SWIFT - Operations Guide

Version 1.3.5

RACKWARE INC.

Contents

About SWIFT

The SWIFT is a container orchestration, backup, and DR product. It is an Any-To-Any DR solution for containers allowing you to seamlessly synchronize between your source and target container platforms, irrespective of where they are located (any of the public clouds or datacenter). Please see subsequent sections for supported cloud container platforms.

The SWIFT works on top of your existing container platforms like Kubernetes and OpenShift. You will need the following items before you execute any supported operations

1. A Cloud-Admin privileged account for the container platform (like Kubernetes or OpenShift cluster), which you will manage with SWIFT.
2. Network connectivity from the SWIFT server to your source or DR container platform, mainly to SWIFT launched transient containers/services.
3. A set of free ports in your container platform. SWIFT will use two of those ports with every sync.

The subsequent sections highlight each of the above requirements in detail.

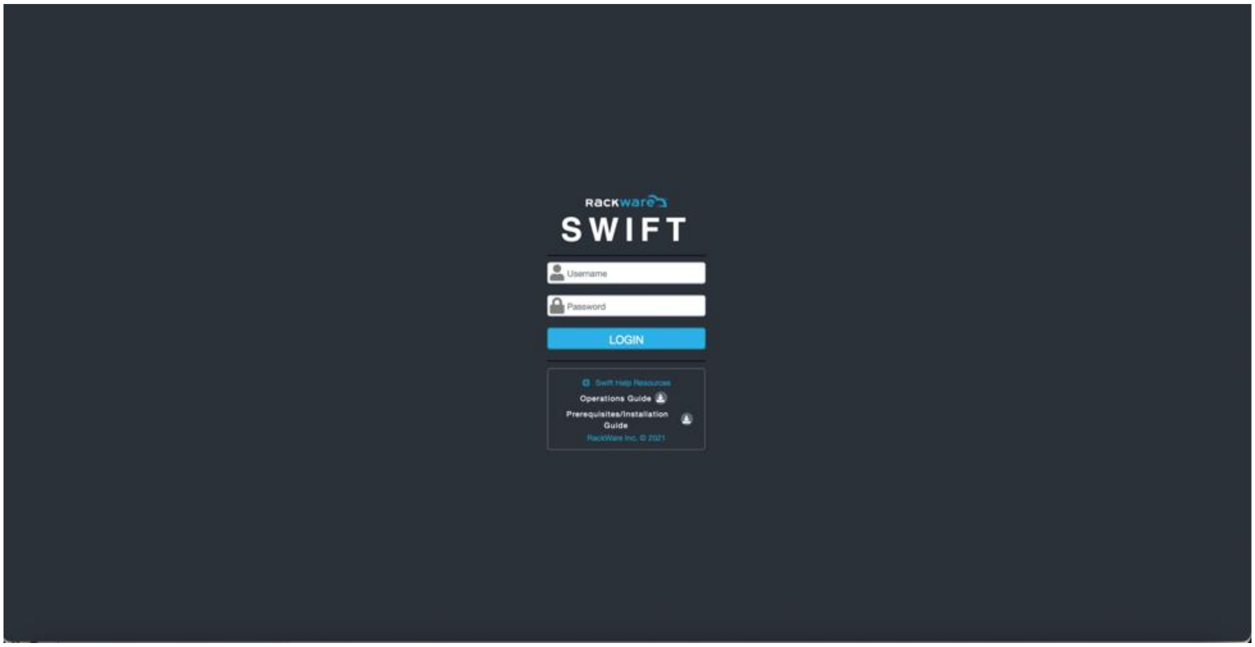
Login to the SWIFT dashboard

Once you complete the installation of the SWIFT, you can do all SWIFT operations through its dashboard. The SWIFT also supports REST and CLI access, which is not covered in this document, but if you are interested in trying those, then please contact RackWare support.

To access the SWIFT dashboard, use the URL:

https://<swift-ip>/swift/dashboard

It will look like this:



To login to the SWIFT dashboard, you would use the 'admin' account password, which you set up with steps mentioned in the next section. The 'admin' account will work like a super-admin within the SWIFT. After logging in with the admin user, you can create more organizations (user-groups) and users from the SWIFT dashboard.

Set password for the admin user

After fresh install, the 'admin' user will be created by the SWIFT in its CMDB. But to login with it, you first need to set the password for it, if not already set during the installation. Login to the SWIFT server using SSH, and then set the admin user password with below command:

```
sudo swiftcli user modify admin --password <yournewpassword>
```

Note: The password must be at least 8 characters long and must contain:

- At least one lower-case letter
- At least one upper-case letter
- At least one special character
- At least one digit

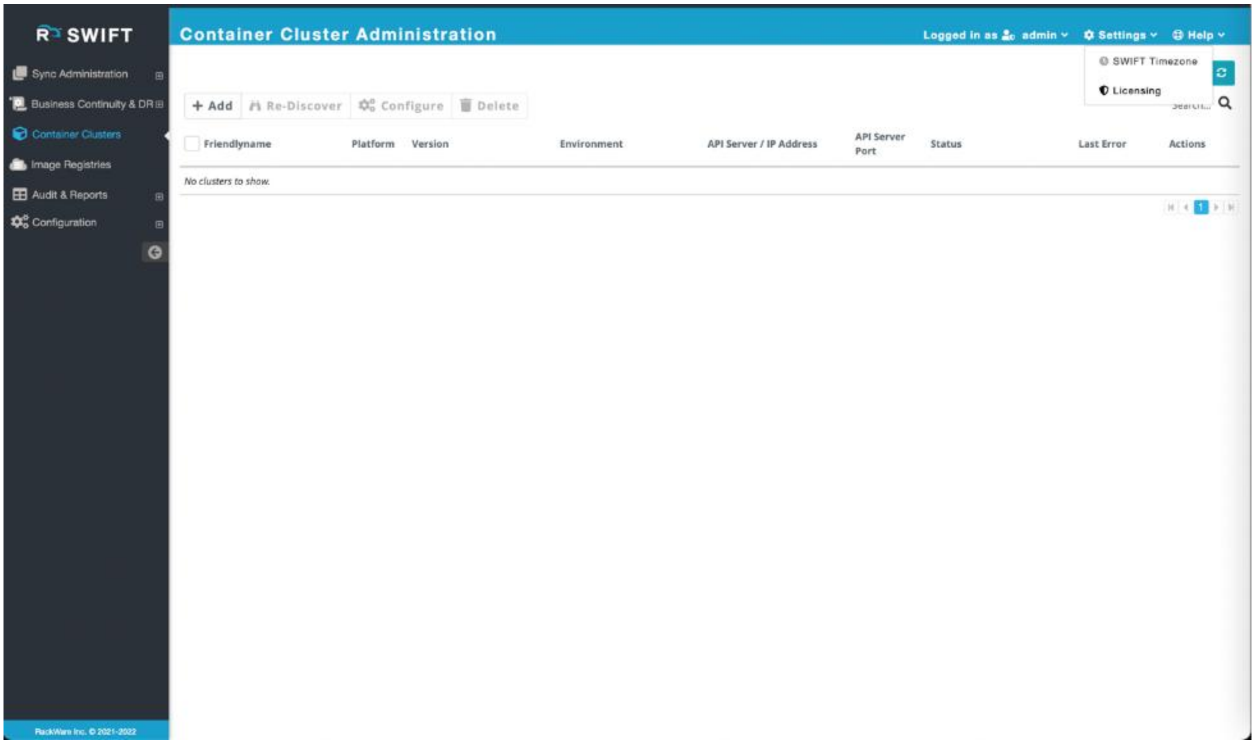
You can then login to the SWIFT dashboard with the 'admin' user and the above set '<yournewpassword>.'

Activating the SWIFT installation with a license

If you have installed a SWIFT product and logged in, then congratulations! It is the first important step towards making your containerization journey seamless. The next step is to activate a license. By default, your SWIFT server is enabled with a 'free-tier' license. Depending on the version of the SWIFT, you will have certain free licenses available and activated so you can start using the SWIFT.

How to check my existing/free license?

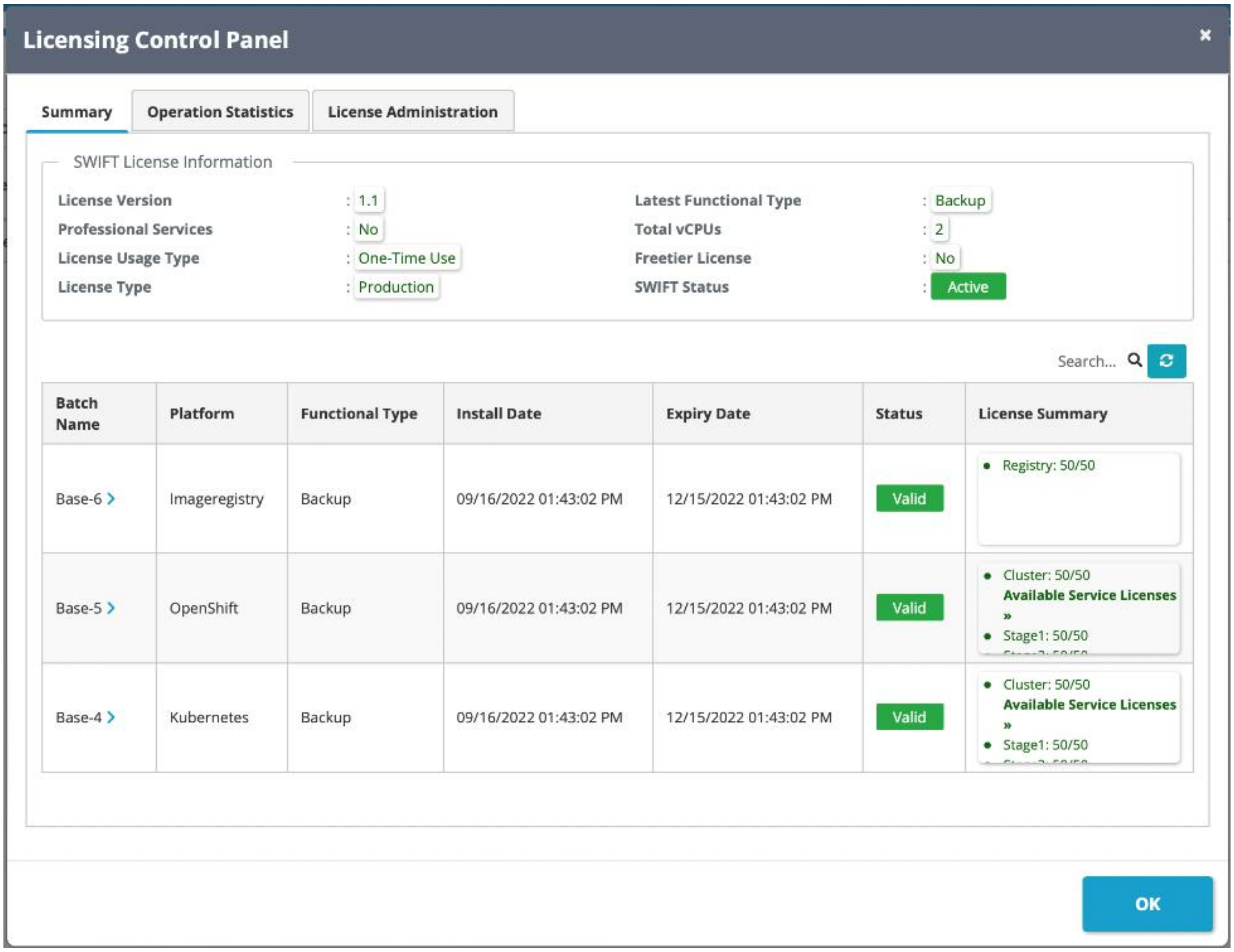
Login to the SWIFT dashboard and click on the 'Settings' menu and then the 'Licensing' submenu.



Friendlyname	Platform	Version	Environment	API Server / IP Address	API Server Port	Status	Last Error	Actions

No clusters to show.

By default, it will show you your existing available licenses.



Licensing Control Panel

SWIFT License Information

License Version	: 1.1	Latest Functional Type	: Backup
Professional Services	: No	Total vCPUs	: 2
License Usage Type	: One-Time Use	Freetier License	: No
License Type	: Production	SWIFT Status	: Active

Batch Name	Platform	Functional Type	Install Date	Expiry Date	Status	License Summary
Base-6 ›	Imageregistry	Backup	09/16/2022 01:43:02 PM	12/15/2022 01:43:02 PM	Valid	● Registry: 50/50
Base-5 ›	OpenShift	Backup	09/16/2022 01:43:02 PM	12/15/2022 01:43:02 PM	Valid	● Cluster: 50/50 **Available Service Licenses »** ● Stage1: 50/50
Base-4 ›	Kubernetes	Backup	09/16/2022 01:43:02 PM	12/15/2022 01:43:02 PM	Valid	● Cluster: 50/50 **Available Service Licenses »** ● Stage1: 50/50

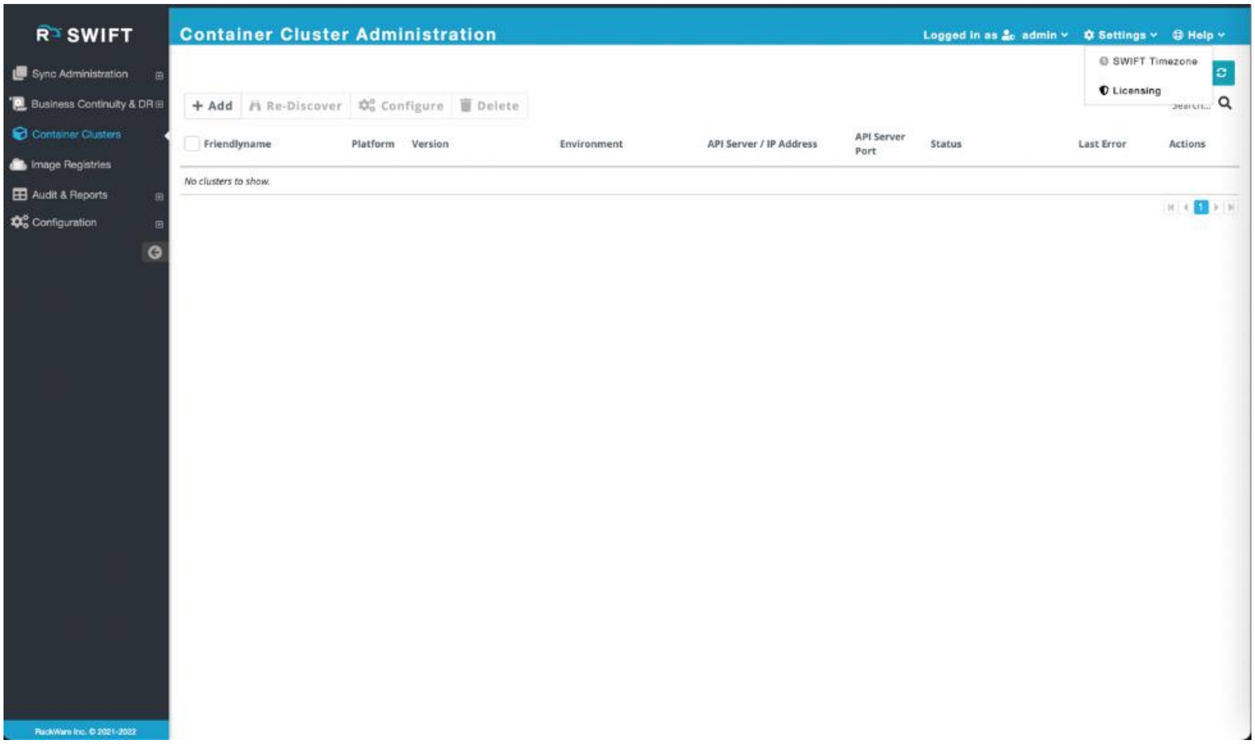
Typically, after a SWIFT install, you will get certain free licenses to try out the SWIFT. Those and any other licenses you applied to the SWIFT so far are all listed on the page above.

You will start with a 'base' license with absolute validity and counts, and then you would apply for an 'add-on' license in the future to extend the base license validity and/or counts.

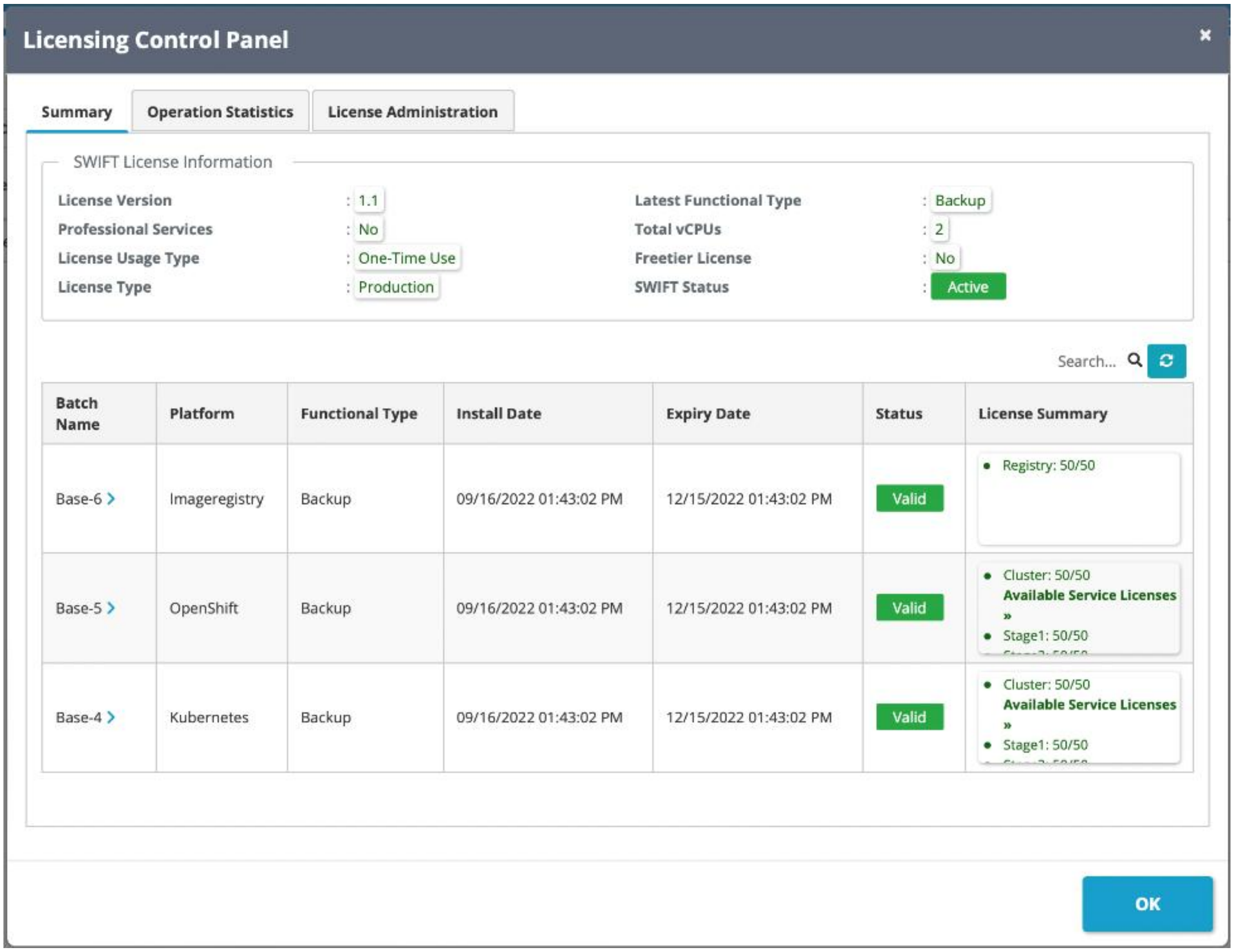
You will see the status of all base and add-on licenses on the above page.

How to request and apply a new production license?

Login to the SWIFT dashboard and click on the 'Settings' menu and then the 'Licensing' submenu.



By default, it will show you your existing available licenses.



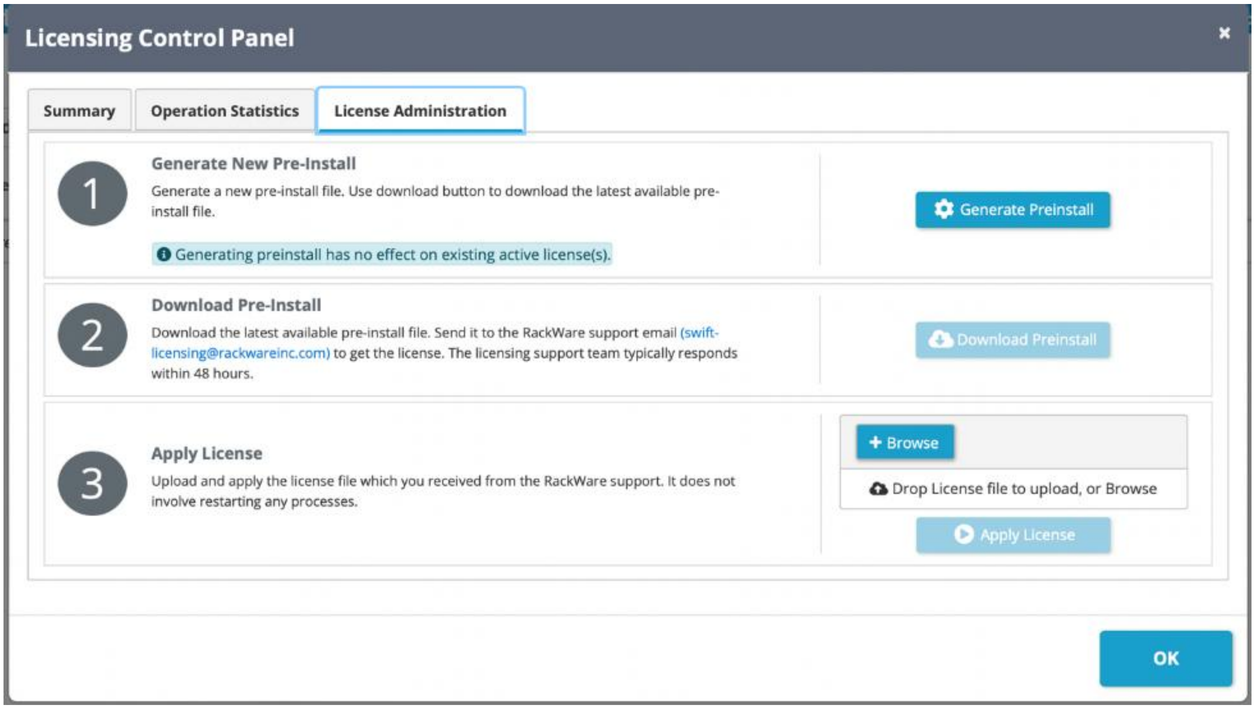
SWIFT License Information

License Version	: 1.1	Latest Functional Type	: Backup
Professional Services	: No	Total vCPUs	: 2
License Usage Type	: One-Time Use	Freetier License	: No
License Type	: Production	SWIFT Status	: Active

Search...

Batch Name	Platform	Functional Type	Install Date	Expiry Date	Status	License Summary
Base-6 ›	Imageregistry	Backup	09/16/2022 01:43:02 PM	12/15/2022 01:43:02 PM	Valid	• Registry: 50/50
Base-5 ›	OpenShift	Backup	09/16/2022 01:43:02 PM	12/15/2022 01:43:02 PM	Valid	• Cluster: 50/50 **Available Service Licenses »** • Stage1: 50/50
Base-4 ›	Kubernetes	Backup	09/16/2022 01:43:02 PM	12/15/2022 01:43:02 PM	Valid	• Cluster: 50/50 **Available Service Licenses »** • Stage1: 50/50

By default, it will show you your existing available licenses. Click on the ‘License Administration’ tab.



Generate New Pre-Install

Generate a new pre-install file. Use download button to download the latest available pre-install file.

Generating preinstall has no effect on existing active license(s).

Download Pre-Install

Download the latest available pre-install file. Send it to the RackWare support email (swift-licensing@rackwareinc.com) to get the license. The licensing support team typically responds within 48 hours.

Apply License

Upload and apply the license file which you received from the RackWare support. It does not involve restarting any processes.

Getting and applying a new production license is a three-step process, as the administration dialog shows.

Step-1: Click on the ‘Generate Preinstall’ button. It will generate a new preinstall file, which is a binary file with some crucial details captured about the SWIFT installation (No sensitive information is captured about the SWIFT server).

Step-2: Download the generated preinstall file and email it to the RackWare support email (swift-licensing@rackwareinc.com) as pointed by the Dashboard. Email to licensing support will automatically create a support ticket for you, and you will get an acknowledgment email within a few minutes after sending the preinstall file. Typically, you will get an email response to your license request within 48 hours. The RackWare Support team will also ship a valid license file to you along with the email response (Note that you must have valid licenses purchased from RackWare).

Alternatively, you can contact the RackWare Sales team and work with your account representative to get the license file. The Sales team will also ask you for the preinstall file, so store it safely. You can generate a fresh preinstall anytime and it doesn’t affect your existing license or its validity.

Step-3: Once you receive a valid license file (in step-2 above), then upload and apply the license file from the above license administration GUI. The new license will be activated immediately. After applying a license, please wait for up to 30 seconds for the new license to get enabled successfully.

SWIFT supported container platforms

Below are SWIFT supported container platforms along with supported versions for each. SWIFT being any-to-any replication and DR solution, you will be able to replicate and achieve DR between any of these platforms used as a source, as well as a target.

Supported Platform	Supported Version
Kubernetes (Opensource/Vanilla)	1.14+
OpenShift Origins (Opensource/Vanilla)	4.5+
OpenShift Dedicated (AWS/GCP)	4.5+
Azure RedHat OpenShift (ARO)	4.5+
IBM OpenShift cloud	4.5+
Oracle OCI OKE	1.14+
Microsoft Azure AKS	1.14+
Amazon AWS EKS	1.14+
Google GCP GKE	1.14+
Oracle OLCNE	1.14+
IBM Kubernetes Service Cloud	1.14+

For any other platform or version that is not listed here in the list, please contact RackWare support at support@rackwareinc.com.

SWIFT supported storage-types for source clusters

Below are SWIFT supported storage vendors for the source cluster. SWIFT being any-to-any replication and DR solution, you will be able to replicate and achieve DR between any of these volume-types. Note that below volume-types only apply to the source cluster and any storage/volume-type is allowed for the target.

Supported Storage Types
Azure Disk, File, and CSI volumes (Premium as well as all SKU combinations supported for all types)
Oracle Block storage and CSI volumes
Amazon EBS, EFS, FSx, and GP volumes (CSI and non-CSI storage types)
Google block storage (CSI and non-CSI storage types)
IBM Classic File and Block Storage (CSI and non-CSI storage types)
IBM VPC Block Storage (CSI and non-CSI storage types)
Ceph storage (CSI and non-CSI storage types)
Any CSI volumes*

***** Any storage used through CSI interface/drivers needs to support snapshot capability to be able to work with SWIFT

SWIFT supported container registries

Below are SWIFT supported container registries. SWIFT being any-to-any replication and DR solution, you will be able to replicate and achieve DR between any of these registry platforms used as a source, as well as a target.

Supported Registry Platform
Azure Container Registry (ACR)
Amazon Elastic Container Registry (ECR)
Google Container Registry (GCR)
Oracle Cloud Infrastructure Registry (OCIR)
Docker Hub

For any other platform or version that is not listed here in the list, please contact RackWare support at support@rackwareinc.com.

SWIFT port usage

Various operations are supported for the SWIFT managed container platform. The required network ports will change depending on the type of the container platform and the type of operation performed. The next section highlights port usage per platform and operation type.

If there is any intermediate firewall between the SWIFT server and your remote container platform, then these below ports also need to be opened in the intermediate firewall.

Kubernetes Discover

Port Number (Default)	Direction	Can it be changed?	Purpose
<Kubernetes-API-Server-IP>:443	SWIFT to cluster	Yes (specify during discover and change from configure operation for the cluster)	Talk to API service

Kubernetes Configure

Port Number (Default)	Direction	Can it be changed?	Purpose
<Kubernetes-API-Server-IP>:443	SWIFT to cluster	Yes (specify during discover and change from configure operation for the cluster)	Talk to API service

Kubernetes Sync

Port Number (Default)	Direction	Can it be changed?	Purpose
<Kubernetes-API-Server-IP>:443	SWIFT to cluster	Yes (specify during discover and change from configure operation for the cluster)	Talk to API service
<TRAI-Service-IP>:<port-1> [Default port-range is typically: 32000-34000)	SWIFT to cluster	Yes (specify during sync else auto picked)	Connect to transient sync staging POD/service within the cluster over HTTPS (Management port)
<TRAI-Service-IP>:<port-2> [Default port-range is typically: 32000-34000)	SWIFT to cluster	Yes (specify during sync else auto picked)	Pull/push data from/to transient sync staging POD/service within the cluster over an encrypted tunnel (Data port)

SWIFT sync run will typically use two unique ports from the cluster's service port range on both sides of clusters. If you do sync on a namespace basis and the selected namespace has more than one region or zone set for persistent volumes on the source cluster side, then SWIFT will run its transient TRAI-Pod for each region/zone. In such cases, you will need 2xTRAI-Pod number of ports per sync. E.g., if your synced source namespace has two regions or zones for persistent volumes and you are syncing the entire namespace as part of a sync run, then the sync will deploy two TRAI-Pods (one for each region or zone of source namespace) and so the sync will need 2x2=4 ports from the source cluster's service port range. These ports can be fixed or can be auto picked by SWIFT. In all cases, SWIFT will always need only two ports per sync run for the target or DR cluster.

Note that any transient TRAI service/Pod ports, either auto picked by SWIFT or specified manually, need to be opened in all intermediate firewalls between the SWIFT server and your cluster. If you want SWIFT to auto pick required ports from the service port range of the cluster, then it is recommended that you whitelist the entire cluster service port range in all intermediate firewalls between the SWIFT server and your cluster.

The <TRAI-Service-IP> can be any of the supported IP types for the cluster services (E.g., ClusterIP, NodePort, LoadBalancer, etc.) and can be optionally specified during the sync.

Depending on the selected TRAI service type, sync will default to auto picked IP for its transient TRAI Kubernetes Service. If you employ NodePort service type for SWIFT's transient TRAI service, then you need to open required sync ports against all cluster node IPs (Also known as NodePort IPs).

OpenShift Discover

Port Number (Default)	Direction	Can it be changed?	Purpose
<OpenShift-API-Server-IP>:443	SWIFT to cluster	Yes (specify during discover and change from configure operation for the cluster)	Talk to API service

OpenShift Configure

Port Number (Default)	Direction	Can it be changed?	Purpose
<OpenShift-API-Server-IP>:443	SWIFT to cluster	Yes (specify during discover and change from configure operation for the cluster)	Talk to API service

OpenShift Sync

Port Number (Default)	Direction	Can it be changed?	Purpose
<OpenShift-API-Server-IP>:443	SWIFT to cluster	Yes (specify during discover and change from configure operation for the cluster)	Talk to API service
<TRAI-Service-IP>:<port-1> [Default port-range is typically: 32000-34000)	SWIFT to cluster	Yes (specify during sync else auto picked)	Connect to transient sync staging POD/service within the cluster over HTTPS (Management port)
<TRAI-Service-IP>:<port-2> [Default port-range is typically: 32000-34000)	SWIFT to cluster	Yes (specify during sync else auto picked)	Pull/push data from/to transient sync staging POD/service within the cluster over an encrypted tunnel (Data port)

SWIFT sync run will typically use two unique ports from the cluster's service port range on both sides of clusters. If you do sync on a namespace basis and the selected namespace has more than one region or zone set for persistent volumes on the source cluster side, then SWIFT will run its transient TRAI-Pod for each region/zone. In such cases, you will need 2xTRAI-Pod number of ports per sync. E.g., if your synced source namespace has two regions or zones for persistent volumes and you are syncing the entire namespace as part of a sync run, then the sync will deploy two TRAI-Pods (one for each region or zone of source namespace) and so the sync will need 2x2=4 ports from the source cluster's service port range. These ports can be fixed or can be auto picked by SWIFT. In all cases, SWIFT will always need only two ports per sync run for the target or DR cluster.

Note that any transient TRAI service/Pod ports, either auto picked by SWIFT or specified manually, need to be opened in all intermediate firewalls between the SWIFT server and your cluster. If you want SWIFT to auto pick required ports from the service port range of the cluster, then it is recommended that you whitelist the entire cluster service port range in all intermediate firewalls between the SWIFT server and your cluster. In all cases, SWIFT will always need only two ports per sync run for the target or DR cluster.

The <TRAI-Service-IP> can be any of the supported IP types for the cluster services (E.g., ClusterIP, NodePort, LoadBalancer, etc.) and can be optionally specified during the sync.

Depending on the selected TRAI service type, sync will default to auto picked IP for its transient TRAI OpenShift Service. If you employ NodePort service type for SWIFT's transient TRAI service, then you need to open required sync ports against all cluster node IPs (Also known as NodePort IPs).

Image Registry Discover

Port Number (Default)	Direction	Can it be changed?	Purpose
<Container-Image-Registry-API-Server-IP>:443	SWIFT to API server	No	Talk to API service

Image Registry Configure

Port Number (Default)	Direction	Can it be changed?	Purpose
<Container-Image-Registry-API-Server-IP>:443	SWIFT to API server	No	Talk to API service

Image Registry Sync

Port Number (Default)	Direction	Can it be changed?	Purpose
<Container-Image-Registry-API-Server-IP>:443	SWIFT to Container Image Registry API server	No	Talk to Container Image Registry API service

The above ports for Image Registry sync are only required if you plan to migrate or set up Disaster Recovery (DR) for your Container Image Registries with SWIFT.

DR Policy Operations

The port usage for DR policy remains the same as respective platform type (discovery and sync) ports documented in the earlier sections, depending on source and/or target platform type selected for the policy and the type of the policy. Additionally, for successful email alerts, the below ports need to be opened from the SWIFT server to the target email server for all configured email accounts where alerts are configured.

Port Number (Default)	Direction	Can it be changed?	Purpose
TCP/587 or custom SMTP port used by the target email server.	SWIFT to remote Email server	Yes (If non-default port used, then specify it in the SWIFT options file for email server options)	Send DR email alerts using local Linux email client

SWIFT uses local Linux email client APIs to relay email alert messages to the configured email accounts for DR policy alerts. If your environment uses non-standard or custom SMTP port, then you should whitelist that port instead of above port. Make sure to specify the custom SMTP port in the SWIFT options file too under the below options, so SWIFT can use that for relaying email alerts for policies. (The SWIFT options file is located on the SWIFT server at: /opt/swift/data/options).

dev_swiftAlertOptions_emailAlertOptions_emailServerUrl=smtp://127.0.0.1:587
dev_swiftAlertOptions_emailAlertOptions_localEmailServer=smtp://127.0.0.1:587

Note: It is important you uncomment the line (by removing # at the start of the line) for set options in the options file, and then restart SWIFT service for the newly set options to take effect.

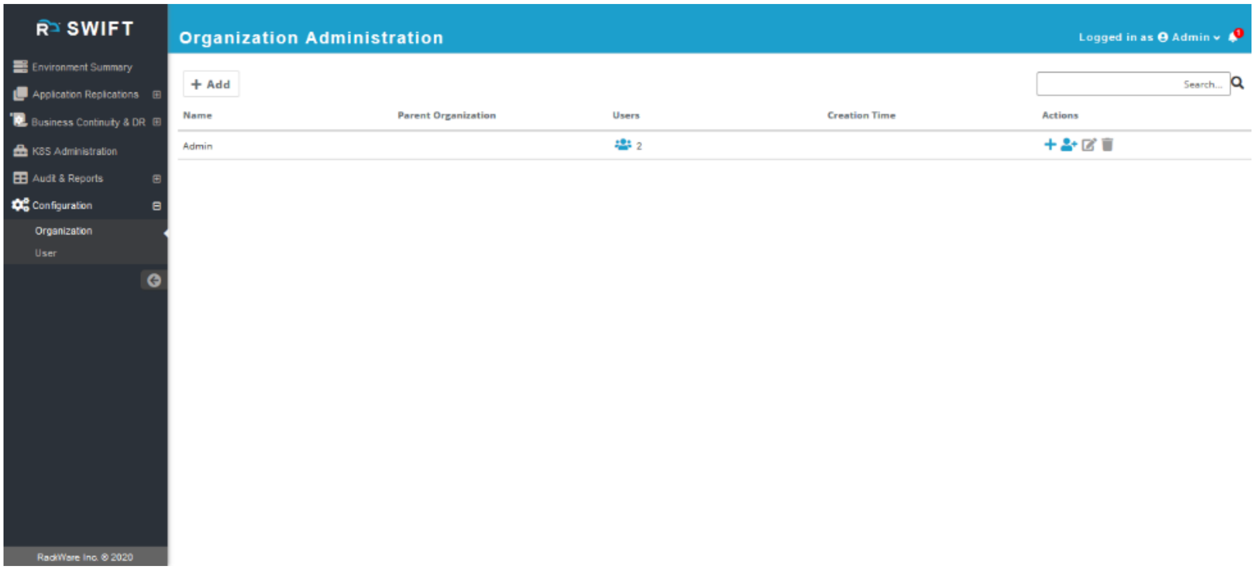
Add more users to the SWIFT

The default admin account for the SWIFT will be the 'admin' user, which is also a local Linux user where the SWIFT is installed. Once you log in initially with this user, you can optionally set up more users and their organizations for access control. An organization is a group of users, and it can also contain one or more child organizations. Within an organization, there can be two types of users:

- Admin users
  These have full access rights to the corresponding organization as well as all child organizations. Admin users can also add or remove child organizations and users within their organization.
- Operator users
  These users can not add or remove any child organization or users from any of those. However, these users have full rights to perform all other regular SWIFT operations.

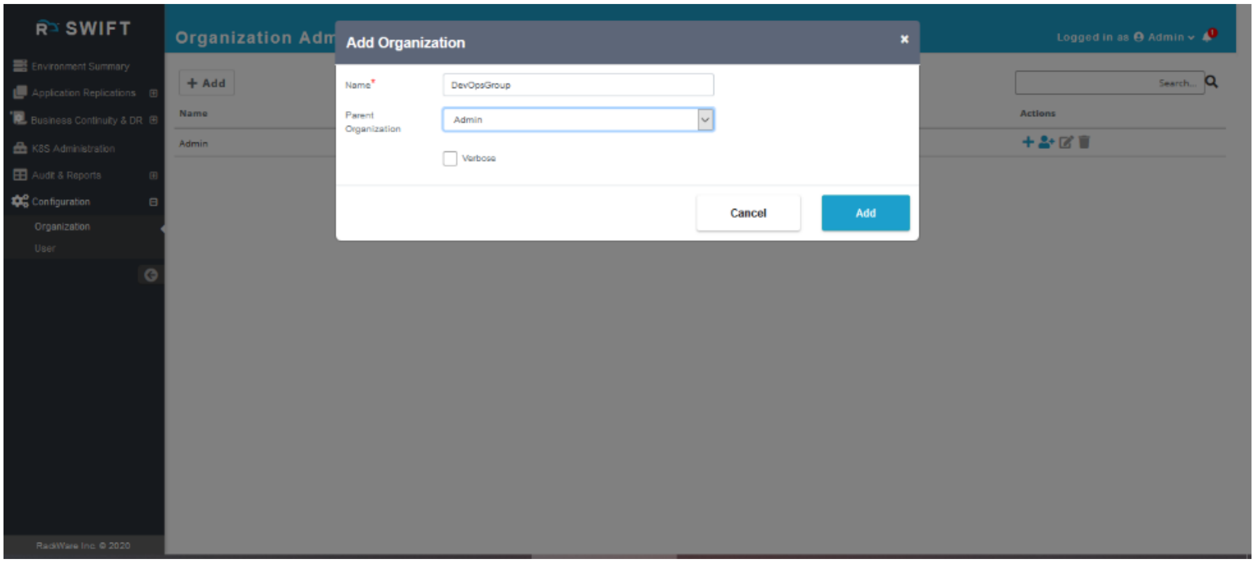
Create a new organization

Login to the SWIFT dashboard and navigate to the Configuration menu and Organization sub-menu.



Name	Parent Organization	Users	Creation Time	Actions
Admin		2		

You will see the built-in ‘Admin’ organization created already. The built-in organization can not be deleted. Press on the ‘+ Add’ button, and enter new organization details.



Add Organization

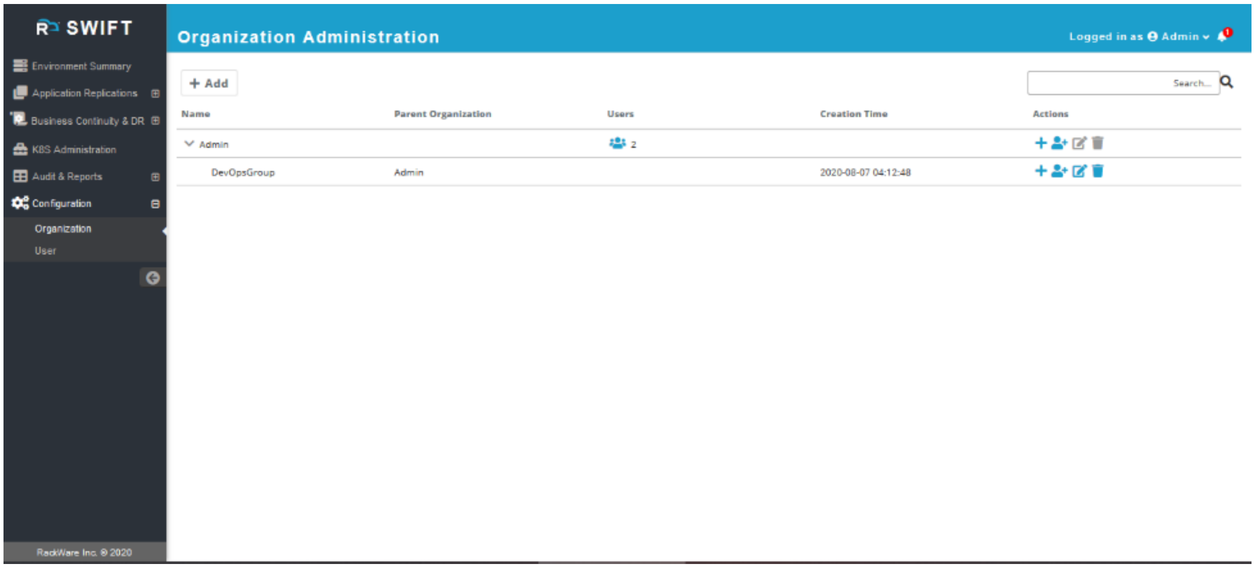
Name*: DevOpsGroup

Parent Organization: Admin

☐ Verbose

Cancel | Add

You will see the newly added organization now.

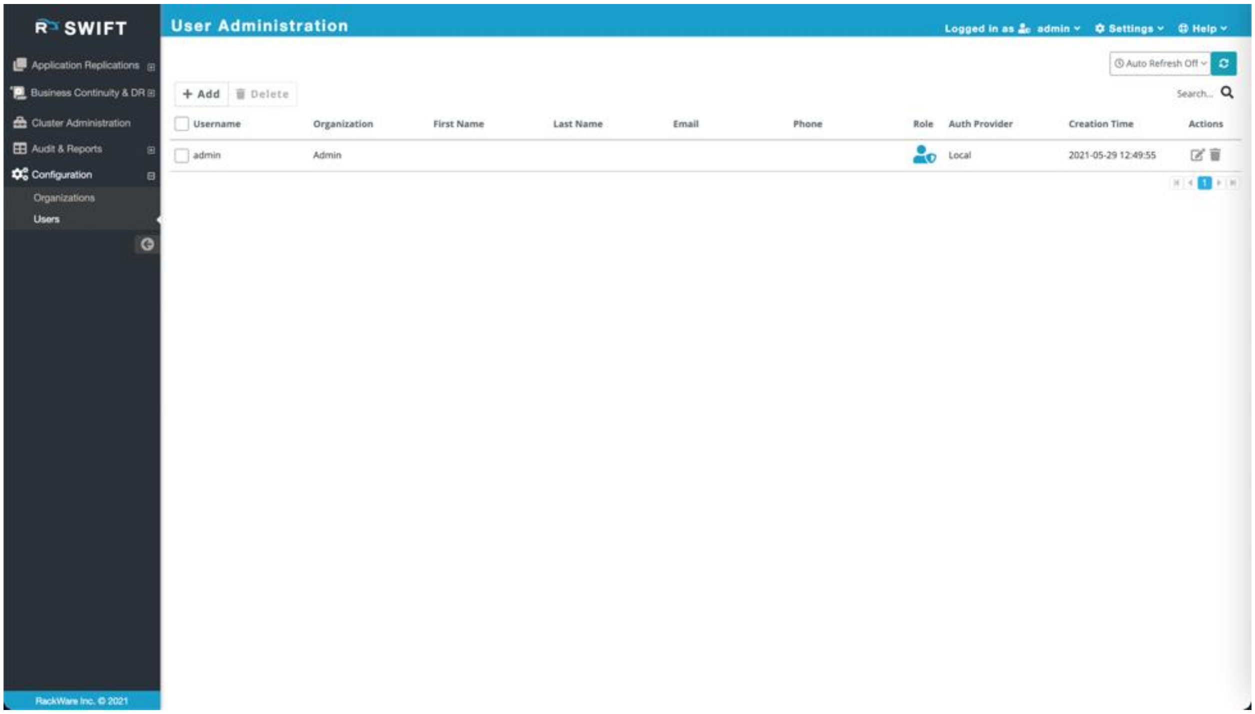


Organization Administration

Name	Parent Organization	Users	Creation Time	Actions
Admin		2		
DevOpsGroup	Admin		2020-08-07 04:12:48	

Create a new user

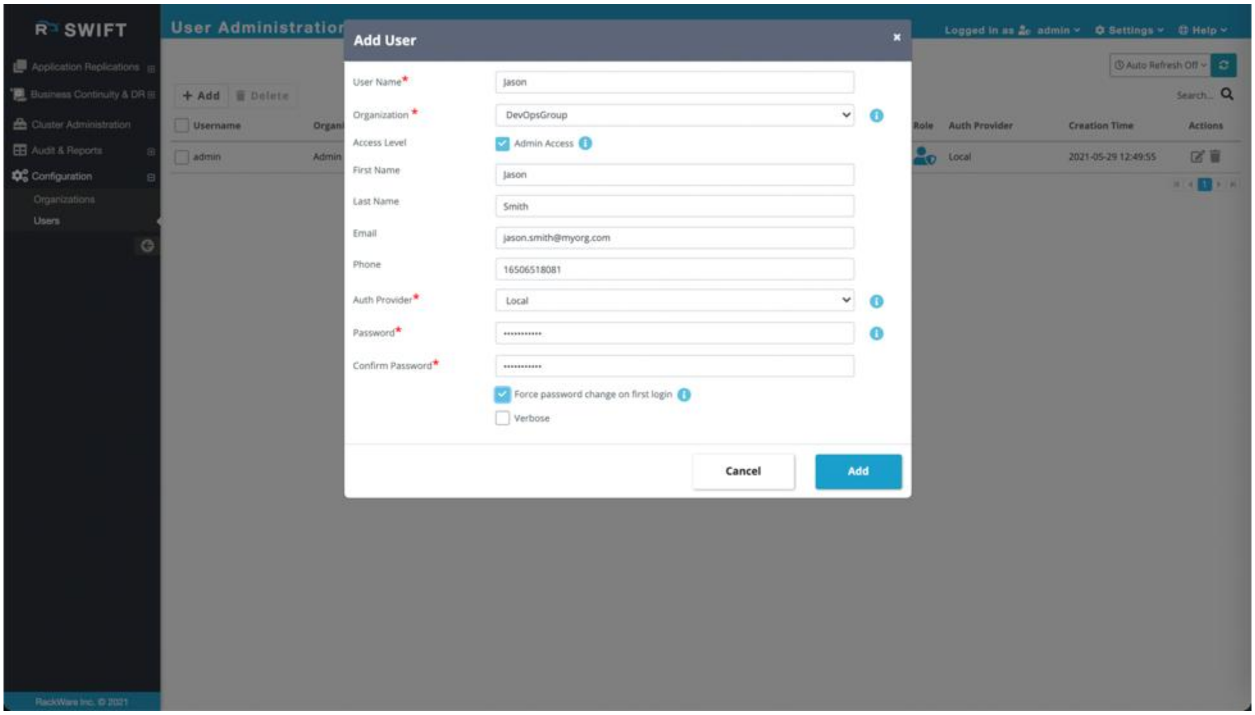
Login to the SWIFT dashboard and navigate to the Configuration menu and User sub-menu. You will always see at least two users – root and admin, which are built-in users and can’t be modified or deleted. The root and admin users also act as admin users for the ‘Admin’ organization.



