification of the configuration file, restart the Syslog service on the Syslog server:

```
# service syslog restart
```

Output:

```
Shutting down syslog services done
Starting syslog services done
```

Platform locking

You can use the ECS Portal to lock remote access to nodes.

ECS can be accessed through the ECS Portal or the ECS Management REST API by management users assigned administration roles. ECS can also be accessed at the node level by a privileged default node user named `admin` that is created during the initial ECS install. This default node user can perform service procedures on the nodes and have access:

- By directly connecting to a node through the management switch with a service laptop and using SSH or the CLI to directly access the node's operating system.
- By remotely connecting to a node over the network using SSH or the CLI to directly access the node's operating system.

For more information about the default `admin` node-level user, see the *ECS Security Guide*, available from the ECS Product Documentation page.

Node locking provides a layer of security against remote node access. Without node locking, the **`admin`** node-level user can remotely access nodes at any time to collect data, configure hardware, and run Linux commands. If all the nodes in a cluster are locked, then remote access can be planned and scheduled for a defined window to minimize the opportunity for unauthorized activity.

You can lock selected nodes in a cluster or all the nodes in the cluster by using the ECS Portal or the ECS Management REST API. Locking affects only the ability to remotely access (SSH to) the locked nodes. Locking does not change the way the ECS Portal and the ECS Management REST APIs access nodes, and it does not affect the ability to directly connect to a node through the management switch.

Maintenance
For node maintenance using remote access, you can unlock a single node to allow remote access to the entire cluster by using SSH as the `admin` user. After the `admin` user successfully logs in to the unlocked node using SSH, the `admin` user can SSH from that node to any other node in the cluster through the private network.

You can unlock nodes to remotely use commands that provide OS-level read-only diagnostics.

Auditing
Node lock and unlock events appear in audit logs and Syslog. Failed attempts to lock or unlock nodes also appear in the logs.

Lock and unlock nodes using the ECS Management REST API

You can use the following APIs to manage node locks.

Table 22 ECS Management REST API calls for managing node locking

Resource	Description
GET /vdc/nodes	Gets the data nodes that are currently configured in the cluster
GET /vdc/lockdown	Gets the locked/unlocked status of a VDC
PUT /vdc/lockdown	Sets the locked/unlocked status of a VDC
PUT /vdc/nodes/{nodeName}/lockdown	Sets the Lock/unlock status of a node
GET /vdc/nodes/{nodeName}/lockdown	Gets the Lock/unlock status of a node

Lock and unlock nodes using the ECS Portal

You can use the ECS Portal to lock and unlock remote SSH access to ECS nodes.

Before you begin

This operation requires the Lock Administrator role assigned to the `emcsecurity` user in ECS.

Locking affects only the ability to remotely access (SSH to) the locked nodes. Locking does not change the way the ECS Portal and the ECS Management REST APIs access nodes, and it does not affect the ability to directly connect to a node through the management switch.

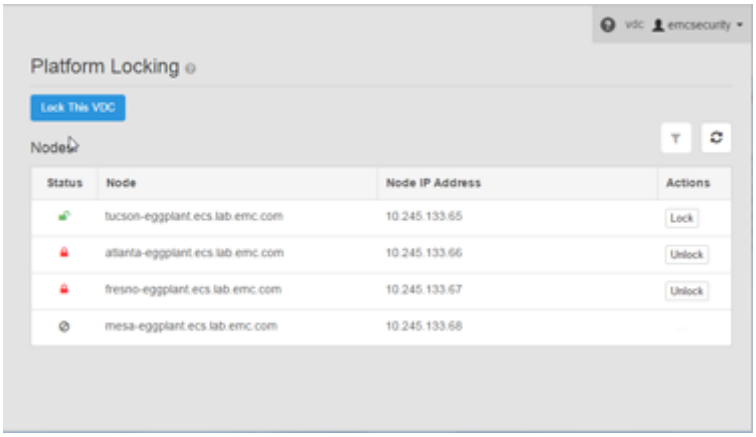
Procedure

1. Log in as the `emcsecurity` user.

   For the initial login for this user, you are prompted to change the password and log back in.

2. In the ECS Portal, select **Settings** > **Platform Locking**.

The screen lists the nodes in the cluster and displays the lock status.



The node states are:

- Unlocked: Displays an open green lock icon and the **Lock** action button.
- Locked: Displays a closed red lock icon and the **Unlock** action button.
- Offline: Displays a circle-with-slash icon but no action button because the node is unreachable and the lock state cannot be determined.

3. Perform any of the following steps.

   a. Click **Lock** in the **Actions** column beside the node you want to lock.

      Any user who is remotely logged in by SSH or CLI has approximately five minutes to exit before their session is terminated. An impending shutdown message appears on the user's terminal screen.

   b. Click **Unlock** in the **Actions** column beside the node you want to unlock.

      The `admin` default node user can remotely log in to the node after a few minutes.

   c. Click **Lock the VDC** if you want to lock all unlocked, online nodes in the VDC.

      It does not set a state where a new or offline node is automatically locked once detected.

Licensing

EMC ECS licensing is capacity-based.

You must obtain at least one ECS license and upload it to the appliance.

Each appliance (rack) has a license.

Obtain the EMC ECS license file

You can obtain a license file (.lic) from the EMC license management website.

Before you begin

To obtain the license file, you must have the License Authorization Code (LAC), which was emailed from EMC. If you have not received the LAC, contact your customer support representative.

Procedure

1. Go to the license page at: `https://support.emc.com/servicecenter/license/`
2. From the list of products, select **ECS Appliance**.
3. On the LAC Request page, enter the LAC code, and then click **Activate**.
4. Select the entitlements to activate, and then click **Start Activation Process**.
5. Select **Add a Machine** to specify any meaningful string for grouping licenses.

   For the machine name, enter any string that will help you keep track of your licenses. (It does not have to be an actual machine name.)
6. Enter the quantities for each entitlement, or select **Activate All**, and then click **Next**.