User Administration

Username	Organization	First Name	Last Name	Email	Phone	Role	Auth Provider	Creation Time	Actions
admin	Admin						Local	2021-05-29 12:49:55	

Click on the '+ Add' button and enter user details.



Add User

User Name* Jason

Organization * DevOpsGroup

Access Level ☑ Admin Access

First Name Jason

Last Name Smith

Email jason.smith@myorg.com

Phone 16506518081

Auth Provider* Local

Password* ••••••••••••

Confirm Password* ••••••••••••

☑ Force password change on first login

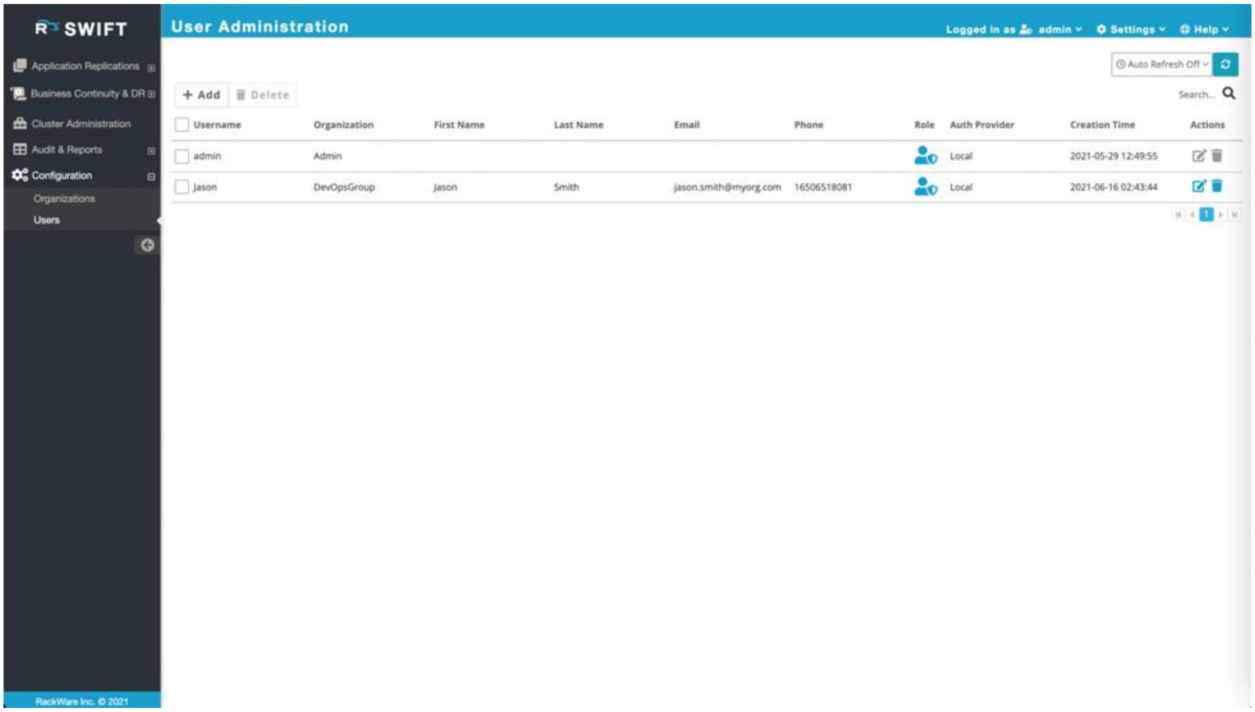
☐ Verbose

Enable the checkbox for 'Admin Access' if you want to grant the new user the admin role for the user's selected organization as well as for all child organizations of the selected organization. Note that you can only add a user to the specific organization if you have admin access for the organization.

Select an auth provider, which is an identity provider configured in the SWIFT. In most cases, you will select the 'Local' identity provider, which means the created users are stored locally in the SWIFT CMDB.

If you are organization admin and creating this user for your group, then you can optionally set 'Force password change on First login' checkbox. Setting this will allow the new user to login to dashboard with temporary password you set here, and then also enforce password change on the first login.

Once the user is added successfully, you will be able to see it listed on the user page.



Username	Organization	First Name	Last Name	Email	Phone	Role	Auth Provider	Creation Time	Actions
admin	Admin						Local	2021-05-29 12:49:55	
Jason	DevOpsGroup	Jason	Smith	jason.smith@myorg.com	16506518081		Local	2021-06-16 02:43:44	

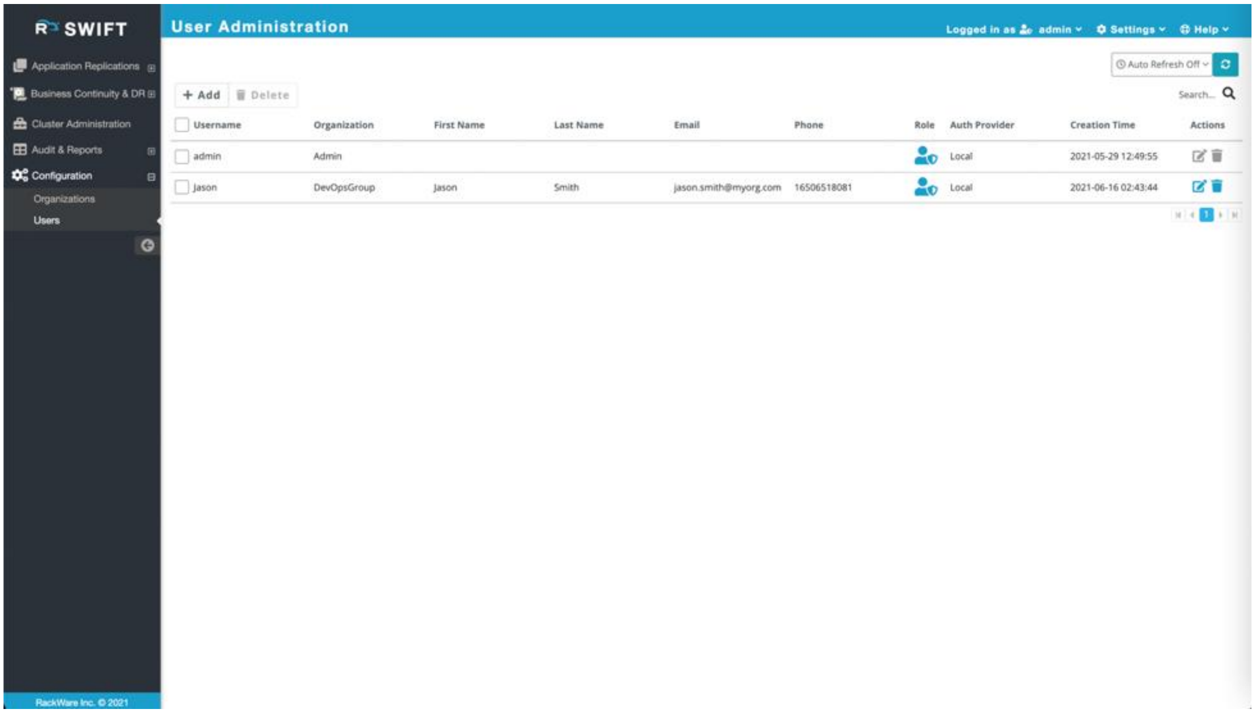
The newly created user can now login to the SWIFT dashboard with the set credentials.

Deleting users from the SWIFT

You can delete users as well as an organization from the SWIFT using the SWIFT dashboard. The below sections below highlight steps for removing both an organization and a user.

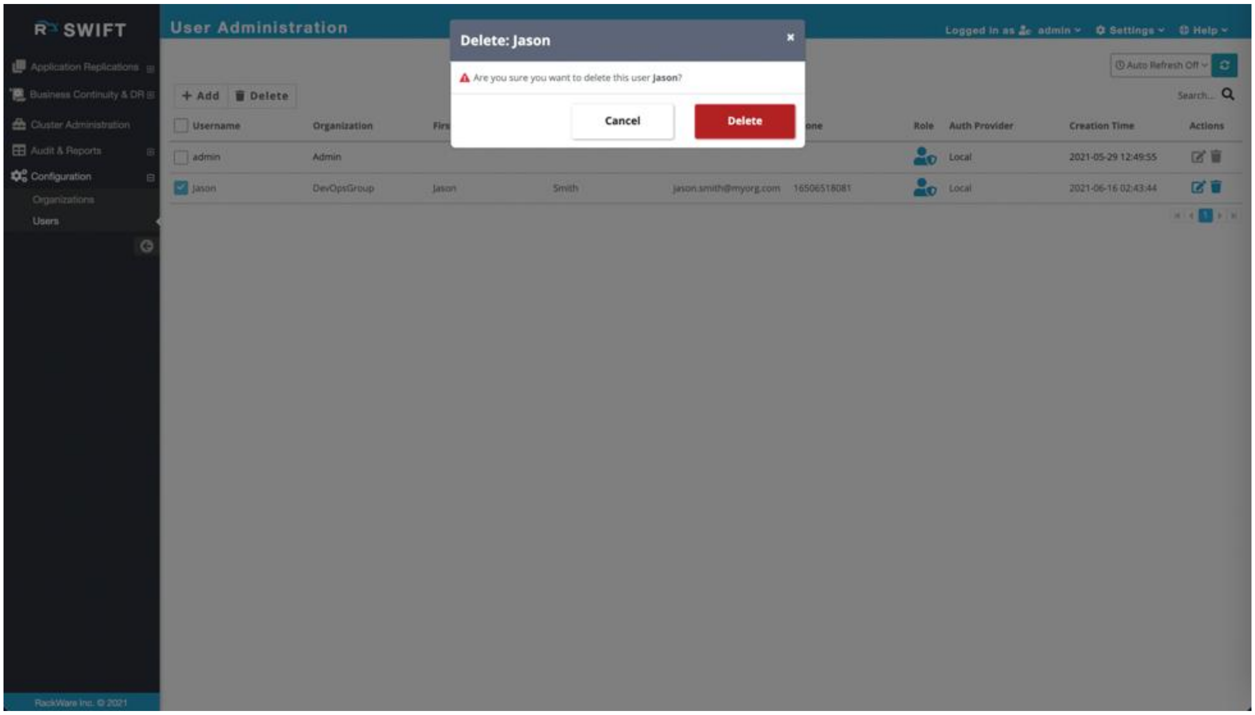
Deleting a user

Login to the SWIFT dashboard and navigate to the Configuration menu and User sub-menu. You will see the list of all users in your current organization as well as those in child organizations.



□ Username	Organization	First Name	Last Name	Email	Phone	Role	Auth Provider	Creation Time	Actions
□ admin	Admin						Local	2021-05-29 12:49:55	
□ Jason	DevOpsGroup	Jason	Smith	jason.smith@myorg.com	16506518081		Local	2021-06-16 02:43:44	

Select the user you want to delete and press the 'Delete' button. The dashboard would ask you for confirmation.

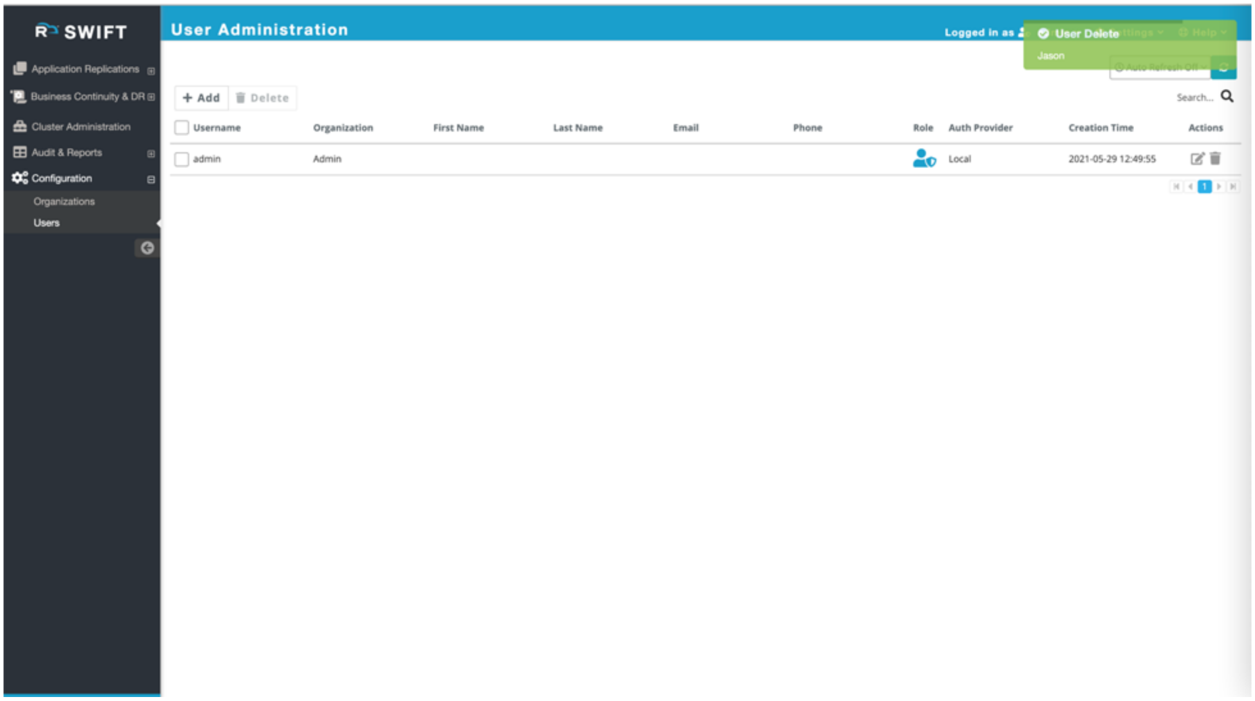


Delete: Jason

Are you sure you want to delete this user **Jason**?

□ Username	Organization	Firs	one	Role	Auth Provider	Creation Time	Actions
□ admin	Admin				Local	2021-05-29 12:49:55	
☑ Jason	DevOpsGroup	Jason	Smith jason.smith@myorg.com 16506518081		Local	2021-06-16 02:43:44	

Confirm, and the user will be deleted permanently.



User Delete

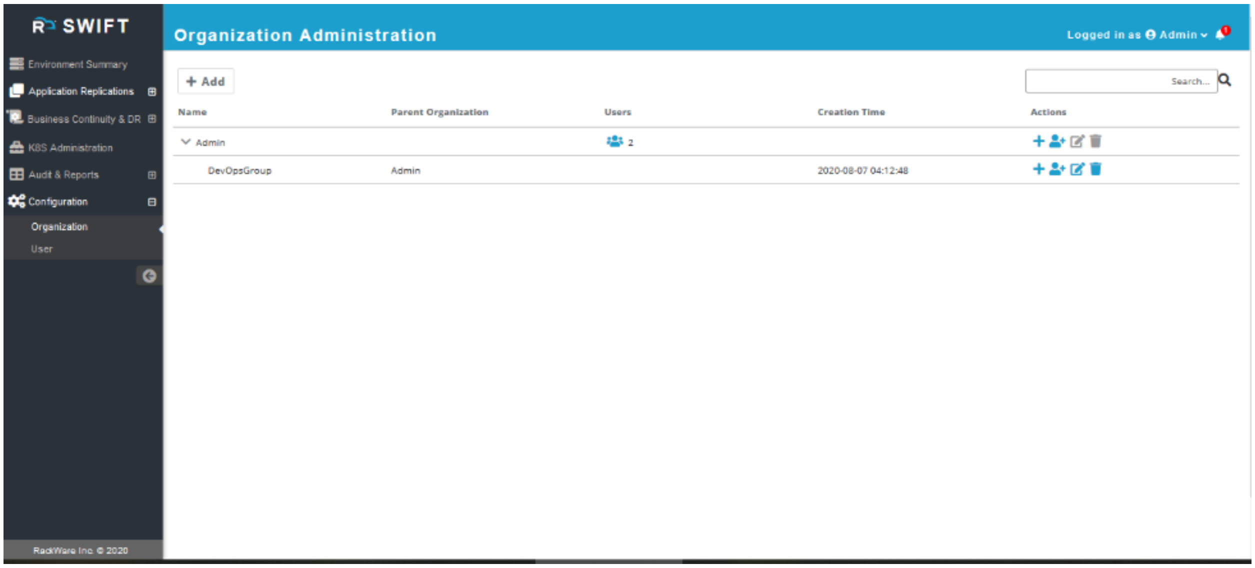
Jason

Username	Organization	First Name	Last Name	Email	Phone	Role	Auth Provider	Creation Time	Actions
□ admin	Admin						Local	2021-05-29 12:49:55	

Note that the audit trail for the user's operations is still retained even if the user is gone.

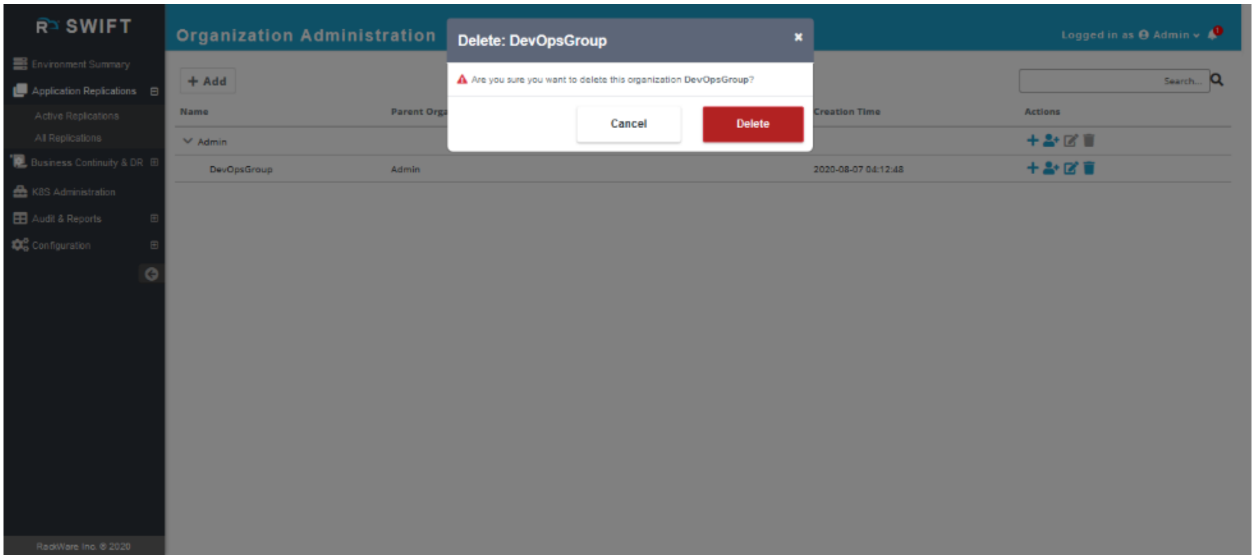
Deleting an organization

Login to the SWIFT dashboard and navigate to the Configuration menu and Organization sub-menu.



Name	Parent Organization	Users	Creation Time	Actions
Admin		2		
DevOpsGroup	Admin		2020-08-07 04:12:48	

Press the delete button (the one with a dustbin icon) next to the organization which you want to delete. You will get a confirmation dialog and press the 'Delete' button to continue. Note that if you have one or more users or child organizations in the selected organization, then the confirmation dialog would notify you of that. In such non-empty organization cases, you will get a 'Force' option on the confirmation dialog, which would then delete all child organizations and users recursively. It is recommended that you remove any child organizations and users individually than using the force option.



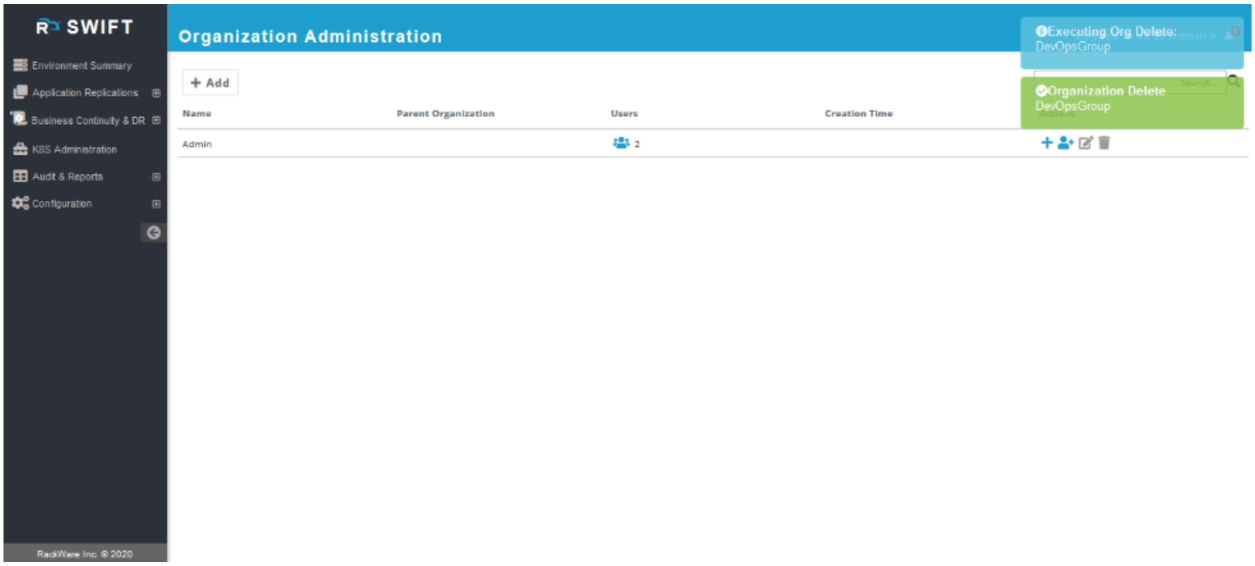
Delete: DevOpsGroup

Are you sure you want to delete this organization DevOpsGroup?

Cancel | Delete

Name	Parent Orga	Creation Time	Actions
Admin			
DevOpsGroup	Admin	2020-08-07 04:12:48	

Once deleted, you will see that the organization is no longer on the page.



Executing Org Delete: DevOpsGroup

Organization Delete DevOpsGroup

Name	Parent Organization	Users	Creation Time
Admin		2	

Creating cluster credentials for SWIFT use

Below sections highlight steps for creating cluster credentials for use with the SWIFT. Note that steps will change for every cloud and non-cloud installs.

The generated credentials are something you would use while configuring cluster details in the SWIFT or using object-storage for the cloud with SWIFT. The same credentials will also be used for the container image registry discover and administration for the respective clouds or platform types.

Adding local Kubernetes cluster service-account for SWIFT use

Before you can add your local (non-cloud) Kubernetes cluster to SWIFT and start managing it, you will need to have a cluster service account created with the necessary permissions.

Create a YAML for the new service account:

```
$ vi swift-admin-sa.yaml
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: swift-admin
  namespace: kube-system
```

Apply the YAML file

```
$ kubectl apply -f swift-admin-sa.yaml
```

Next, add the ‘cluster-admin’ role to the newly created account.

```
$ vi swift-admin-roles.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: swift-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: swift-admin
  namespace: kube-system
---
apiVersion: v1
kind: Secret
```

```
metadata:
  name: swift-admin
  namespace: kube-system
  annotations:
        kubernetes.io/service-account.name: swift-admin
type: kubernetes.io/service-account-token
```

Apply the YAML file

```
$ kubectl apply -f swift-admin-roles.yaml
```

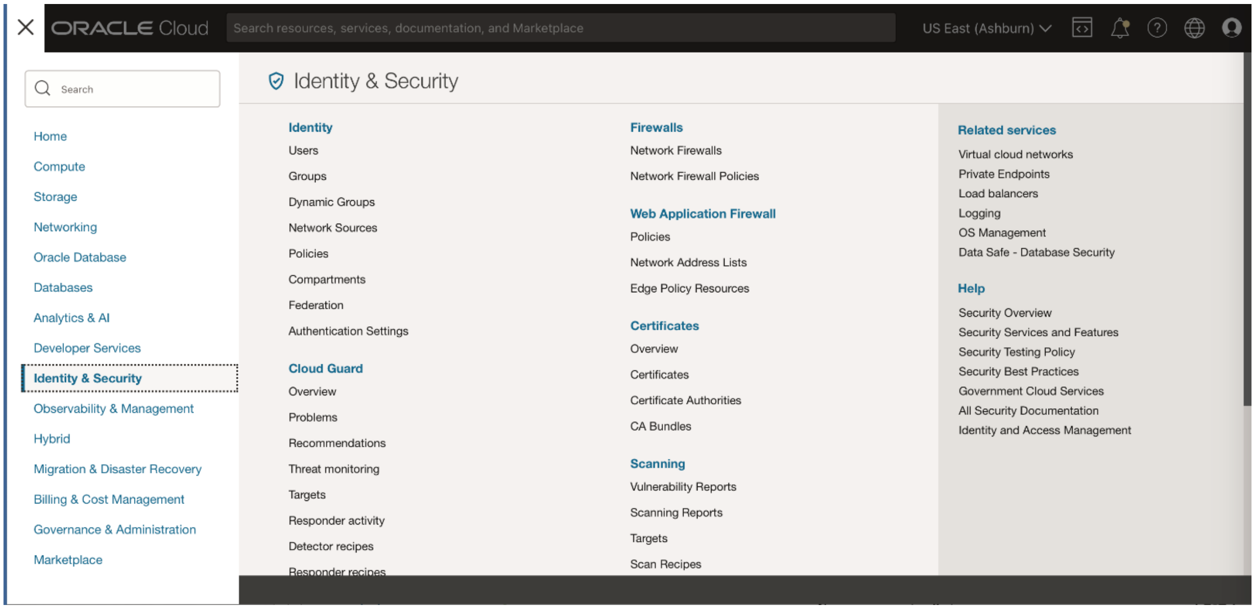
To get the service-account token, you can use a command as below. The command would print the 'token' key. You will use this output token later while adding the cluster to the SWIFT.

```
$ kubectl -n kube-system describe secret $(kubectl -n kube-system get secret | grep "swift-admin" | awk
'{print $1}')
```

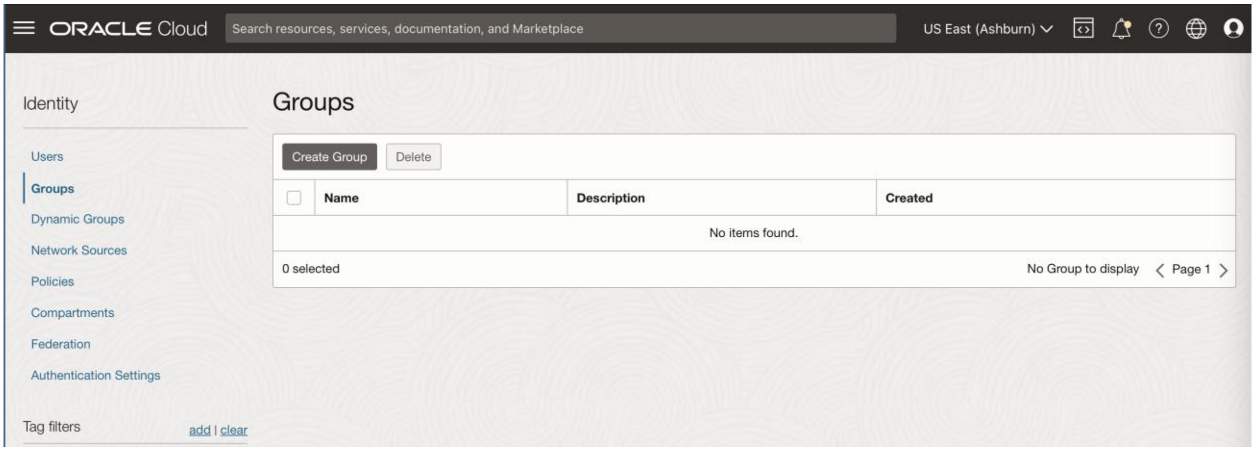
Adding Oracle Cloud Infrastructure (OCI) user for SWIFT use

This section highlights the steps to create an account under your OCI cloud tenancy, which you can use later to configure the cluster details under your installed SWIFT. The same credentials can also be used later to discover an Oracle Cloud Infrastructure Container Registry (OCIR) instance or add an OCI cloud object storage under your SWIFT.

Login to OCI console. Select the 'Identity & Security' submenu from the top left menu and then select 'Groups' option. We will create a Group, a Policy, and then finally a User.



Press the ‘Create Group’ option.

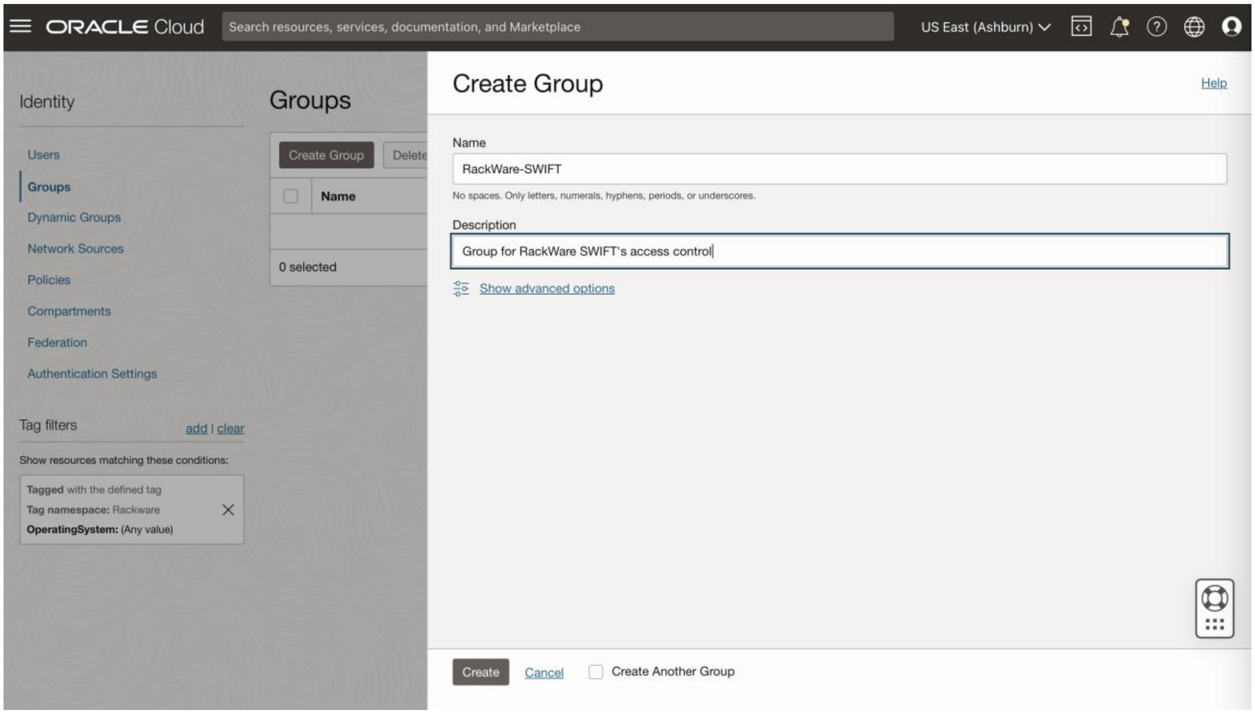


□	Name	Description	Created
		No items found.	

0 selected

No Group to display ‹ Page 1 ›

In the new ‘Create Group’ wizard, set appropriate name and description. In this example case, we will name it ‘RackWare-SWIFT.’



Show resources matching these conditions:

Tagged with the defined tag
Tag namespace: Rackware
OperatingSystem: (Any value)

Create Group

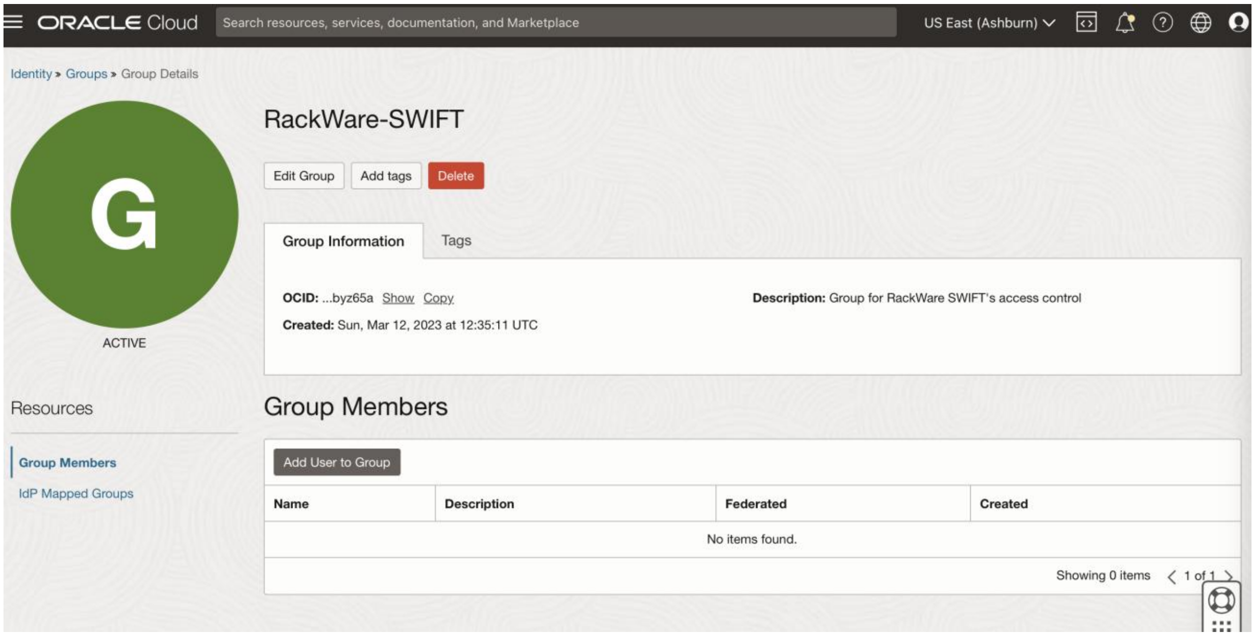
Name: RackWare-SWIFT

No spaces. Only letters, numerals, hyphens, periods, or underscores.

Description: Group for RackWare SWIFT's access control

□ Create Another Group

Once the group is created, we are ready for the next step of creating a Policy.



RackWare-SWIFT

ACTIVE

OCID: ...byz65a Show Copy

Created: Sun, Mar 12, 2023 at 12:35:11 UTC

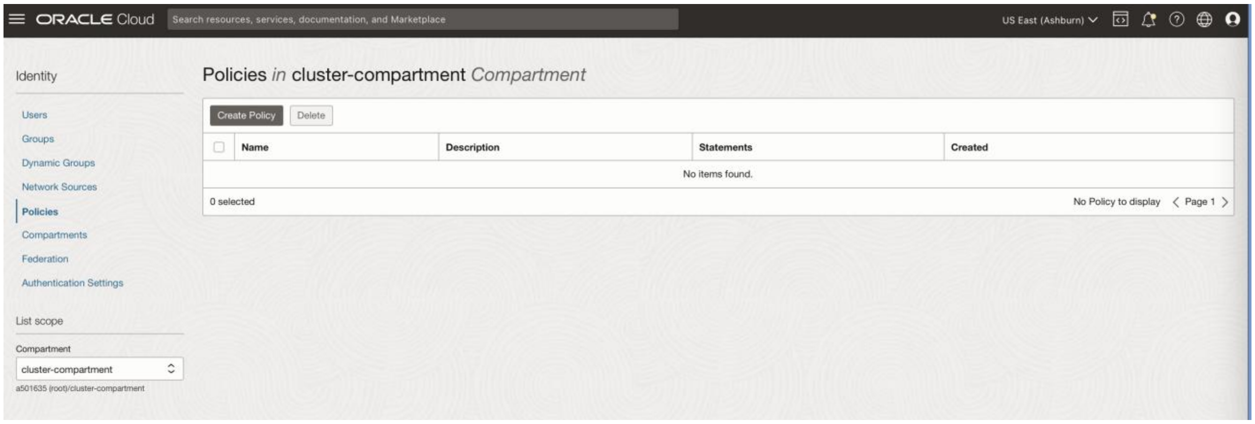
Description: Group for RackWare SWIFT's access control

Group Members

Name	Description	Federated	Created
No items found.			

Showing 0 items < 1 of 1 >

From the Identity menu, select ‘Policies’ submenu. Select compartment where OKE clusters or OCIR registries are located. Then press the ‘Create Policy’ button.



Policies *in* cluster-compartment *Compartment*

□	Name	Description	Statements	Created
No items found.				

0 selected — No Policy to display < Page 1 >

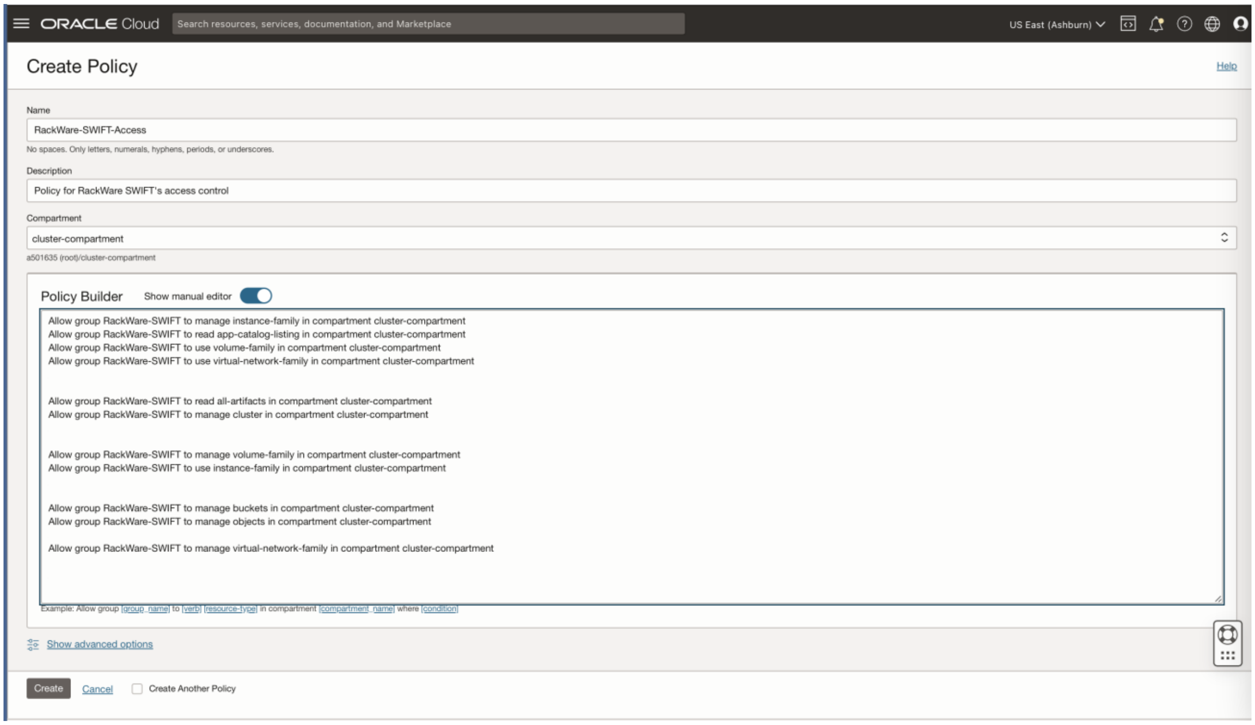
List scope

Compartment

cluster-compartment

a501635 (root)/cluster-compartment

For the new ‘Create Policy’ wizard, give policy a name and description. Then enable ‘Show manual editor’ slider. It will show you a textbox to edit policy rules that you can use to enter rules shown below.



Create Policy

Name: RackWare-SWIFT-Access

No spaces. Only letters, numerals, hyphens, periods, or underscores.

Description: Policy for RackWare SWIFT's access control

Compartment: cluster-compartment

a501635 (root)/cluster-compartment

Policy Builder — Show manual editor

```
Allow group RackWare-SWIFT to manage instance-family in compartment cluster-compartment
Allow group RackWare-SWIFT to read app-catalog-listing in compartment cluster-compartment
Allow group RackWare-SWIFT to use volume-family in compartment cluster-compartment
Allow group RackWare-SWIFT to use virtual-network-family in compartment cluster-compartment


Allow group RackWare-SWIFT to read all-artifacts in compartment cluster-compartment
Allow group RackWare-SWIFT to manage cluster in compartment cluster-compartment


Allow group RackWare-SWIFT to manage volume-family in compartment cluster-compartment
Allow group RackWare-SWIFT to use instance-family in compartment cluster-compartment


Allow group RackWare-SWIFT to manage buckets in compartment cluster-compartment
Allow group RackWare-SWIFT to manage objects in compartment cluster-compartment

Allow group RackWare-SWIFT to manage virtual-network-family in compartment cluster-compartment
```

Example: Allow group [group_name] to [verb] [resource-type] in compartment [compartment_name] where [condition]

In the policy editor textbox for rules, you will enter below rules. Note that some rules are optional depending on use-cases needed with the SWIFT. Replace '{group name}' and '{compartment name}' in below rules with user group created earlier and compartment name where OKE clusters are located respectively.

Instance access control rules - Mandatory

```
Allow group {group name} to manage instance-family in compartment {compartment name}

Allow group {group name} to read app-catalog-listing in compartment {compartment name}

Allow group {group name} to use volume-family in compartment {compartment name}

Allow group {group name} to use virtual-network-family in compartment {compartment name}
```

Storage access control for snapshots rules - Mandatory

```
Allow group {group name} to manage volume-family in compartment {compartment name}

Allow group {group name} to use instance-family in compartment {compartment name}
```

Sync to/from OKE cluster rules – Mandatory

```
Allow group {group name} to read all-artifacts in compartment {compartment name}
Allow group {group name} to manage cluster in compartment {compartment name}

Allow group {group name} to manage instance-family in compartment {compartment name}

Allow group {group name} to manage volume-family in compartment {compartment name}

Allow group {group name} to use virtual-network-family in compartment {compartment name}

Allow group {group name} to manage objects in compartment {compartment name}

Allow group {group name} to inspect instance-family in tenancy
```

Backup to Object storage control rules – Only needed if you are planning to backup to OCI Object Storage with SWIFT

```
Allow group {group name} to manage volume-family in compartment {compartment name}

Allow group {group name} to manage buckets in compartment {compartment name}

Allow group {group name} to manage objects in compartment {compartment name}

Allow group {group name} to manage virtual-network-family in compartment {compartment name}
```

OKE Dynamic cluster provisioning support rules – Only needed if you are planning to dynamically provision DR OKE clusters with SWIFT

```
Allow group {group name} to manage compartments in tenancy

Allow group {group name} to manage vcns in compartment {compartment name}

Allow group {group name} to manage subnets in compartment {compartment name}

Allow group {group name} to use vnics in compartment {compartment name}

Allow group {group name} to use private-ips in compartment {compartment name}

Allow group {group name} to manage public-ips in compartment {compartment name}

Allow group {group name} to use cluster-node-pools in compartment {compartment name}

Allow group {group name} to inspect instance-family in tenancy

Allow group {group name} to manage cluster-family in compartment {compartment name}
```

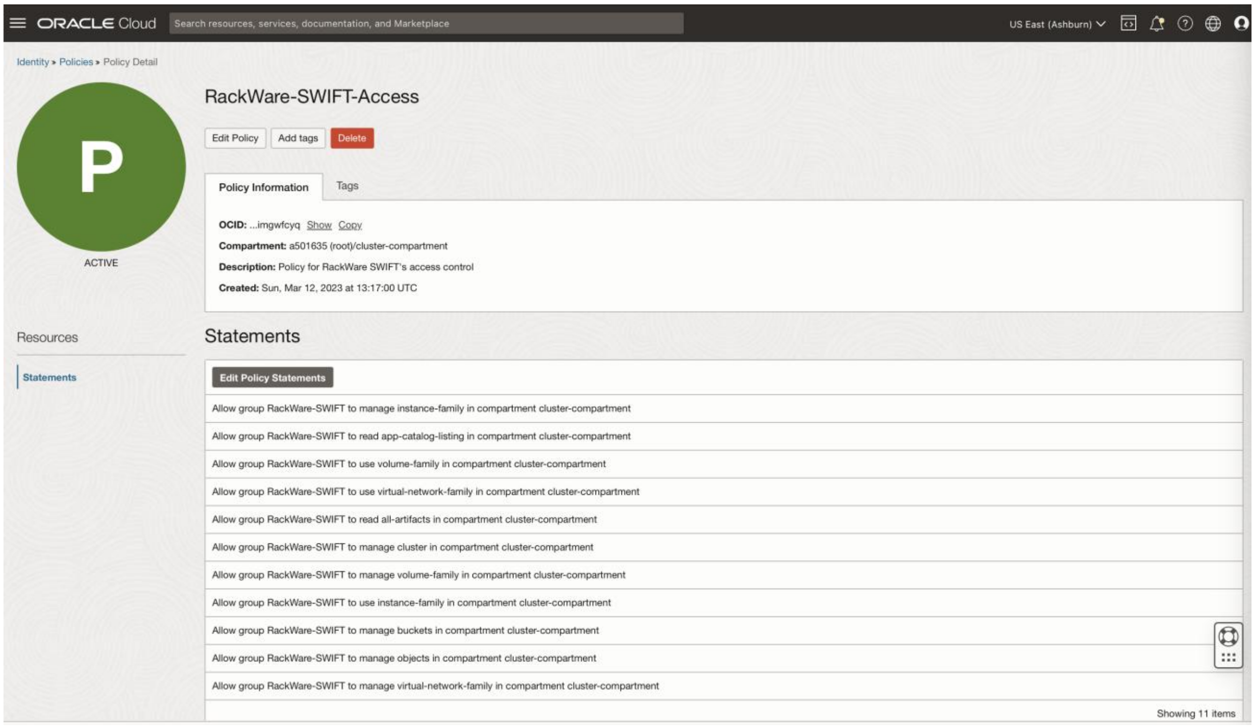
Oracle Container Registry (OCIR) sync rules – Only needed if you are planning to sync to/from Oracle OCI Container Registries (OCIR) with SWIFT

```
Allow group {group name} to manage volume-family in compartment {compartment name}
Allow group {group name} to manage buckets in compartment {compartment name}

Allow group {group name} to manage objects in compartment {compartment name}

Allow group {group name} to manage virtual-network-family in compartment {compartment name}
```

Once you enter required rules above, create the policy.