   For more than one site in a geo-federated system, distribute the controllers as appropriate, to obtain individual license files for each virtual data center (VDC).
7. Optionally, specify an addressee to receive an email summary of the activation transaction.
8. Click **Finish**.
9. Click **Save to File** to save the license file (.lic) to a folder on your computer.

   This is the license file that is needed during initial setup of ECS, or when adding a new license later in the ECS Portal.

Upload the ECS license file

You can upload the ECS license file from the ECS Portal.

Before you begin

- This operation requires the System Administrator role in ECS.
- Ensure that you have a valid license file. You can follow the instructions provided in Obtain the EMC ECS license file on page 161 to obtain a license.

Where you are installing more than one site in a geo-federated system, ensure that the licensing scheme across sites is the same. If the existing cluster has an encryption-enabled license, any new site added to it should have the same. Similarly, if existing sites do not have encryption-enabled licenses, the new sites that are added to the cluster must follow the same model.

Procedure

1. In the ECS Portal, select **Settings** > **Licensing**.
2. On the **Licensing** page, in the **Upload a New License File** field, click **Browse** to navigate to your local copy of the license file.
3. Click **Upload** to add the license.

   The license appears in the list of licenses.

About this VDC

You can view information about software version numbers for the current node or other nodes in the VDC on the **About this VDC** page.

You can view information related to the node you are currently connected to on the **About** tab. You can view the names, IP addresses, rack IDs, and software versions of

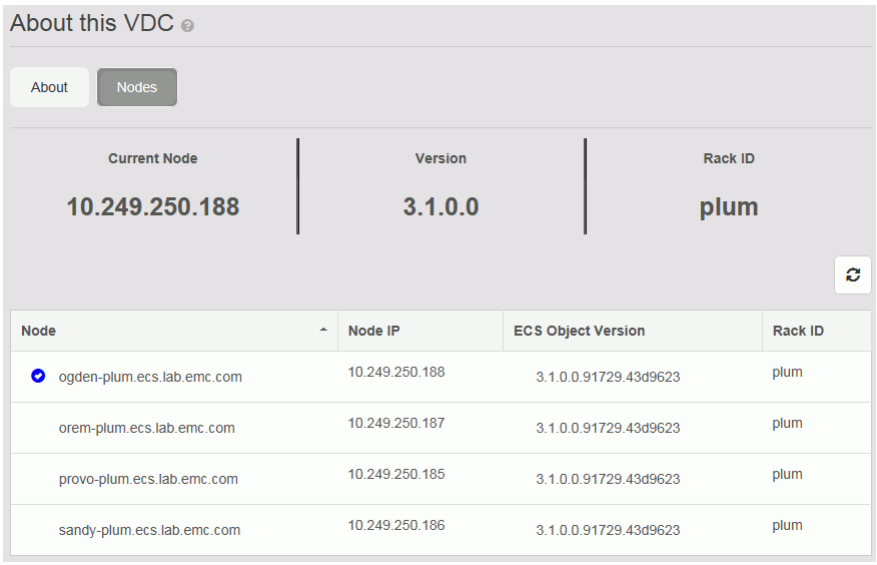
the nodes available in the VDC on the **Nodes** tab. You can identify any nodes that are not at the same software version as the node you are connected to on the **Nodes** tab.

Procedure

1. In the ECS Portal, select **Settings** > **About this VDC**.

   The **About this VDC** page appears with the **About** tab open. This page displays information about the ECS software version and ECS Object service version for the current node.

2. On the **About this VDC** page, to view the software version for the reachable nodes in the cluster, click the **Nodes** tab.



   The blue checkmark indicates the current node. A star indicates the nodes that have a different software version.

CHAPTER 11

ECS Outage and Recovery

Introduction to ECS site outage and recovery

ECS is designed to provide protection when a site outage occurs due to a disaster or other problem that causes a site to go offline or become disconnected from the other sites in a geo-federated deployment.

Site outages can be classified as a temporary site outage (TSO) or a permanent site outage (PSO). A TSO is a failure of the WAN connection between two sites, or a temporary failure of an entire site (for example, a power failure). A site can be brought back online after a TSO. ECS can detect and automatically handle these types of temporary site failures.

A PSO is when an entire site becomes permanently unrecoverable, such as when a disaster occurs. In this case, the System Administrator must permanently fail over the site from the federation to initiate failover processing, as described in Delete a VDC and fail over a site on page 36.

TSO and PSO behavior is described in the following topics:

- TSO behavior on page 166
- TSO considerations on page 174
- NFS file system access during a TSO on page 174
- PSO behavior on page 174

Note

For more information on TSO and PSO behavior, see the *Elastic Cloud Storage High Availability Design* white paper.

ECS recovery and data balancing behavior is described in these topics:

- Recovery on disk and node failures on page 175
- Data rebalancing after adding new nodes on page 176

TSO behavior

VDCs in a geo-replicated environment have a heartbeat mechanism. Sustained loss of heartbeats for a configurable duration (by default, 15 minutes) indicates a network or site outage and the system transitions to identify the TSO.

ECS marks the unreachable site as TSO and the site status displays as `Temporarily unavailable` in the **Replication Group Management** page in the ECS Portal.

There are two important concepts that determine how the ECS system behaves during a TSO.

- *Owner*: If a bucket or object is created within a namespace in Site A, then Site A is the owner of that bucket or object. When a TSO occurs, the behavior for read/write requests differs depending on whether the request is made from the site that owns the bucket or object, or from a non-owner site that does not own the primary copy of the object.
- *Access During Outage (ADO) bucket property*: Access to buckets and the objects within them during a TSO differs depending on whether the ADO property is enabled on buckets. The ADO property can be set at the bucket level; meaning you can enable this option for some buckets and not for others.

- If the ADO property is disabled on a bucket, strong consistency is maintained during a TSO by continuing to allow access to data owned by accessible sites and preventing access to data owned by a failed site.
- If the ADO property is enabled on a bucket, read and optionally write access to all geo-replicated data is allowed, including the data that is owned by the failed site. During a TSO the data in the ADO-enabled bucket temporarily switches to eventual consistency; once all sites are back online it will revert back to strong consistency.

For more information, see the following topics:

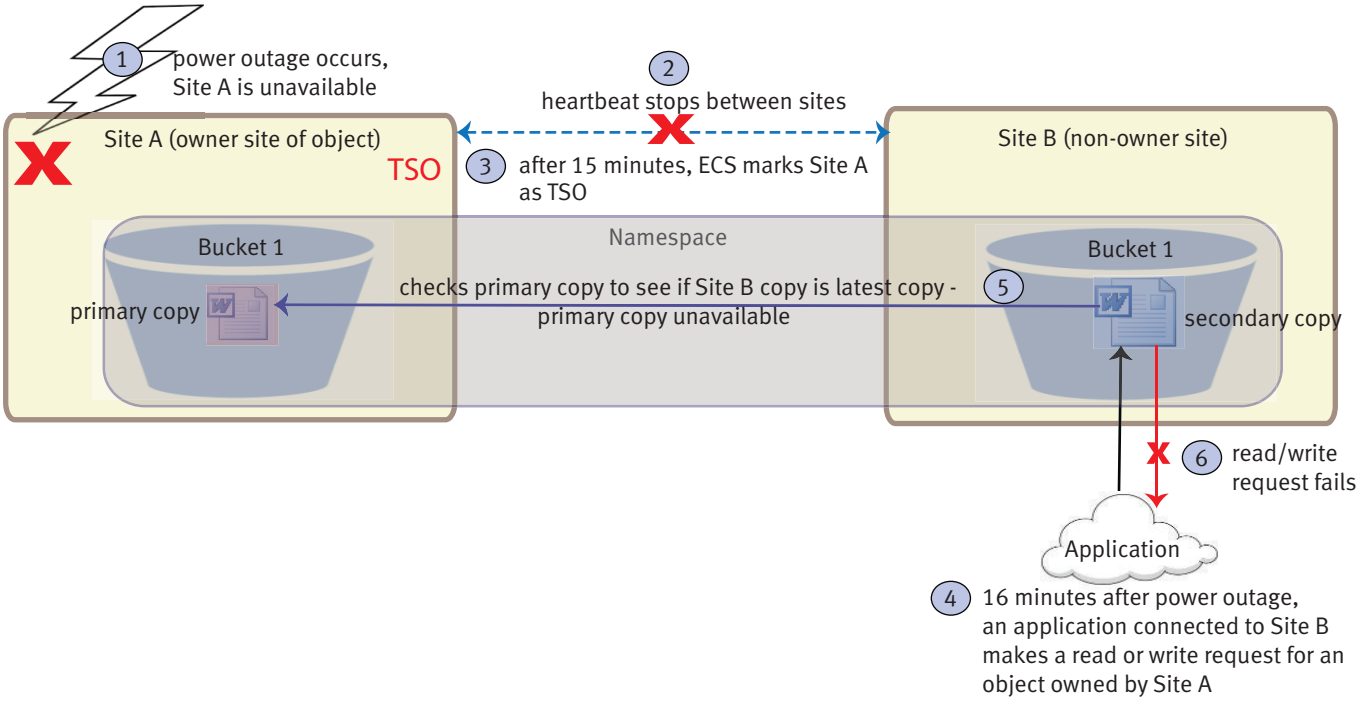
- TSO behavior with the ADO bucket property disabled on page 167
- TSO behavior with the ADO bucket property enabled on page 168

TSO behavior with the ADO bucket property disabled

If the Access During Outage (ADO) property is not set on a bucket, during a TSO you will only be able to access the data in that bucket if it is owned by an available site. You will not be able to access data in a bucket that is owned by a failed site.

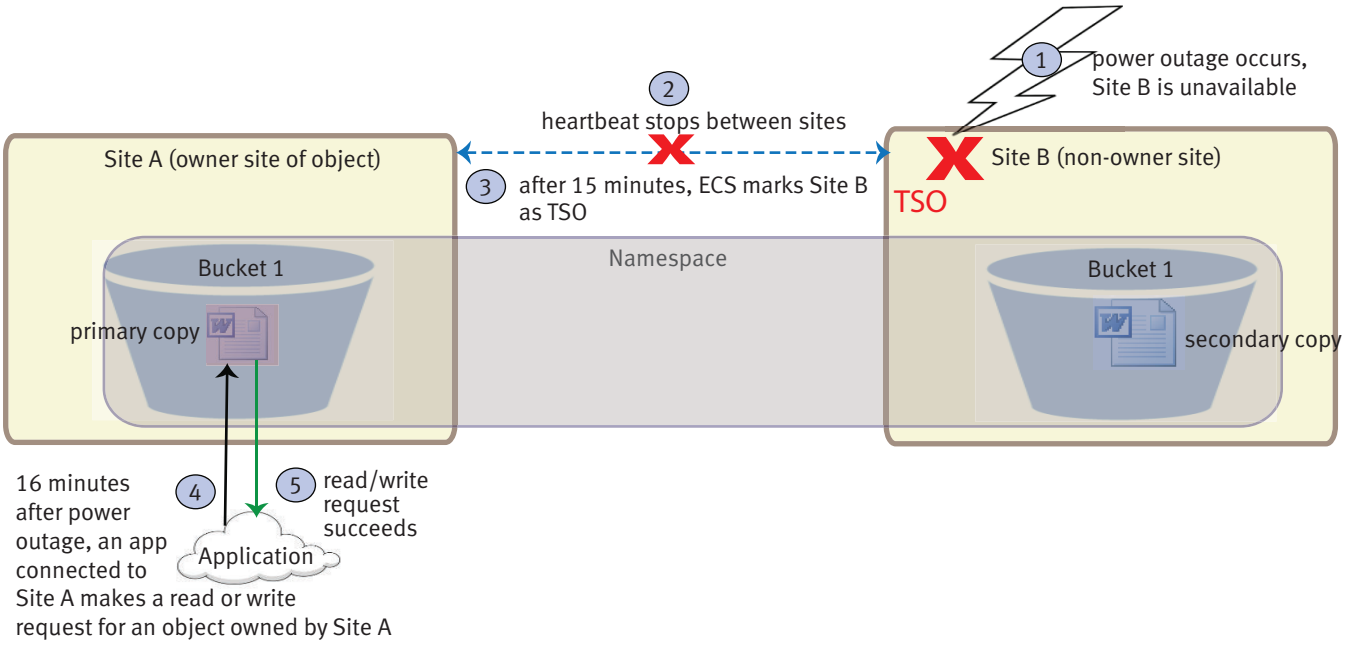
In the ECS system example shown in the following figure, Site A is marked as TSO and is unavailable. Site A is the owner of Bucket 1, because that is where the bucket (and the objects within it) was created. At the time Bucket 1 was created, the ADO property was disabled. The read/write requests for objects in that bucket made by applications connected to Site A will fail. When an application tries to access an object in that bucket from a non-owner site (Site B), the read/write request will also fail. Note that the scenario would be the same if the request was made before the site was officially marked as TSO by the system (which occurs after the heartbeat is lost for a sustained period of time, which is set at 15 minutes by default). In other words, if a read/write request was made from an application connected to Site B within 15 minutes of the power outage, the request would still fail.