RackWare-SWIFT-Access

OCID: ...imgwfcyq Show Copy

Compartment: a501635 (root)/cluster-compartment

Description: Policy for RackWare SWIFT's access control

Created: Sun, Mar 12, 2023 at 13:17:00 UTC

Statements

Allow group RackWare-SWIFT to manage instance-family in compartment cluster-compartment

Allow group RackWare-SWIFT to read app-catalog-listing in compartment cluster-compartment

Allow group RackWare-SWIFT to use volume-family in compartment cluster-compartment

Allow group RackWare-SWIFT to use virtual-network-family in compartment cluster-compartment

Allow group RackWare-SWIFT to read all-artifacts in compartment cluster-compartment

Allow group RackWare-SWIFT to manage cluster in compartment cluster-compartment

Allow group RackWare-SWIFT to manage volume-family in compartment cluster-compartment

Allow group RackWare-SWIFT to use instance-family in compartment cluster-compartment

Allow group RackWare-SWIFT to manage buckets in compartment cluster-compartment

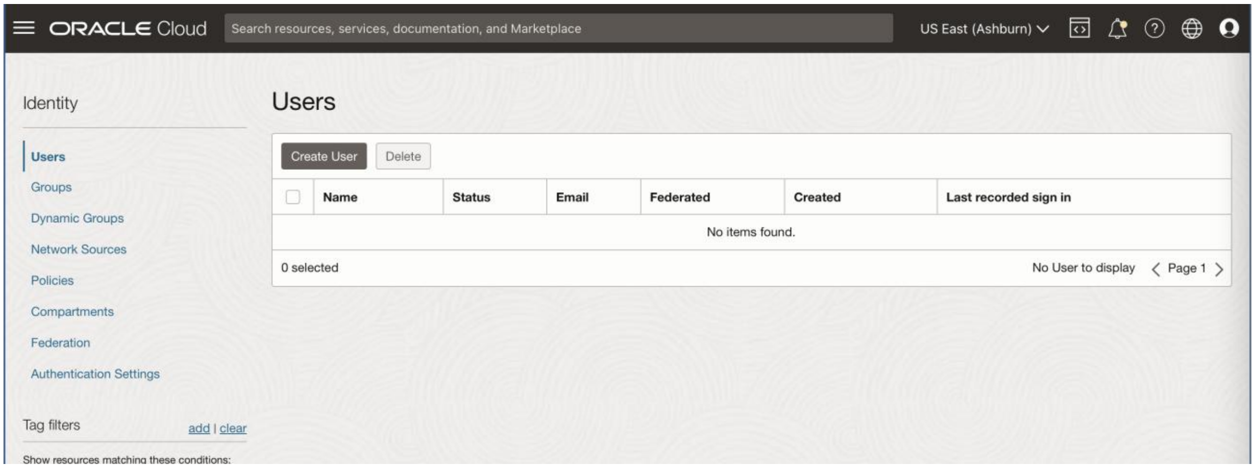
Allow group RackWare-SWIFT to manage objects in compartment cluster-compartment

Allow group RackWare-SWIFT to manage virtual-network-family in compartment cluster-compartment

Showing 11 items

Let's now create a User and add it to the Group created earlier, where we also applied access policy now.

From the Identity menu, select the Users submenu.

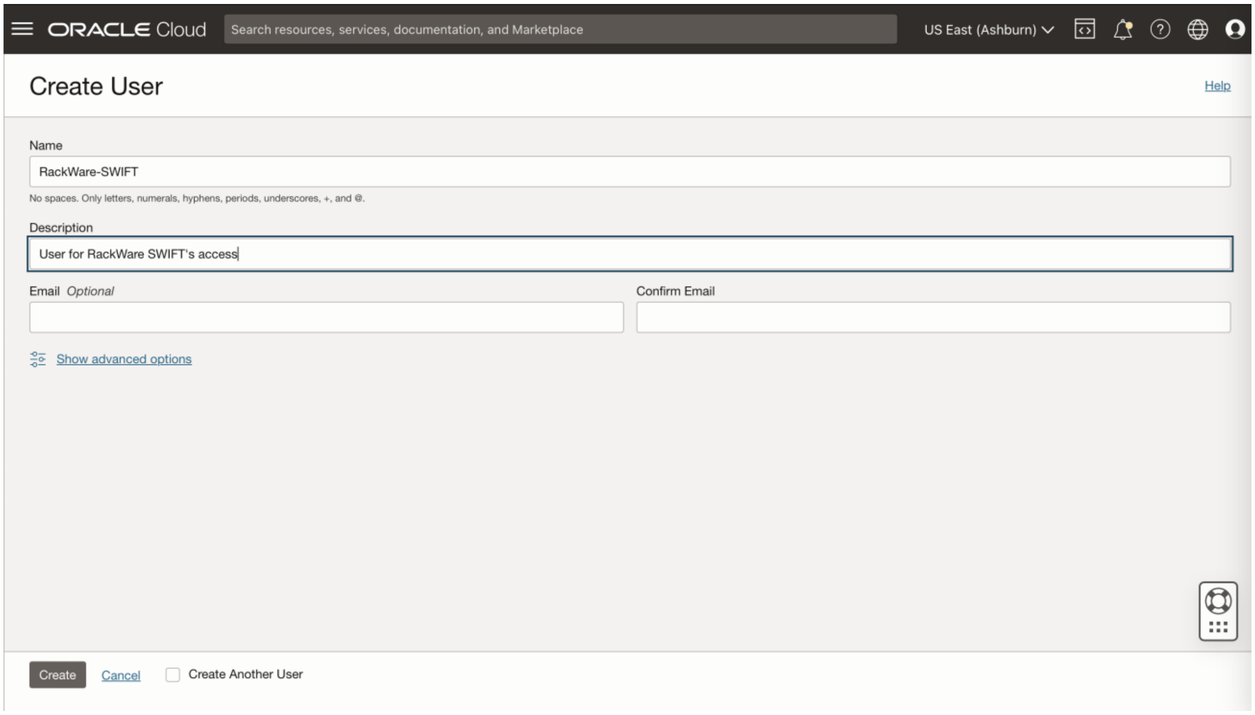


Users

□	Name	Status	Email	Federated	Created	Last recorded sign in
No items found.						

0 selected — No User to display — Page 1

Select the ‘Create User’ option. Then on new user creation wizard, set name and description for the User. We will use ‘RackWare-SWIFT’ as a name in the example case.



Create User

Name: RackWare-SWIFT

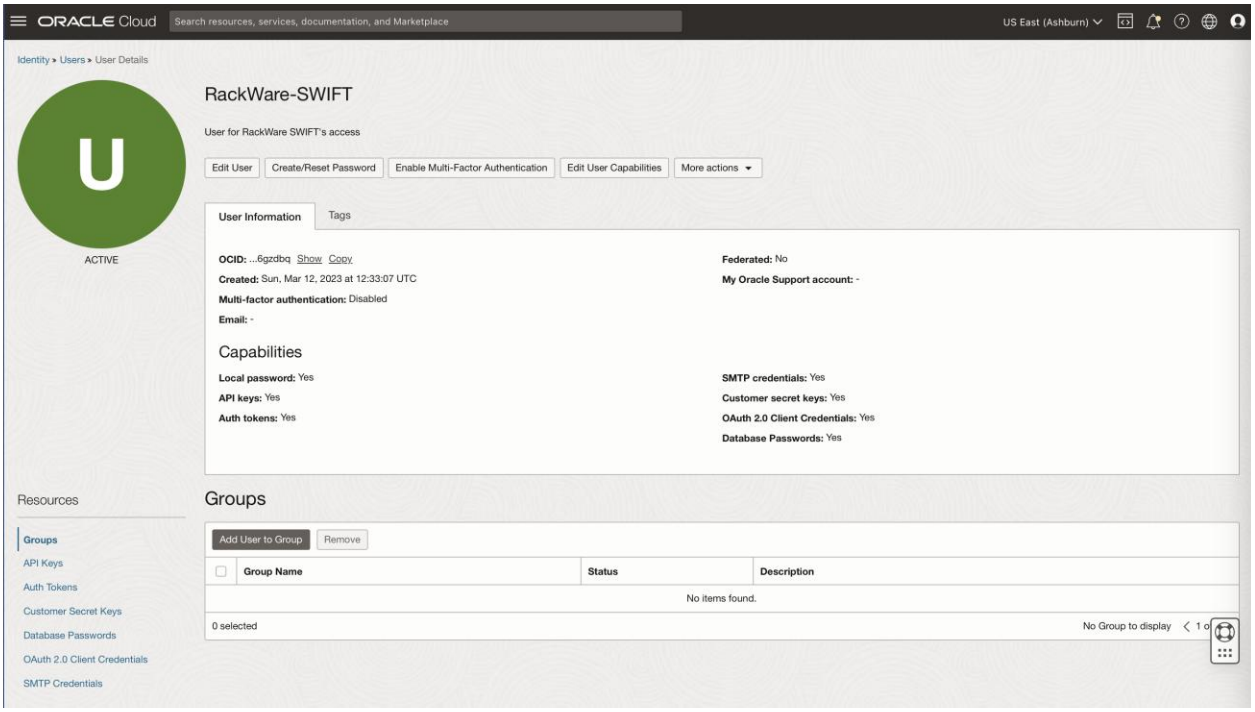
No spaces. Only letters, numerals, hyphens, periods, underscores, +, and @.

Description: User for RackWare SWIFT's access

Email *Optional*

Confirm Email

Once the user is created, select ‘Add User to Group’ option from the Groups tab.



ORACLE Cloud
Search resources, services, documentation, and Marketplace
US East (Ashburn)
Identity » Users » User Details
U
ACTIVE

RackWare-SWIFT

User for RackWare SWIFT's access

OCID: ...6gzdbq Show Copy

Created: Sun, Mar 12, 2023 at 12:33:07 UTC

Multi-factor authentication: Disabled

Email: -

Federated: No

My Oracle Support account: -

Capabilities

Local password: Yes

API keys: Yes

Auth tokens: Yes

SMTP credentials: Yes

Customer secret keys: Yes

OAuth 2.0 Client Credentials: Yes

Database Passwords: Yes

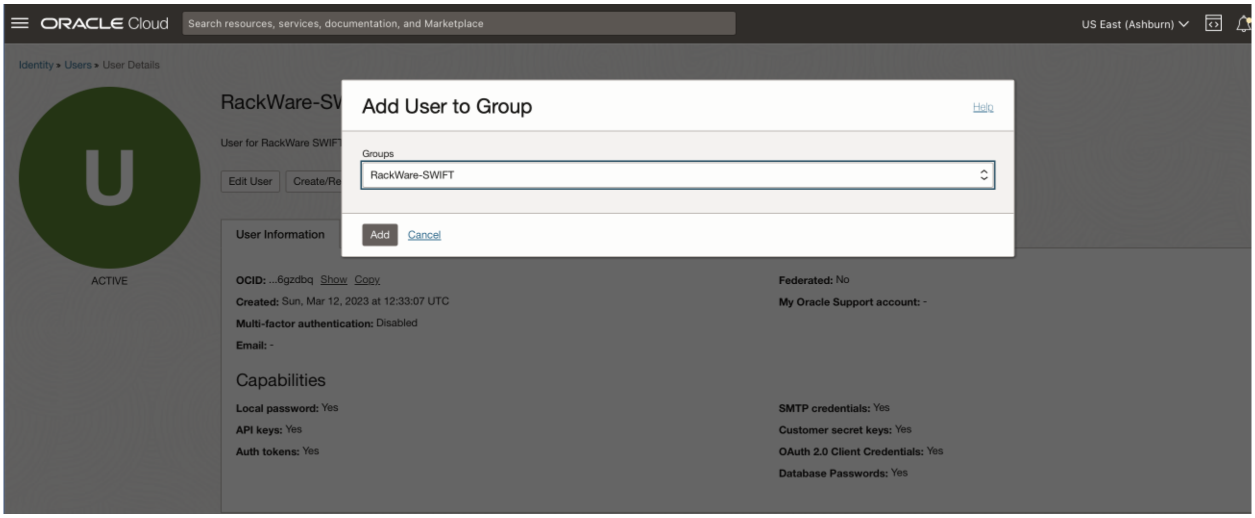
Groups

□	Group Name	Status	Description
No items found.			

0 selected

No Group to display < 1 o

Add it to the 'RackWare-SWIFT' group created earlier in the flow, where new policy is also applied. Adding user to this group would restrict this user's access to OCI with the earlier applied restrictive policy.



Add User to Group

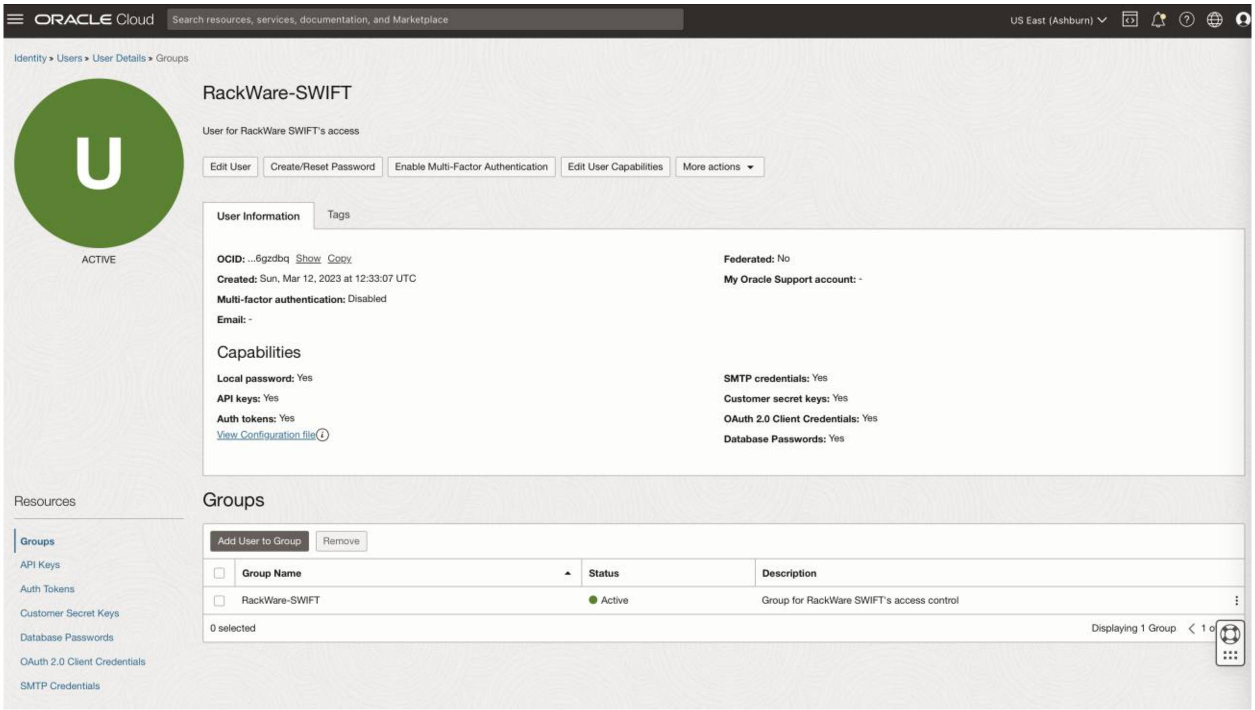
Groups: RackWare-SWIFT

OCID: ...6gzdbq Show Copy
Created: Sun, Mar 12, 2023 at 12:33:07 UTC
Multi-factor authentication: Disabled
Email: -
Federated: No
My Oracle Support account: -

Capabilities

Local password: Yes
API keys: Yes
Auth tokens: Yes
SMTP credentials: Yes
Customer secret keys: Yes
OAuth 2.0 Client Credentials: Yes
Database Passwords: Yes

Once the user is added to the group, you will see the new group listed in the Groups tab.



RackWare-SWIFT

User for RackWare SWIFT's access

OCID: ...6gzdbq Show Copy
Created: Sun, Mar 12, 2023 at 12:33:07 UTC
Multi-factor authentication: Disabled
Email: -
Federated: No
My Oracle Support account: -

Capabilities

Local password: Yes
API keys: Yes
Auth tokens: Yes
View Configuration file
SMTP credentials: Yes
Customer secret keys: Yes
OAuth 2.0 Client Credentials: Yes
Database Passwords: Yes

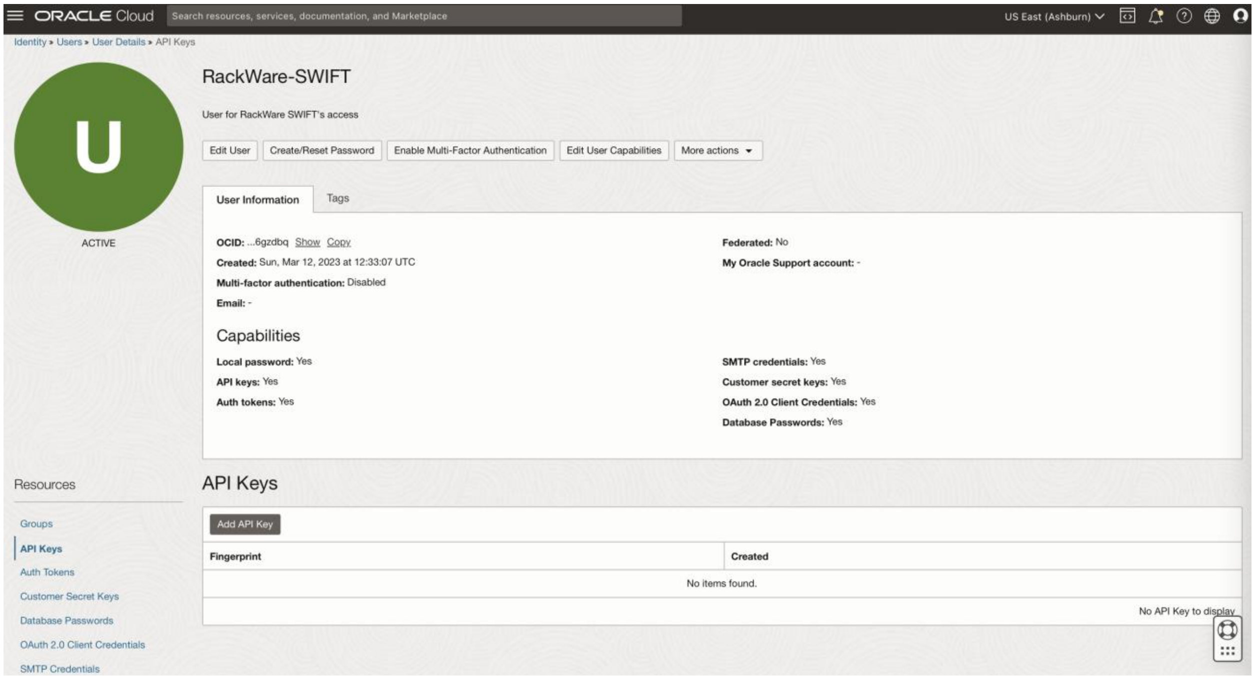
Groups

Group Name	Status	Description
RackWare-SWIFT	● Active	Group for RackWare SWIFT's access control

0 selected   Displaying 1 Group   1 o

Now, let's generate an API key for the user that later you will use with SWIFT for OKE cluster discovery, OCIR syncs, OCI object storage addition to SWIFT, or OKE dynamic provisioning configuration in SWIFT.

Click on the ‘API Keys’ tab under user configuration page.



RackWare-SWIFT

User for RackWare SWIFT's access

OCID: ...6gzdbq
Created: Sun, Mar 12, 2023 at 12:33:07 UTC
Multi-factor authentication: Disabled
Email: -
Federated: No
My Oracle Support account: -

Capabilities

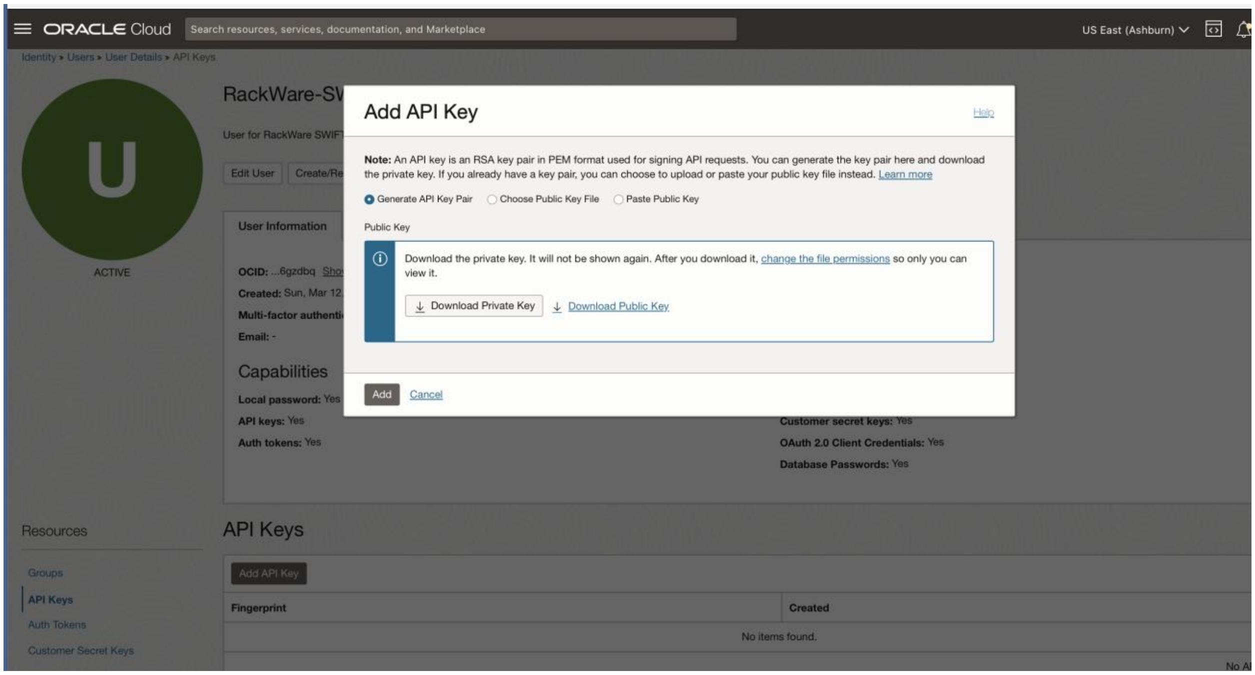
Local password: Yes
API keys: Yes
Auth tokens: Yes
SMTP credentials: Yes
Customer secret keys: Yes
OAuth 2.0 Client Credentials: Yes
Database Passwords: Yes

API Keys

Fingerprint	Created
No items found.	

No API Key to display

Select ‘Add API Key’ button. On the new key wizard, you can either upload keypair or let OCI generate one for you. Make sure to download both keys if you let OCI generate keypair for you, as you can’t later download these keys later.



Add API Key

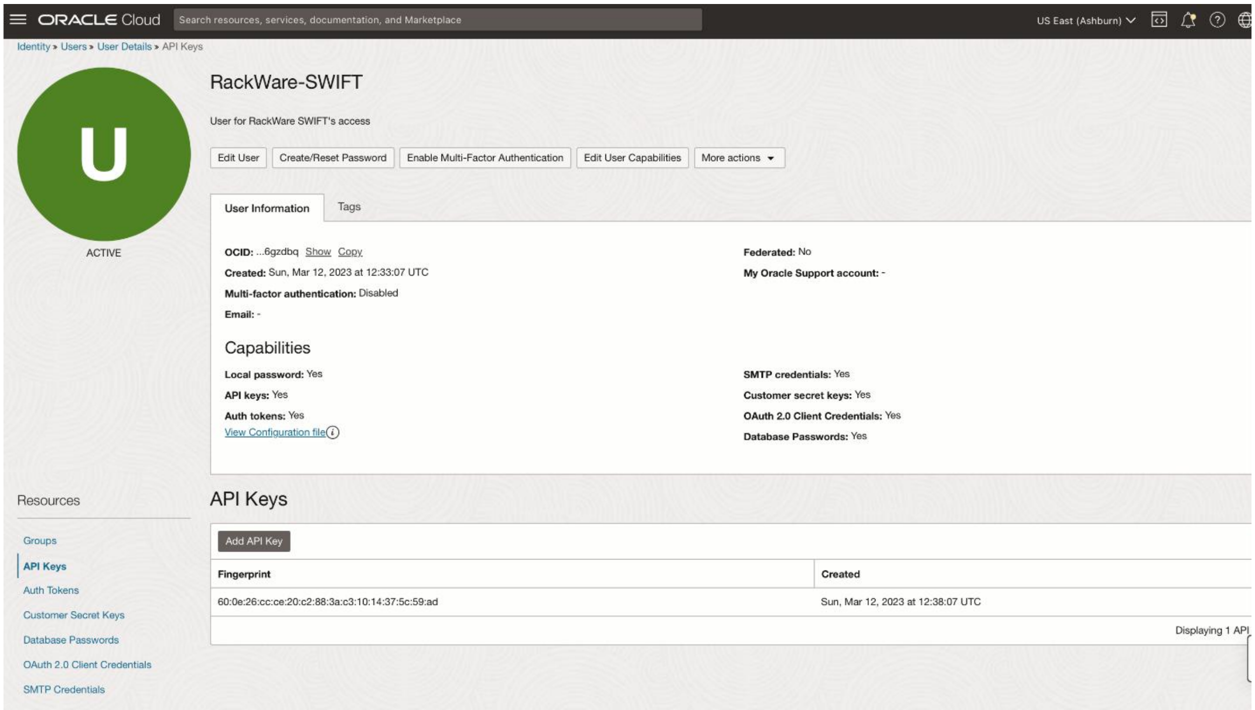
Note: An API key is an RSA key pair in PEM format used for signing API requests. You can generate the key pair here and download the private key. If you already have a key pair, you can choose to upload or paste your public key file instead. Learn more

● Generate API Key Pair ○ Choose Public Key File ○ Paste Public Key

Public Key

Download the private key. It will not be shown again. After you download it, change the file permissions so only you can view it.

Note the generate key’s fingerprint too, as you will later need it for various OKE/OCIR specific configurations in SWIFT, including OKE/OCIR cluster discovery.



RackWare-SWIFT

User for RackWare SWIFT's access

ACTIVE

OCID: ...6gzdbq Show Copy
Created: Sun, Mar 12, 2023 at 12:33:07 UTC
Multi-factor authentication: Disabled
Email: -

Federated: No
My Oracle Support account: -

Capabilities

Local password: Yes
API keys: Yes
Auth tokens: Yes
View Configuration file

SMTP credentials: Yes
Customer secret keys: Yes
OAuth 2.0 Client Credentials: Yes
Database Passwords: Yes

API Keys

Fingerprint	Created
60:0e:26:cc:ce:20:c2:88:3a:c3:10:14:37:5c:59:ad	Sun, Mar 12, 2023 at 12:38:07 UTC

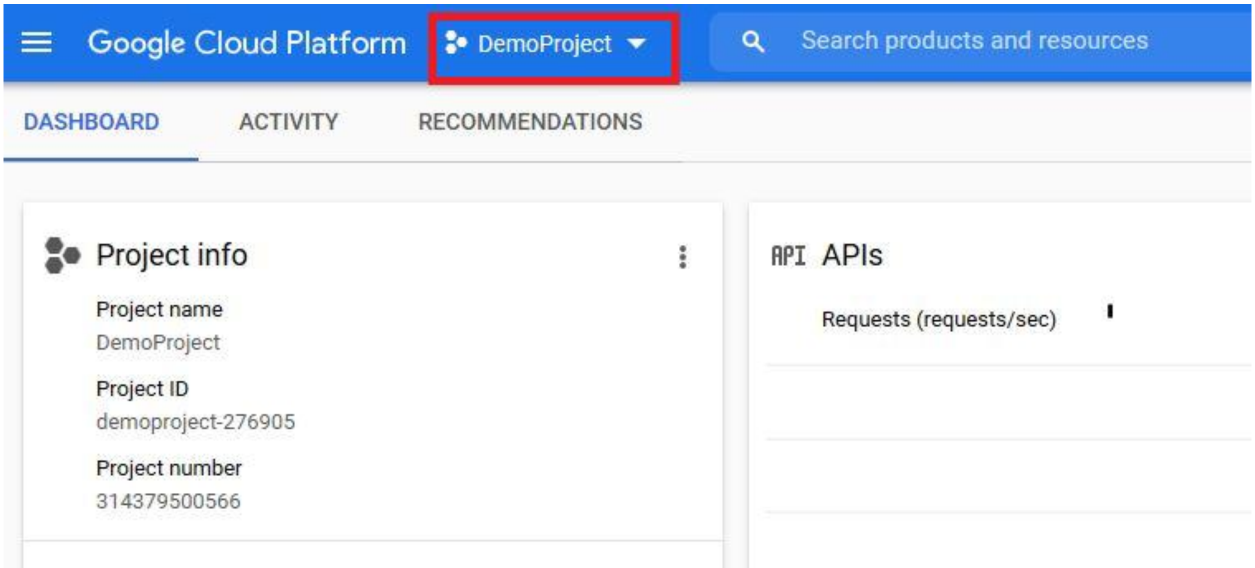
Displaying 1 API

That’s it! You can now use the generated API key and fingerprint along with other details like compartment id and user id with SWIFT to discover an OKE cluster or sync OCIR registries under the OCI account. The new API key will have same access that access policy we created and applied above for the new user allows.

Adding Google GCP service-account for SWIFT use

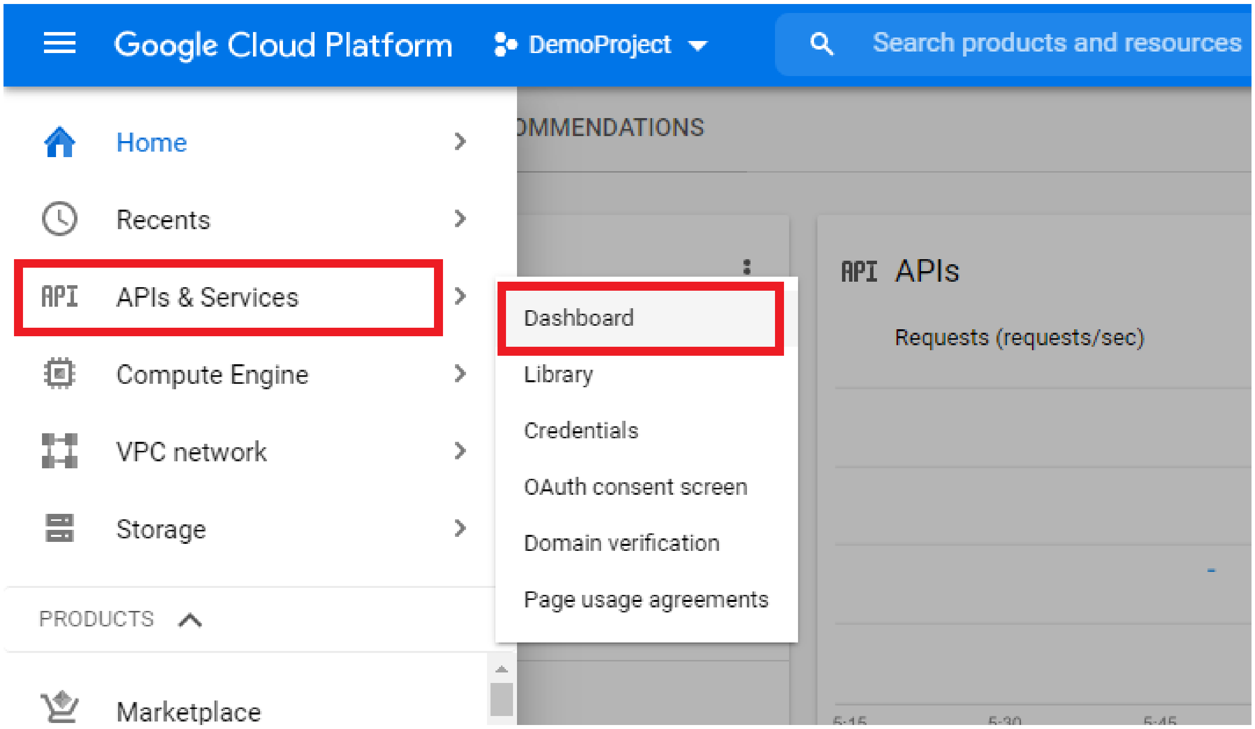
This section highlights the steps to create an account under your Google cloud project, which you can use later to configure the cluster details under your installed SWIFT. The same credentials can also be used later to discover a Google Container Registry (GCR) instance or add a GCP cloud object storage under your SWIFT.

Login to Google cloud console, and select the required project. Note that you will have to repeat these steps for every project, where you have a cluster located and which you want to manage under the SWIFT.

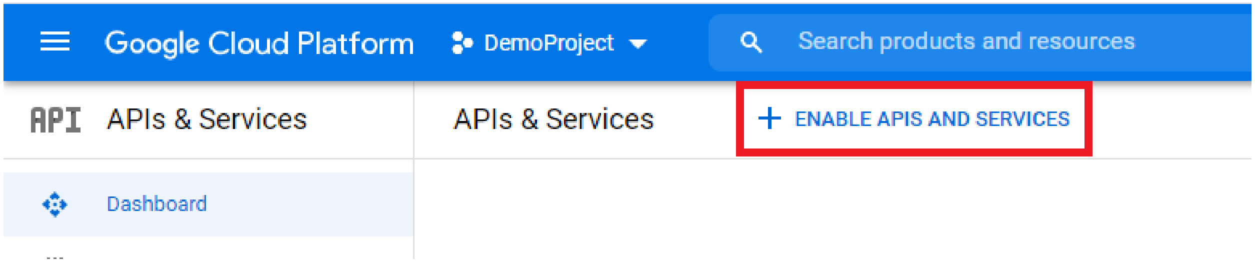


Project name
DemoProject
Project ID
demoproject-276905
Project number
314379500566

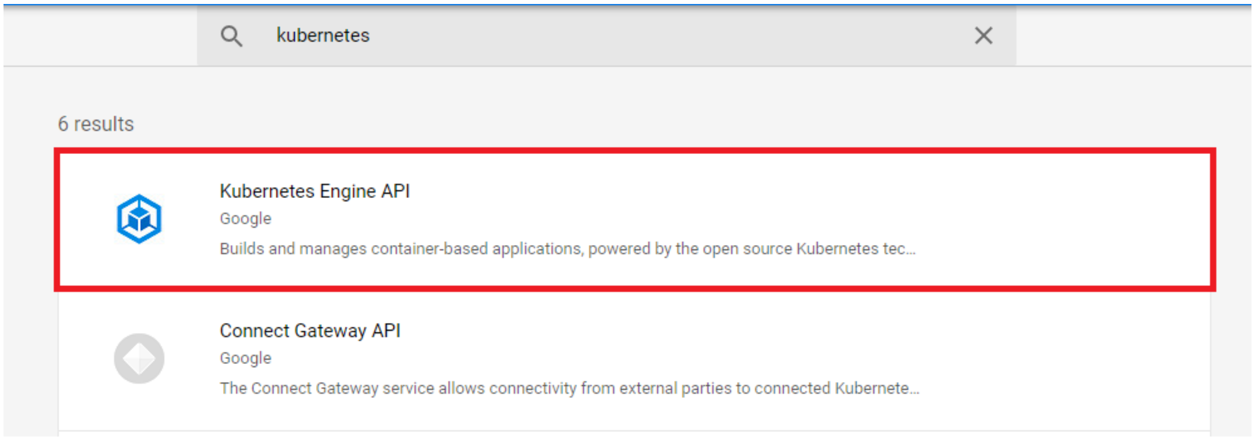
From your menu options, click on the ‘APIs & Services’ option and then on the ‘Dashboard’ submenu.



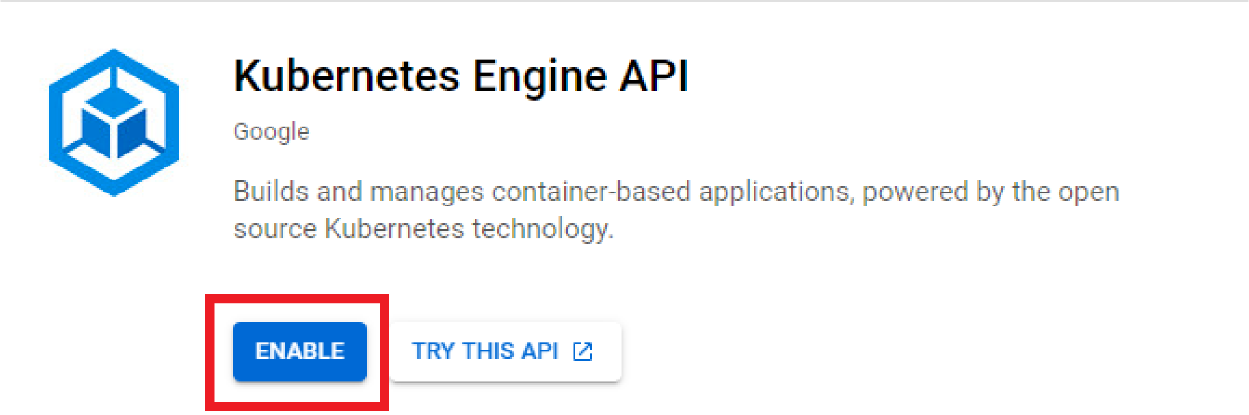
Click on the ‘Enable APIs and Services’ option.



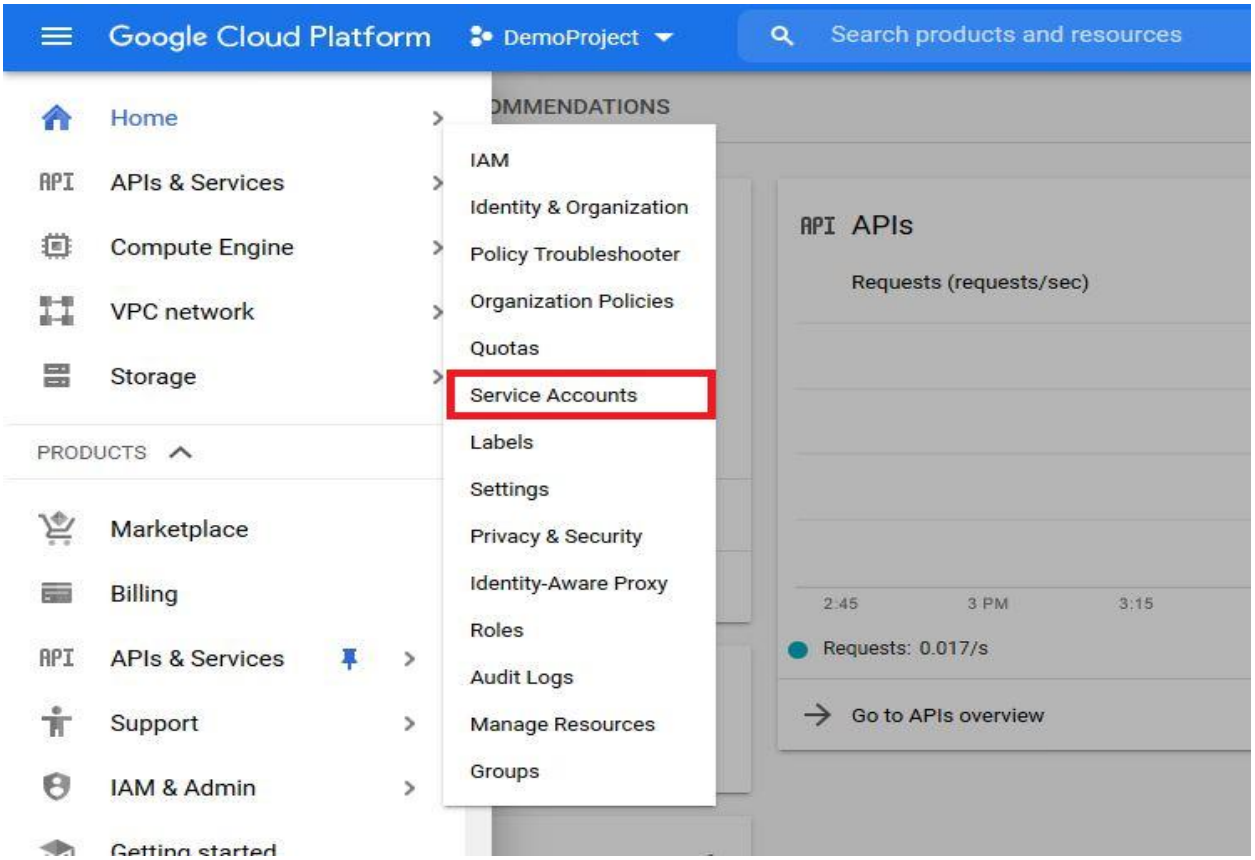
Search For 'Kubernetes Engine API' option.


kubernetes
6 results
Kubernetes Engine API
Google
Builds and manages container-based applications, powered by the open source Kubernetes tec...
Connect Gateway API
Google
The Connect Gateway service allows connectivity from external parties to connected Kubernete...

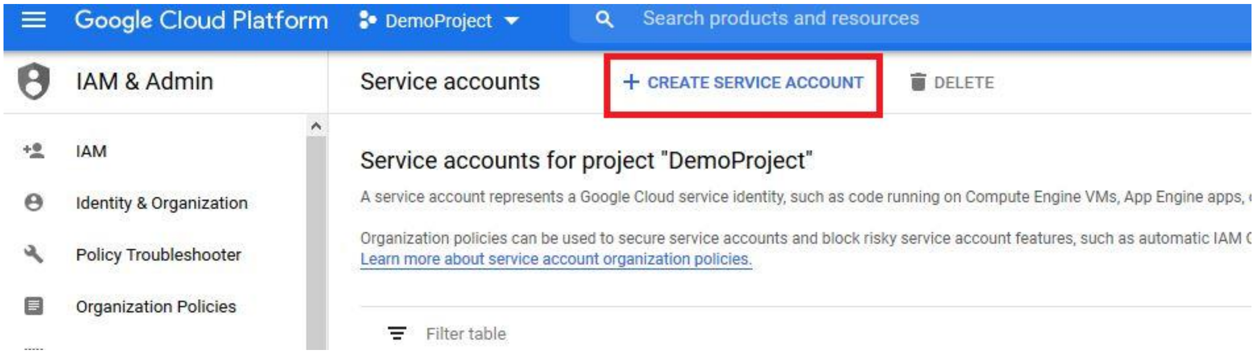
Click on the 'Enable' button.


Kubernetes Engine API
Google
Builds and manages container-based applications, powered by the open source Kubernetes technology.
ENABLE
TRY THIS API

From your menu options, click on the 'IAM & Admin' option and then on the 'Service Accounts' submenu.



Click on the ‘Create Service Account’ option.

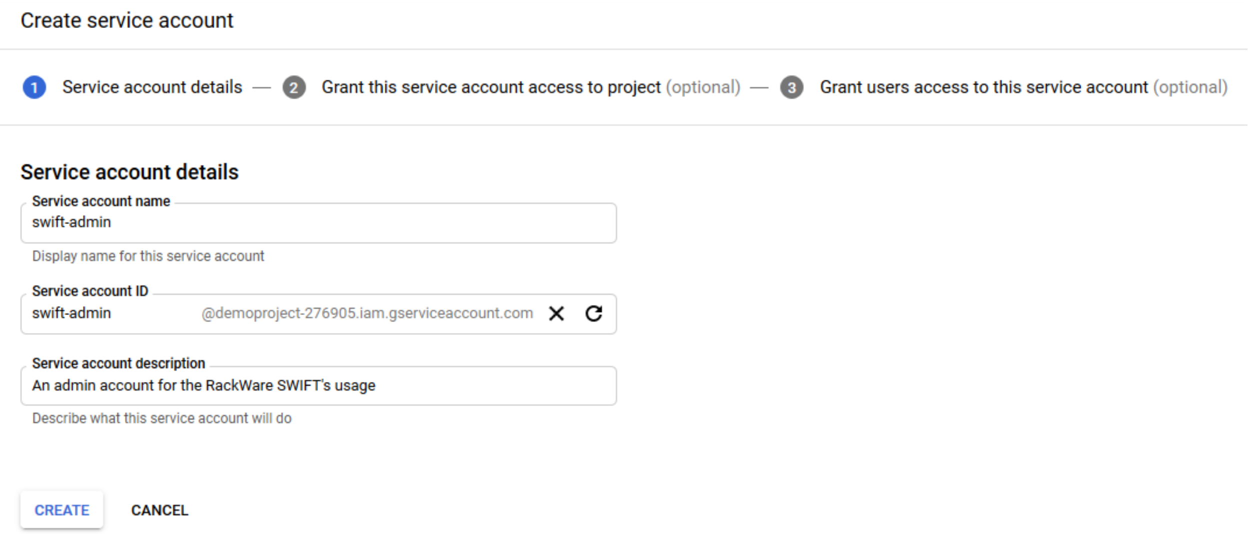


Service accounts for project "DemoProject"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps,

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM
Learn more about service account organization policies.

Fill out the necessary account name and description details.



Create service account

1 Service account details — 2 Grant this service account access to project (optional) — 3 Grant users access to this service account (optional)

Service account details

Service account name: swift-admin

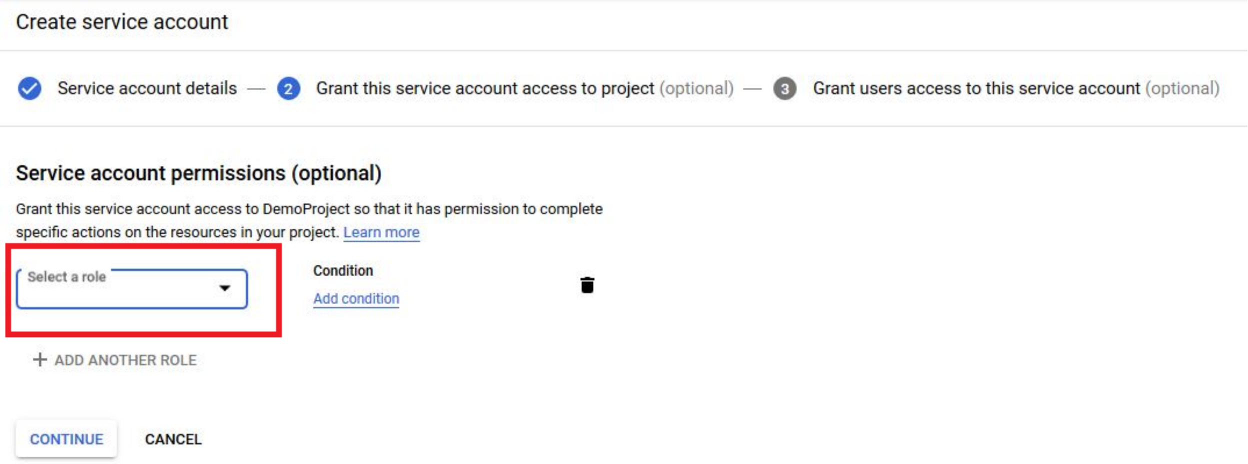
Display name for this service account

Service account ID: swift-admin @demoproject-276905.iam.gserviceaccount.com

Service account description: An admin account for the RackWare SWIFT's usage

Describe what this service account will do

After the account is created, select roles to add for the new account, as shown below. You need to add Kubernetes engine developer, Kubernetes Engine admin, Compute Instance Admin (beta) and Compute Instance Admin (v1).