Figure 22 Read/write request fails during TSO when data is accessed from non-owner site and owner site is TSO



The following figure shows a non-owner site that is marked as TSO with the ADO property disabled on the bucket. When an application tries to access the primary copy at the owner site, the read/write request made to the owner site will be successful. A read/write request made from an application connected to the non-owner site will fail.

Figure 23 Read/write request succeeds during TSO when data is accessed from owner site and non-owner site is TSO



TSO behavior with the ADO bucket property enabled

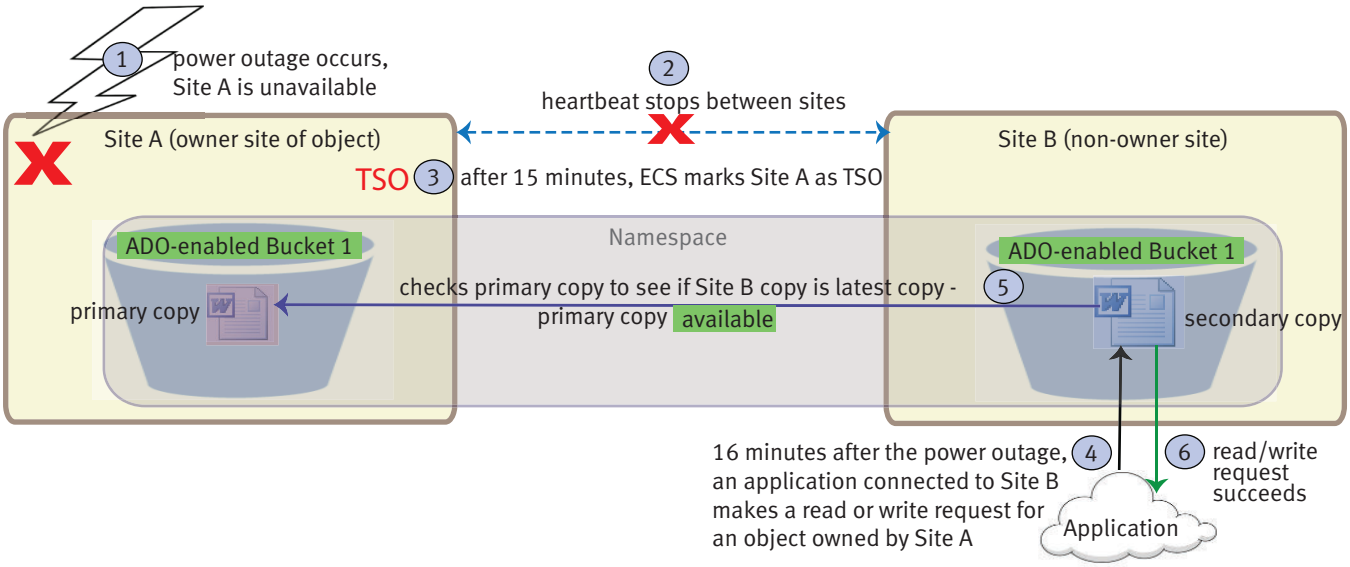
ECS provides a mechanism to ensure that data is accessible during a TSO. You can set the Access During Outage (ADO) property on a bucket so that the primary copies of the objects in that bucket are available, even when the site that owns the bucket fails. If you do not set the ADO property on a bucket, the read/write requests for objects in the bucket that is owned by a failed site cannot be made from the other sites.

When buckets are configured with the **Access During Outage** property set to **On**, applications can read the objects in those buckets while connected to any site. With the ADO property enabled on buckets and upon detecting a temporary outage, read/write requests from applications connected to a non-owner site are accepted and honored, as shown in the following figure.

Note

When an application is connected to a site that is not the bucket owner, the application cannot list the buckets in the namespace, so the access to the bucket or object must be explicit. For more information, see TSO considerations on page 174.

Figure 24 Read/write request succeeds during TSO when ADO-enabled data is accessed from non-owner site and owner site is TSO



The ECS system operates under the eventual consistency model during a TSO with ADO enabled on buckets. When a change is made to an object at one site, it will be *eventually* consistent across all copies of that object at other sites. Until enough time elapses to replicate the change to other sites, the value might be inconsistent across multiple copies of the data at a particular point in time.

An important factor to consider is that enabling ADO on buckets has performance consequences; ADO-enabled buckets have slower read/write performance than buckets with ADO disabled. The performance difference is due to the fact that when a bucket is enabled for ADO, ECS must first resolve object ownership in order to provide strong consistency when all sites become available after a TSO. When ADO is not enabled on a bucket, ECS does not have to resolve object ownership because the bucket does not allow change of object ownership during a TSO.

The benefit of the ADO property is that it allows you to access data during temporary site outages; the disadvantage is that the data returned may be outdated and read/write performance on ADO buckets will be slower.

You can define whether or not there will be access to the objects in a bucket during a TSO by choosing to enable or disable the **Access During Outage** (ADO) property, and you can further define the type of access you have to the objects in a bucket during a site outage by enabling or disabling the **Read-Only Access During Outage** property.

Note

It is also the case that ADO-enabled buckets with the **Read-Only Access During Outage** property enabled have slower read/write performance than buckets with ADO disabled.

By default, **Access During Outage** is disabled, because there is a risk that object data retrieved during a TSO is not the most recent. Enabling **Access During Outage** marks the bucket, and all of the objects in the bucket, as available during an outage. You can enable **Access During Outage** when creating a bucket, and you can change this property after the bucket is created (provided that all sites are online.)

When you enable the **Access During Outage** property, the following occurs during a TSO:

- By default, object data is accessible for both read and write operations during the outage.
- File systems within file system-enabled (HDFS/NFS) buckets that are owned by the unavailable site are read-only during an outage.

When you create a bucket and enable the **Access During Outage** property, you also have the option of enabling the **Read-Only Access During Outage** property on the bucket. You can only set the **Read-Only Access During Outage** property while creating the bucket; you cannot change this property after the bucket has been created.

When the **Read-Only Access During Outage** property is enabled, the following occurs during a TSO:

- Creation of new objects in the bucket is restricted.
- Access to file systems is not impacted because they are automatically in read-only mode when **Access During Outage** is set on the file system buckets.

You can set the **Access During Outage** and **Read-Only Access During Outage** properties when creating a bucket from the following interfaces:

- ECS Portal (see Create a bucket on page 88)
- ECS Management REST API
- ECS CLI
- Object API REST interfaces such as S3, Swift, and Atmos

TSO behavior with ADO-enabled buckets is described for the following ECS system configurations:

- Two-site geo-federated deployment with ADO-enabled buckets on page 170
- Three-site Geo-Active federated deployment with ADO-enabled buckets on page 171
- Three-site Geo-Passive federated deployment with ADO-enabled buckets on page 172

Two-site geo-federated deployment with ADO-enabled buckets

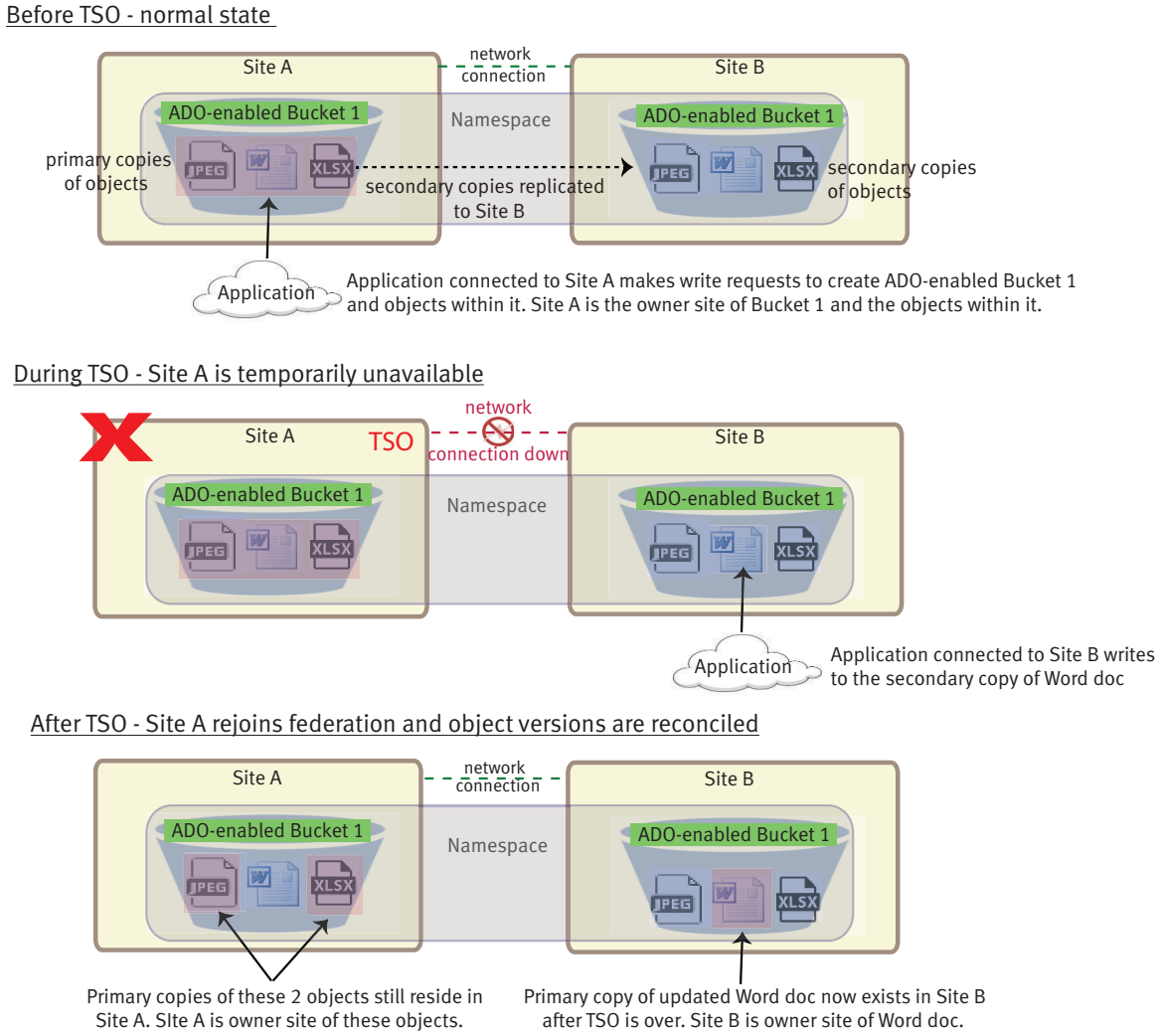
When an application is connected to a non-owner site, and it modifies an object within an ADO-enabled bucket during a network outage, ECS transfers ownership of the object to the site where the object was modified.