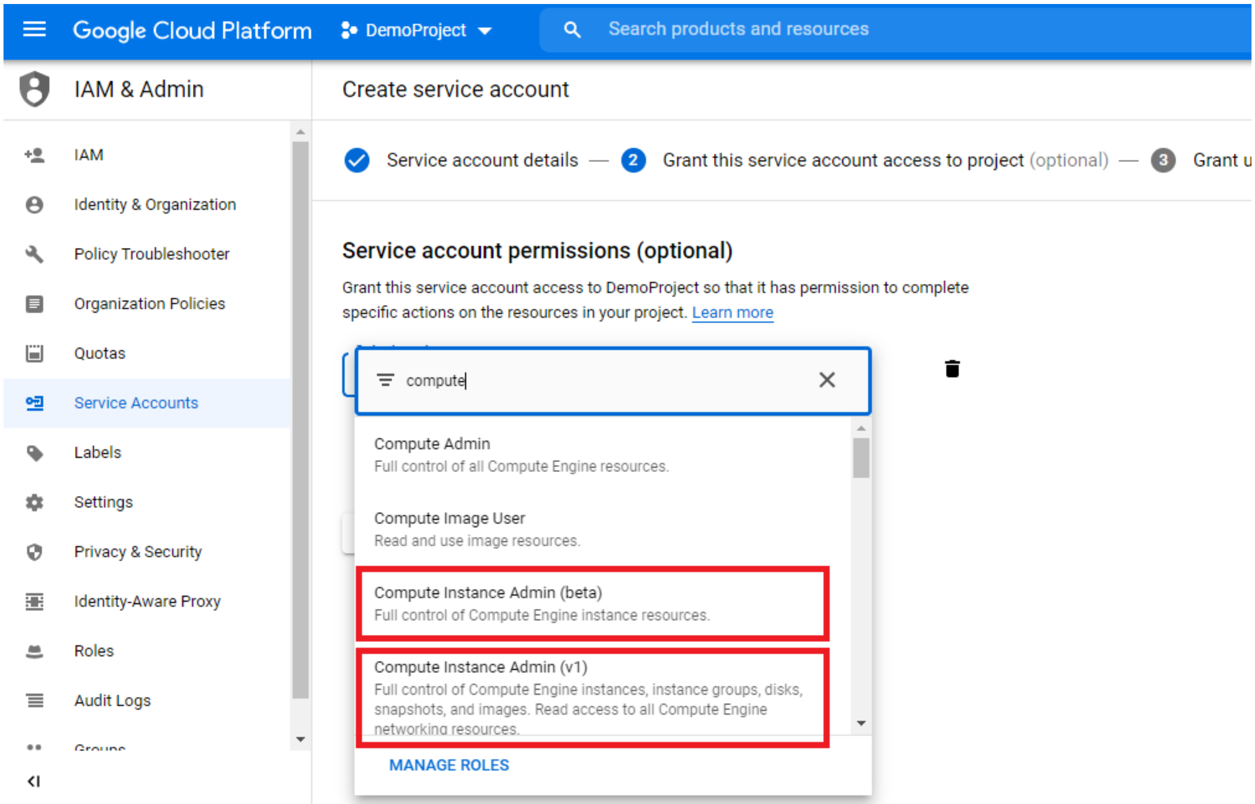


Create service account

✓ Service account details — 2 Grant this service account access to project (optional) — 3 Grant users access to this service account (optional)

Service account permissions (optional)

Grant this service account access to DemoProject so that it has permission to complete specific actions on the resources in your project. Learn more



Service account permissions (optional)

Grant this service account access to DemoProject so that it has permission to complete specific actions on the resources in your project. Learn more

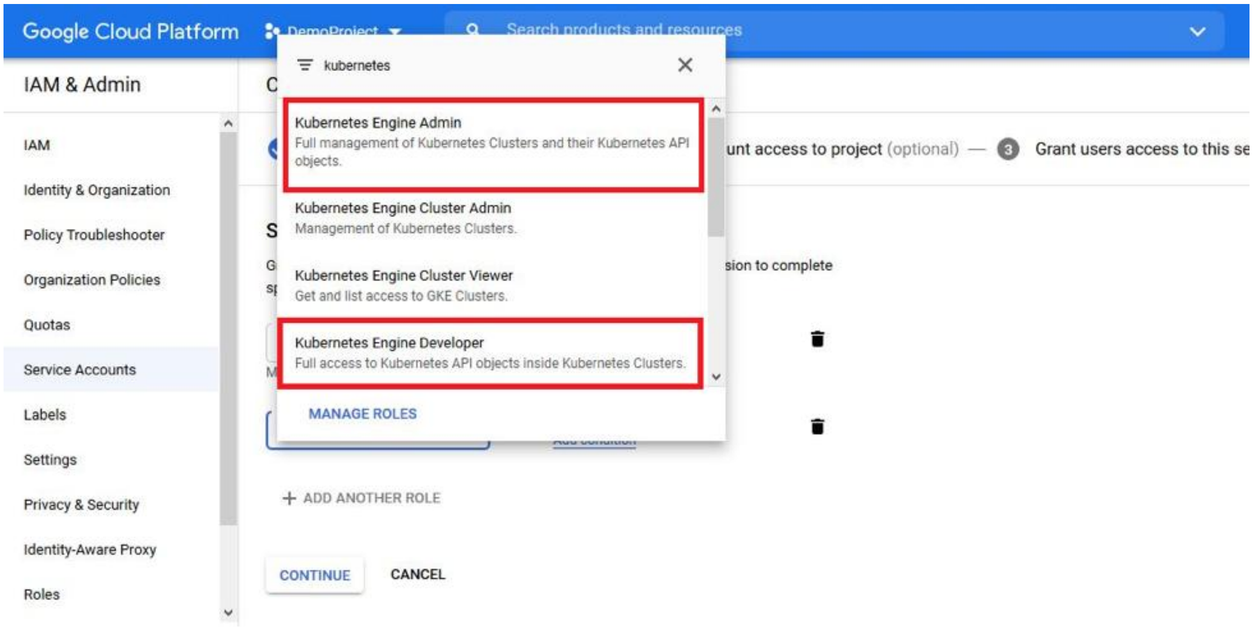
compute

Compute Admin
Full control of all Compute Engine resources.

Compute Image User
Read and use image resources.

Compute Instance Admin (beta)
Full control of Compute Engine instance resources.

Compute Instance Admin (v1)
Full control of Compute Engine instances, instance groups, disks, snapshots, and images. Read access to all Compute Engine networking resources.



kubernetes

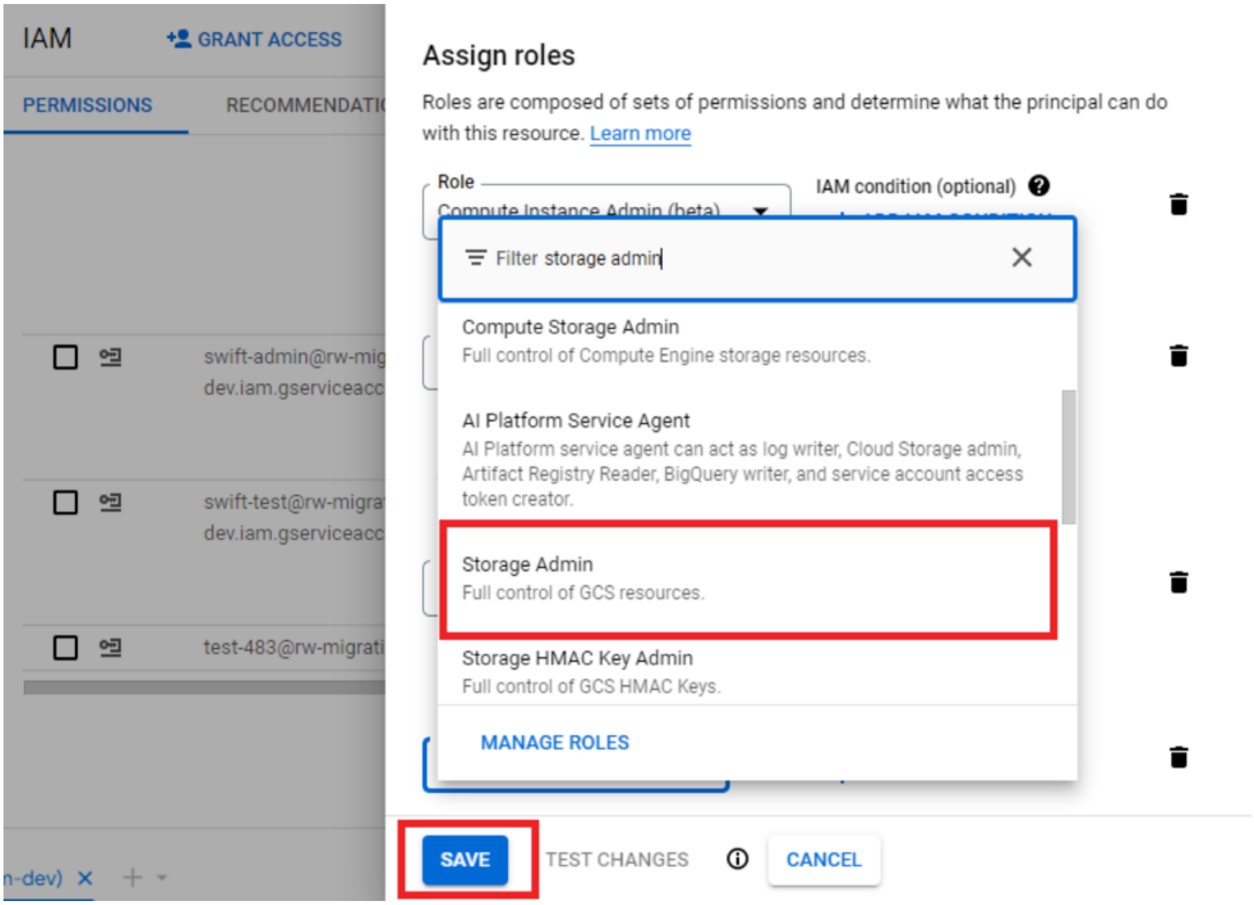
Kubernetes Engine Admin
Full management of Kubernetes Clusters and their Kubernetes API objects.

Kubernetes Engine Cluster Admin
Management of Kubernetes Clusters.

Kubernetes Engine Cluster Viewer
Get and list access to GKE Clusters.

Kubernetes Engine Developer
Full access to Kubernetes API objects inside Kubernetes Clusters.

Also add the ‘Storage Admin’ role to the service account if you plan to use this account with the SWIFT for adding a GCP object storage in the SWIFT. If you do not plan to use GCP object storge with SWIFT for backups, then this role addition is an optional step.



Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. Learn more

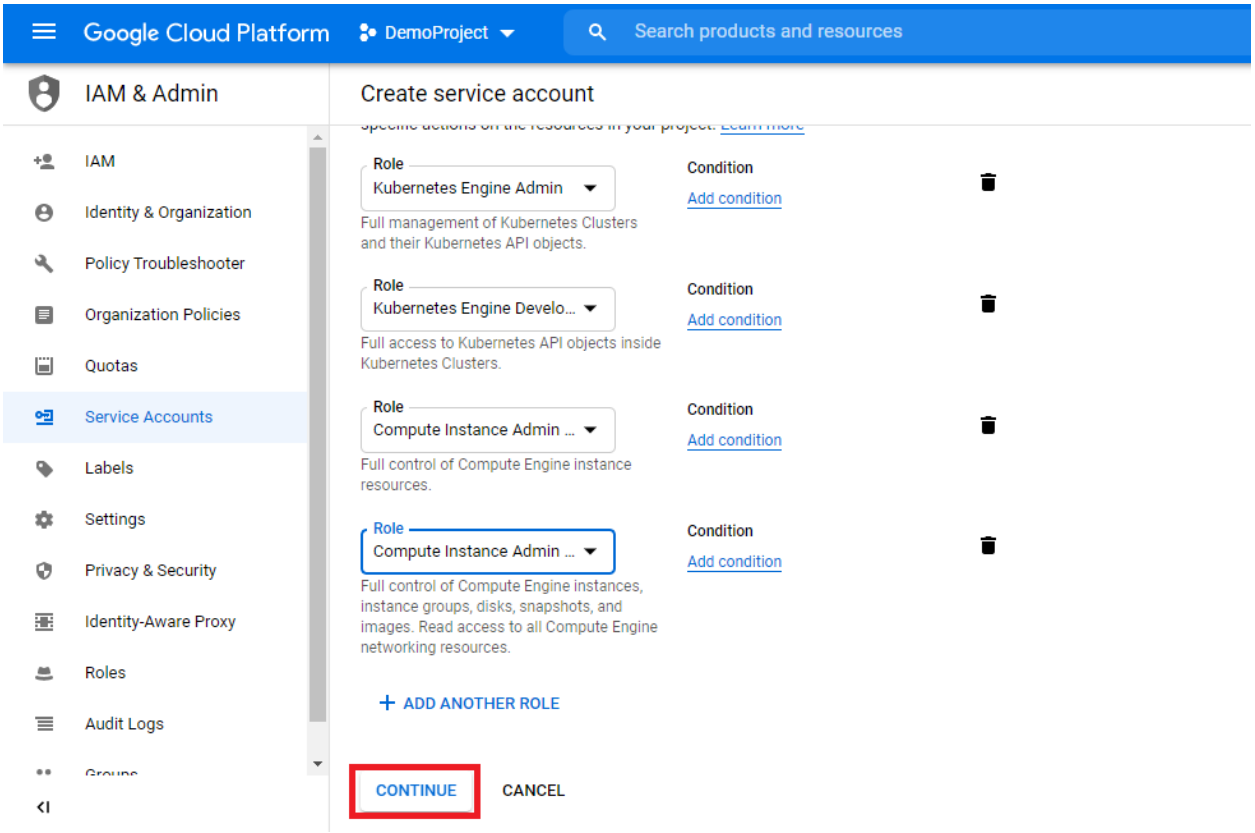
Role: Compute Instance Admin (beta)

IAM condition (optional)

Filter storage admin

- Compute Storage Admin — Full control of Compute Engine storage resources.
- AI Platform Service Agent — AI Platform service agent can act as log writer, Cloud Storage admin, Artifact Registry Reader, BigQuery writer, and service account access token creator.
- Storage Admin — Full control of GCS resources.
- Storage HMAC Key Admin — Full control of GCS HMAC Keys.

Once the required roles are added, press the ‘Continue’ button.



Create service account

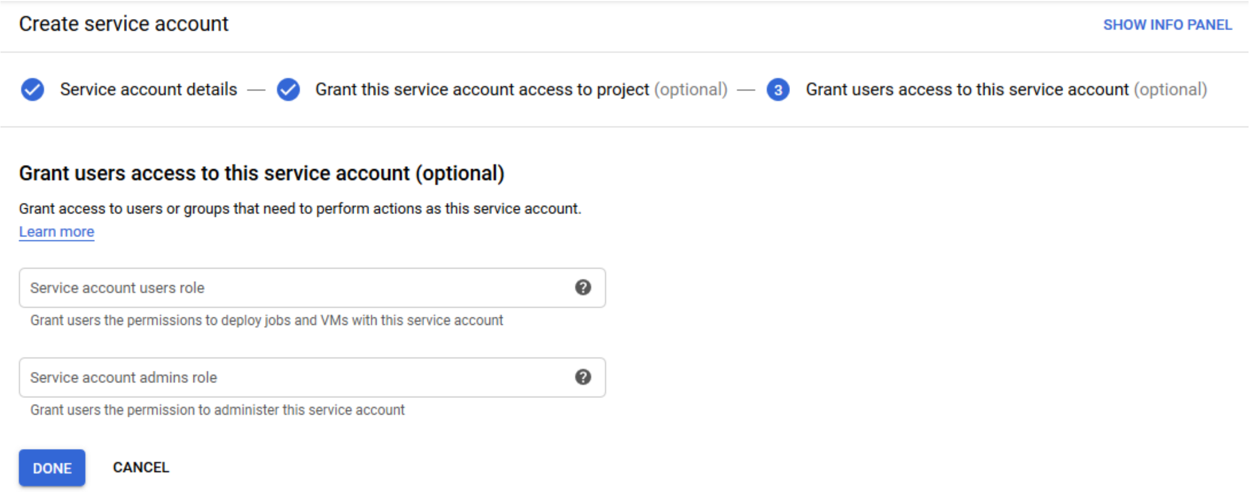
Role: Kubernetes Engine Admin — Full management of Kubernetes Clusters and their Kubernetes API objects. — Condition: Add condition

Role: Kubernetes Engine Develo... — Full access to Kubernetes API objects inside Kubernetes Clusters. — Condition: Add condition

Role: Compute Instance Admin ... — Full control of Compute Engine instance resources. — Condition: Add condition

Role: Compute Instance Admin ... — Full control of Compute Engine instances, instance groups, disks, snapshots, and images. Read access to all Compute Engine networking resources. — Condition: Add condition

Add any user roles for the account. Adding any user roles is optional for this account. Press the 'Done' button to complete the creation of the user.



Create service account

SHOW INFO PANEL
Service account details — Grant this service account access to project (optional) — 3 Grant users access to this service account (optional)

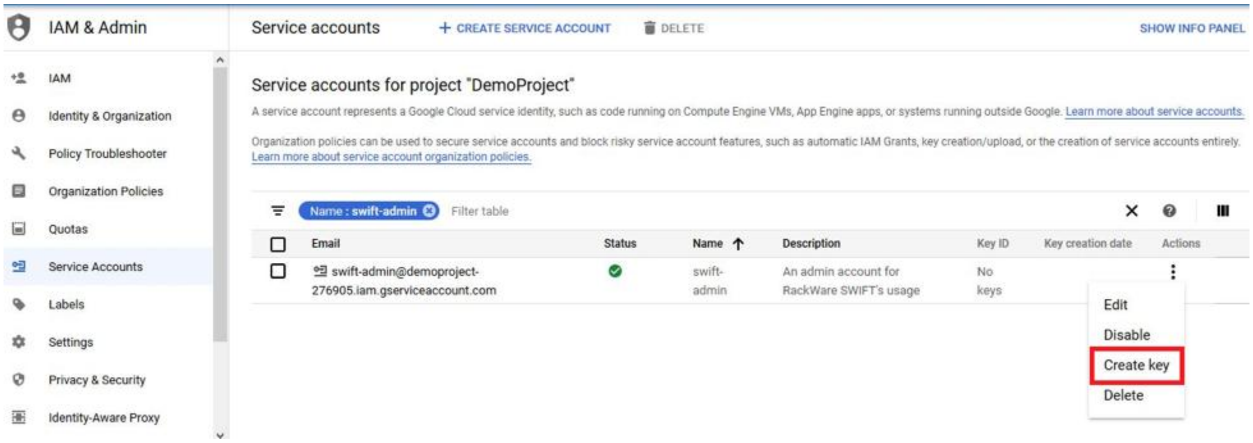
Grant users access to this service account (optional)

Grant access to users or groups that need to perform actions as this service account.
Learn more

Service account users role
Grant users the permissions to deploy jobs and VMs with this service account

Service account admins role
Grant users the permission to administer this service account

Now the new account will appear on the service accounts page. Filter, if necessary, to find the newly created user and generate a key for it from the account menu.



Service accounts for project "DemoProject"

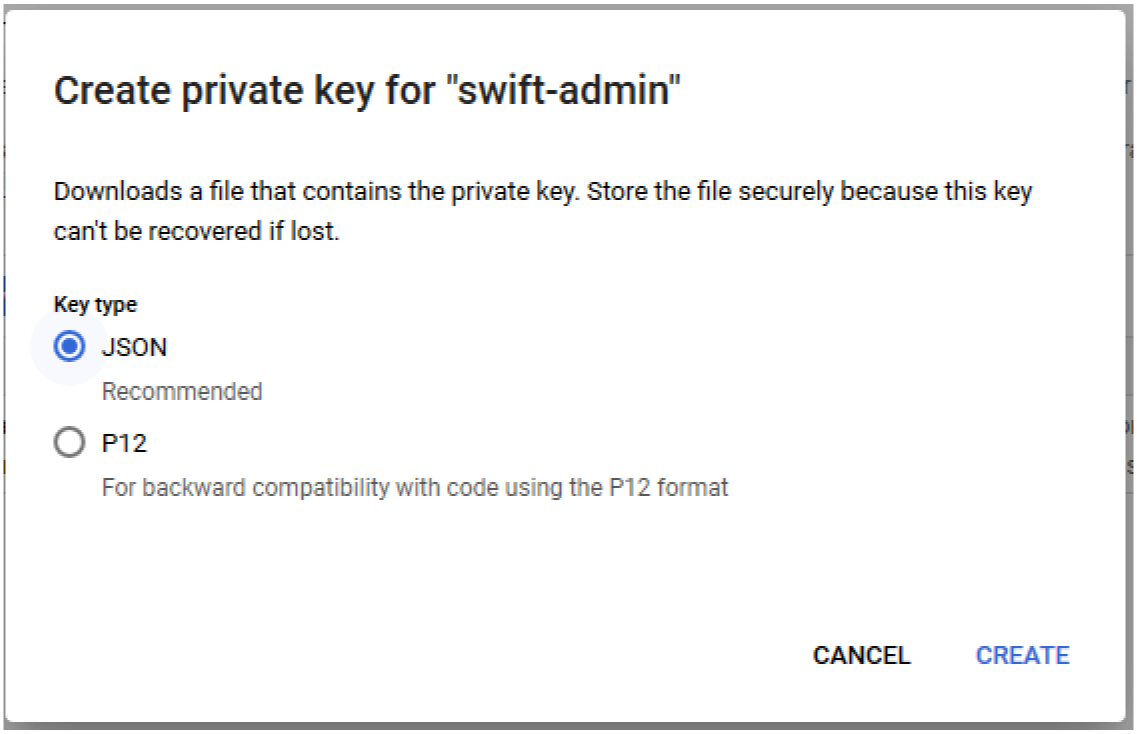
A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. Learn more about service accounts.

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. Learn more about service account organization policies.

Name : swift-admin Filter table

□	Email	Status	Name ↑	Description	Key ID	Key creation date	Actions
□	swift-admin@demoproject-276905.iam.gserviceaccount.com	✓	swift-admin	An admin account for RackWare SWIFT's usage	No keys		⋮

Select the JSON format for the key, and then download the JSON key file.



Create private key for "swift-admin"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

● JSON

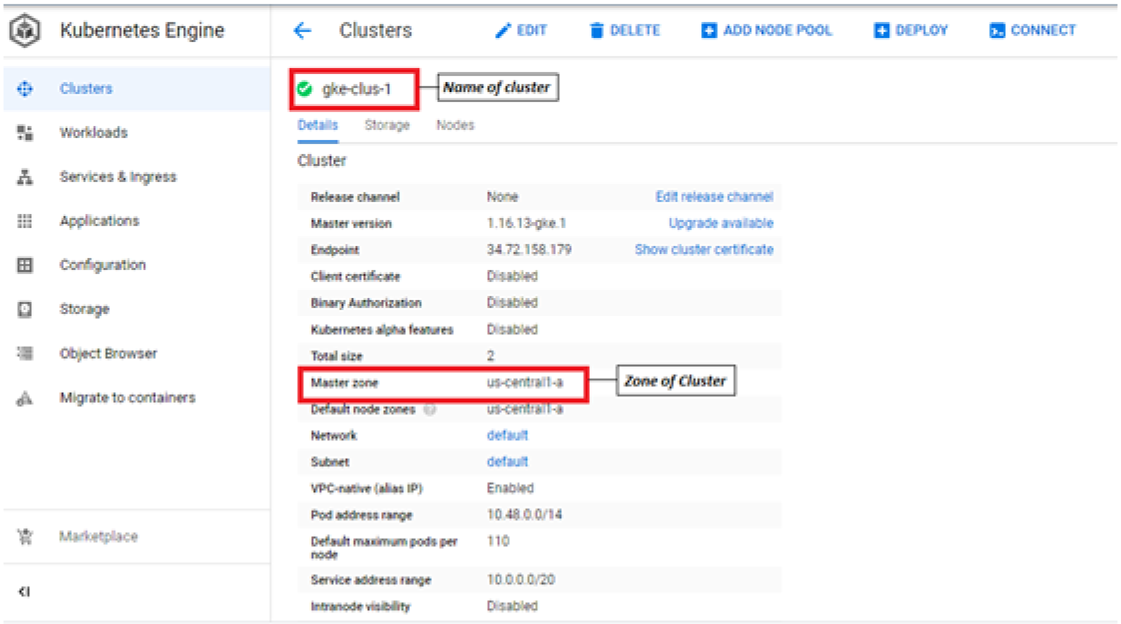
Recommended

○ P12

For backward compatibility with code using the P12 format

Store the downloaded JSON file securely, as it will contain API private key and a few other sensitive details. Later, when you are ready to add your cluster to the SWIFT, you will need to upload this downloaded JSON file.

Additionally, you will also need to specify your cloud cluster details to the SWIFT while configuring the GKE cluster under SWIFT. You will need a few necessary details, like the cluster name, the project under which this cluster is created, etc. You can find those on the cluster details page, as shown below.



gke-clus-1

Name of cluster

Cluster

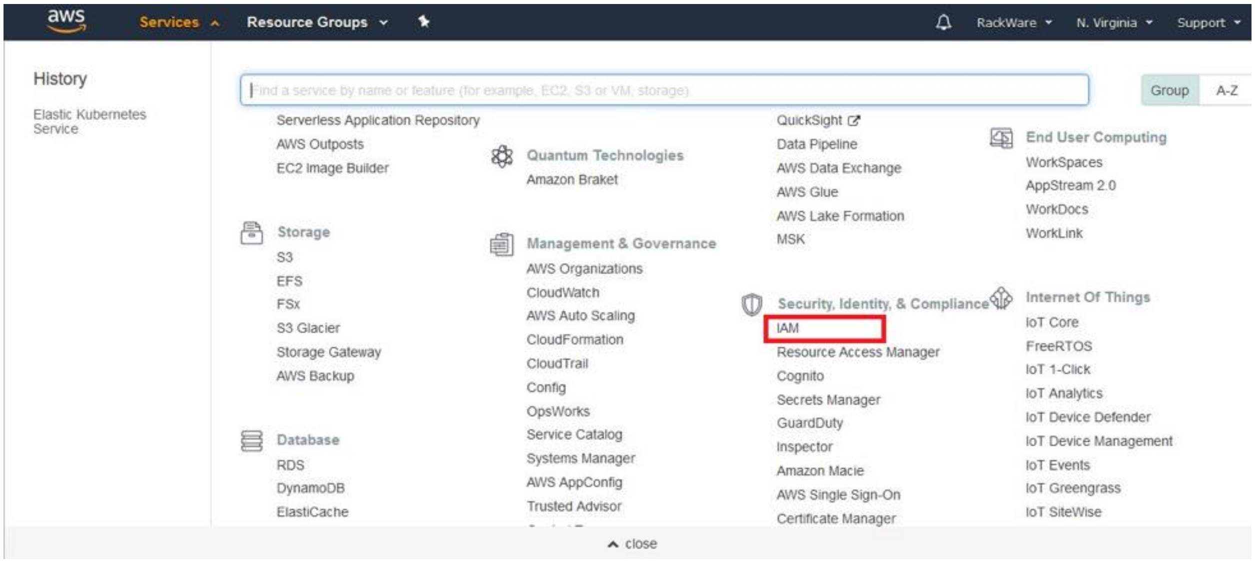
Release channel	None	Edit release channel
Master version	1.16.13-gke.1	Upgrade available
Endpoint	34.72.158.179	Show cluster certificate
Client certificate	Disabled	
Binary Authorization	Disabled	
Kubernetes alpha features	Disabled	
Total size	2	
Master zone	us-central1-a	
Default node zones	us-central1-a	
Network	default	
Subnet	default	
VPC-native (alias IP)	Enabled	
Pod address range	10.48.0.0/14	
Default maximum pods per node	110	
Service address range	10.0.0.0/20	
Intranode visibility	Disabled	

Zone of Cluster

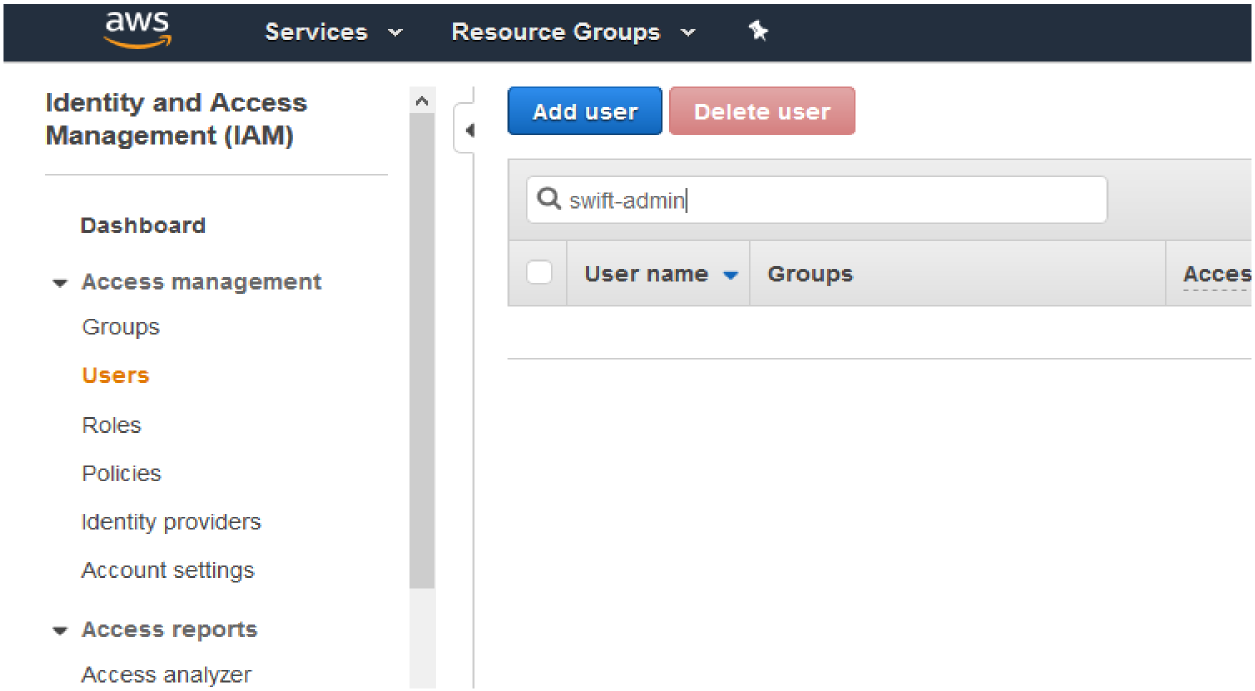
Adding Amazon AWS user for SWIFT use

This section highlights the steps to create an account under your Amazon AWS cloud, which you can use later to configure the EKS cluster details or discover Elastic Container Registry (ECR) instance, or AWS cloud object storage under your installed SWIFT.

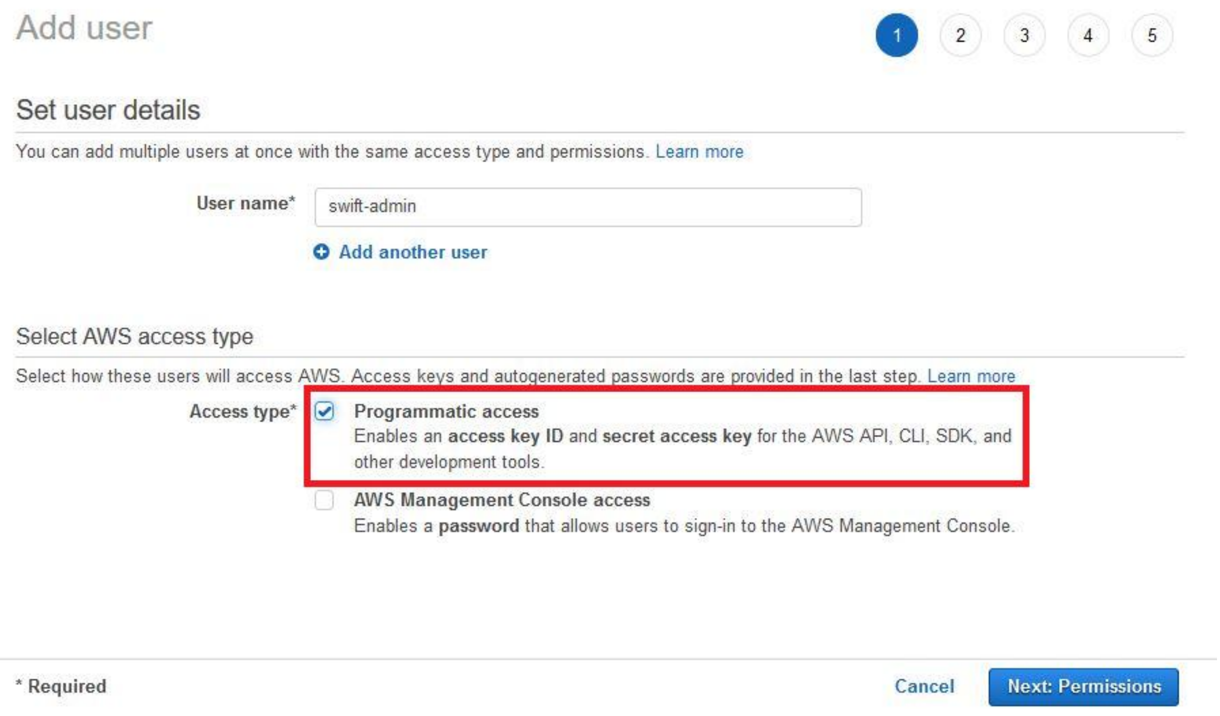
Login to AWS cloud console, and then from the 'Services' menu, select the 'IAM' service.



Traverse to the 'Users' menu on the left and then click on the 'Add User' button.



Fill up the user name and then select the ‘Programmatic Access’ for the access-type.



Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* swift-admin

Select AWS access type

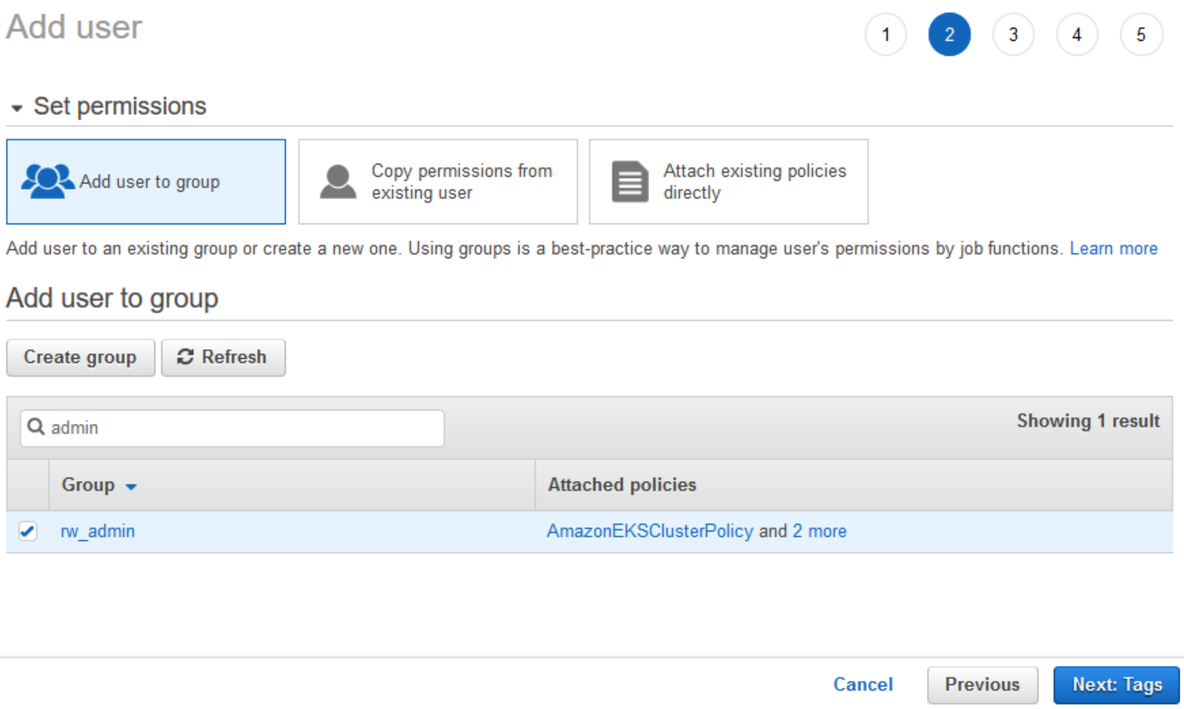
Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

Access type* ☑ **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

□ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required

On the next page, select necessary admin groups where this new user would need to be added. In this example case below, the ‘rw-admin’ group allows access to essential policies, which in turn enable the admin access to EKS clusters.


Add user

1 2 3 4 5

Set permissions

Add user to group
Copy permissions from existing user
Attach existing policies directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Add user to group

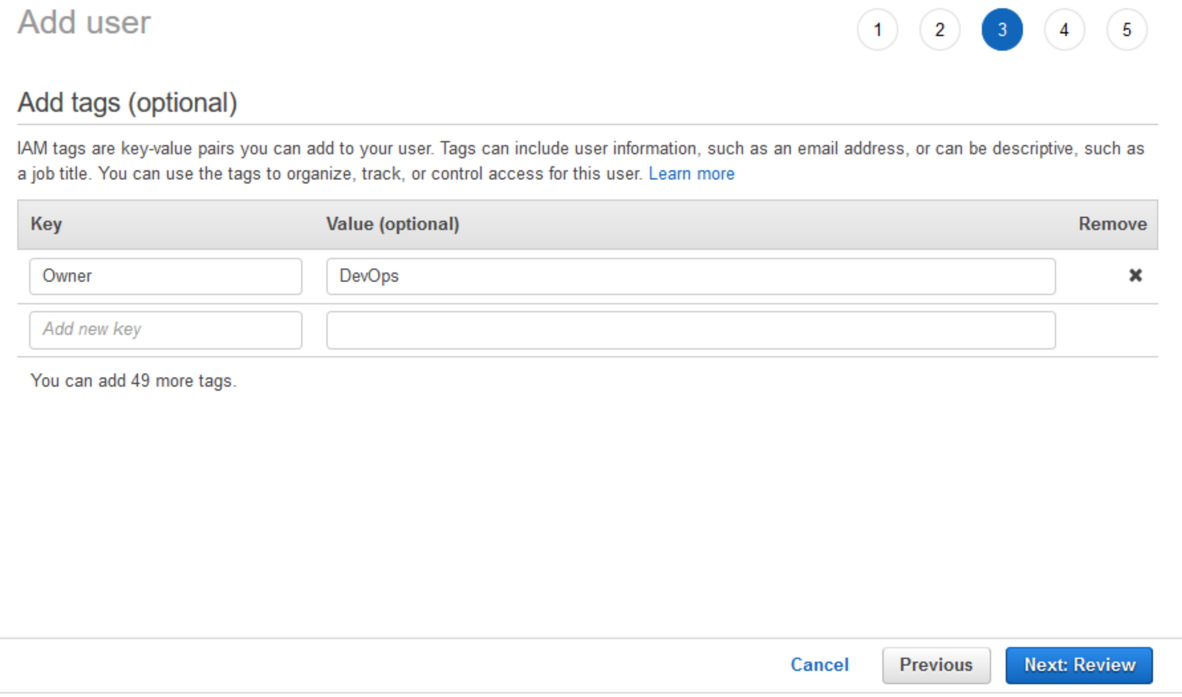
admin

Showing 1 result

	Group	Attached policies
☑	rw_admin	AmazonEKSClusterPolicy and 2 more

Cancel
Previous
Next: Tags

On the next page, select any tags you want to set for the new user.


Add user

1 2 3 4 5

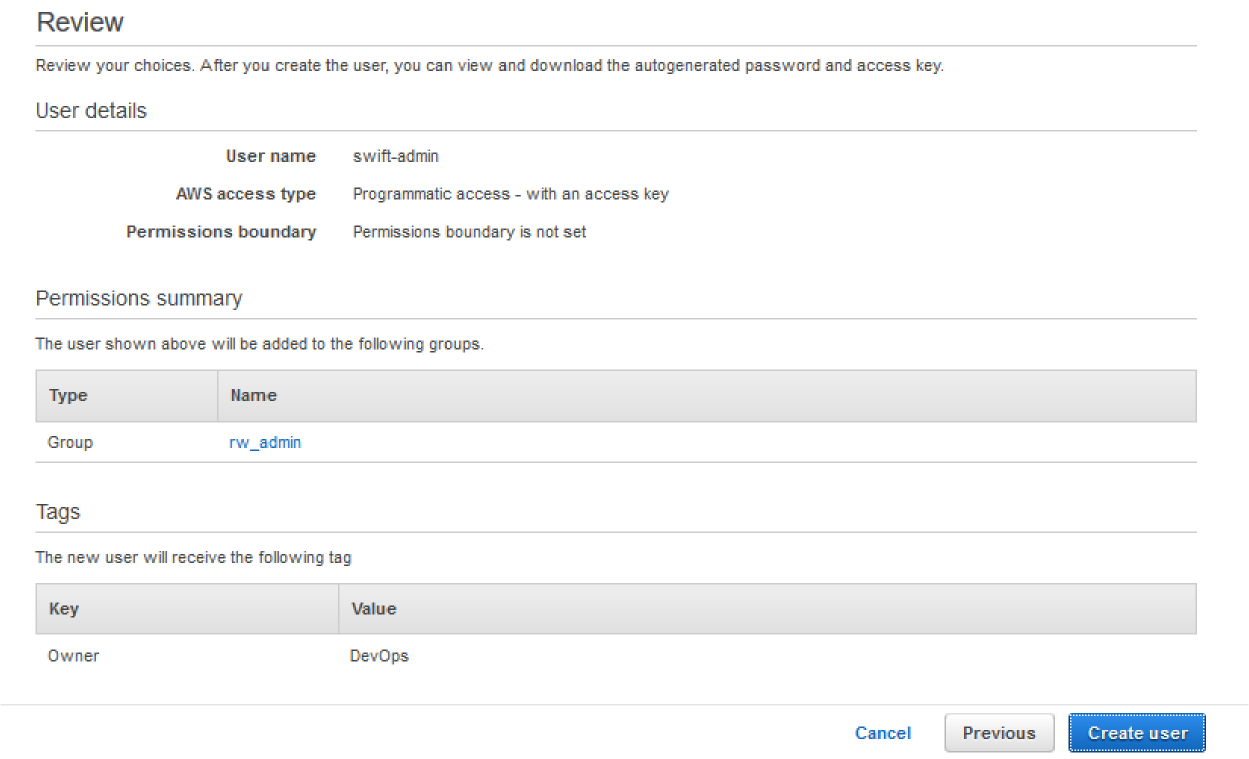
Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. Learn more

Key	Value (optional)	Remove
Owner	DevOps	✖
Add new key		

You can add 49 more tags.

Next, then review all settings and create the user.



Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	swift-admin
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

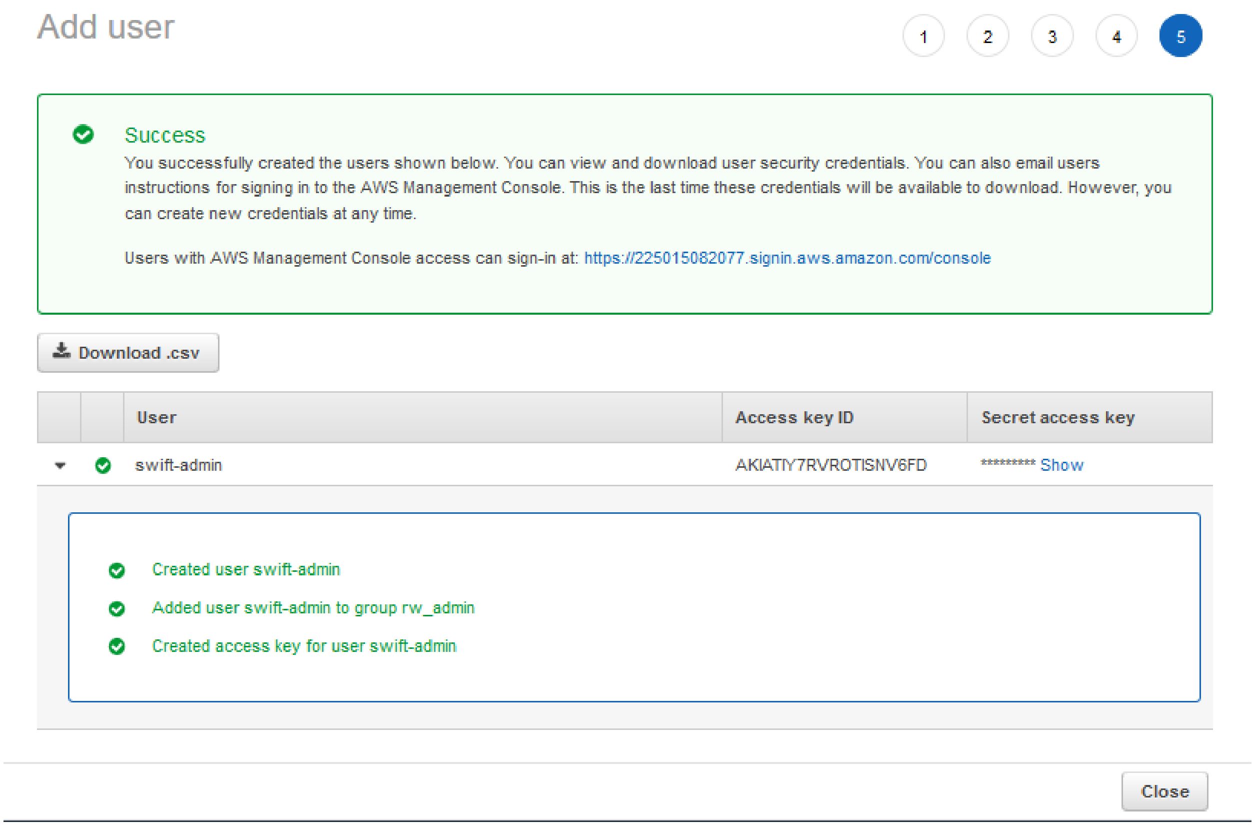
The user shown above will be added to the following groups.