The following figure shows how a write to a non-owner site causes the non-owner site to take ownership of the object during a TSO in a two-site geo-federated deployment. This functionality allows applications connected to each site to continue to read and write objects from buckets in a shared namespace.

When the same object is modified in *both* Site A and Site B during a TSO, the copy on the non-owner site is the authoritative copy. When an object that is owned by Site B is modified in both Site A and Site B during a network outage, the copy on Site A is the authoritative copy that is kept, and the other copy is overwritten.

When network connectivity between two sites is restored, the heartbeat mechanism automatically detects connectivity, restores service, and reconciles objects from the two sites. This synchronization operation is done in the background and can be monitored on the **Monitor** > **Recovery Status** page in the ECS Portal.

Figure 25 Object ownership example for a write during a TSO in a two-site federation

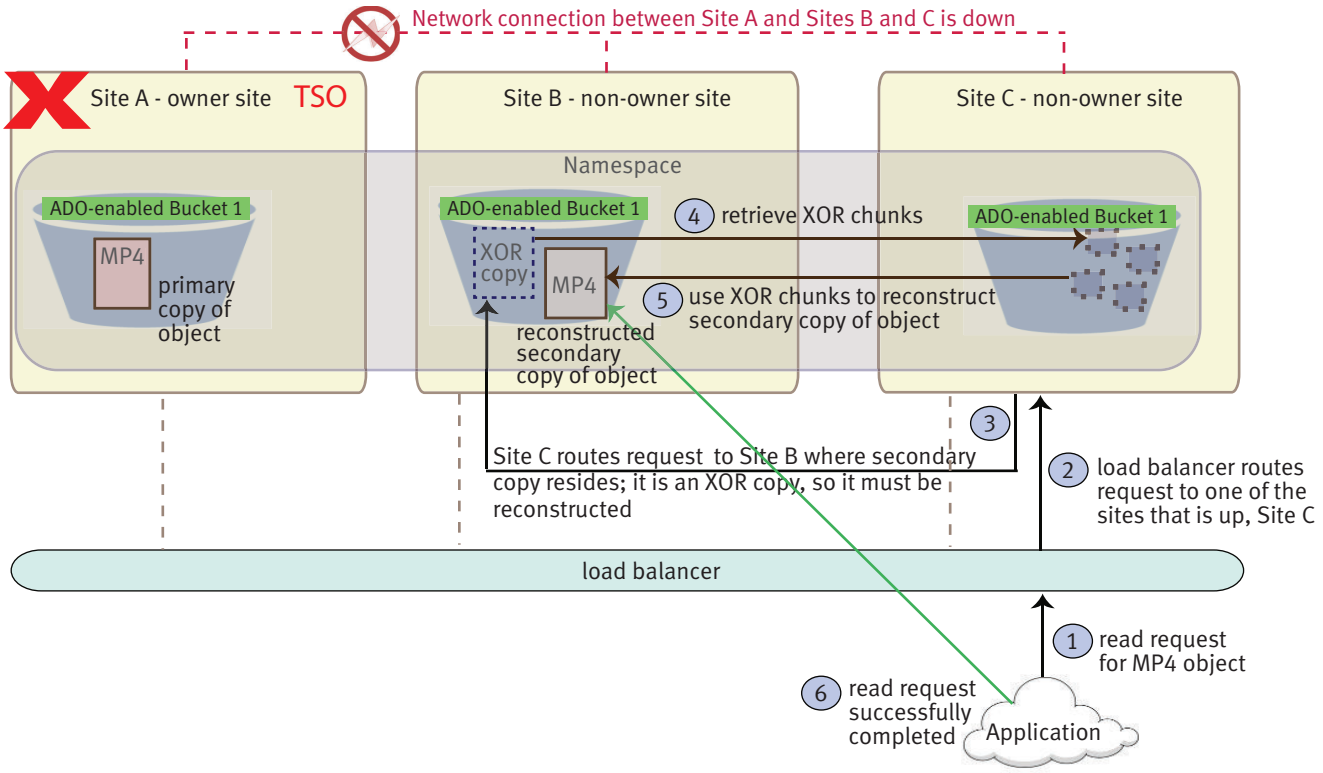


Three-site Geo-Active federated deployment with ADO-enabled buckets

When more than two sites are part of a replication group, and if network connectivity is interrupted between one site and the other two, then write/update/ownership operations continue just as they would with two sites, but the process for responding to read requests is more complex.

If an application requests an object that is owned by a site that is not reachable, ECS sends the request to the site with the secondary copy of the object. The secondary copy might have been subject to a data contraction operation, which is an XOR between two different data sets that produces a new data set. The site with the secondary copy must retrieve the chunks of the object included in the original XOR operation, and it must XOR those chunks with the recovery copy. This operation returns the contents of the chunk originally stored on the owner site. The chunks from the recovered object can then be reassembled and returned. When the chunks are reconstructed, they are also cached so that the site can respond more quickly to subsequent requests. Reconstruction is time consuming. More sites in a replication group imply more chunks that must be retrieved from other sites, and hence reconstructing the object takes longer. The following figure shows the process for responding to read requests in a three-site federation.