Type	Name
Group	rw_admin

Tags

The new user will receive the following tag

Key	Value
Owner	DevOps



Add user

✓ Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: https://225015082077.signin.aws.amazon.com/console

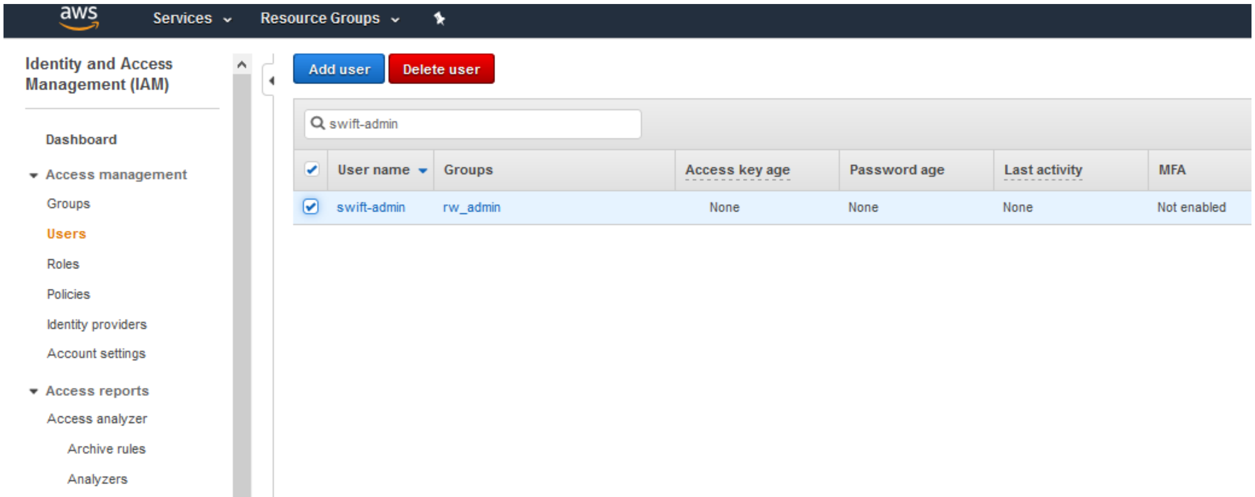
		User	Access key ID	Secret access key
▾	✓	swift-admin	AKIATIY7RVROTISNV6FD	********* Show

✓ Created user swift-admin

✓ Added user swift-admin to group rw_admin

✓ Created access key for user swift-admin

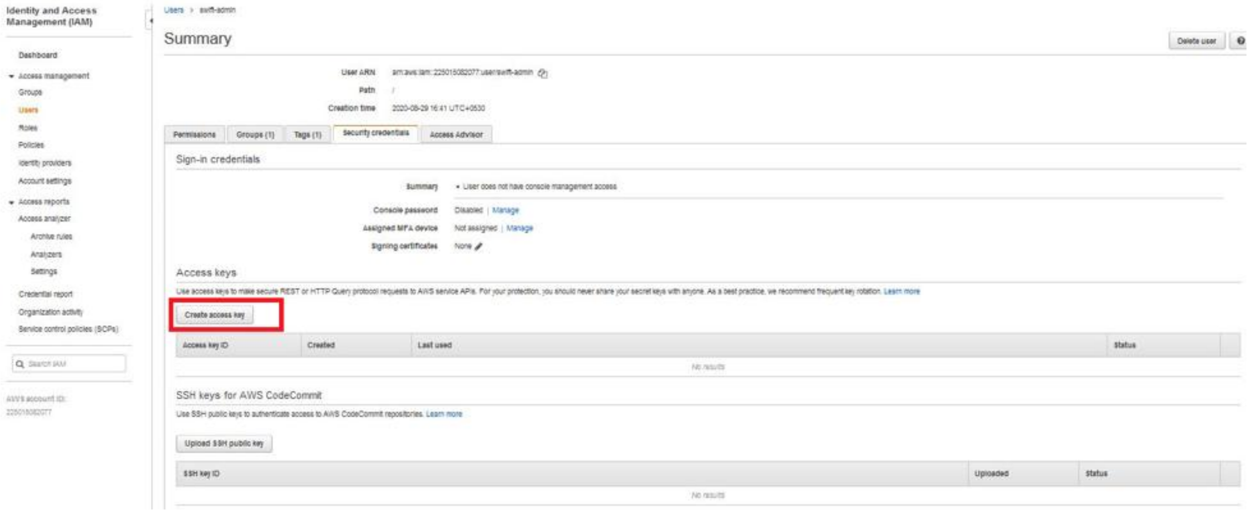
Next, click on the generated user from the ‘Users’ menu. You will create an access key for it.



swift-admin

☑	User name	Groups	Access key age	Password age	Last activity	MFA
☑	swift-admin	rw_admin	None	None	None	Not enabled

Traverse to the ‘Security Credentials’ tab under user details. You would use the ‘Create access key’ button to generate a new access key.



AWS account ID: 225015082077

Summary

User ARN arn:aws:iam::225015082077:user/swift-admin

Path /

Creation time 2020-08-29 16:41 UTC+0530

Sign-in credentials

Summary • User does not have console management access

Console password Disabled | Manage

Assigned MFA device Not assigned | Manage

Signing certificates None

Access keys

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. Learn more

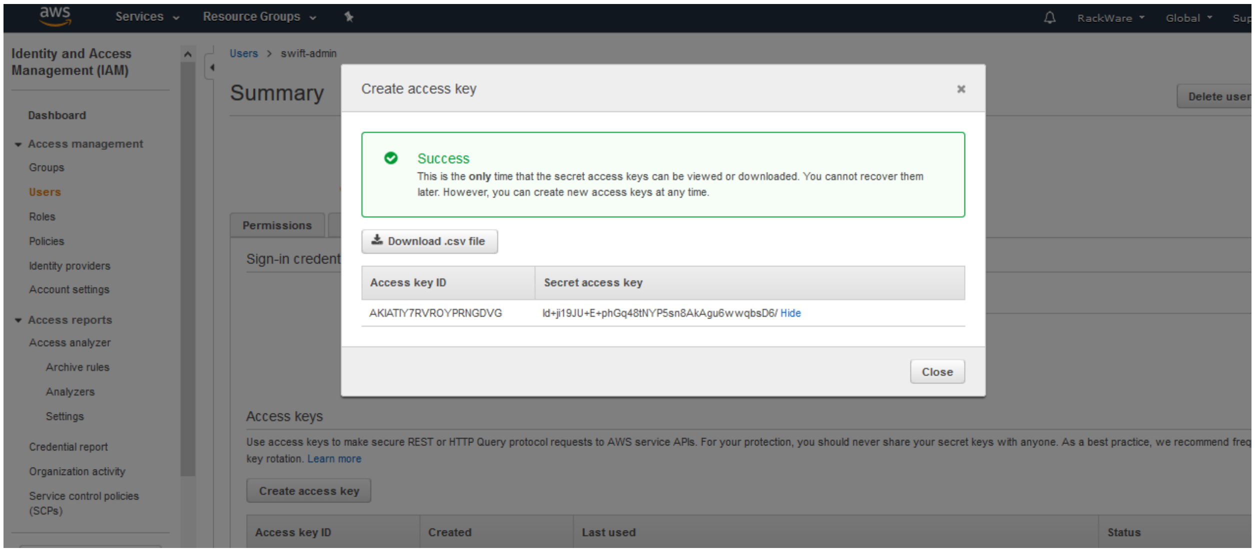
Access key ID	Created	Last used	Status
No results			

SSH keys for AWS CodeCommit

Use SSH public keys to authenticate access to AWS CodeCommit repositories. Learn more

SSH key ID	Uploaded	Status
No results		

Note the generated access key and store it securely. You will need it later while configuring your EKS cluster details under the SWIFT.



Create access key

Success
This is the **only** time that the secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time.

Download .csv file

Access key ID	Secret access key
AKIATIY7RVROYPRNGDVG	Id+ji19JU+E+phGq48tNYP5sn8AkAgu6wwqbsD6/ Hide

Close

Access keys

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend freq key rotation. Learn more

Note:

Once you create a new user and set its credentials/access-key, you will also need to whitelist this user for access to the Kubernetes (EKS) service instance, which is managed with the SWIFT. To do that, you have to edit the ConfigMap named ‘aws-auth,’ which is located under the ‘kube-system’ namespace on your EKS instance, and then add the ARN of the newly created user there in the below format:

```
apiVersion: v1
data:
  mapRoles: |
    - groups:
      - system:bootstrappers
      - system:nodes
      rolearn: arn:aws:iam::225015082077:role/eksctl-EKSCluster1-nodegroup-ng-a-NodeInstanceRole-1QTOCK6OPERZS
      username: system:node:{{EC2PrivateDNSName}}
  mapUsers: |
    - userarn: arn:aws:iam::225015082077:user/swift-admin
      username: swift-admin
      groups:
        - system:masters
kind: ConfigMap
metadata:
  name: aws-auth
  namespace: kube-system
```

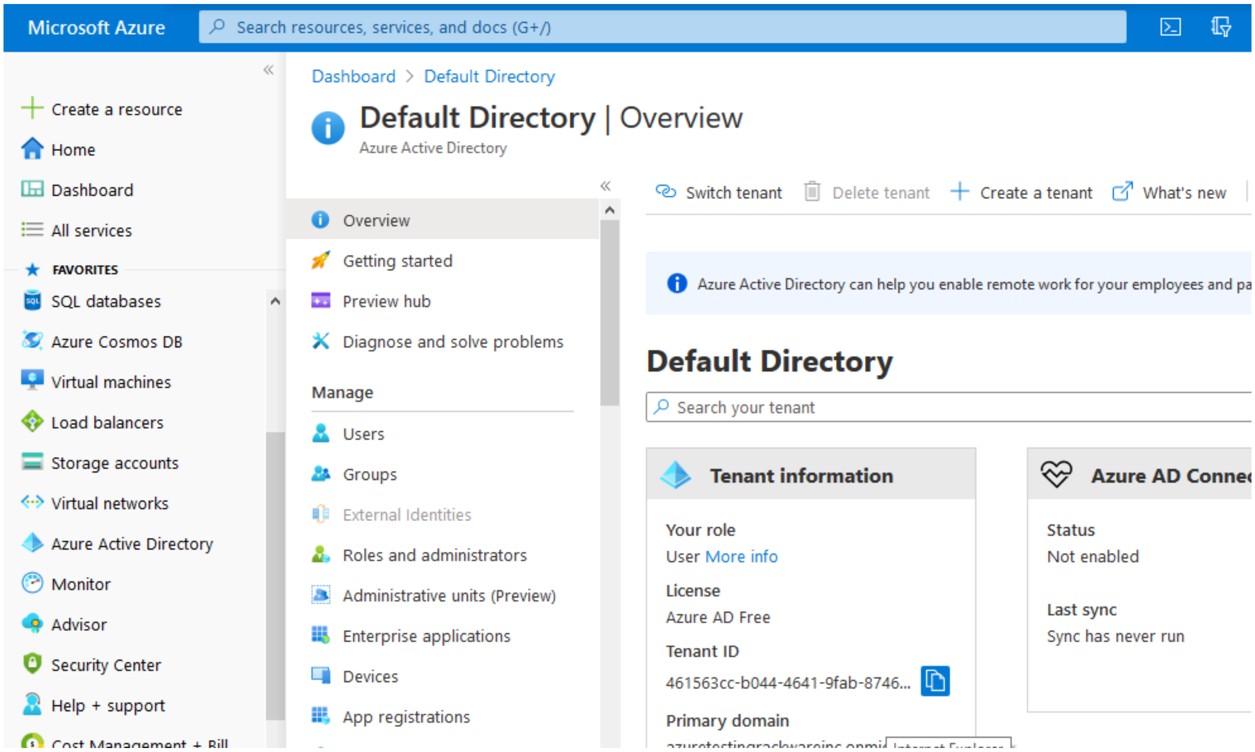
You will need to do the ConfigMap edit step above using the existing cluster-creator user's credentials. You can use the kubectl utility (along with cluster creator credentials) to do the edit. The detailed steps can be found in this KB.

Note that you will have to repeat this whitelisting step for every EKS instance, which you want the SWIFT to manage with the newly created user. If you skip this step, your newly created user will not be able to access the corresponding EKS service instance.

Adding Azure AAD application for SWIFT use

This section highlights the steps to create an AAD application under your Azure cloud. You will later use the AAD application credentials to configure the AKS cluster details under your installed SWIFT. The same credentials could also be used for Azure Container Registry (ACR) or Azure Object Storage discovery in SWIFT.

Log in to the Azure cloud console, and from menus on the left, select the 'Azure Active Directory' menu option.



Azure Active Directory can help you enable remote work for your employees and pa

Default Directory

Tenant information

Your role
User More info

License
Azure AD Free

Tenant ID
461563cc-b044-4641-9fab-8746...

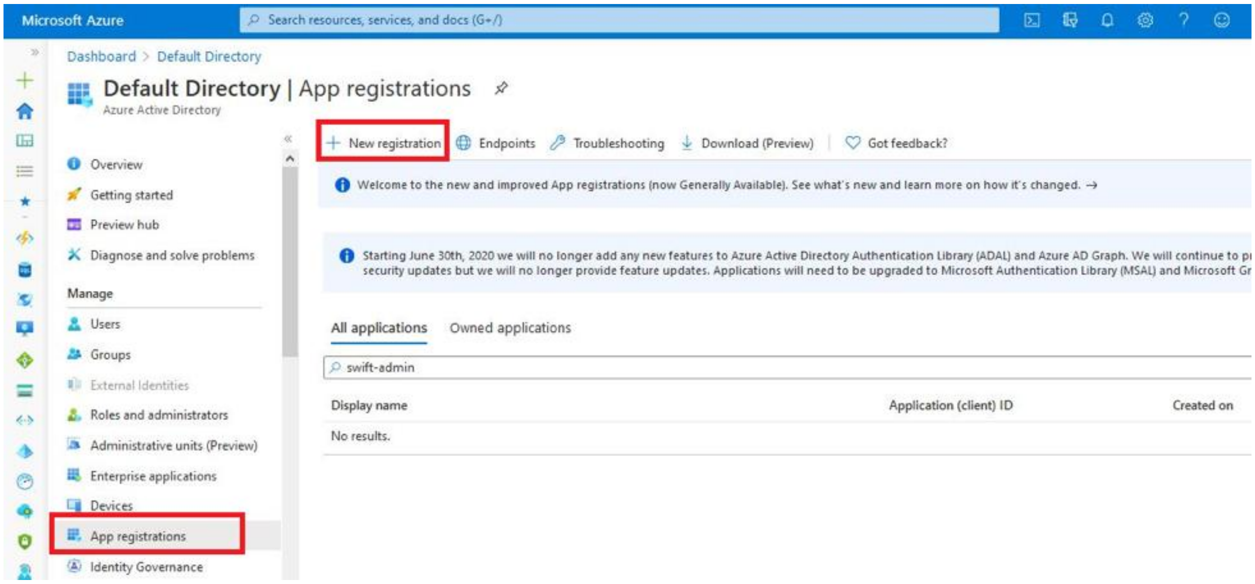
Primary domain
azuretestingrackwareinc.onmi

Azure AD Connec

Status
Not enabled

Last sync
Sync has never run

Click on the 'App Registrations' menu and then the 'New Registration' option.



Default Directory | App registrations

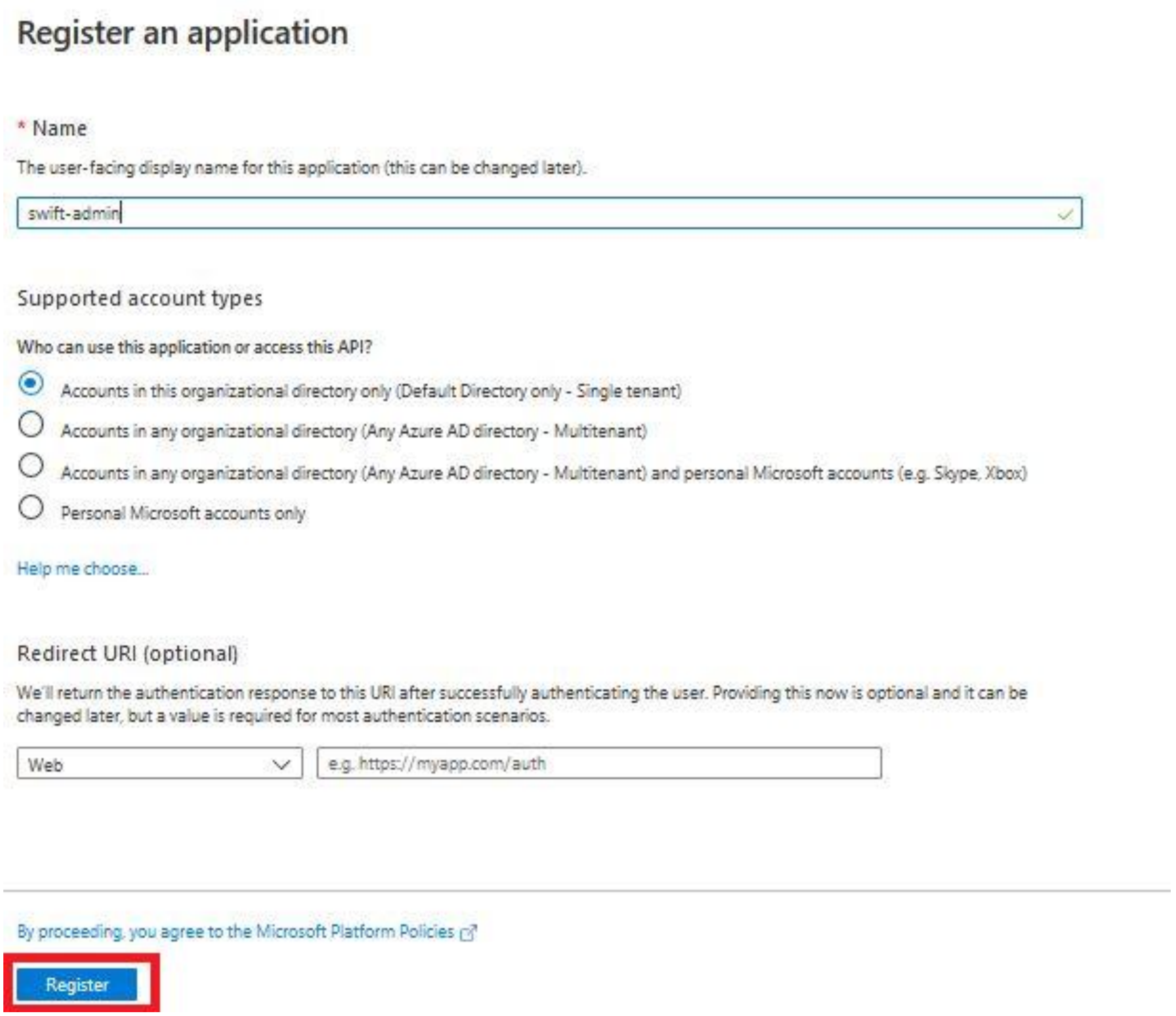
Azure Active Directory

Welcome to the new and improved App registrations (now Generally Available). See what's new and learn more on how it's changed. →

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to pr security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Gr

Display name	Application (client) ID	Created on
No results.		

Create a new app registration and give it a name.



Register an application

* Name

The user-facing display name for this application (this can be changed later).

swift-admin

Supported account types

Who can use this application or access this API?

● Accounts in this organizational directory only (Default Directory only - Single tenant)

○ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

○ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

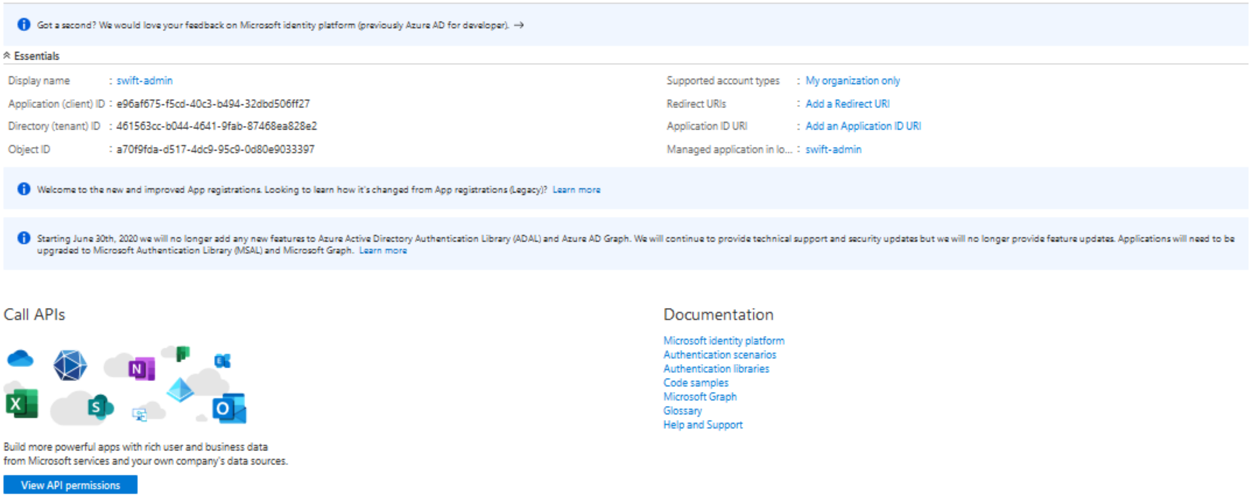
○ Personal Microsoft accounts only

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web | e.g. https://myapp.com/auth

Once the AAD application is created, note the app-id and directory-id. You will need these details later while configuring your AKS clusters under the SWIFT.



Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Essentials

Display name : swift-admin

Application (client) ID : e96af675-f5cd-40c3-b494-32dbd506ff27

Directory (tenant) ID : 461563cc-b044-4641-9fab-87468ea828e2

Object ID : a70f9fda-d517-4dc9-95c9-0d80e9033397

Supported account types : My organization only

Redirect URIs : Add a Redirect URI

Application ID URI : Add an Application ID URI

Managed application in lo... : swift-admin

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

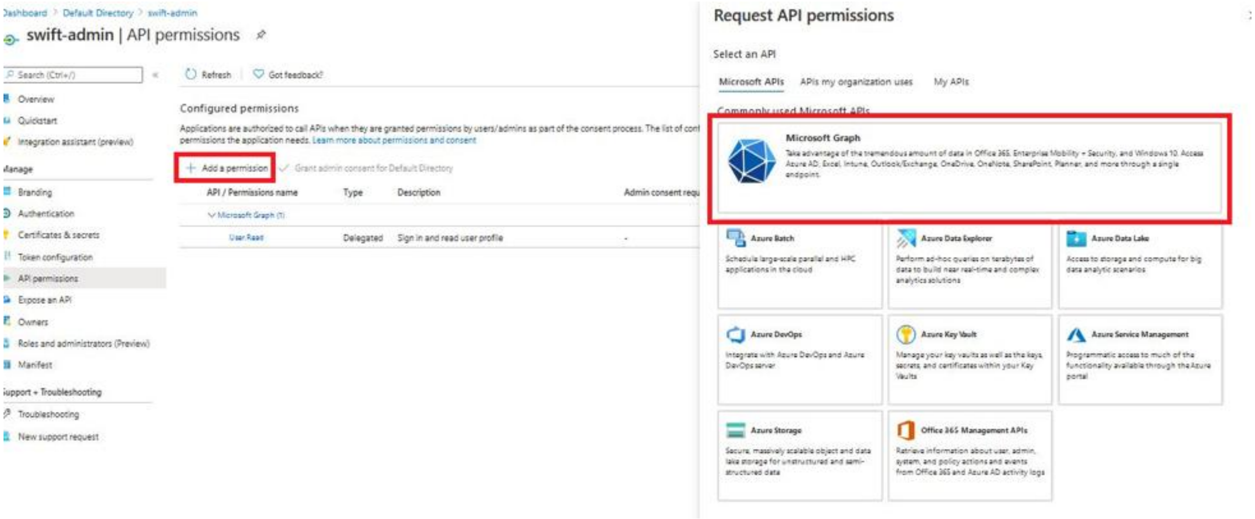
Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

Call APIs

Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

Documentation

For the new application, click on the 'API permissions' menu on the left, and then select the 'Add Permission' and the 'Microsoft Graph' API permissions.



Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of con permissions the application needs. Learn more about permissions and consent

API / Permissions name	Type	Description	Admin consent requ
Microsoft Graph (1)			
User.Read	Delegated	Sign in and read user profile	-

Request API permissions

Select an API

Microsoft Graph: Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Azure AD, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.

Azure Batch: Schedule large-scale parallel and HPC applications in the cloud

Azure Data Explorer: Perform ad-hoc queries on terabytes of data to build near real-time and complex analytics solutions

Azure Data Lake: Access to storage and compute for big data analytic scenarios

Azure DevOps: Integrate with Azure DevOps and Azure DevOps server

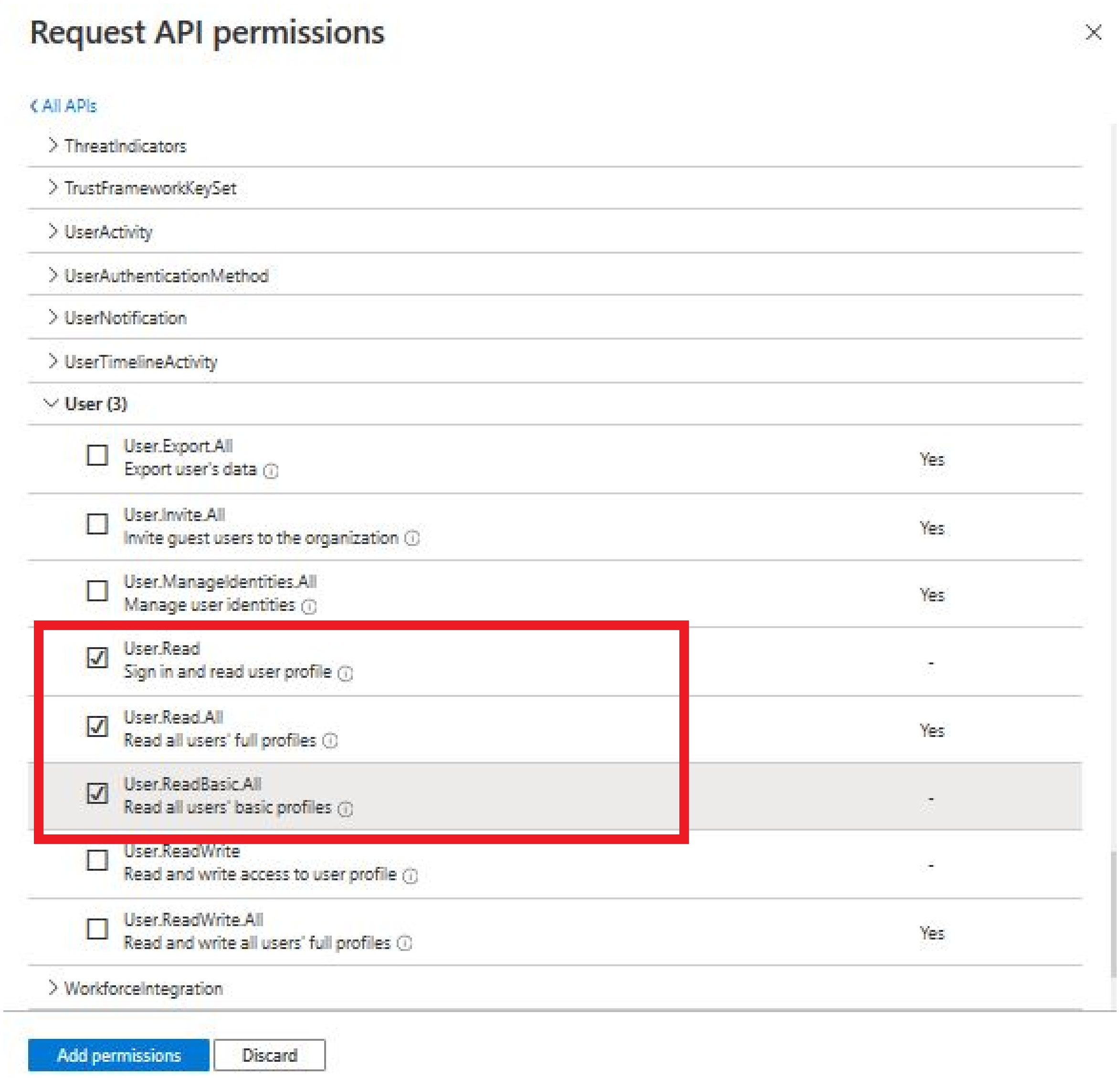
Azure Key Vault: Manage your key vaults as well as the keys, secrets, and certificates within your Key Vaults

Azure Service Management: Programmatic access to much of the functionality available through the Azure portal

Azure Storage: Secure, massively scalable object and data lake storage for unstructured and semi-structured data

Office 365 Management APIs: Retrieve information about user, admin, system, and policy actions and events from Office 365 and Azure AD activity logs

Under the 'Request API permissions' section, select 'Delegated permissions' and check all Read permissions as shown in the below screenshot, and then click on the 'Add permission' button to finish adding those.



Request API permissions ×

‹ All APIs

› ThreatIndicators

› TrustFrameworkKeySet

› UserActivity

› UserAuthenticationMethod

› UserNotification

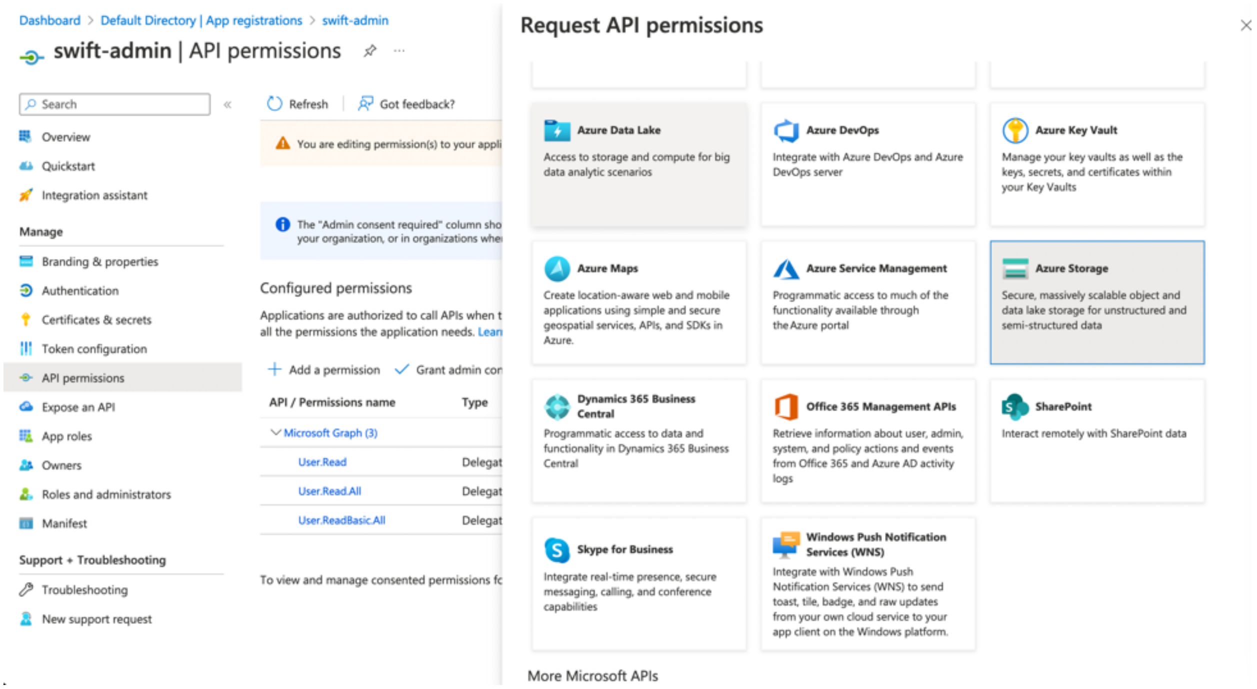
› UserTimelineActivity

⌄ User (3)

□	User.Export.All Export user's data ⓘ	Yes
□	User.Invite.All Invite guest users to the organization ⓘ	Yes
□	User.ManageIdentities.All Manage user identities ⓘ	Yes
☑	User.Read Sign in and read user profile ⓘ	-
☑	User.Read.All Read all users' full profiles ⓘ	Yes
☑	User.ReadBasic.All Read all users' basic profiles ⓘ	-
□	User.ReadWrite Read and write access to user profile ⓘ	-
□	User.ReadWrite.All Read and write all users' full profiles ⓘ	Yes

› WorkforceIntegration

Select ‘Add a permission’ again and this time select ‘Azure Storage’ category.



Dashboard > Default Directory | App registrations > swift-admin

swift-admin | API permissions

Search

- Overview
- Quickstart
- Integration assistant

Manage

- Branding & properties
- Authentication
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

Refresh | Got feedback?

You are editing permission(s) to your appli

The "Admin consent required" column sho your organization, or in organizations whe

Configured permissions

Applications are authorized to call APIs when t all the permissions the application needs. Lear

+ Add a permission ✓ Grant admin con

API / Permissions name	Type
Microsoft Graph (3)	
User.Read	Delegat
User.Read.All	Delegat
User.ReadBasic.All	Delegat

To view and manage consented permissions fc

Request API permissions

Azure Data Lake
Access to storage and compute for big data analytic scenarios

Azure DevOps
Integrate with Azure DevOps and Azure DevOps server

Azure Key Vault
Manage your key vaults as well as the keys, secrets, and certificates within your Key Vaults

Azure Maps
Create location-aware web and mobile applications using simple and secure geospatial services, APIs, and SDKs in Azure.

Azure Service Management
Programmatic access to much of the functionality available through the Azure portal

Azure Storage
Secure, massively scalable object and data lake storage for unstructured and semi-structured data

Dynamics 365 Business Central
Programmatic access to data and functionality in Dynamics 365 Business Central

Office 365 Management APIs
Retrieve information about user, admin, system, and policy actions and events from Office 365 and Azure AD activity logs

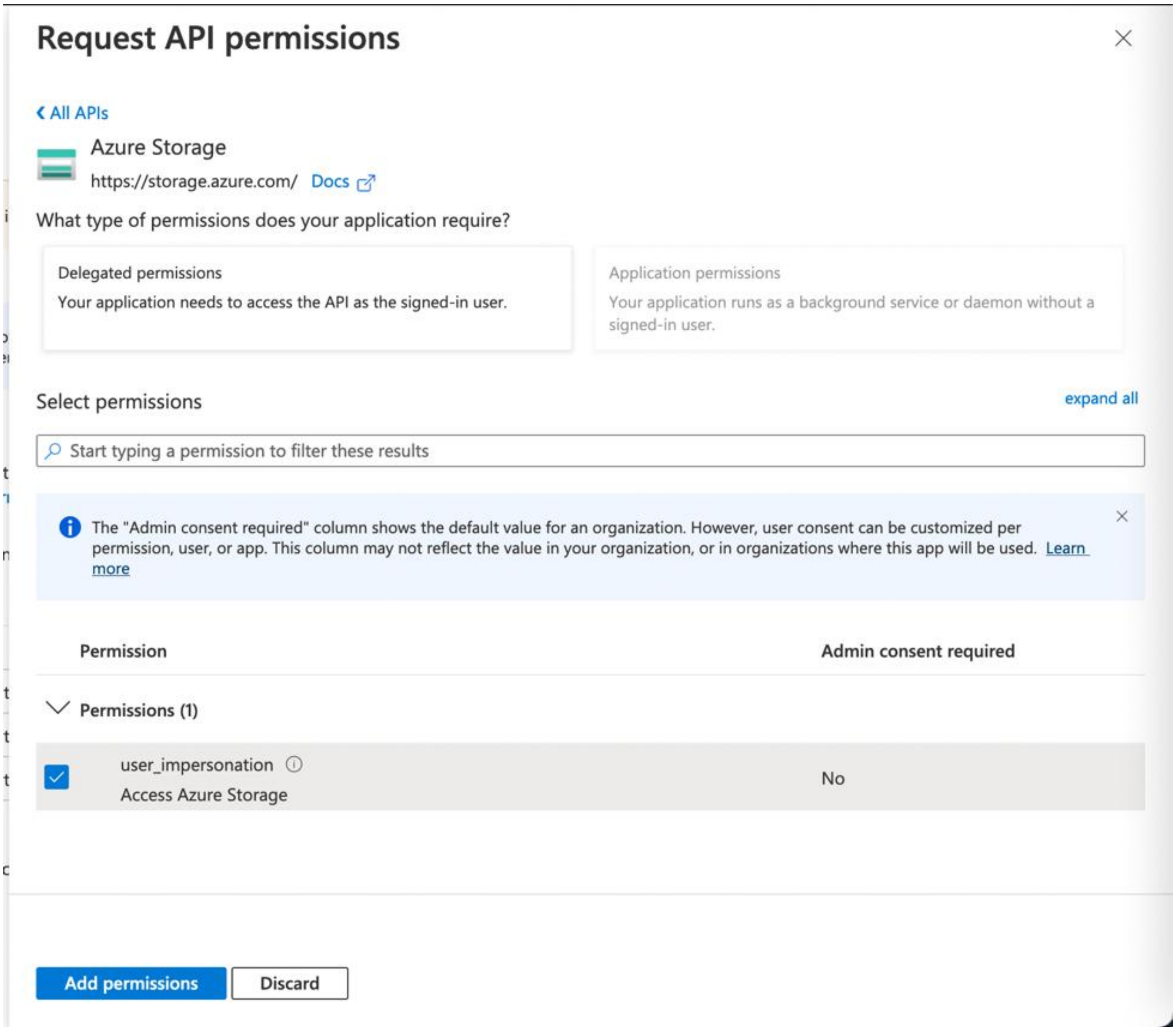
SharePoint
Interact remotely with SharePoint data

Skype for Business
Integrate real-time presence, secure messaging, calling, and conference capabilities

Windows Push Notification Services (WNS)
Integrate with Windows Push Notification Services (WNS) to send toast, tile, badge, and raw updates from your own cloud service to your app client on the Windows platform.

More Microsoft APIs

Select permissions shown below and then add it.



Request API permissions

Azure Storage

https://storage.azure.com/ Docs

What type of permissions does your application require?

Delegated permissions
Your application needs to access the API as the signed-in user.

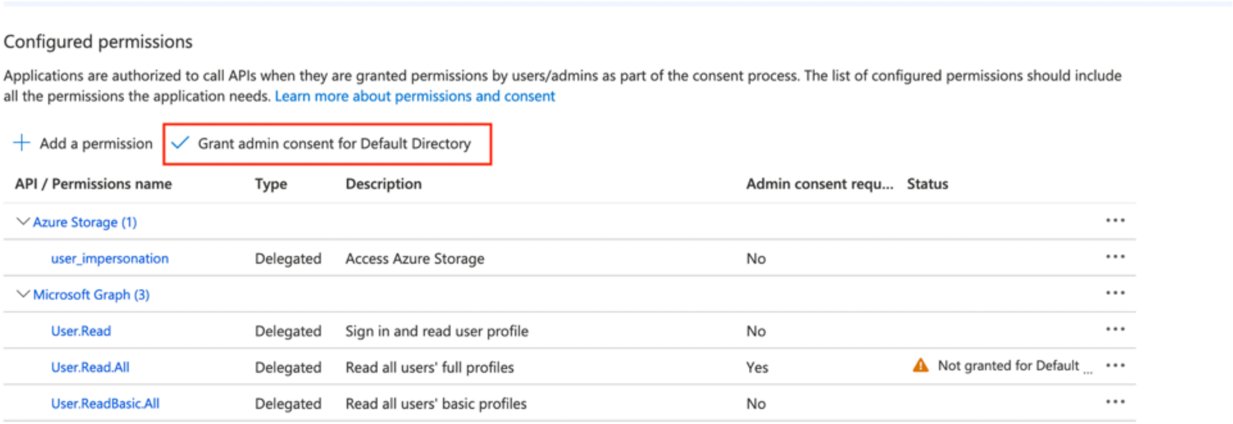
Application permissions
Your application runs as a background service or daemon without a signed-in user.

Select permissions

The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. Learn more

Permission	Admin consent required
Permissions (1)	
☑ user_impersonation Access Azure Storage	No

Once the permissions are added, select the admin consent grant option, and then grant the permissions.

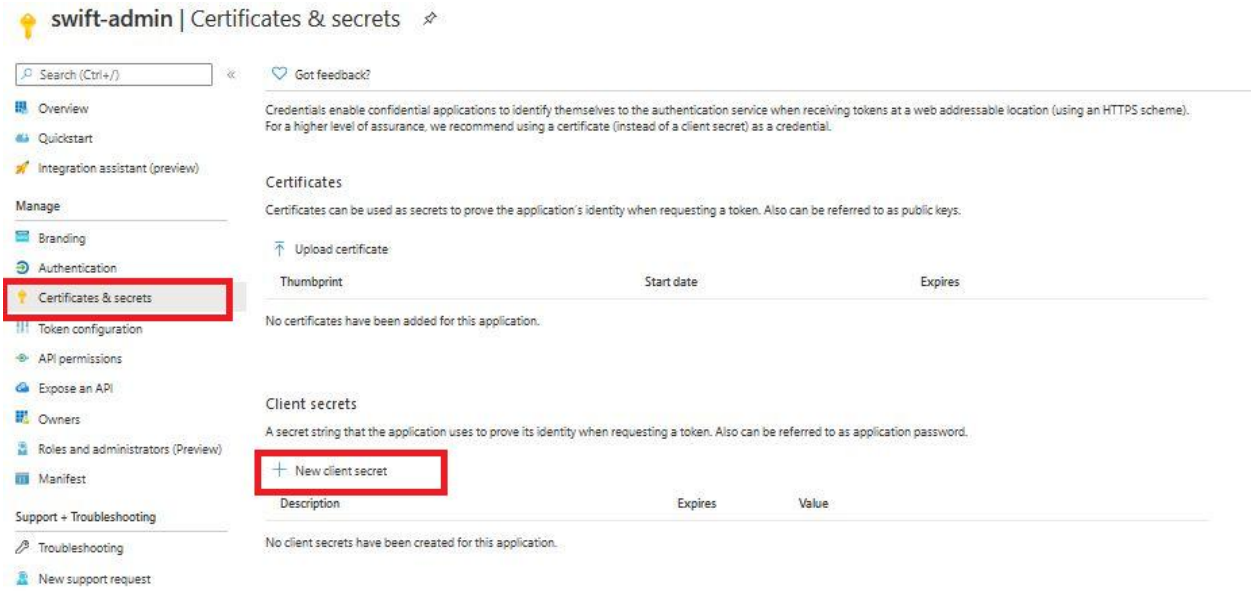


Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

API / Permissions name	Type	Description	Admin consent requ...	Status
Azure Storage (1)				
user_impersonation	Delegated	Access Azure Storage	No	
Microsoft Graph (3)				
User.Read	Delegated	Sign in and read user profile	No	
User.Read.All	Delegated	Read all users' full profiles	Yes	⚠ Not granted for Default ...
User.ReadBasic.All	Delegated	Read all users' basic profiles	No	

Next, click on the ‘Certificates & secrets’ menu, and then select the ‘New client secret’ option.



Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Thumbprint	Start date	Expires

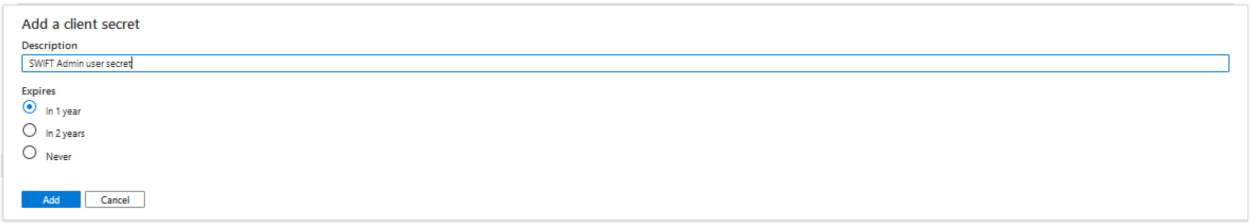
No certificates have been added for this application.

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value

No client secrets have been created for this application.



Add a client secret

Description

SWIFT Admin user secret

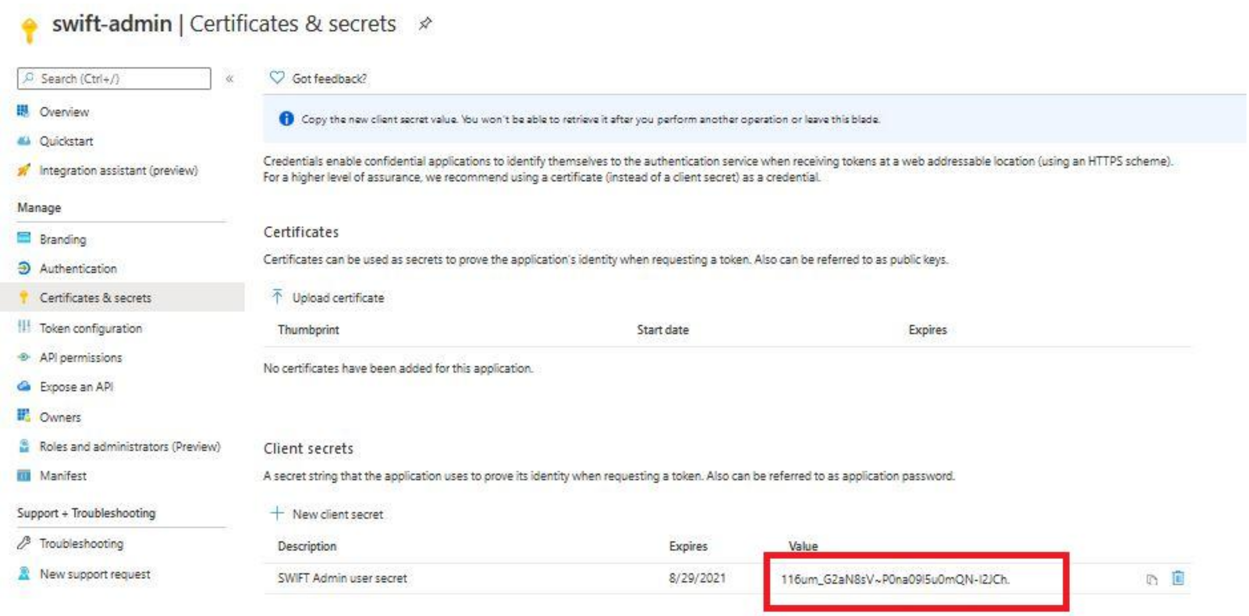
Expires

● In 1 year

○ In 2 years

○ Never

Note the generated secret key and store it safely. The key will not be shown later for security reasons. Also, you will need this key later while configuring your AKS cluster or ACR registry or Azure object storage details under the SWIFT.



Copy the new client secret value. You won't be able to retrieve it after you perform another operation or leave this blade.

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Thumbprint	Start date	Expires

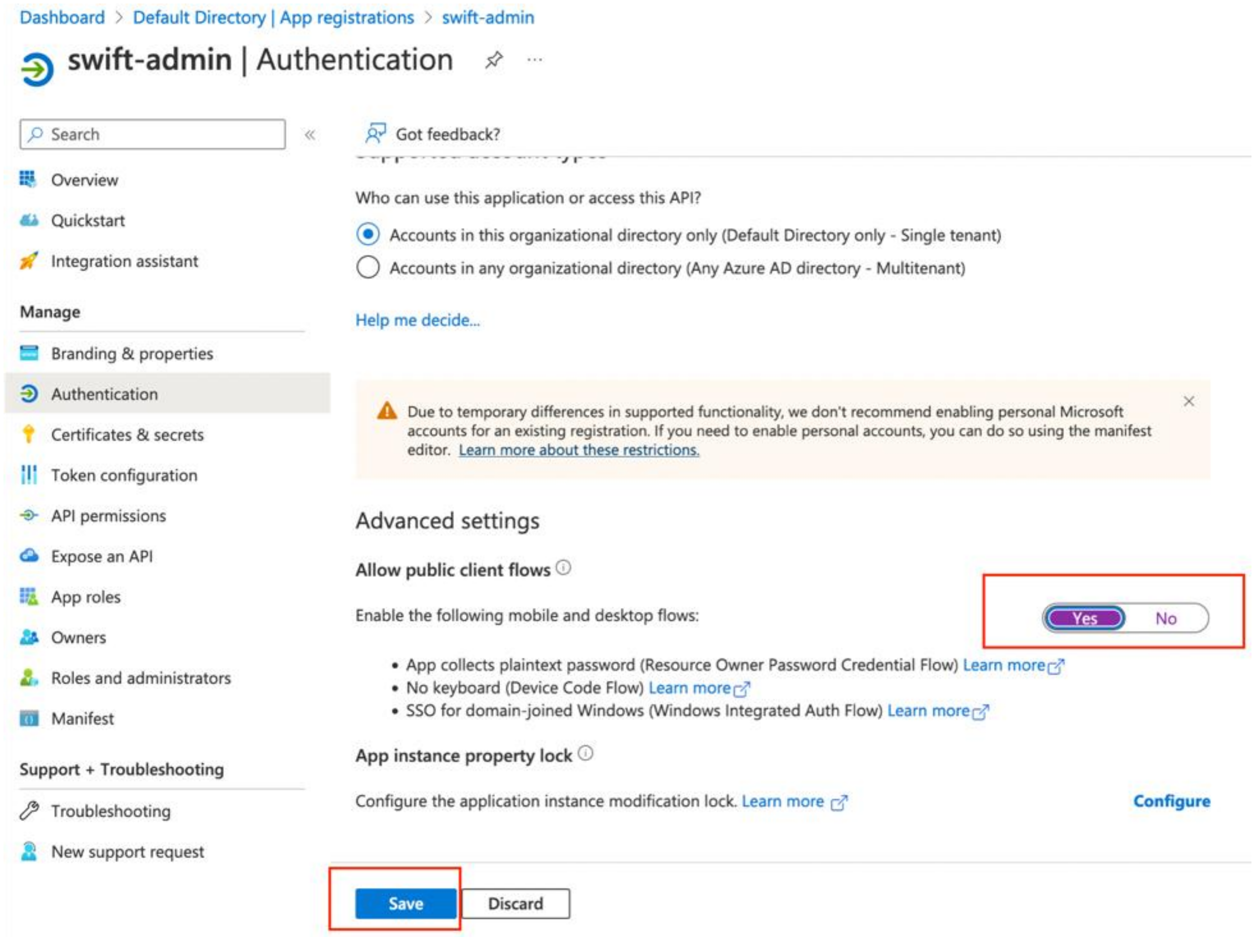
No certificates have been added for this application.