Figure 26 Read request workflow example during a TSO in a three-site federation



Three-site Geo-Passive federated deployment with ADO-enabled buckets

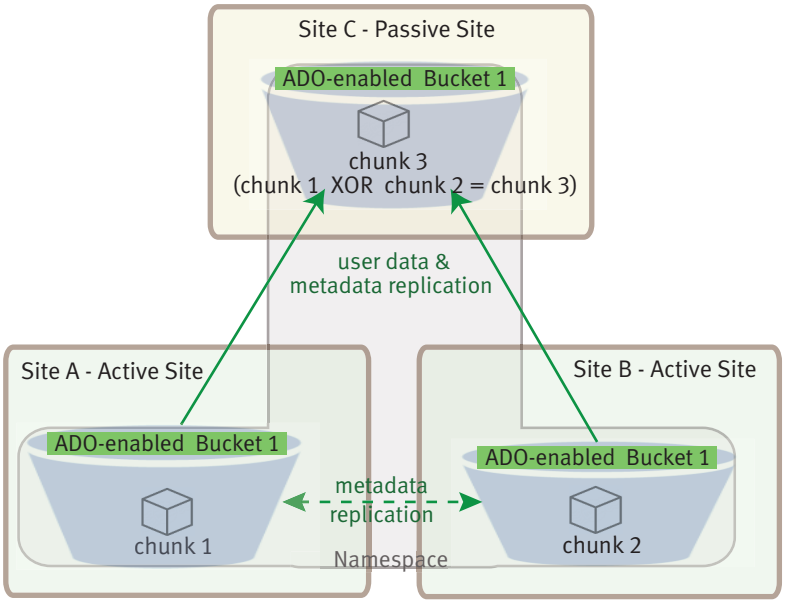
When ECS is deployed in a three-site Geo-Passive configuration, the TSO behavior is the same as described in Three-site Geo-Active federated deployment with ADO-enabled buckets on page 171, with one difference. If a network connection fails between an active site and the passive site, ECS always marks the passive site as TSO (not the active site).

When the network connection fails between the two active sites, the following normal TSO behavior occurs:

1. ECS marks one of the active sites as TSO (unavailable). For example, owner Site B.
2. Read/write/update requests are rendered from the site that is up (Site A).
3. For a read request, Site A requests the object from the passive site (Site C).
4. Site C decodes (undo XOR) the XOR chunks and sends to Site A.
5. Site A reconstructs a copy of the object to honor the read request.
6. In the case of a write/update request, Site A becomes the owner of the object and keeps the ownership after the outage.

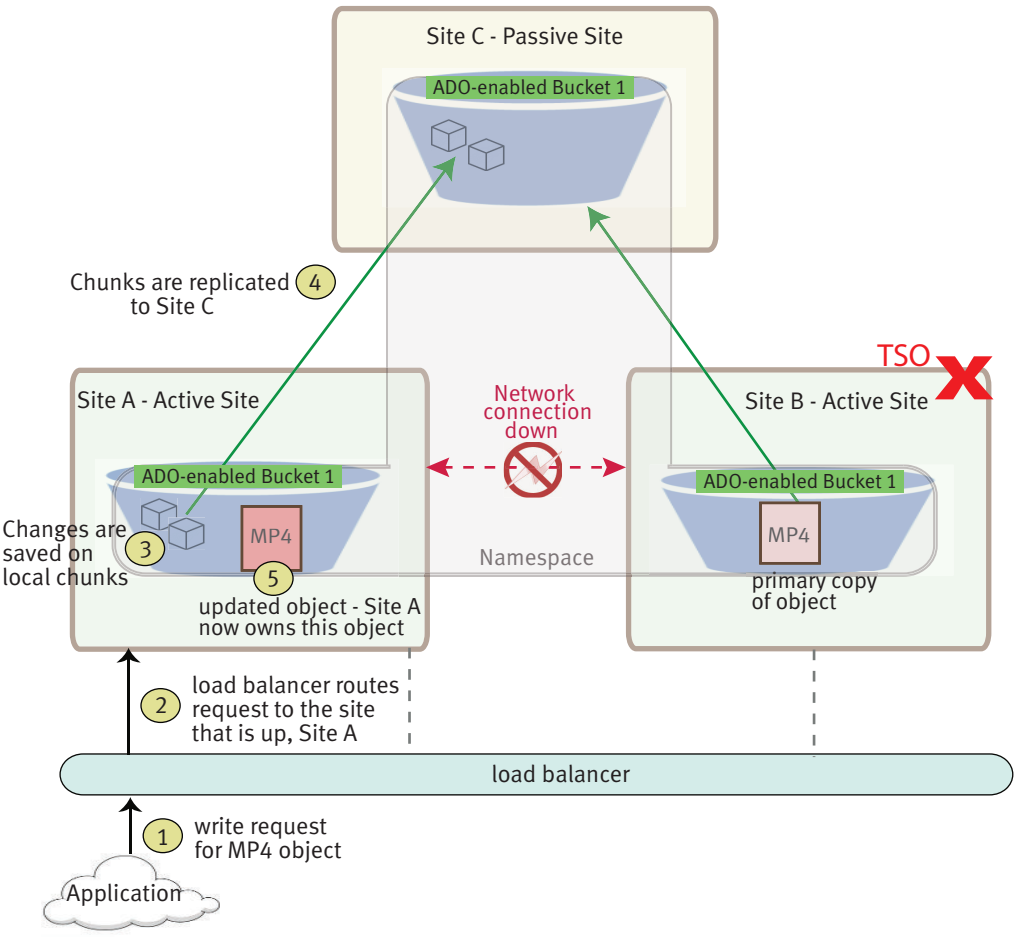
The following figure shows a Geo-Passive configuration in a normal state; users can read and write to active Sites A and B and the data and metadata is replicated one way to the passive Site C. Site C XORs the data from the active sites.

Figure 27 Geo-Passive replication in normal state



The following figure shows the workflow for a write request made during a TSO in a three-site Geo-Passive configuration.

Figure 28 TSO for Geo-Passive replication



TSO considerations

You can perform many object operations during a TSO. You cannot perform create, delete, or update operations on the following entities at any site in the geo-federation until the temporary failure is resolved, regardless of the ADO bucket setting:

- Namespaces
- Buckets
- Object users
- Authentication providers
- Replication groups (you can remove a VDC from a replication group for a site failover)
- NFS user and group mappings

The following limitations apply to buckets during a TSO:

- You cannot list buckets for a namespace when the namespace owner site is not reachable. You can list objects within buckets that are owned by available sites and you can list objects within ADO-enabled buckets that are owned by the unavailable site. Listing objects in ADO-enabled buckets owned by the unavailable site will return only replicated objects, and the list may be incomplete.
- File systems within file system-enabled (HDFS/NFS) buckets that are owned by the unavailable site are read-only.
- When you copy an object from a bucket owned by the unavailable site, the copy is a full copy of the source object. This means that the same object's data is stored more than once. Under normal non-TSO circumstances, the object copy consists of the data indices of the object, not a full duplicate of the object's data.
- OpenStack Swift users cannot log in to OpenStack during a TSO because ECS cannot authenticate Swift users during the TSO. After the TSO, Swift users must re-authenticate.

NFS file system access during a TSO

NFS provides a single namespace across all ECS nodes and can continue to operate in the event of a TSO. When you mount an NFS export, you can specify any of the ECS nodes as the NFS server or you can specify the address of a load balancer. Whichever node you point at, the ECS system is able to resolve the file system path.

In the event of a TSO, if your load balancer is able to redirect traffic to a different site, your NFS export continues to be available. Otherwise, you must remount the export from another, non-failed site.

When the owner site fails, and ECS is required to reconfigure to point at a non-owner site, data can be lost due to NFS asynchronous writes and also due to unfinished ECS data replication operations.

For more information on how to access NFS-enabled buckets, see Introduction to file access on page 106.

PSO behavior

If a disaster occurs, an entire site can become unrecoverable; this is referred to in ECS as a permanent site outage (PSO). ECS treats the unrecoverable site as a temporary site failure, but only if the entire site is down or completely unreachable over the

WAN. If the failure is permanent, the System Administrator must permanently fail over the site from the federation to initiate failover processing; this initiates resynchronization and re-protection of the objects stored on the failed site. The recovery tasks run as a background process. For more information on how to perform the failover procedure in the ECS Portal, see Delete a VDC and fail over a site on page 36.

Before you initiate a PSO in the ECS Portal, it is advised to contact your customer support representative, so that the representative can validate the cluster health. Data is not accessible until the failover processing is completed. You can monitor the progress of the failover processing on the **Monitor** > **Geo Replication** > **Failover Processing** tab in the ECS Portal. After the failover process is completed, this tab does not show status. While the recovery background tasks are running, but after failover processing has completed, some data from the removed site might not be read back until the recovery tasks fully complete.

Recovery on disk and node failures

ECS continuously monitors the health of the nodes, their disks, and objects stored in the cluster. ECS disperses data protection responsibilities across the cluster and automatically re-protects at-risk objects when nodes or disks fail.

Disk health

ECS reports disk health as Good, Suspect, or Bad.

- Good: The disk's partitions can be read from and written to.
- Suspect: The disk has not yet met the threshold to be considered bad.
- Bad: A certain threshold of declining hardware performance has been met. When met, no data can be read or written.

ECS writes only to disks in good health. ECS does not write to disks in suspect or bad health. ECS reads from good disks and suspect disks. When two of an object's chunks are located on suspect disks, ECS writes the chunks to other nodes.

Node health

ECS reports node health as Good, Suspect, or Bad.

- Good: The node is available and responding to I/O requests in a timely manner.
- Suspect: The node has been unavailable for more than 30 minutes.
- Bad: The node has been unavailable for more than an hour.

ECS writes to reachable nodes regardless of the node health state. When two of an object's chunks are located on suspect nodes, ECS writes two new chunks of it to other nodes.

Data recovery

When there is a failure of a node or drive in the site, the storage engine:

1. Identifies the chunks or erasure coded fragments affected by the failure.
2. Writes copies of the affected chunks or erasure coded fragments to good nodes and disks that do not currently have copies.

NFS file system access during a node failure

NFS provides a single namespace across all ECS nodes and can continue to operate in the event of node failure. When you mount an NFS export, you can specify any of the ECS nodes as the NFS server or you can specify the address of a load balancer. Whichever node you point at, the ECS system resolves the file system path.

In the event of a node failure, ECS recovers data using its data fragments. If your NFS export is configured for asynchronous writes, you run the risk of losing data related to any transactions that have not yet been written to disk. This is the same with any NFS implementation.

If you mounted the file system by pointing at an ECS node and that node fails, you must remount the export by specifying a different node as the NFS server. If you mounted the export by using the load balancer address, failure of the node is handled by the load balancer which automatically directs requests to a different node.

Data rebalancing after adding new nodes

When the number of nodes at a site is expanded due to the addition of new racks or storage nodes, new erasure coded chunks are allocated to the new storage and existing data chunks are redistributed (rebalanced) across the new nodes. Four or more nodes must exist for erasure coding of chunks to take place. Addition of new nodes over and above the required four nodes results in erasure coding rebalancing.

The redistribution of erasure coded fragments is performed as a background task so that the chunk data is accessible during the redistribution process. In addition, the new fragment data is distributed as a low priority to minimize network bandwidth consumption.

Fragments are redistributed according to the same erasure coding scheme with which they were originally encoded. If a chunk was written using the cold storage erasure coding scheme, ECS uses the cold storage scheme when creating the new fragments for redistribution.