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value
SWIFT Admin user secret	8/29/2021	116um_G2aN8sV~P0na09l5u0mQN-l2JCh.

For the new application, from the 'Authentication' menu, enable public client flows. Make sure to press the save button to save the changes.



Who can use this application or access this API?

● Accounts in this organizational directory only (Default Directory only - Single tenant)

○ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

Due to temporary differences in supported functionality, we don't recommend enabling personal Microsoft accounts for an existing registration. If you need to enable personal accounts, you can do so using the manifest editor. Learn more about these restrictions.

Advanced settings

Allow public client flows

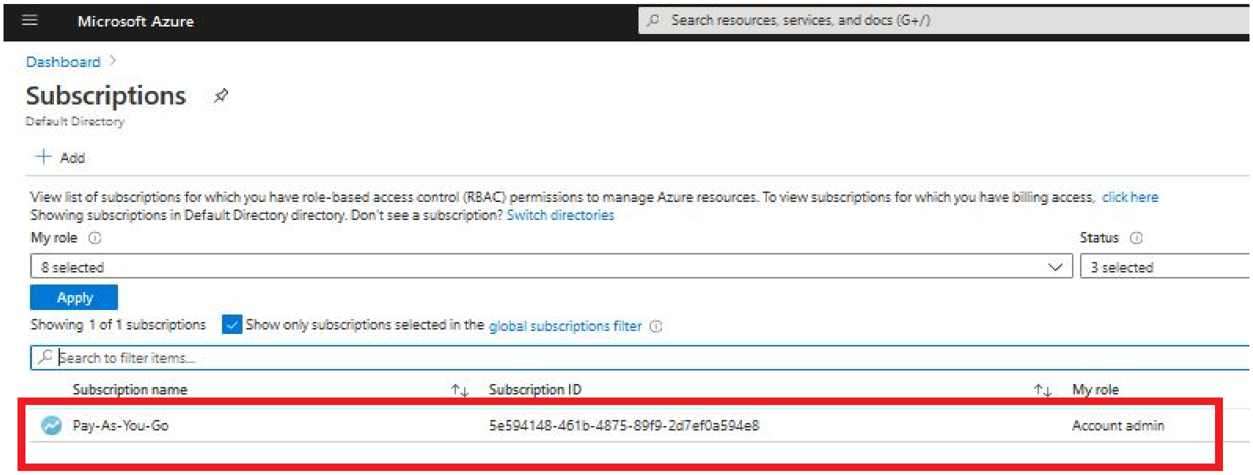
Enable the following mobile and desktop flows: [✓ Yes] No

- App collects plaintext password (Resource Owner Password Credential Flow) Learn more
- No keyboard (Device Code Flow) Learn more
- SSO for domain-joined Windows (Windows Integrated Auth Flow) Learn more

App instance property lock

Configure the application instance modification lock. Learn more

Grant subscription ‘Contributor’ role to the new application. To do that, click on the ‘Subscriptions’ menu and select your subscription where you have your AKS clusters provisioned.



Subscriptions

Default Directory

View list of subscriptions for which you have role-based access control (RBAC) permissions to manage Azure resources. To view subscriptions for which you have billing access, click here
Showing subscriptions in Default Directory directory. Don't see a subscription? Switch directories

My role: 8 selected

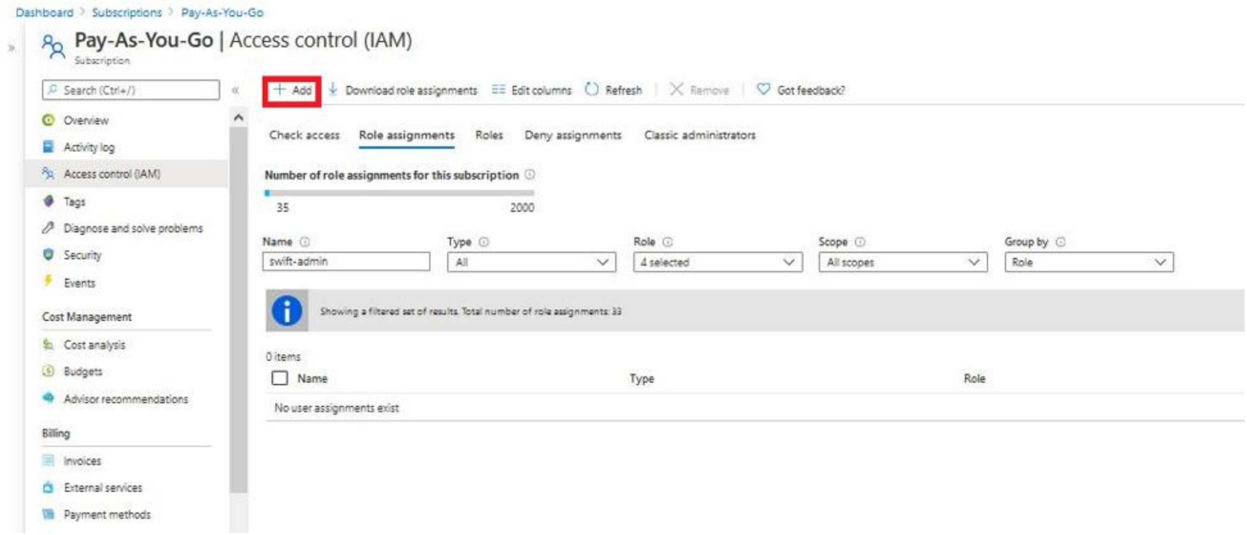
Status: 3 selected

Showing 1 of 1 subscriptions ☑ Show only subscriptions selected in the global subscriptions filter

Search to filter items...

Subscription name	Subscription ID	My role
Pay-As-You-Go	5e594148-461b-4875-89f9-2d7ef0a594e8	Account admin

While on the above page, you may also want to note down your subscription id, as you will need it later while configuring your AKS cluster details under the SWIFT.



Pay-As-You-Go | Access control (IAM)

Subscription

Number of role assignments for this subscription

35 2000

Name: swift-admin

Type: All

Role: 4 selected

Scope: All scopes

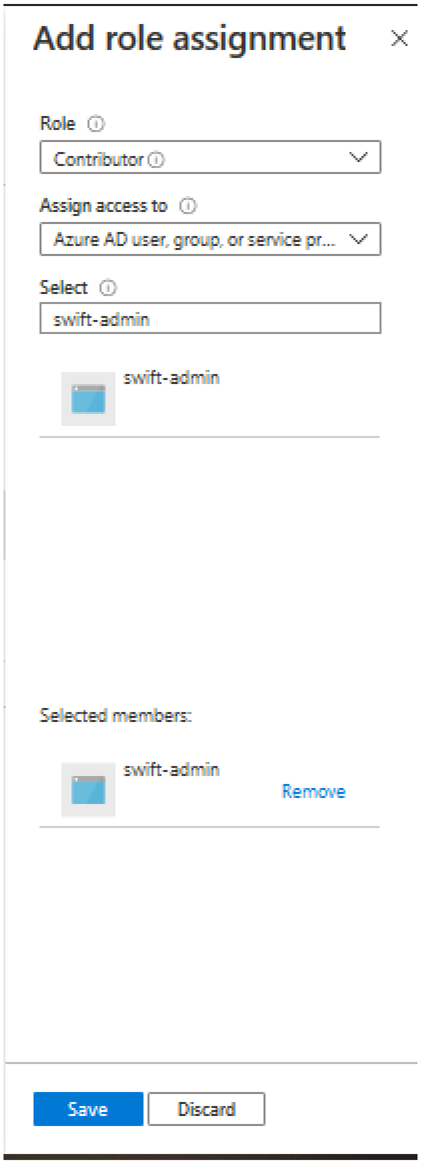
Group by: Role

Showing a filtered set of results. Total number of role assignments: 33

0 items

□ Name	Type	Role
No user assignments exist		

On add dialog, select the newly created application and the contributor role, and then save it.



Add role assignment

Role: Contributor

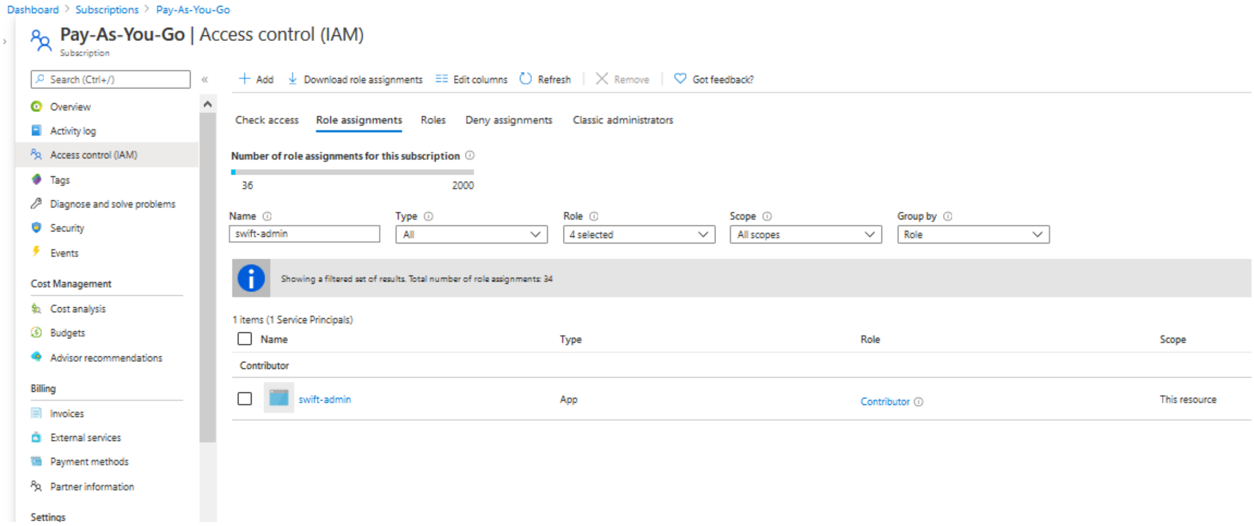
Assign access to: Azure AD user, group, or service pr...

Select: swift-admin

swift-admin

Selected members:

swift-admin



Dashboard > Subscriptions > Pay-As-You-Go

Pay-As-You-Go | Access control (IAM)

Subscription

Number of role assignments for this subscription

36 2000

Name: swift-admin

Type: All

Role: 4 selected

Scope: All scopes

Group by: Role

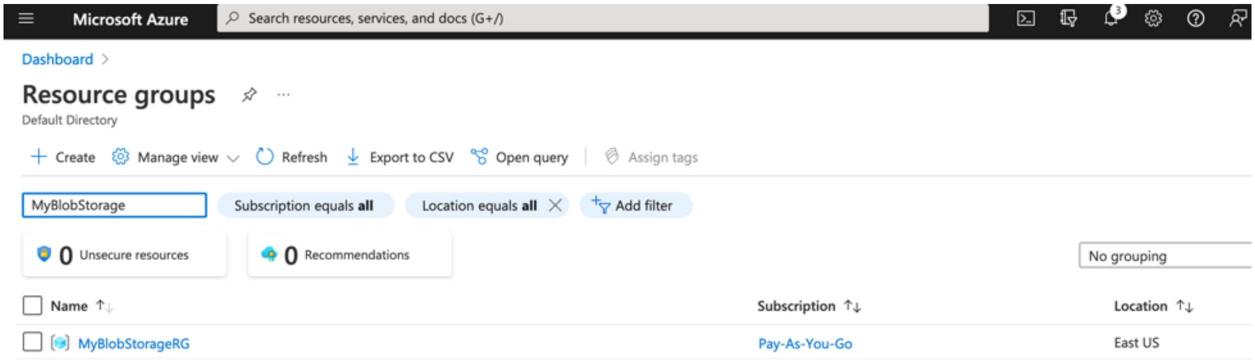
Showing a filtered set of results. Total number of role assignments: 34

1 items (1 Service Principals)

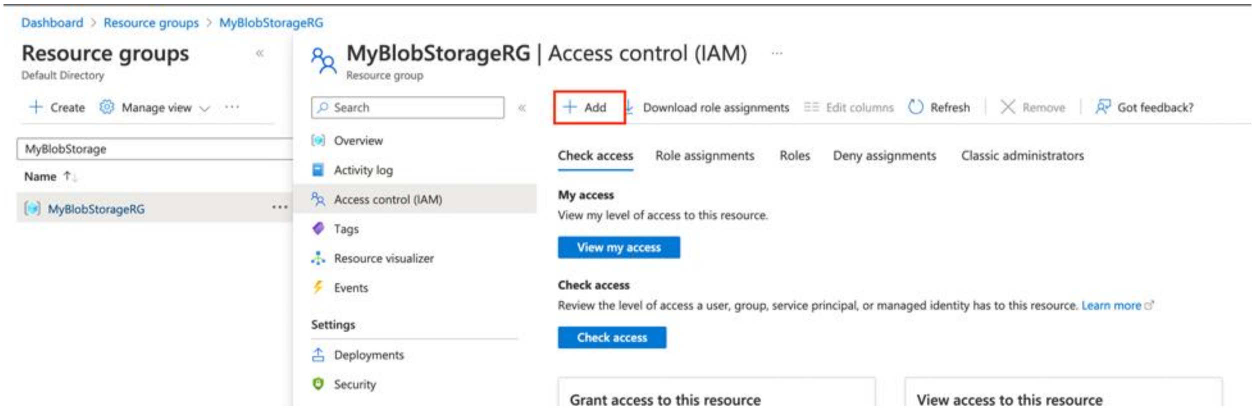
Name	Type	Role	Scope
Contributor			
swift-admin	App	Contributor	This resource

Lastly, you need to assign ‘Storage Blob Data Contributor’ role to the new app or client we created above at all resource group levels where you have either cloud object storage created or want SWIFT to use those resource groups for longer term backups. Note that this is optional step if you do not want SWIFT to use Azure object storage for backups.

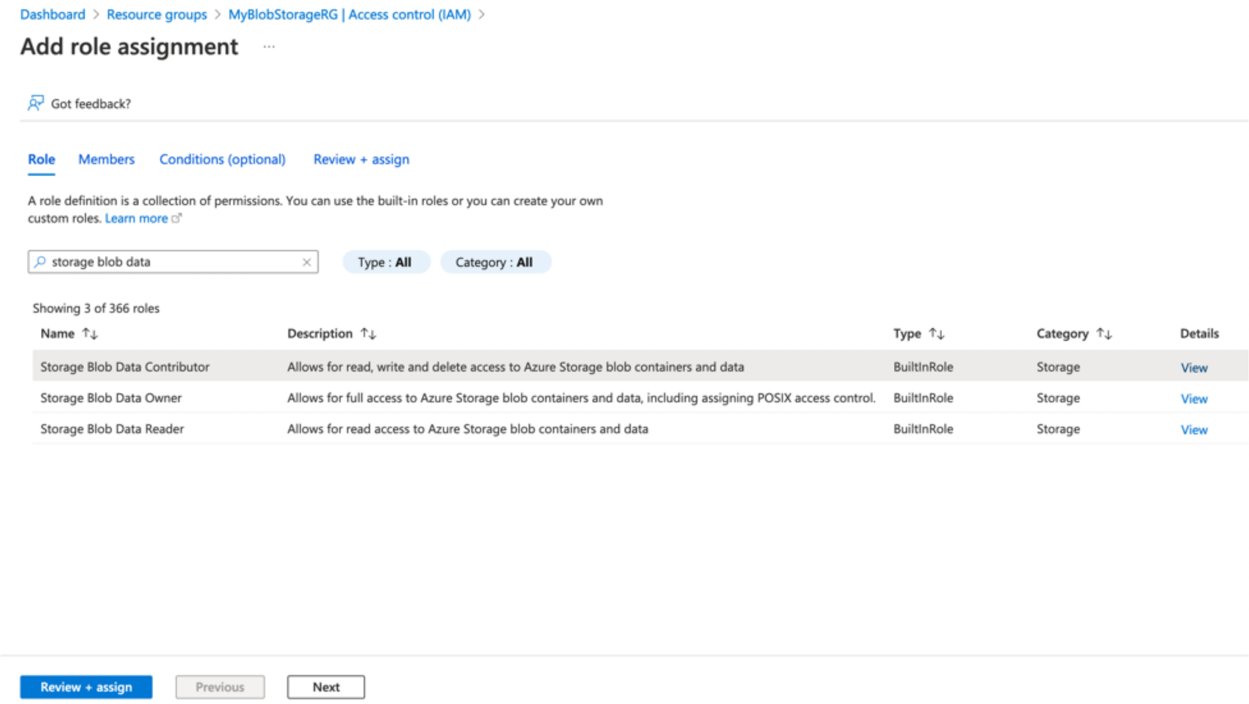
Select required resource group from ‘Resource Groups’ menu.



Select the Resource Group and then ‘Access Control (IAM)’ properties for it. Then select the ‘Add’ button and then ‘Add role assignment’ option.



Select the ‘Storage Blob Data Contributor’ role.



Dashboard > Resource groups > MyBlobStorageRG | Access control (IAM) >

Add role assignment

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. Learn more

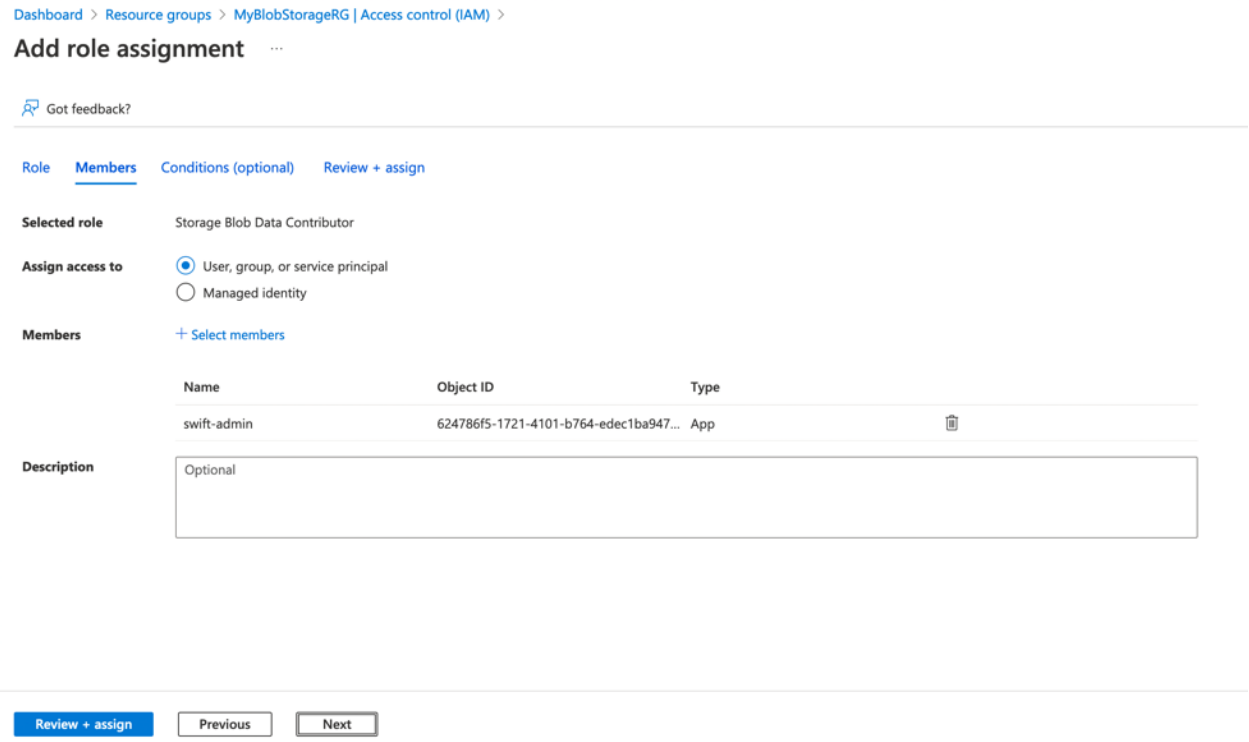
storage blob data

Type : **All** Category : **All**

Showing 3 of 366 roles

Name ↑↓	Description ↑↓	Type ↑↓	Category ↑↓	Details
Storage Blob Data Contributor	Allows for read, write and delete access to Azure Storage blob containers and data	BuiltInRole	Storage	View
Storage Blob Data Owner	Allows for full access to Azure Storage blob containers and data, including assigning POSIX access control.	BuiltInRole	Storage	View
Storage Blob Data Reader	Allows for read access to Azure Storage blob containers and data	BuiltInRole	Storage	View

Select the app or client you created in earlier steps. For this example case, that is ‘swift-admin’ app.



Dashboard > Resource groups > MyBlobStorageRG | Access control (IAM) >

Add role assignment

Selected role Storage Blob Data Contributor

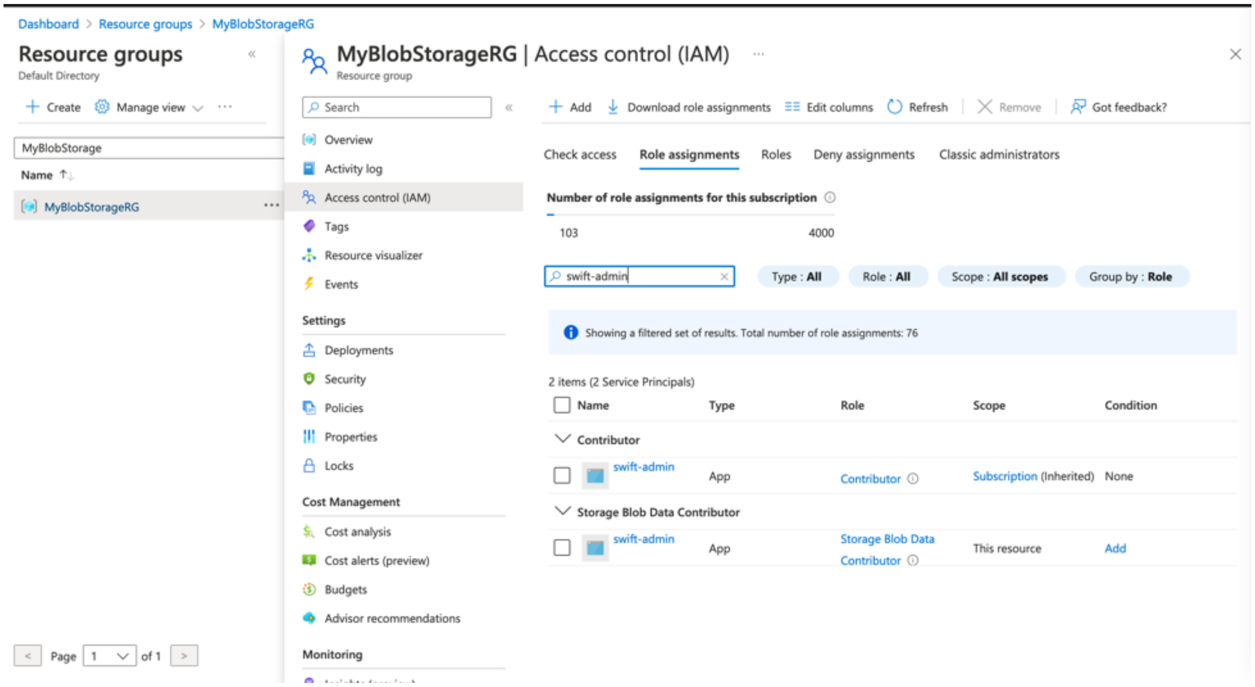
Assign access to
● User, group, or service principal
○ Managed identity

Members + Select members

Name	Object ID	Type
swift-admin	624786f5-1721-4101-b764-edec1ba947...	App

Description Optional

Press the ‘Next’ button and add permissions after a review. You will see newly added permissions under IAM menu for the resource group.



Number of role assignments for this subscription

103 4000

swift-admin

Type : All | Role : All | Scope : All scopes | Group by : Role

Showing a filtered set of results. Total number of role assignments: 76

2 items (2 Service Principals)

Name	Type	Role	Scope	Condition
Contributor				
swift-admin	App	Contributor	Subscription (Inherited)	None
Storage Blob Data Contributor				
swift-admin	App	Storage Blob Data Contributor	This resource	Add

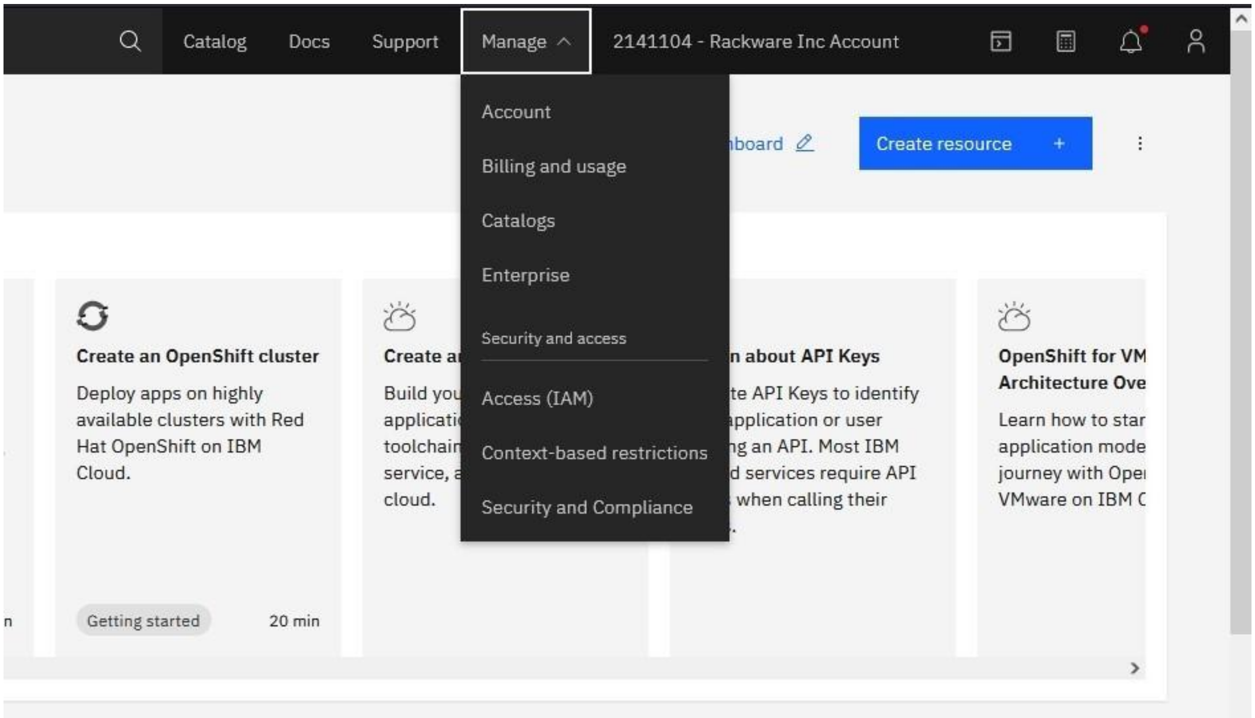
Please now note the below details for the account and the newly created AAD application, which you will need later while configuring your AKS clusters, object storages, or container image registries under the SWIFT:

- Subscription id
- Tenant id (Default directory id)
- Newly created app name and id (client id)
- Newly created app secret (client secret)

Adding IBM cloud user for SWIFT use

This section highlights the steps to create a user account under your IBM cloud. You will later use these user credentials to configure the IBM Kubernetes Service (IKS) as well as IBM OpenShift cluster details under your installed SWIFT. The same credentials could also be used later to discover an IBM Cloud Container Registry or an object storage under SWIFT.

Log in to the IBM cloud console, and from the top section, select the 'Manage' and then the 'Access (IAM)' menu option.



Create an OpenShift cluster

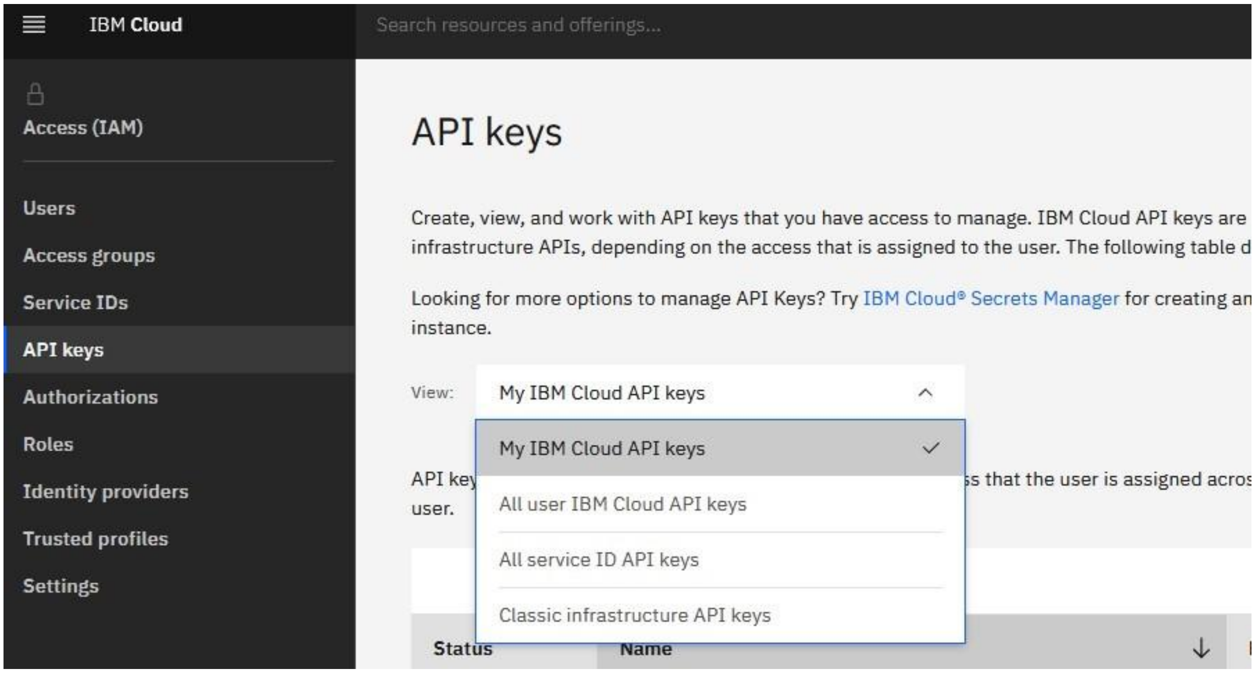
Deploy apps on highly available clusters with Red Hat OpenShift on IBM Cloud.

Create a
Build you
applicati
toolchai
service, a
cloud.

n about API Keys
te API Keys to identify
application or user
ng an API. Most IBM
d services require API
when calling their
.

OpenShift for VM
Architecture Ove
Learn how to star
application mode
journey with Ope
VMware on IBM C

Click on the 'API keys' option. Make sure the 'My IBM Cloud API keys' option is selected.



API keys

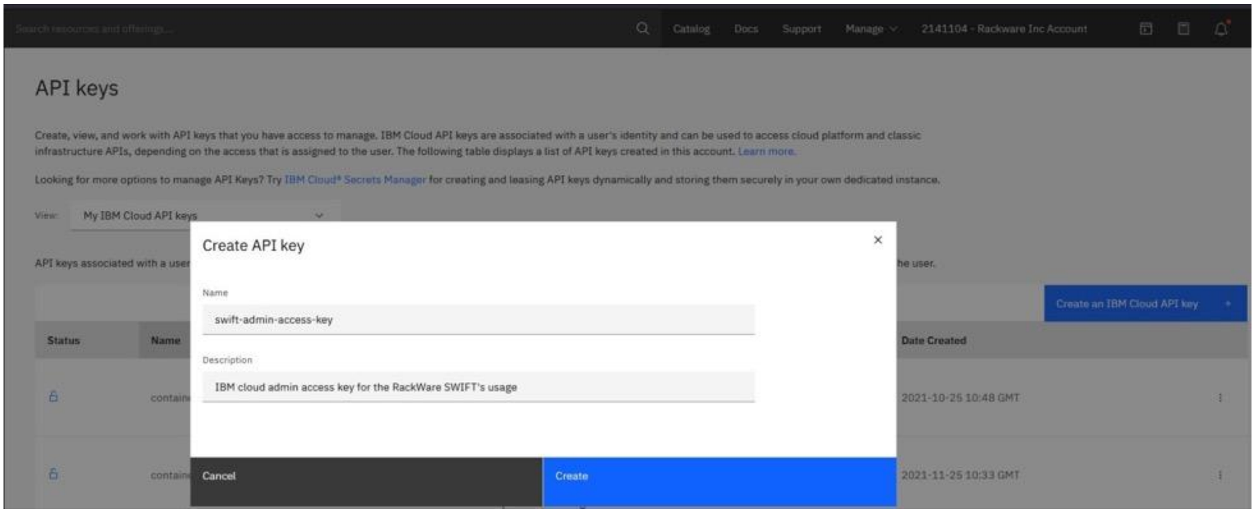
Create, view, and work with API keys that you have access to manage. IBM Cloud API keys are
infrastructure APIs, depending on the access that is assigned to the user. The following table d

Looking for more options to manage API Keys? Try IBM Cloud® Secrets Manager for creating an
instance.

API key
user.
s that the user is assigned acros

To create an API key, select the 'Create an IBM Cloud API Key' button on the right side of the webpage.

Fill in the name of the key file name and relevant description, and then click on the 'Create' button.


Search resources and offerings...
Catalog
Docs
Support
Manage
2141104 - Rackware Inc Account
Create an IBM Cloud API key +
Cancel
Create

API keys

Create, view, and work with API keys that you have access to manage. IBM Cloud API keys are associated with a user's identity and can be used to access cloud platform and classic infrastructure APIs, depending on the access that is assigned to the user. The following table displays a list of API keys created in this account. Learn more.

Looking for more options to manage API Keys? Try IBM Cloud® Secrets Manager for creating and leasing API keys dynamically and storing them securely in your own dedicated instance.

View: My IBM Cloud API keys

API keys associated with a user ... he user.

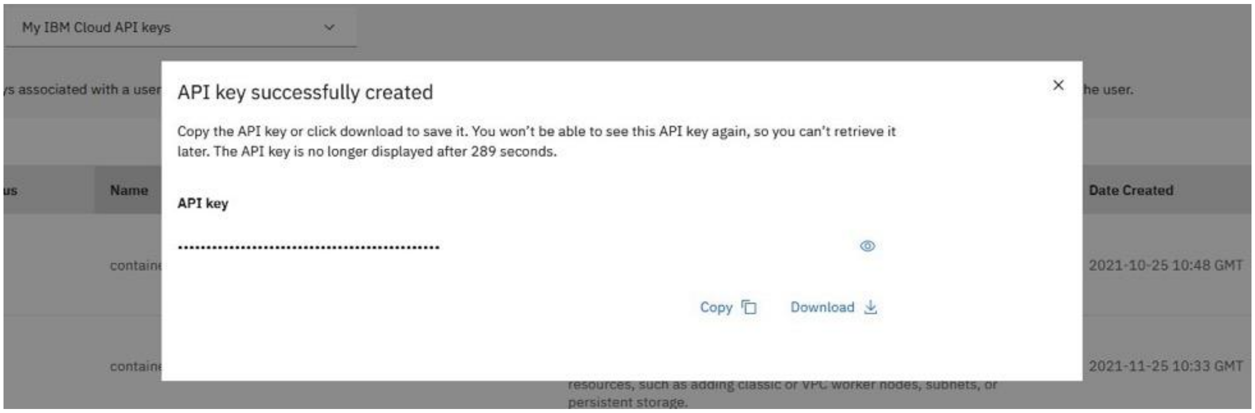
Create API key

Name: swift-admin-access-key

Description: IBM cloud admin access key for the RackWare SWIFT's usage

Status	Name	Date Created
	contain	2021-10-25 10:48 GMT
	contain	2021-11-25 10:33 GMT

Download and save the generated API key safely. This key will be needed later while adding an IBM IKS or OpenShift cluster to the SWIFT.



API key successfully created

Copy the API key or click download to save it. You won't be able to see this API key again, so you can't retrieve it later. The API key is no longer displayed after 289 seconds.

API key

••

us	Name	Date Created
	containe	2021-10-25 10:48 GMT
	containe	2021-11-25 10:33 GMT

resources, such as adding classic or VPC worker nodes, subnets, or persistent storage.

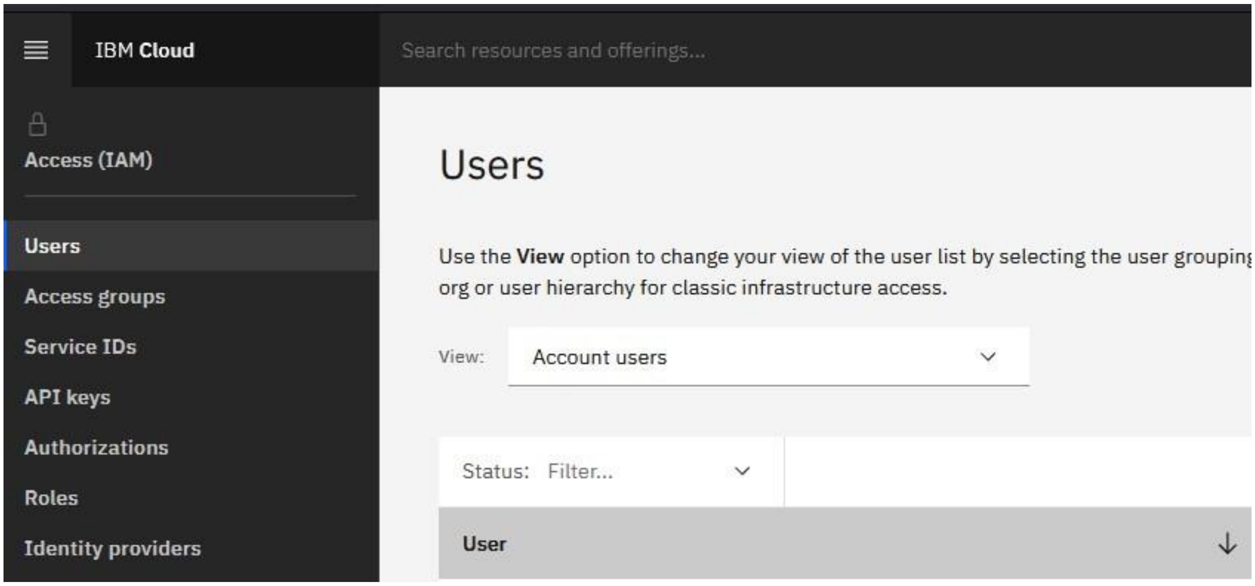
Note: The generated IBM API key inherits all the privileges of the logged in user. If you're logged in with an 'owner' role user and then generated a key, then the generated API key will have full owner rights and if that is the case then you can skip the next set of steps.

However, if you're logged in as a regular user and created a key, then you need to assign some specific privileges for the user that are mentioned in the next section. Without those privileges for the current user, the generated key will not work correctly with your SWIFT server.

Set permissions of the IBM cloud user for SWIFT usage

Select the 'Users' option from the left side menu and then select the current user that generated the key.

Additional information can be found here.



Users

Use the **View** option to change your view of the user list by selecting the user grouping org or user hierarchy for classic infrastructure access.

View: Account users

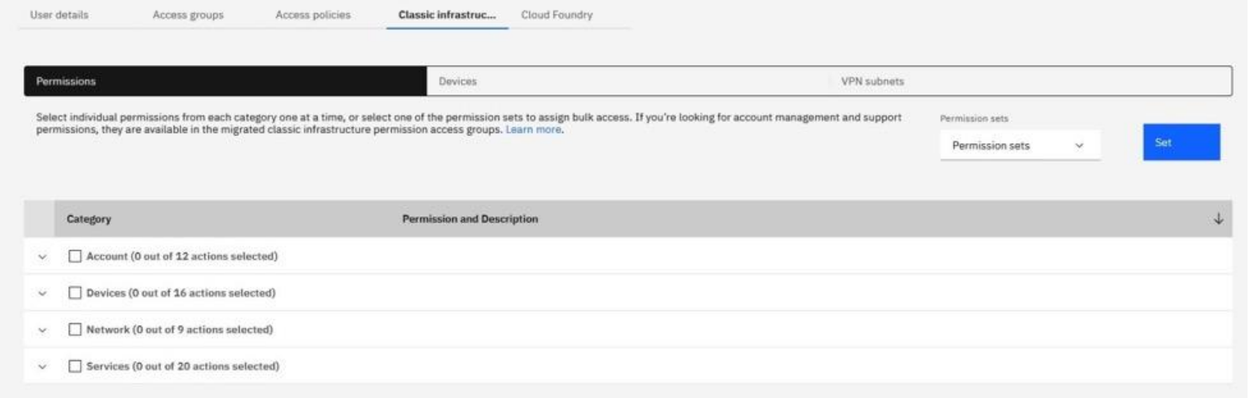
Status: Filter...

User ↓

Add the Classic Infrastructure permissions

To assign the classic infrastructure permissions to the user, select the 'Classic Infrastructure' tab.

Additional information can be found here.

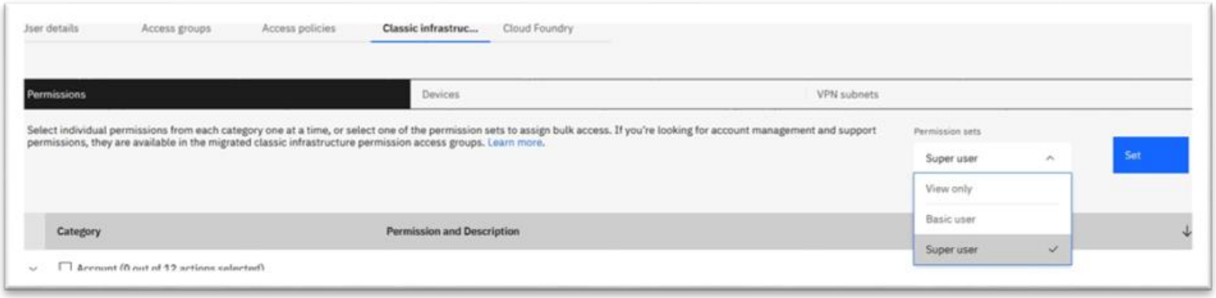


Select individual permissions from each category one at a time, or select one of the permission sets to assign bulk access. If you're looking for account management and support permissions, they are available in the migrated classic infrastructure permission access groups. Learn more.

Category	Permission and Description
□ Account (0 out of 12 actions selected)	
□ Devices (0 out of 16 actions selected)	
□ Network (0 out of 9 actions selected)	
□ Services (0 out of 20 actions selected)	

There are 2 options for assigning classic infrastructure access:

1. Add super user access



Select individual permissions from each category one at a time, or select one of the permission sets to assign bulk access. If you're looking for account management and support permissions, they are available in the migrated classic infrastructure permission access groups. Learn more.

Permission sets: Super user ✓

□ Account (0 out of 12 actions selected)

2. Add individual permissions as below:

Under the 'Permissions' tab, expand and check the following permissions:

Account:

- Add/Upgrade Storage (Storage Layer)

Devices:

- Edit Hostname/Domain
- Manage Port Control

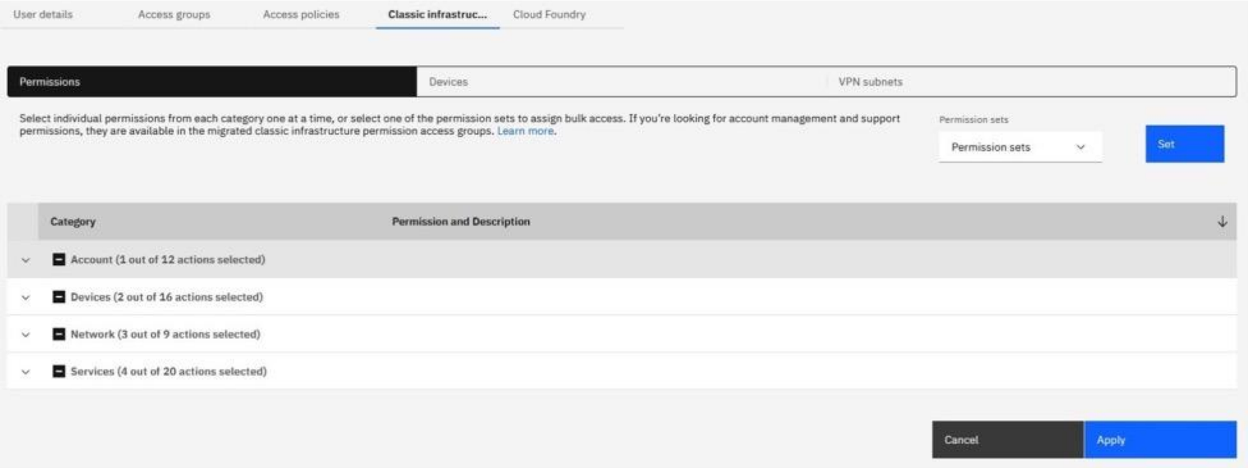
Network:

- Add IP Addresses
- Manage Network Subnet Routes
- Add Compute with Public Network Port

Services:

- Manage DNS
- Manage Certificates (SSL)
- View Certificates (SSL)
- Storage Manage

After selecting the above permissions, click on the 'Apply' button.

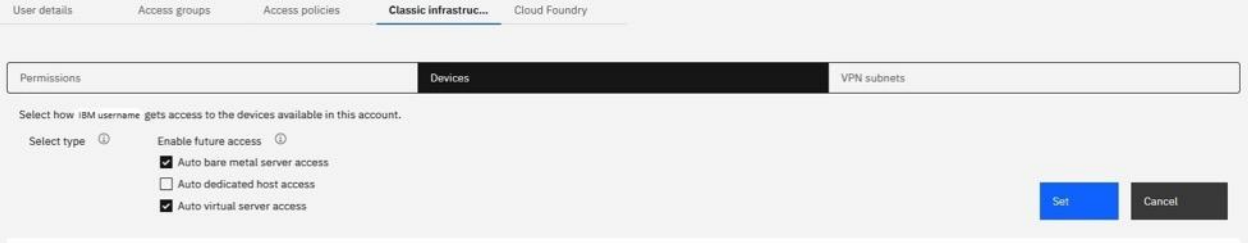


Select individual permissions from each category one at a time, or select one of the permission sets to assign bulk access. If you're looking for account management and support permissions, they are available in the migrated classic infrastructure permission access groups. Learn more.

Permission sets: Permission sets

Category	Permission and Description
Account (1 out of 12 actions selected)	
Devices (2 out of 16 actions selected)	
Network (3 out of 9 actions selected)	
Services (4 out of 20 actions selected)	

To add/check worker node specific permissions, go to the ‘Devices’ tab and make sure the following permissions are set.



Select how IBM username gets access to the devices available in this account.

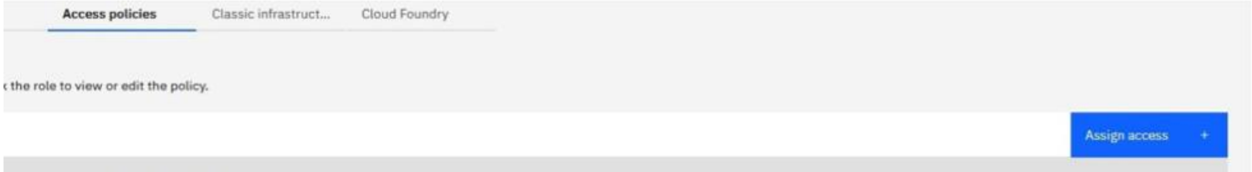
Select type ⓘ

Enable future access ⓘ

☑ Auto bare metal server access
☐ Auto dedicated host access
☑ Auto virtual server access

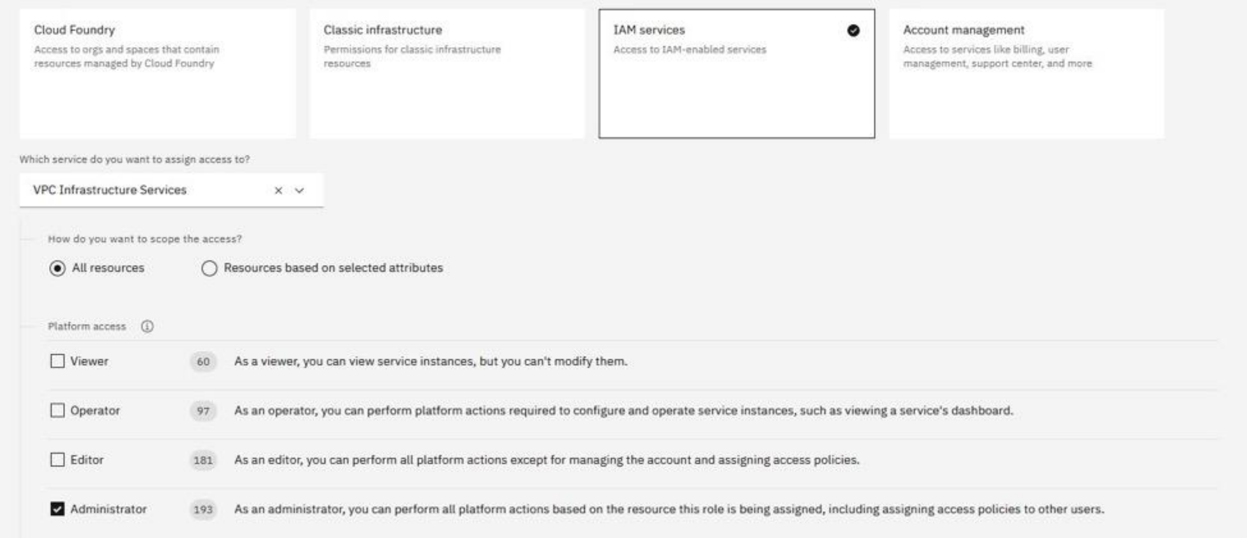
Add VPC Infrastructure permissions

If you have a VPC cluster then the user needs to have the ‘Administrator’ platform access. To assign the access, click on ‘Assign access’ button under the ‘Access policies’ tab.



‹ the role to view or edit the policy.

Search for ‘VPC Infrastructure Services’ and select the related drop-down option. Select the ‘Administrator’ role and click on the ‘Add’ button below.



Cloud Foundry
Access to orgs and spaces that contain resources managed by Cloud Foundry

Classic infrastructure
Permissions for classic infrastructure resources

IAM services ●
Access to IAM-enabled services

Account management
Access to services like billing, user management, support center, and more

Which service do you want to assign access to?

VPC Infrastructure Services

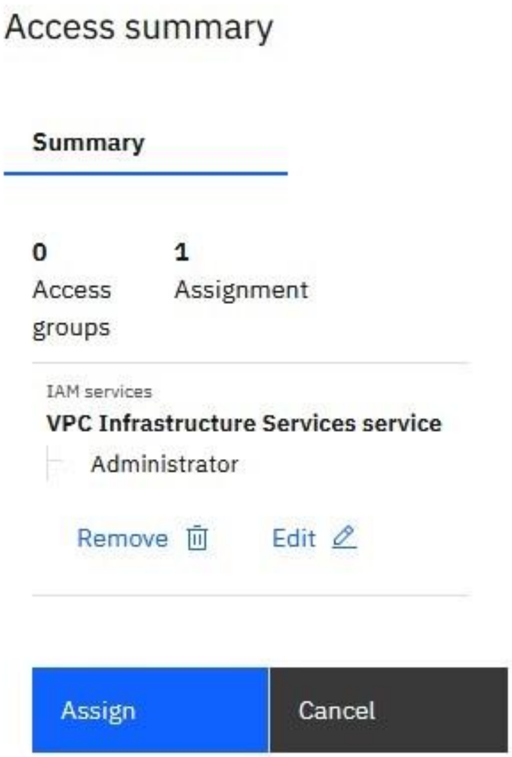
How do you want to scope the access?

● All resources ○ Resources based on selected attributes

Platform access

□ Viewer	60	As a viewer, you can view service instances, but you can't modify them.
□ Operator	97	As an operator, you can perform platform actions required to configure and operate service instances, such as viewing a service's dashboard.
□ Editor	181	As an editor, you can perform all platform actions except for managing the account and assigning access policies.
☑ Administrator	193	As an administrator, you can perform all platform actions based on the resource this role is being assigned, including assigning access policies to other users.

Click on the ‘Assign’ button on the right to assign the selected role.



Access summary

Summary

0 Access groups

1 Assignment

IAM services
VPC Infrastructure Services service
Administrator

Now you can add your IBM IKS or OpenShift cluster to SWIFT by using the username and generated API key.

Adding OpenShift cluster service-account for SWIFT use

Before you can add your local or cloud based OpenShift cluster to SWIFT and start managing it, you will need to have a cluster service account created with the necessary permissions.

Create a YAML for the new service account:

```
$ vi swift-admin-sa.yaml
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: swift-admin
  namespace: kube-system
```

Apply the YAML file

```
$ oc apply -f swift-admin-sa.yaml
```

Next, add the ‘cluster-admin’ role to the newly created account.

```
$ vi swift-admin-roles.yaml
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: swift-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
  - kind: ServiceAccount
    name: swift-admin
    namespace: kube-system
```

Apply the YAML file

```
$ oc apply -f swift-admin-roles.yaml
```

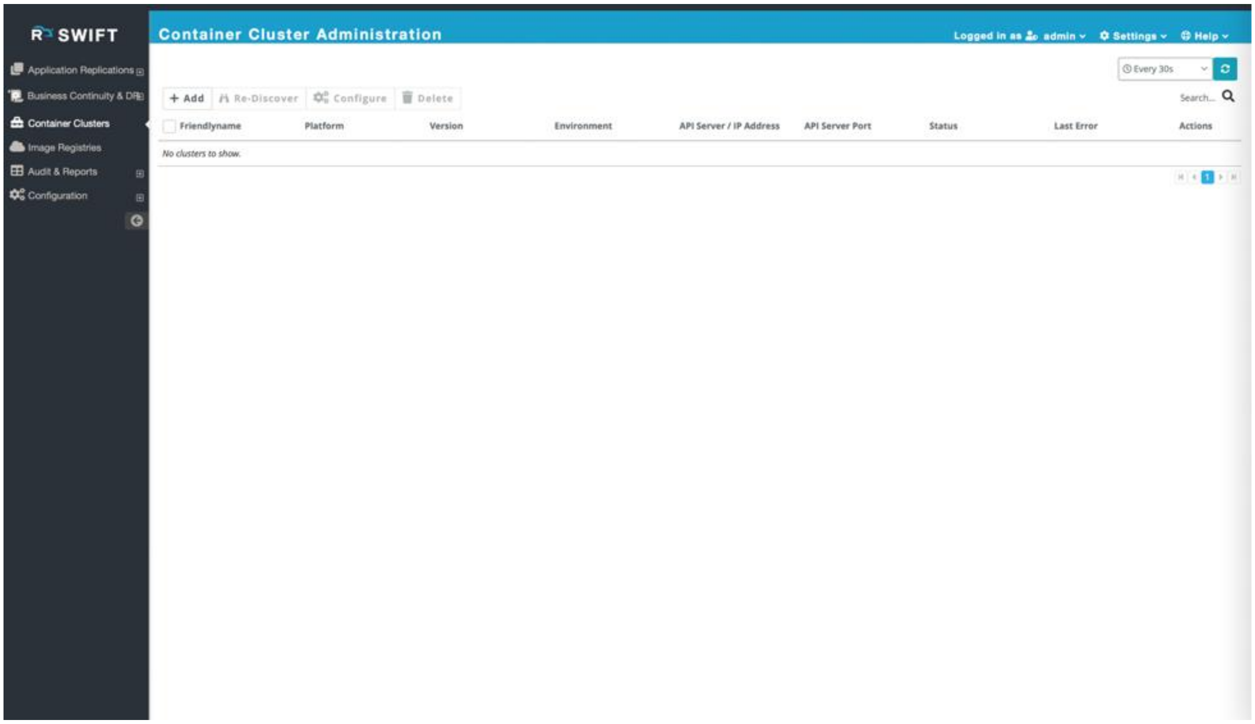
To get the service-account token, you can use a command as below. The command would print the ‘token’ key. You will use this output token later while adding the cluster to the SWIFT.

```
$ oc -n kube-system describe secret $(oc -n kube-system get secret | grep "swift-admin-token" | awk
'{print $1}')
```

Add new Kubernetes cluster details to SWIFT

SWIFT can work with different container clusters. Below sections highlight how to add a new container cluster under SWIFT management for various cloud types. The cloud credentials used will remain the same as what you will use for respective cloud container registry discovery and administration with the SWIFT. Please refer to cloud credentials generation section for detailed steps on how to generate cloud credentials with necessary permissions for using with SWIFT.

To add new cluster details to the SWIFT, login to the SWIFT dashboard first. Then click on the ‘Container Clusters’ menu.



Friendlyname	Platform	Version	Environment	API Server / IP Address	API Server Port	Status	Last Error	Actions

No clusters to show.

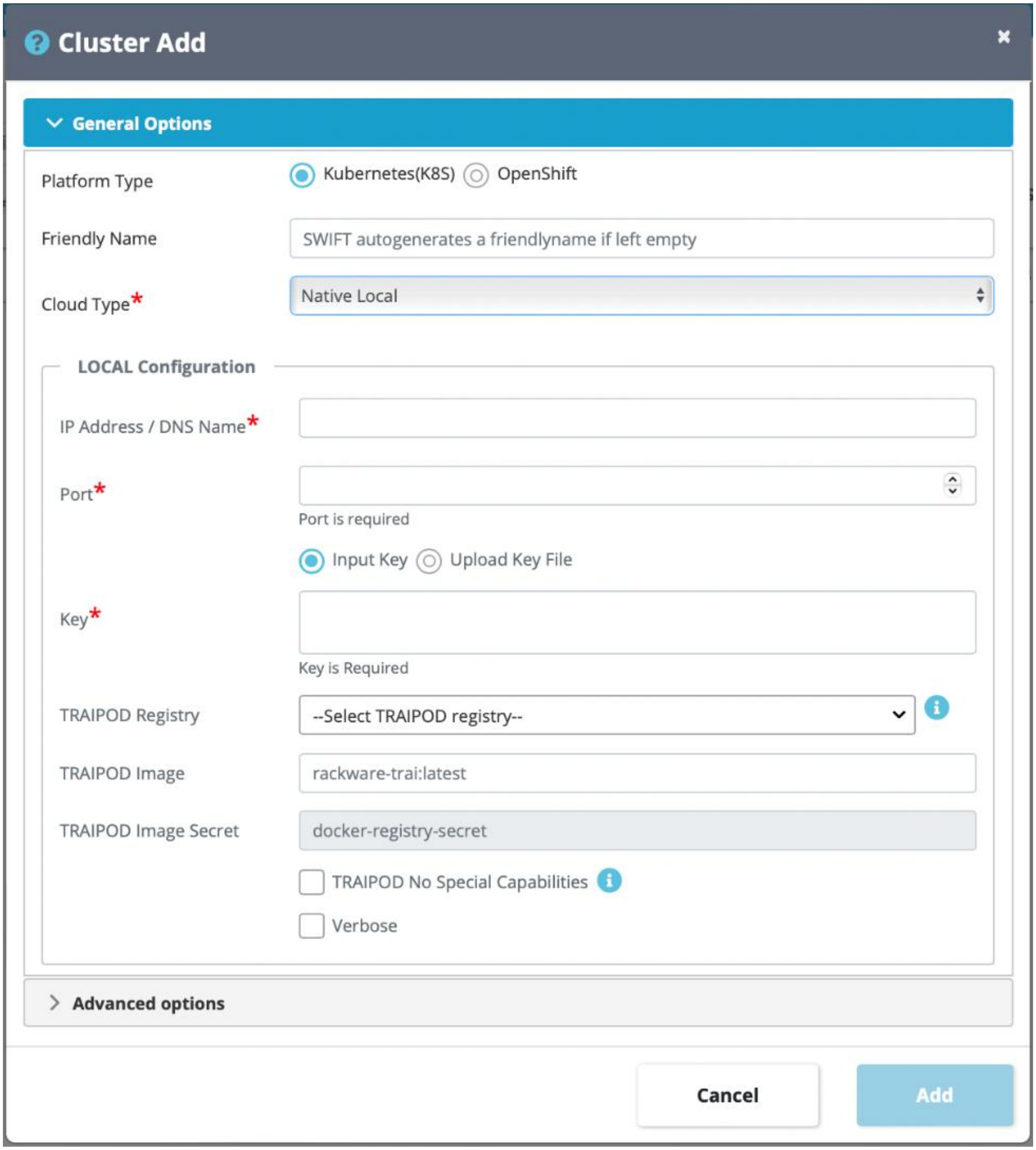
Click on the ‘+ Add’ button and then enter cluster friendlyname and other details. Choose the cloud-type for the cluster from the corresponding ‘Cloud Type’ drop-down.

The steps remain the same for other platforms like OpenShift clusters too.

The below sections highlight various cluster details to configure under the SWIFT depending on the cloud-type for the cluster.

Local Cluster

For a local cluster, you would need a service-account token with cluster-admin privileges.



Cluster Add

General Options

Platform Type: ● Kubernetes(K8S) ○ OpenShift

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Cloud Type*: Native Local

LOCAL Configuration

IP Address / DNS Name*:

Port*:

Port is required

● Input Key ○ Upload Key File

Key*:

Key is Required

TRAIPOD Registry: --Select TRAIPOD registry--

TRAIPOD Image: rackware-trai:latest

TRAIPOD Image Secret: docker-registry-secret

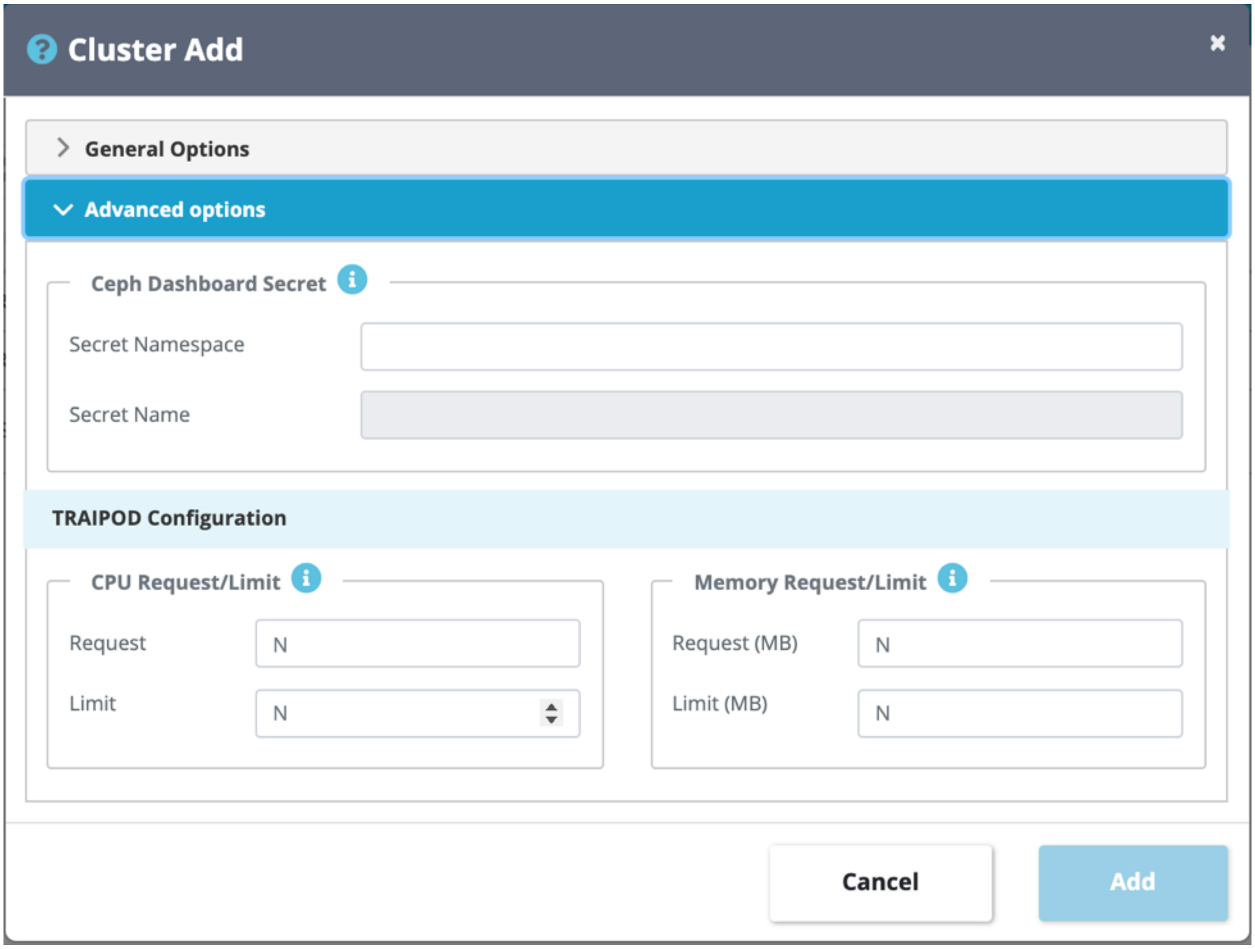
□ TRAIPOD No Special Capabilities

□ Verbose

Enter the IP address or name of the cluster API server along with the relevant port. The key would be your service account token (you can refer to the earlier section for creating such a service account).

TRAIPOD details can be filled here or later through cluster configuration menu. You will enter TRAI image name and secret or select discovered Image Registry where SWIFT will upload TRAI image and orchestrate secret creation as part of a sync.

You can optionally set some advanced options here, or later from the cluster configuration menu. These advanced options are some of the sync defaults used for syncs to/from this cluster.



Cluster Add

General Options

Advanced options

Ceph Dashboard Secret

Secret Namespace

Secret Name

TRAIPOD Configuration

CPU Request/Limit

Request: N

Limit: N

Memory Request/Limit

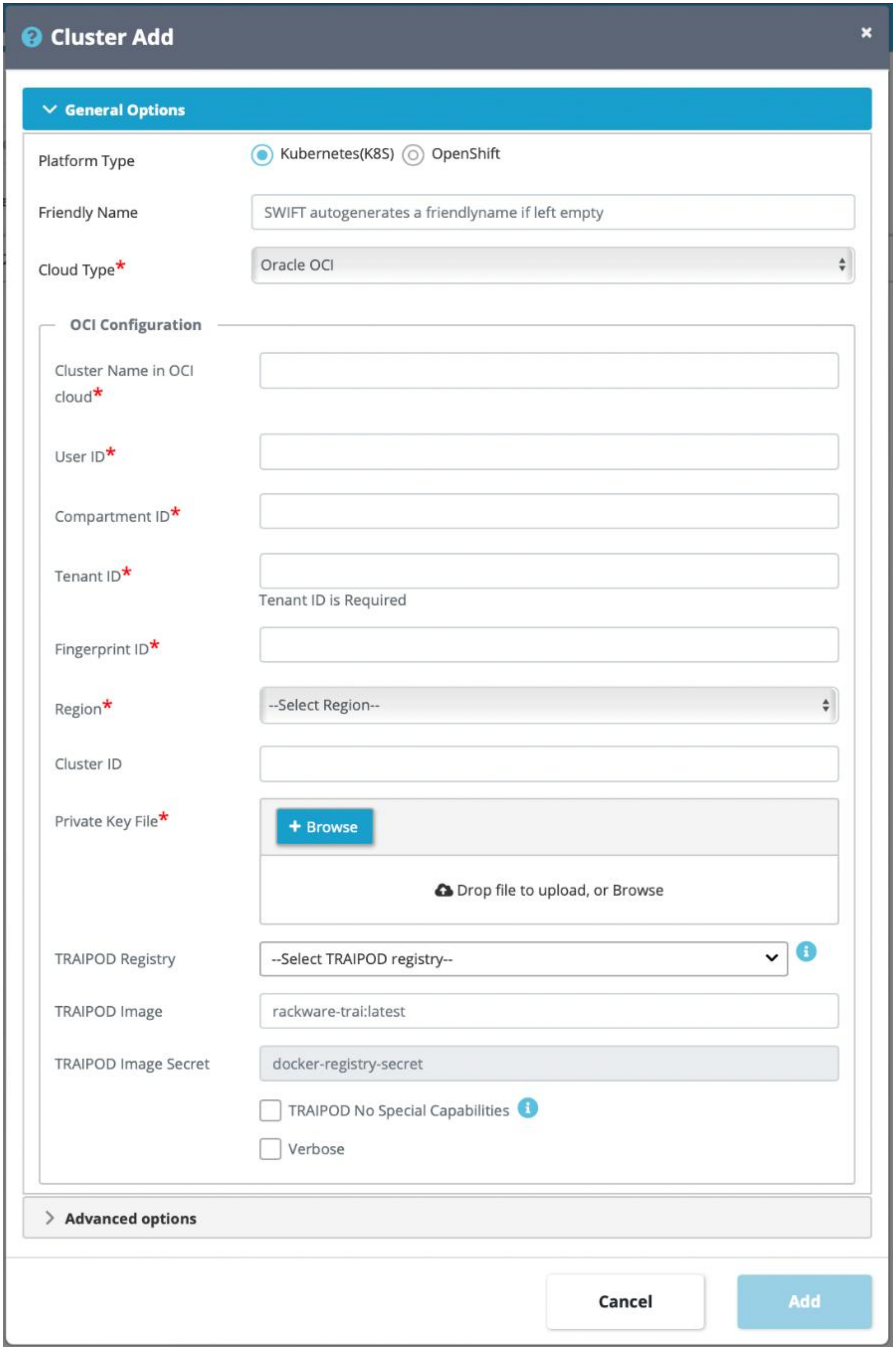
Request (MB): N

Limit (MB): N

Oracle OKE Cluster

For an Oracle OKE cluster, you will need the following values for the Oracle Cloud Infrastructure (OCI) account to configure the cluster in the SWIFT:

1. Tenant id
2. Compartment id
3. An API key with fingerprint
4. Region of the cluster
5. Cluster name



Cluster Add

General Options

Platform Type: ● Kubernetes(K8S) ○ OpenShift

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Cloud Type*: Oracle OCI

OCI Configuration

Cluster Name in OCI cloud*:

User ID*:

Compartment ID*:

Tenant ID*:

Tenant ID is Required

Fingerprint ID*:

Region*: --Select Region--

Cluster ID:

Private Key File*: + Browse

Drop file to upload, or Browse

TRAIPOD Registry: --Select TRAIPOD registry--

TRAIPOD Image: rackware-trai:latest

TRAIPOD Image Secret: docker-registry-secret

□ TRAIPOD No Special Capabilities

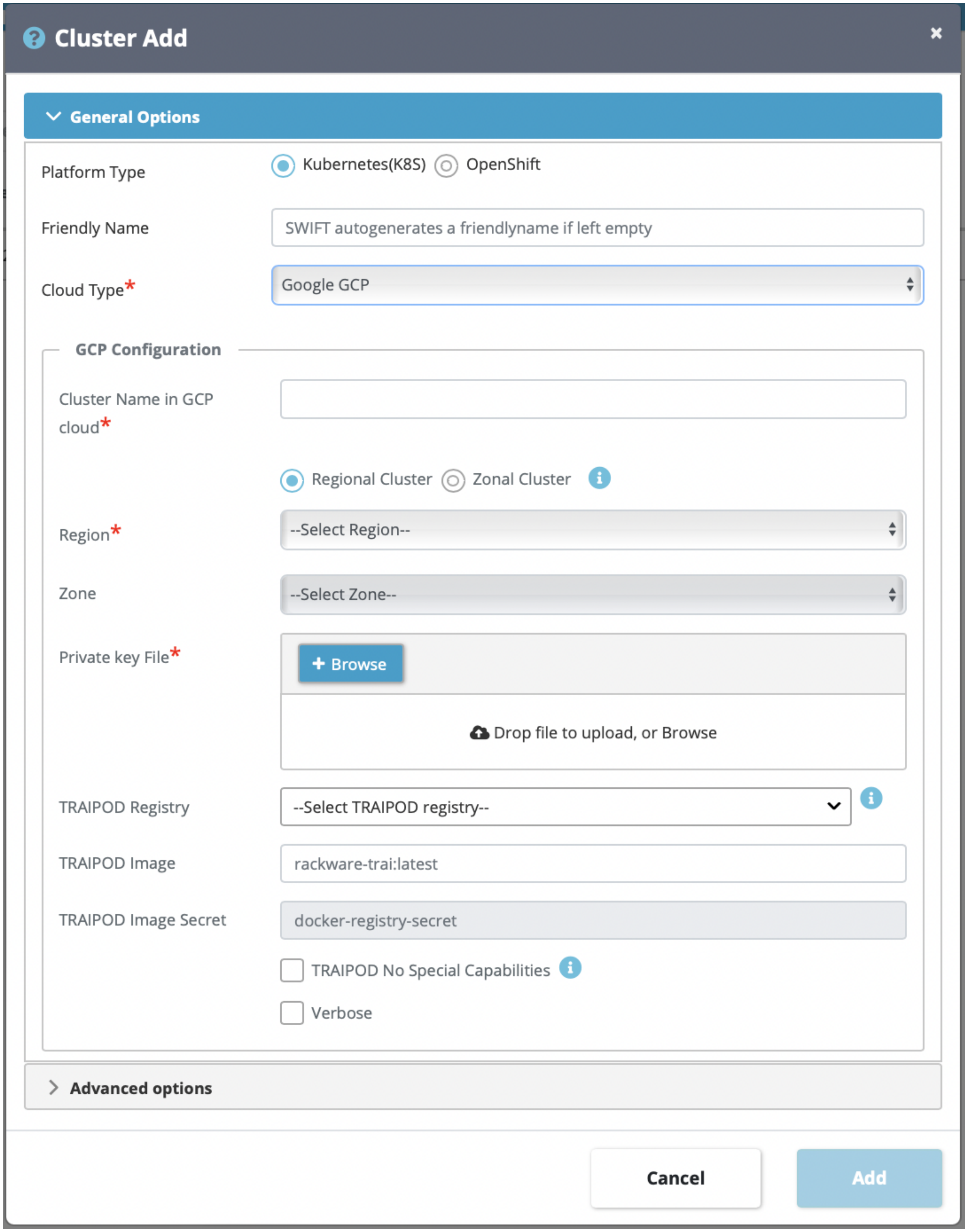
□ Verbose

Cluster id is an optional input. You can refer to the OCI documentation for generating an API key and fingerprint from here.

Google GKE Cluster and GCP OpenShift Clusters

For a Google cloud GKE cluster, you will need the following values for the Google cloud account to configure the cluster in the SWIFT:

1. Cluster name (Display name of the GKE cluster)
2. GCP Region
3. GCP Zone
4. The private key of the GCP service account



Cluster Add

General Options

Platform Type ● Kubernetes(K8S) ○ OpenShift

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Cloud Type*: Google GCP

GCP Configuration

Cluster Name in GCP cloud*:

● Regional Cluster ○ Zonal Cluster

Region*: --Select Region--

Zone: --Select Zone--

Private key File*: + Browse

Drop file to upload, or Browse

TRAIPOD Registry: --Select TRAIPOD registry--

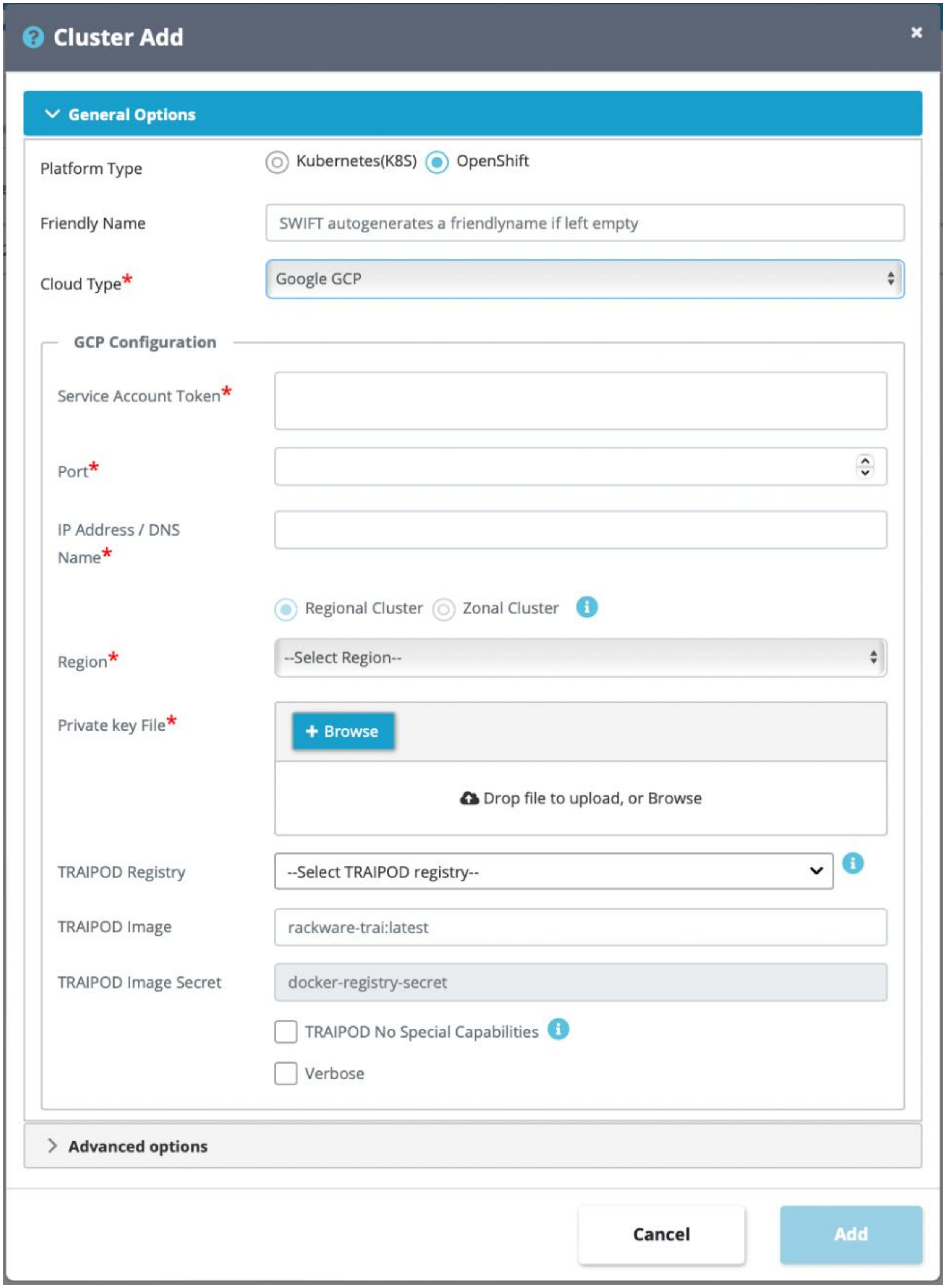
TRAIPOD Image: rackware-trai:latest

TRAIPOD Image Secret: docker-registry-secret

□ TRAIPOD No Special Capabilities

□ Verbose

A very similar input is needed for GCP based OpenShift clusters. Only difference is that you need to input both GCP credentials and service-account token. The OpenShift can be any variant like OpenShift Container platform, dedicated, or OKD.



Platform Type ○ Kubernetes(K8S) ● OpenShift

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Cloud Type*: Google GCP

GCP Configuration

Service Account Token*:

Port*:

IP Address / DNS Name*:

● Regional Cluster ○ Zonal Cluster

Region*: --Select Region--

Private key File*:

TRAIPOD Registry: --Select TRAIPOD registry--

TRAIPOD Image: rackware-trai:latest

TRAIPOD Image Secret: docker-registry-secret

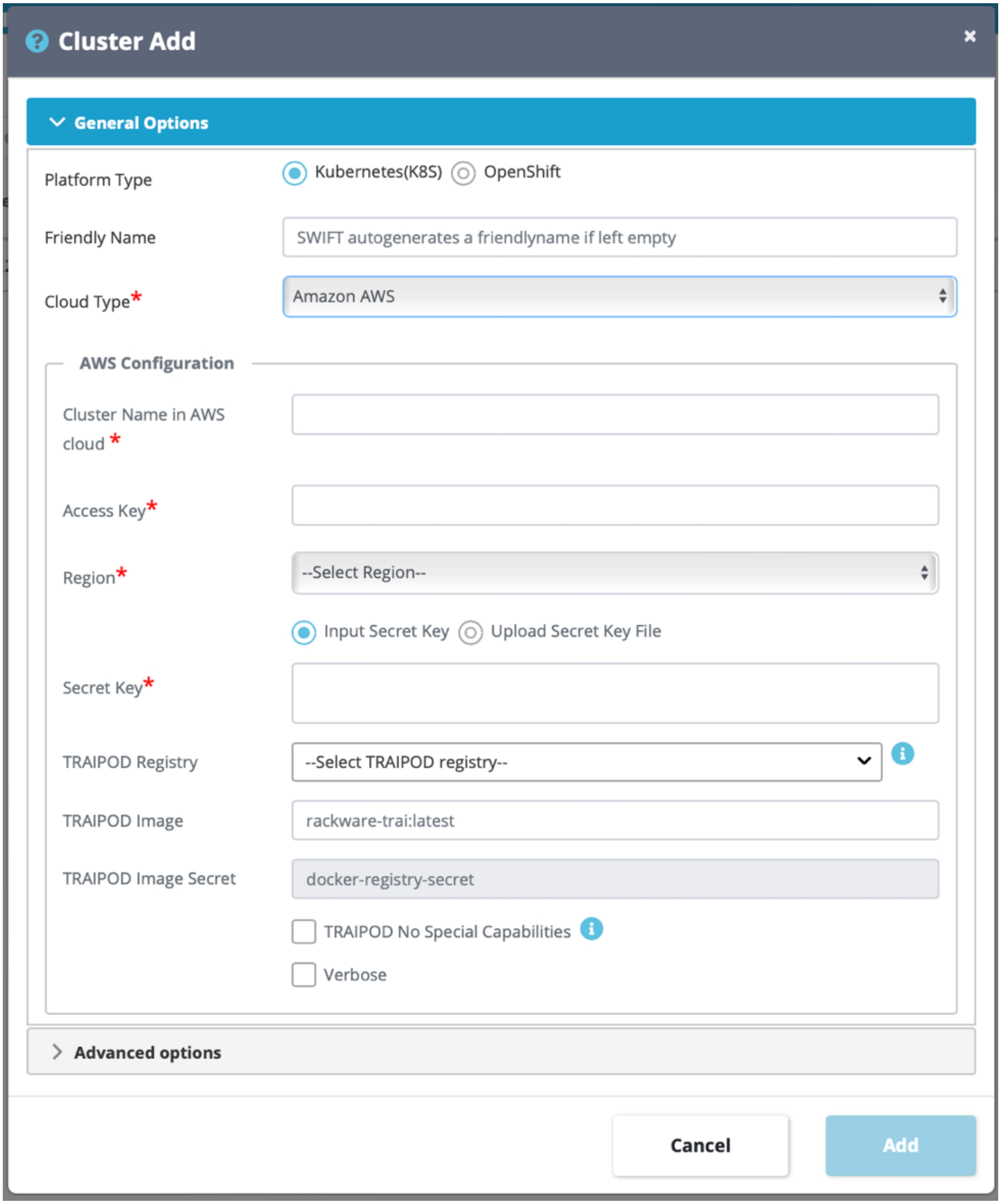
□ TRAIPOD No Special Capabilities

□ Verbose

Amazon EKS Cluster and Amazon OpenShift Cluster

For an Amazon EKS cluster, you will need the following values for the AWS account to configure the cluster in the SWIFT:

1. Cluster name (The display name of the EKS cluster)
2. Access-key id for AWS account
3. Secret access key for AWS account
4. AWS Region



Cluster Add

General Options

Platform Type ● Kubernetes(K8S) ○ OpenShift

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Cloud Type*: Amazon AWS

AWS Configuration

Cluster Name in AWS cloud *:

Access Key*:

Region*: --Select Region--

● Input Secret Key ○ Upload Secret Key File

Secret Key*:

TRAIPOD Registry: --Select TRAIPOD registry--

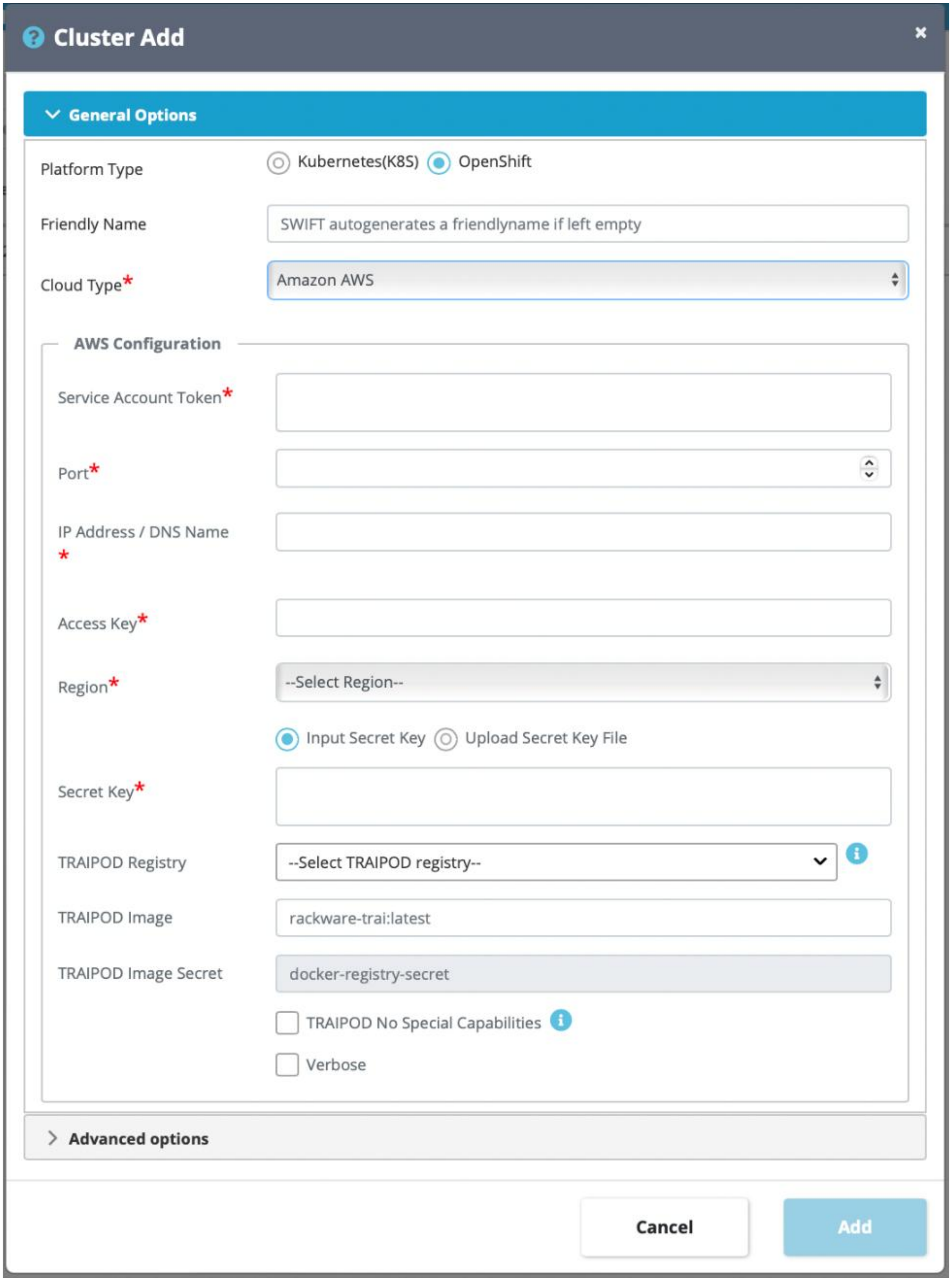
TRAIPOD Image: rackware-trai:latest

TRAIPOD Image Secret: docker-registry-secret

□ TRAIPOD No Special Capabilities

□ Verbose

A very similar input is needed for AWS based OpenShift clusters. Only difference is that you need to input both AWS credentials and service-account token. The OpenShift can be any variant like OpenShift Container platform, dedicated, or OKD.



Cluster Add

General Options

Platform Type ○ Kubernetes(K8S) ● OpenShift

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Cloud Type*: Amazon AWS

AWS Configuration

Service Account Token*:

Port*:

IP Address / DNS Name *:

Access Key*:

Region*: --Select Region--

● Input Secret Key ○ Upload Secret Key File

Secret Key*:

TRAIPOD Registry: --Select TRAIPOD registry--

TRAIPOD Image: rackware-trai:latest

TRAIPOD Image Secret: docker-registry-secret

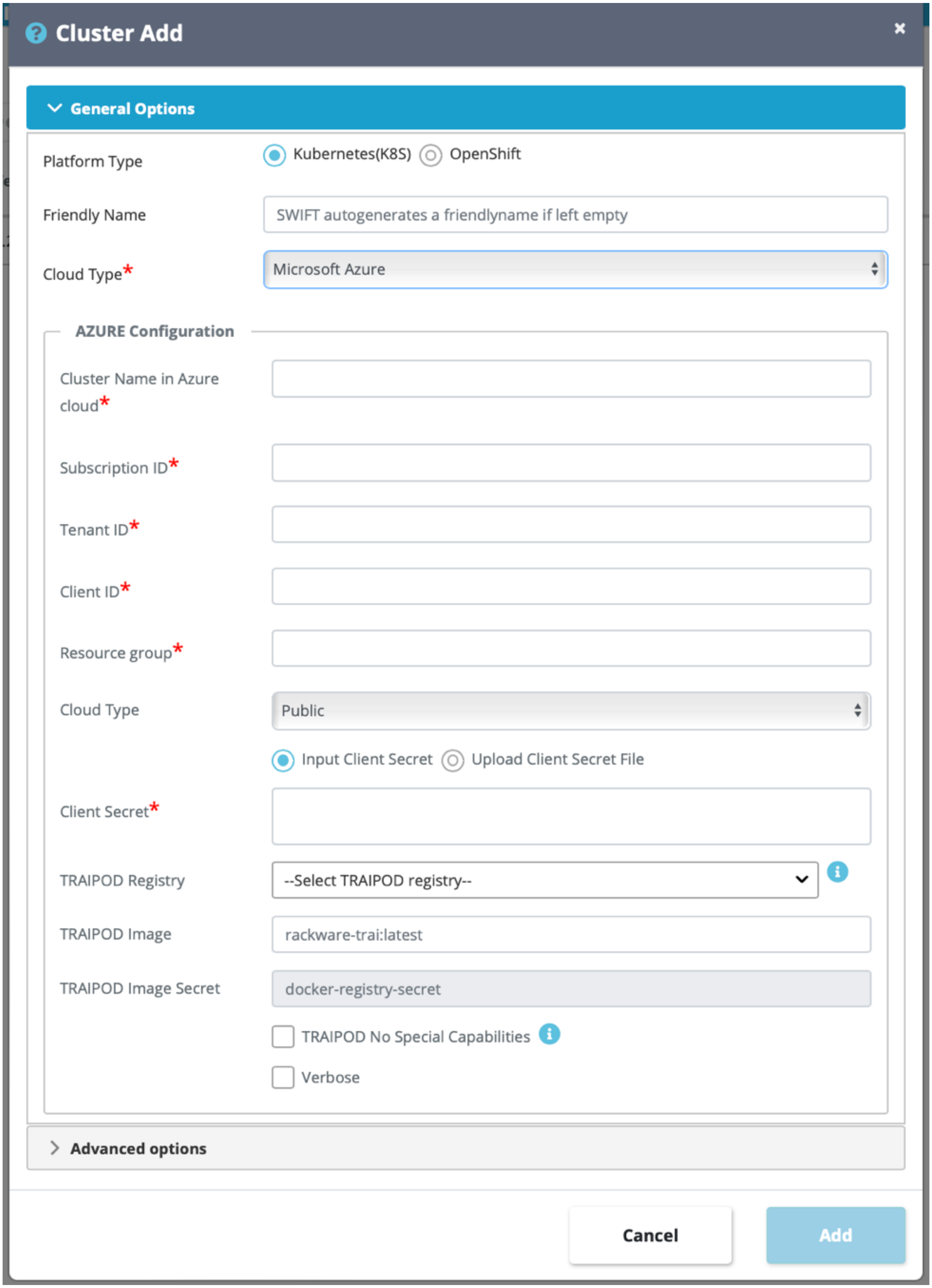
□ TRAIPOD No Special Capabilities

□ Verbose

Azure AKS Cluster and Azure OpenShift Cluster

For an Azure AKS cluster, you will need the following values for the Azure cloud account to configure the cluster in the SWIFT:

1. Cluster name (The display name of the AKS cluster)
2. Subscription id
3. Tenant id
4. App id (client id)
5. App secret (client secret)
6. Resource group name
7. Cloud type (Public/Government/China)



Cluster Add

General Options

Platform Type ● Kubernetes(K8S) ○ OpenShift

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Cloud Type*: Microsoft Azure

AZURE Configuration

Cluster Name in Azure cloud*:

Subscription ID*:

Tenant ID*:

Client ID*:

Resource group*:

Cloud Type: Public

● Input Client Secret ○ Upload Client Secret File

Client Secret*:

TRAIPOD Registry: --Select TRAIPOD registry--

TRAIPOD Image: rackware-trai:latest

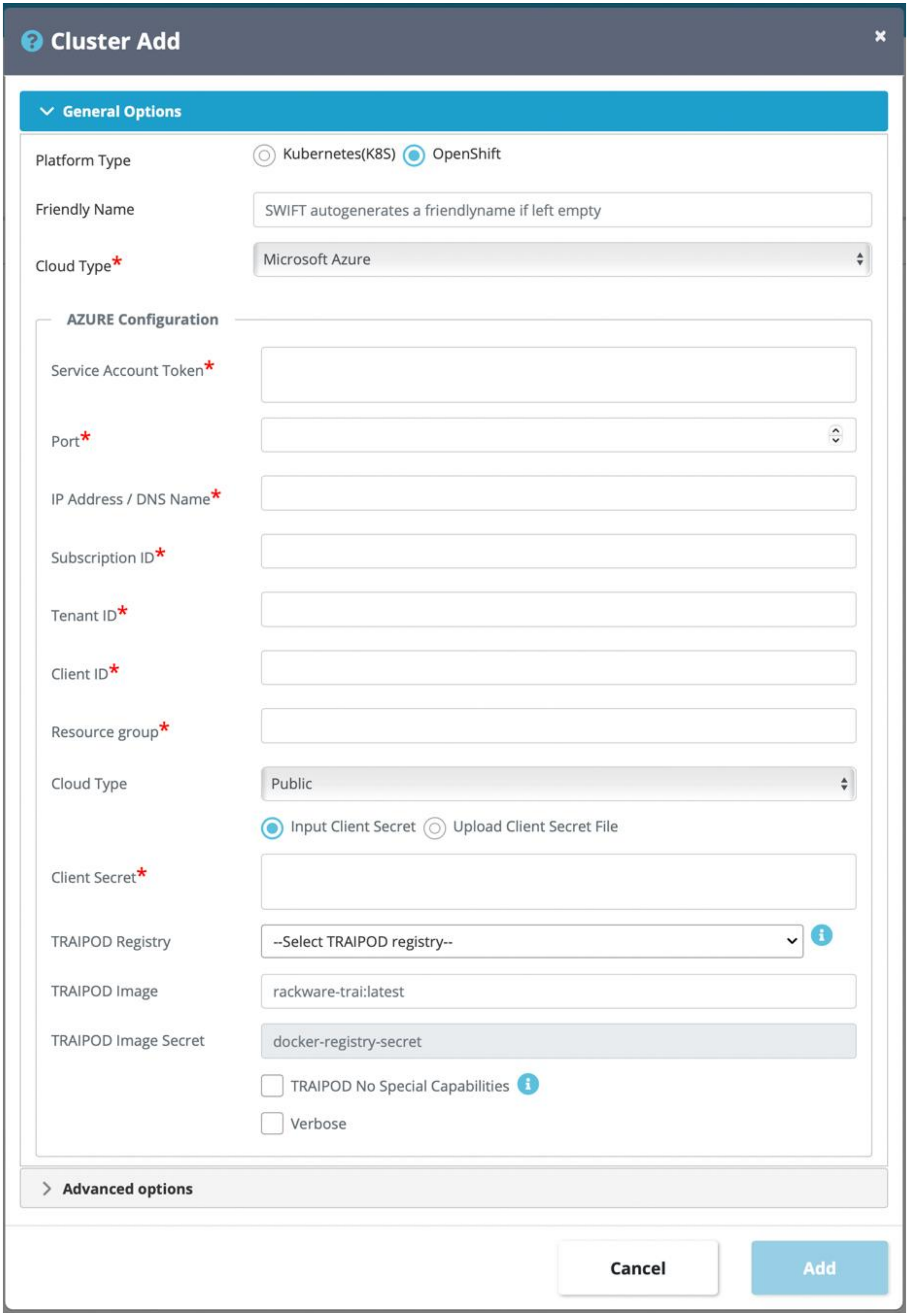
TRAIPOD Image Secret: docker-registry-secret

□ TRAIPOD No Special Capabilities

□ Verbose

Advanced options

A very similar input is needed for Azure based OpenShift clusters. Only difference is that you need to input both Azure credentials and service-account token. The OpenShift can be any variant like OpenShift Container platform, dedicated, or OKD.



Cluster Add

General Options

Platform Type ○ Kubernetes(K8S) ● OpenShift

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Cloud Type*: Microsoft Azure

AZURE Configuration

Service Account Token*:

Port*:

IP Address / DNS Name*:

Subscription ID*:

Tenant ID*:

Client ID*:

Resource group*:

Cloud Type: Public

● Input Client Secret ○ Upload Client Secret File

Client Secret*:

TRAIPOD Registry: --Select TRAIPOD registry--

TRAIPOD Image: rackware-trai:latest

TRAIPOD Image Secret: docker-registry-secret

□ TRAIPOD No Special Capabilities

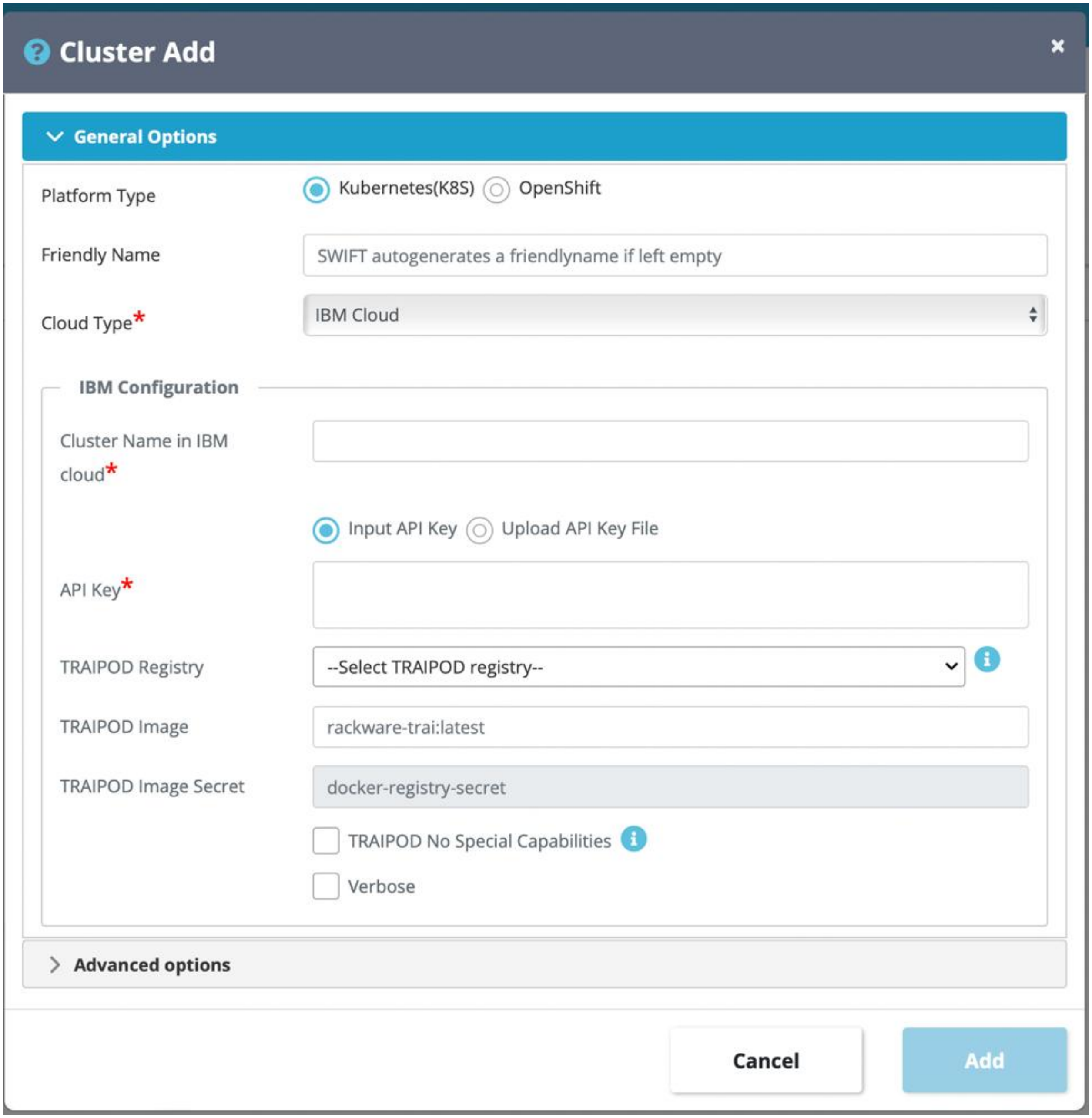
□ Verbose

Advanced options

IBM Kubernetes Service (IKS) Cluster and IBM OpenShift clusters

For an Azure AKS cluster, you will need the following values for the Azure cloud account to configure the cluster in the SWIFT:

1. Cluster name (The display name of the AKS cluster)
2. API key



Cluster Add

General Options

Platform Type ● Kubernetes(K8S) ○ OpenShift

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Cloud Type*: IBM Cloud

IBM Configuration

Cluster Name in IBM cloud*:

● Input API Key ○ Upload API Key File

API Key*:

TRAIPOD Registry: --Select TRAIPOD registry--

TRAIPOD Image: rackware-trai:latest

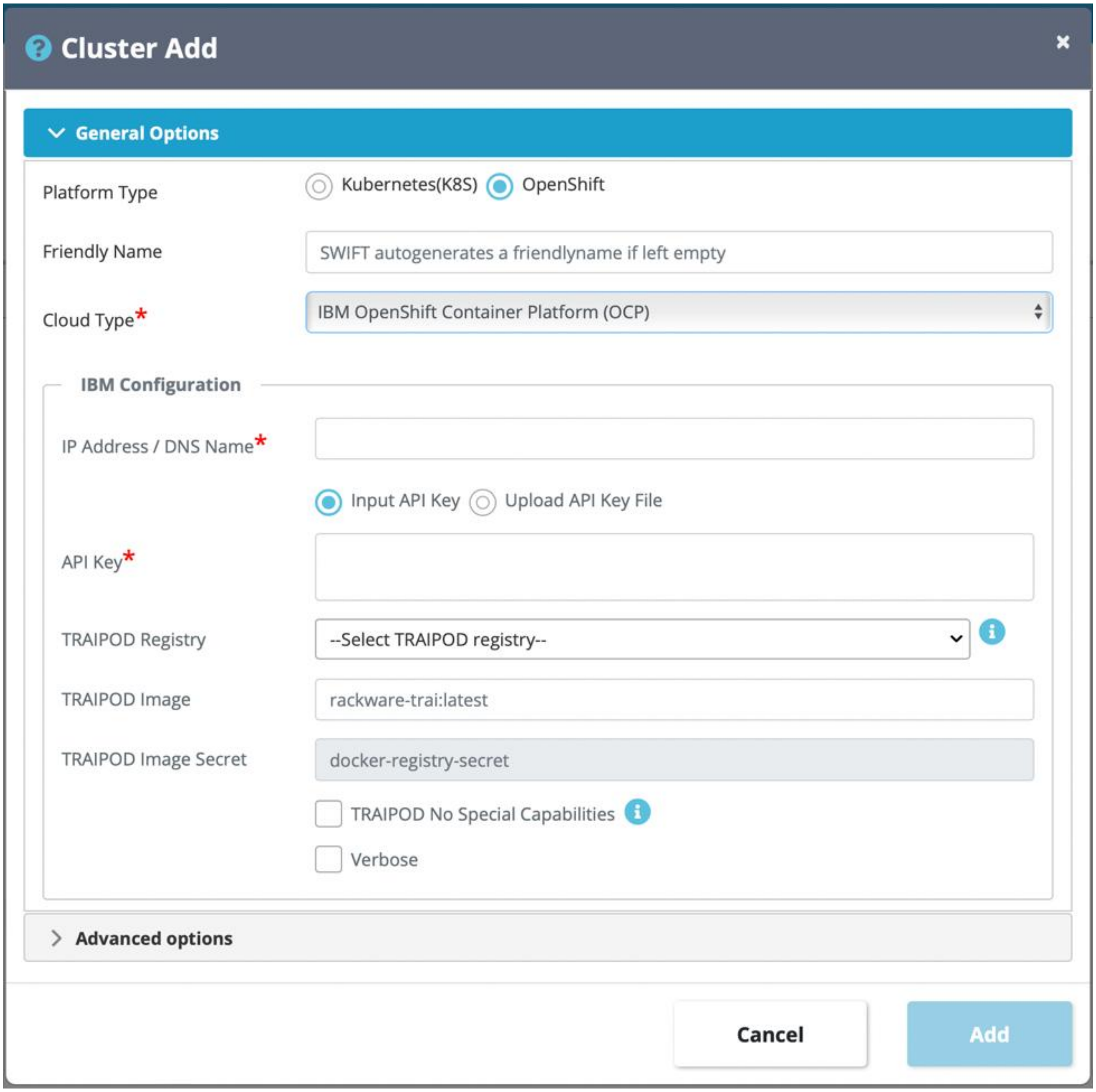
TRAIPOD Image Secret: docker-registry-secret

□ TRAIPOD No Special Capabilities

□ Verbose

Advanced options

A very similar input is needed for IBM OpenShift clusters. Only difference is that you need to input both IBM API key and cluster service-account token.



Cluster Add

General Options

Platform Type ○ Kubernetes(K8S) ● OpenShift

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Cloud Type*: IBM OpenShift Container Platform (OCP)

IBM Configuration

IP Address / DNS Name*:

● Input API Key ○ Upload API Key File

API Key*:

TRAIPOD Registry: --Select TRAIPOD registry--

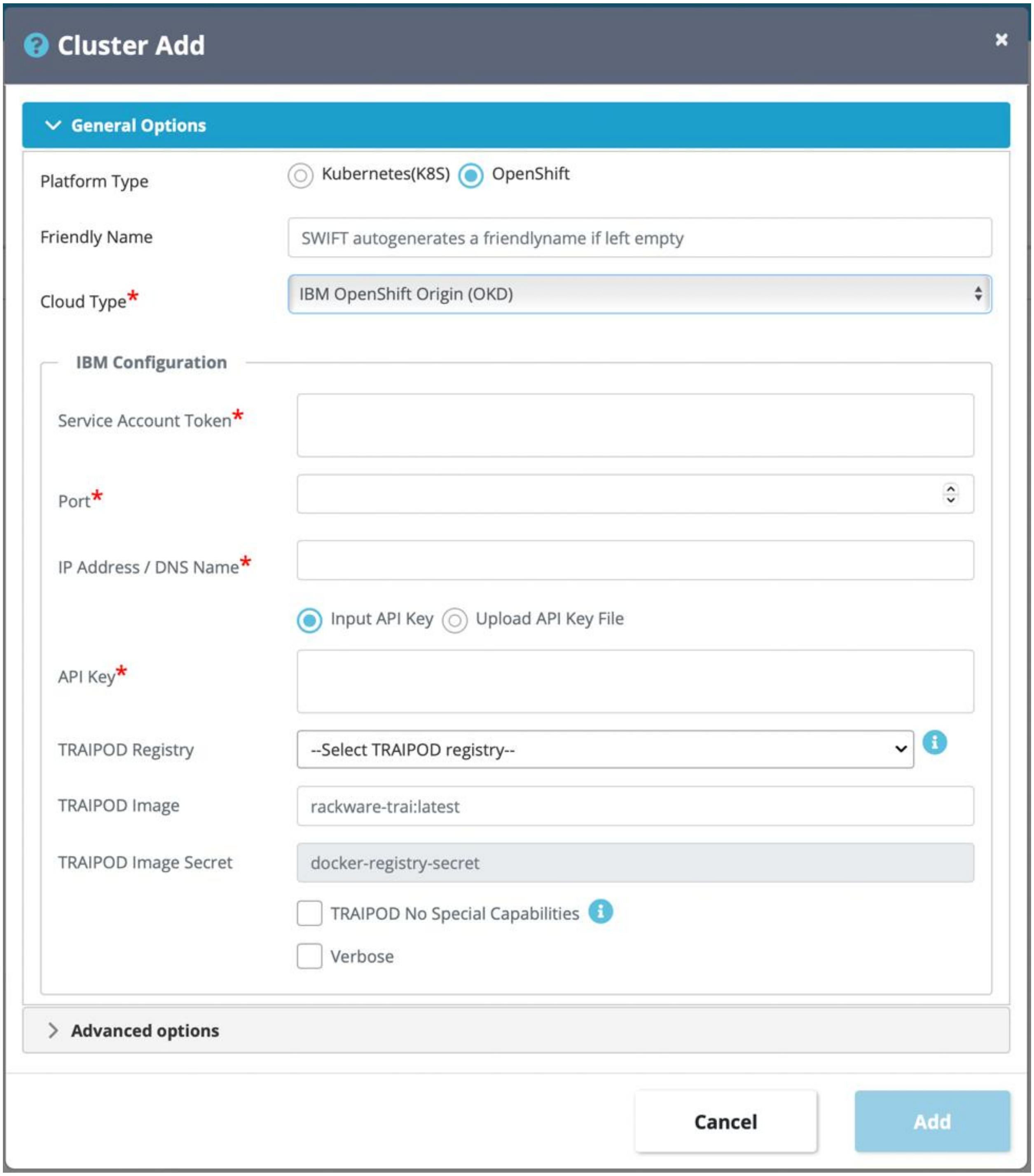
TRAIPOD Image: rackware-trai:latest

TRAIPOD Image Secret: docker-registry-secret

□ TRAIPOD No Special Capabilities

□ Verbose

For IBM cloud, you can also optionally deploy OpenShift Origins (OKD) that is opensource variant of the OpenShift. IF you have deployed Origins, then make sure you select its type correctly during SWIFT's cluster add step. The cluster add inputs remain almost the same for Origins as IBM OpenShift container platform (OCP), only additional input is Service Account (SA) token with admin rights.



Cluster Add

General Options

Platform Type ○ Kubernetes(K8S) ● OpenShift

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Cloud Type*: IBM OpenShift Origin (OKD)

IBM Configuration

Service Account Token*:

Port*:

IP Address / DNS Name*:

● Input API Key ○ Upload API Key File

API Key*:

TRAIPOD Registry: --Select TRAIPOD registry--

TRAIPOD Image: rackware-trai:latest

TRAIPOD Image Secret: docker-registry-secret

□ TRAIPOD No Special Capabilities

□ Verbose

Other Common Inputs

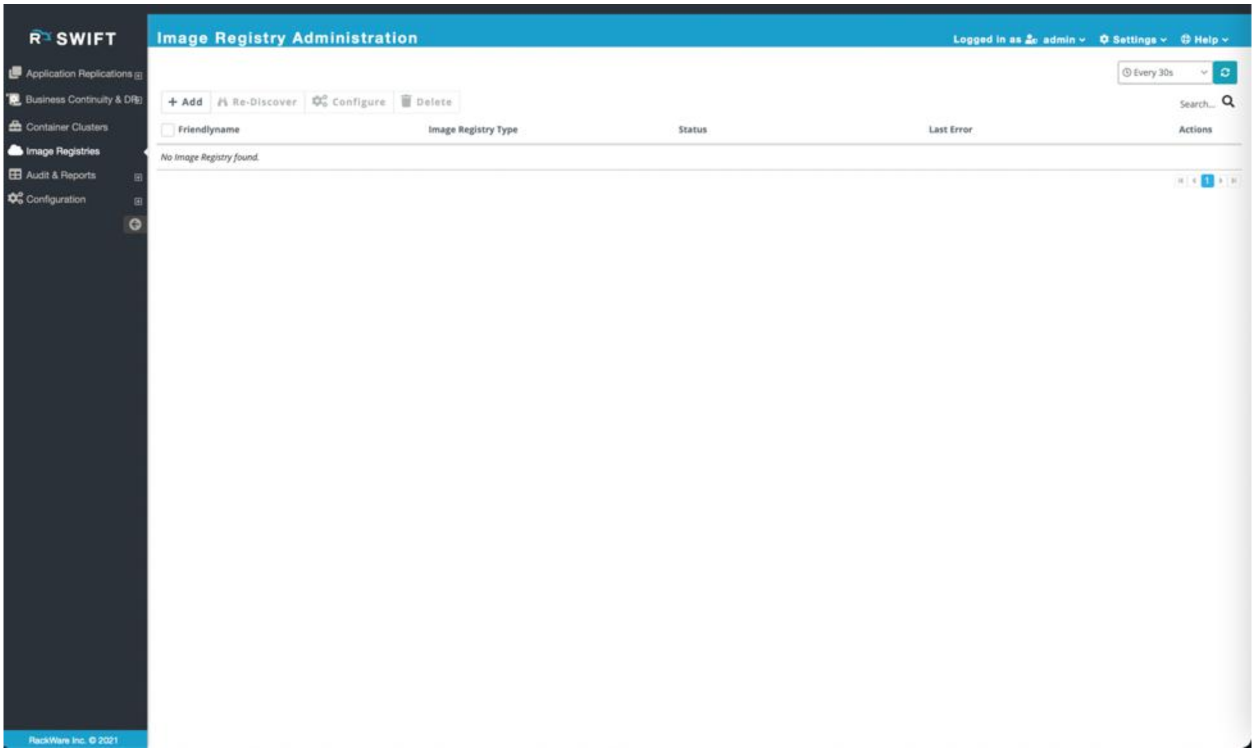
Transient RackWare Agent Image POD (TRAIPOD) Image and Image-Secret defaults are optional inputs. They can be configured for each sync, and the values for the sync will override the defaults set for the cluster. These defaults can be changed at any time using the ‘Configure’ button from the ‘K8S

Administration' page. Please see the TRAI section for more details on what this image is and how it is used during syncs.

Add new Image Registry details to SWIFT

SWIFT can work with different container image registries. Below sections highlight how to add a new container registry under SWIFT management for various cloud types. The cloud credentials used will remain the same as what you will use for respective cloud cluster discovery and administration with the SWIFT. Please refer to cloud credentials generation section for detailed steps on how to generate cloud credentials with necessary permissions for using with SWIFT.

To add new image registry details to the SWIFT, login to the SWIFT dashboard first. Then click on the 'Image Registries' menu.



Friendlyname	Image Registry Type	Status	Last Error	Actions
No Image Registry found.				

Click on the '+ Add' button and then enter registry friendlyname and other details. Choose the cloud-type for the cluster from the corresponding 'Cloud Type' drop-down.

The below sections now highlight various registry details to configure under the SWIFT depending on the cloud or registry-type.

Amazon Elastic Container Registry (ECR)

For Amazon registry, you will need the following values for the AWS account to configure the registry in the SWIFT:

1. Access-key id for AWS account
2. Secret access key for AWS account

3. AWS Region

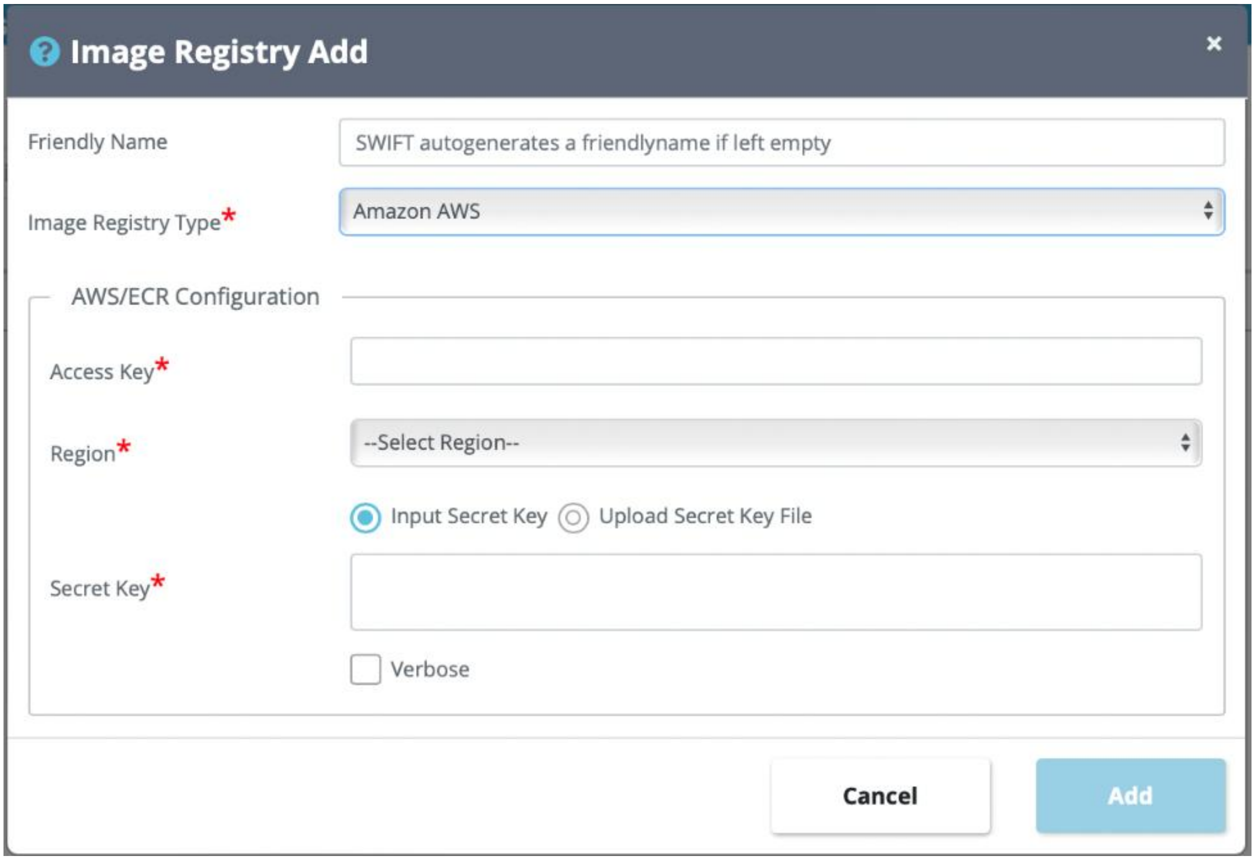


Image Registry Add

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Image Registry Type*: Amazon AWS

AWS/ECR Configuration

Access Key*:

Region*: --Select Region--

● Input Secret Key ○ Upload Secret Key File

Secret Key*:

□ Verbose

Azure Container Registry (ACR)

For an Azure ACR registry, you will need the following values for the Azure cloud account to configure the registry in the SWIFT:

1. Subscription id
2. Tenant id
3. App id (client id)
4. App secret (client secret)
5. Resource group name
6. Cloud type (Public/Government/China)
7. ECR registry display name in Azure
8. ECR registry password in Azure

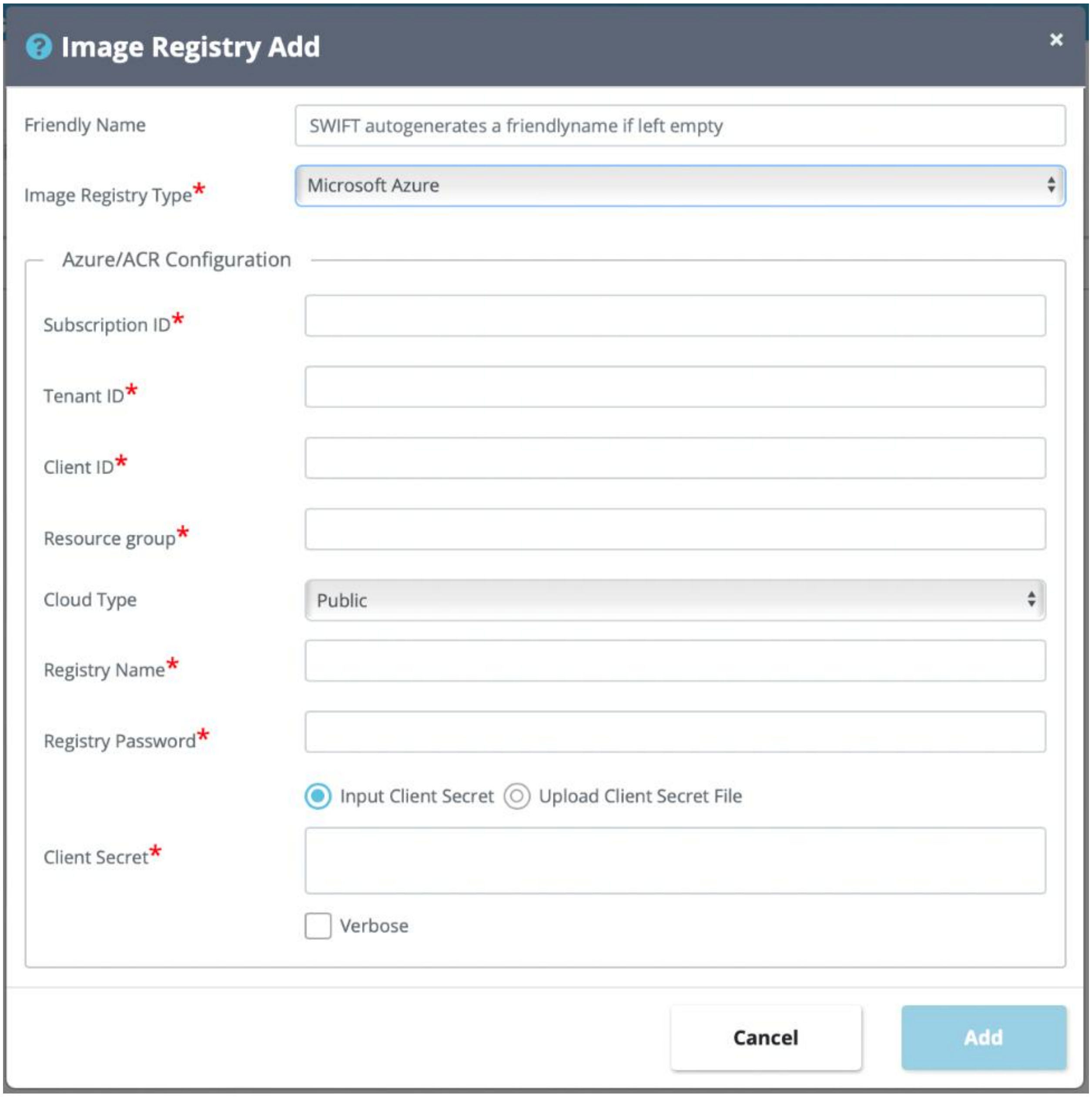


Image Registry Add

Friendly Name	SWIFT autogenerates a friendlyname if left empty
Image Registry Type*	Microsoft Azure

Azure/ACR Configuration

Subscription ID*	
Tenant ID*	
Client ID*	
Resource group*	
Cloud Type	Public
Registry Name*	
Registry Password*	
	● Input Client Secret ○ Upload Client Secret File
Client Secret*	
	□ Verbose

Oracle Cloud Infrastructure Container Registry (OCIR)

For Oracle OCIR registry, you will need the following values for the Oracle Cloud Infrastructure (OCI) cloud account to configure the registry in the SWIFT:

1. Tenant id
2. Compartment id
3. An API key with fingerprint
4. Region of the registry
5. OCI user’s id (OCI Id)

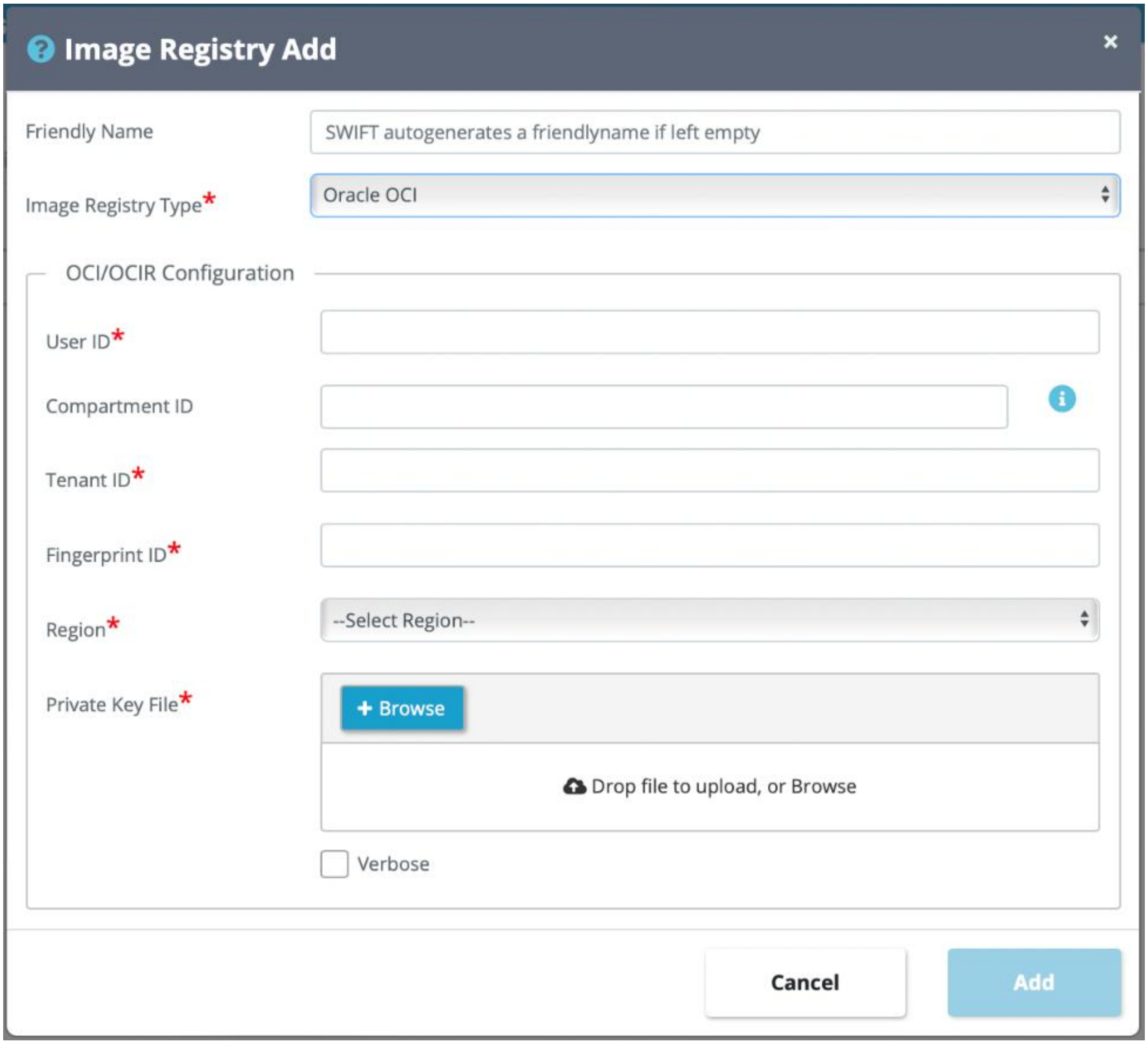


Image Registry Add

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Image Registry Type*: Oracle OCI

OCI/OCIR Configuration

User ID*

Compartment ID

Tenant ID*

Fingerprint ID*

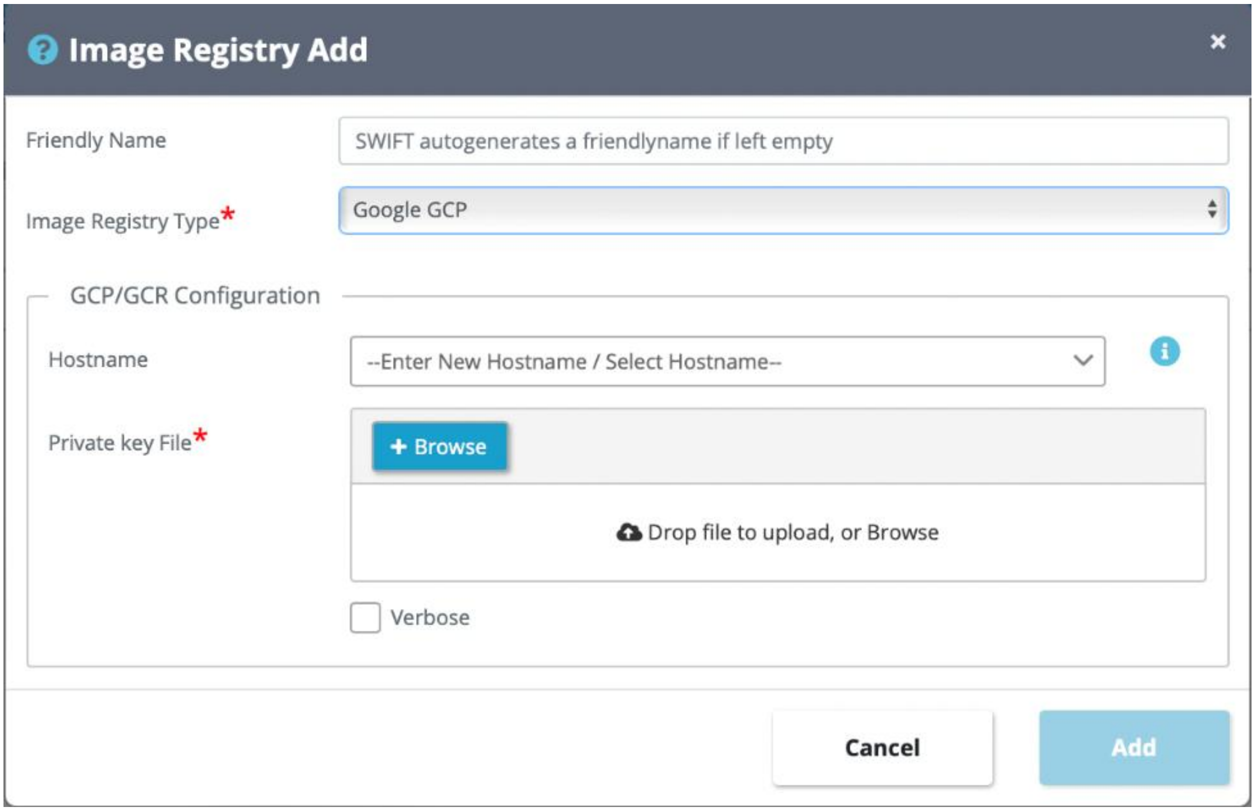
Region*: --Select Region--

Private Key File*

Google Container Registry (GCR)

For Google Container Registry (GCR), you will need the following values for the Google Cloud Platform (GCP) account to configure the registry in the SWIFT:

1. GCR hostname
2. The private key of the GCP service account



Friendly Name	SWIFT autogenerates a friendlyname if left empty
Image Registry Type*	Google GCP

GCP/GCR Configuration

Hostname	--Enter New Hostname / Select Hostname--
Private key File*	

□ Verbose

Docker Hub Container Registry

For Docker Hub Container Registry, you will need the following values for the Docker account to configure the registry in the SWIFT:

1. Docker account username
2. Docker account password

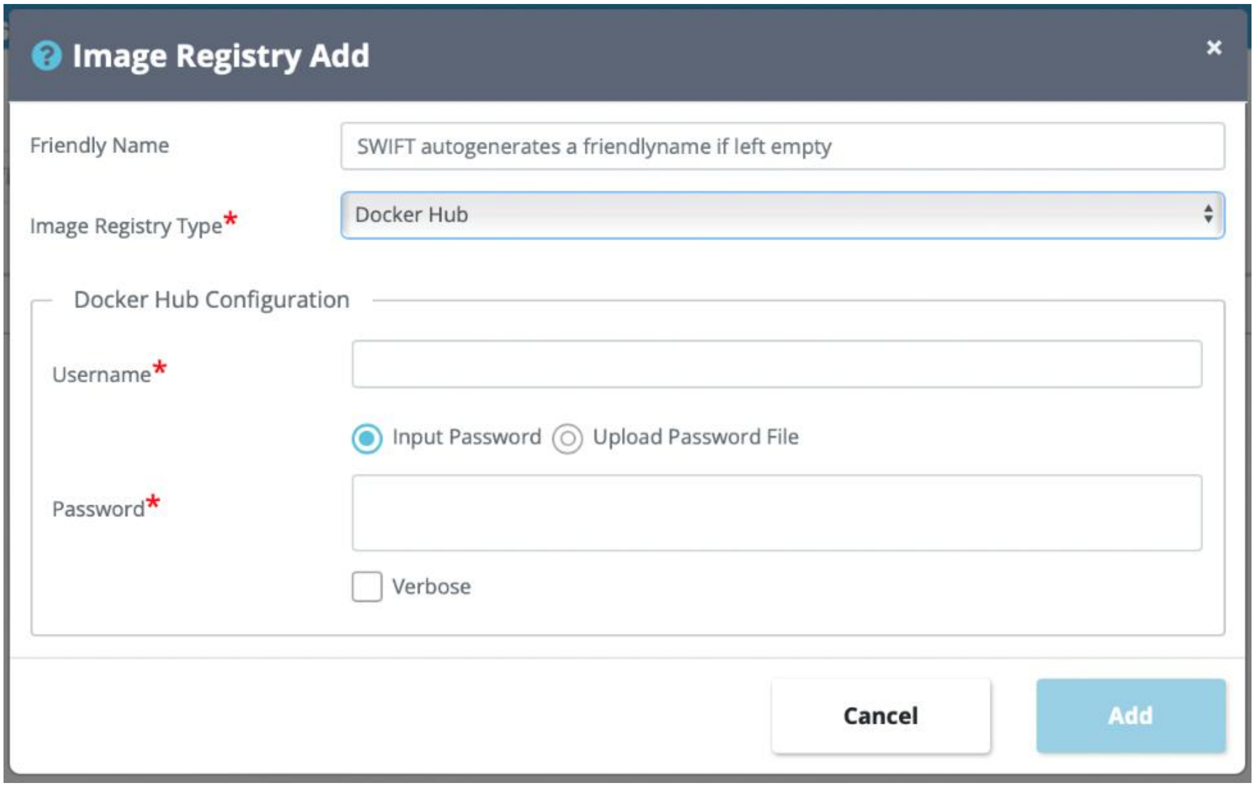


Image Registry Add

Friendly Name: SWIFT autogenerates a friendlyname if left empty

Image Registry Type*: Docker Hub

Docker Hub Configuration

Username*:

● Input Password ○ Upload Password File

Password*:

□ Verbose

Configuring Storage details for SWIFT’s use

If K8S/OpenShift cluster is using flex/native volumes from external storage server, then it needs to be configured so that SWIFT can connect to it and use it for snapshotting volumes.

Ceph Storage

If Ceph is deployed

- On K8S/OpenShift cluster, deploy the Ceph dashboard and expose it using either ‘NodePort’ or ‘LoadBalancer’. You can refer this link for deploying and exposing the Ceph dashboard.
- Outside K8S/OpenShift cluster, refer this link to deploy the Ceph dashboard.

Once the Ceph dashboard is deployed, create a secret on K8S/ OpenShift with following command:

On K8S:

```
$ kubectl create secret generic <secret name> /
                --from-literal=dashboard-username=<username> /
                --from-literal=dashboard-password=<password> /
                --from-literal=dashboard-address=<DNS hostname or IP address> /
                [--from-literal=dashboard-port=<port>]
```

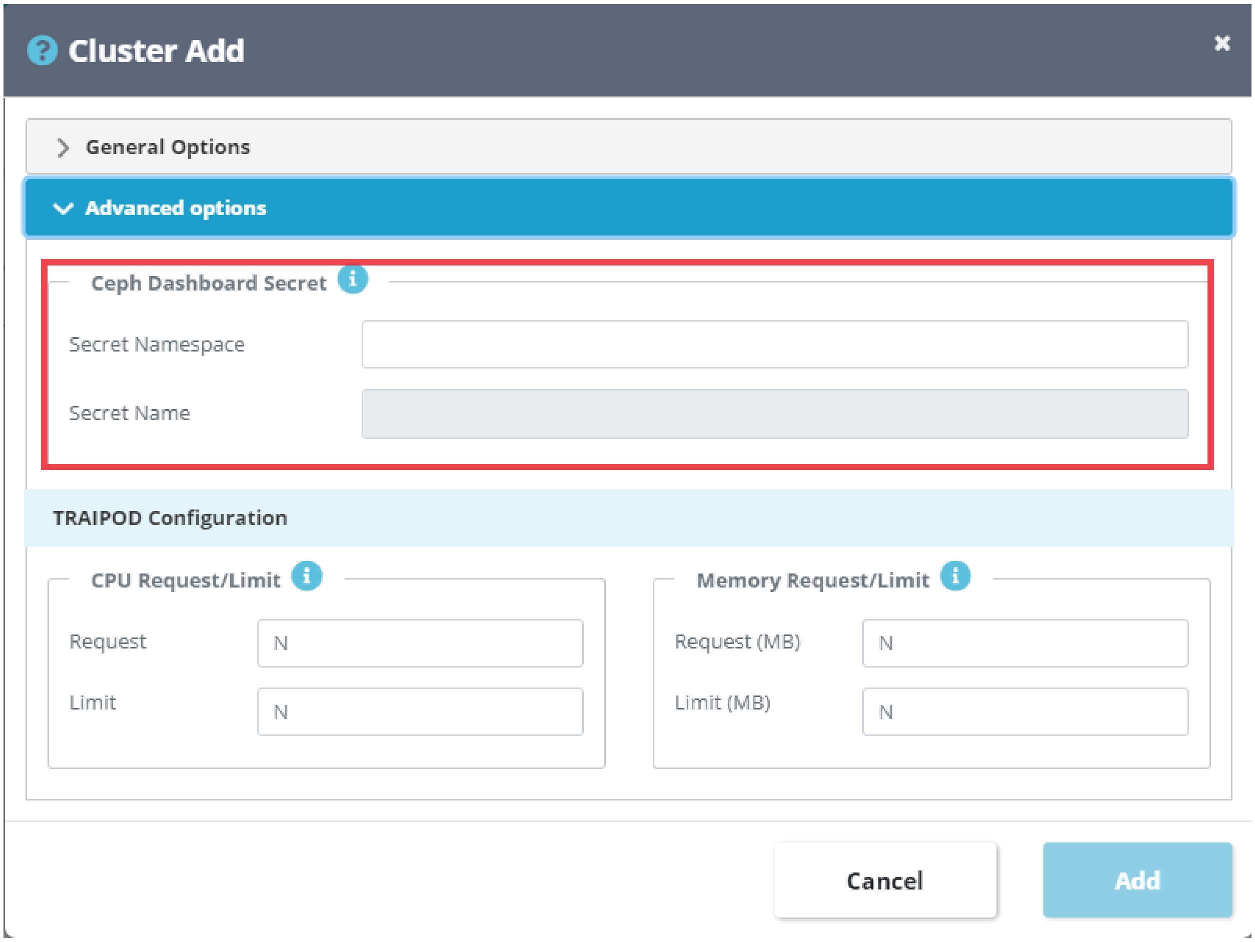
On OpenShift:

```
$ oc create secret generic <secret name> /
                    --from-literal=dashboard-username=<username> /
                    --from-literal=dashboard-password=<password> /
                    --from-literal=dashboard-address=<DNS hostname or IP address> /
                    [--from-literal=dashboard-port=<port>]
```

- <username>: username corresponds to any username from the Ceph dashboard having required privileges to perform CRUD operations for volumes.
- <password>: password of given username.
- <DNS hostname or IP address>: Address of the Ceph dashboard which should be accessible from SWIFT.
- <port>: Ceph dashboard port.

Note: The dashboard-port is optional if the Ceph dashboard is exposed using a K8S Ingress or OpenShift Route object.

Once the secret is created, provide its namespace and name during discovering the K8S/OpenShift cluster.



Cluster Add

General Options

Advanced options

Ceph Dashboard Secret

Secret Namespace

Secret Name

TRAIPOD Configuration

CPU Request/Limit

Request: N

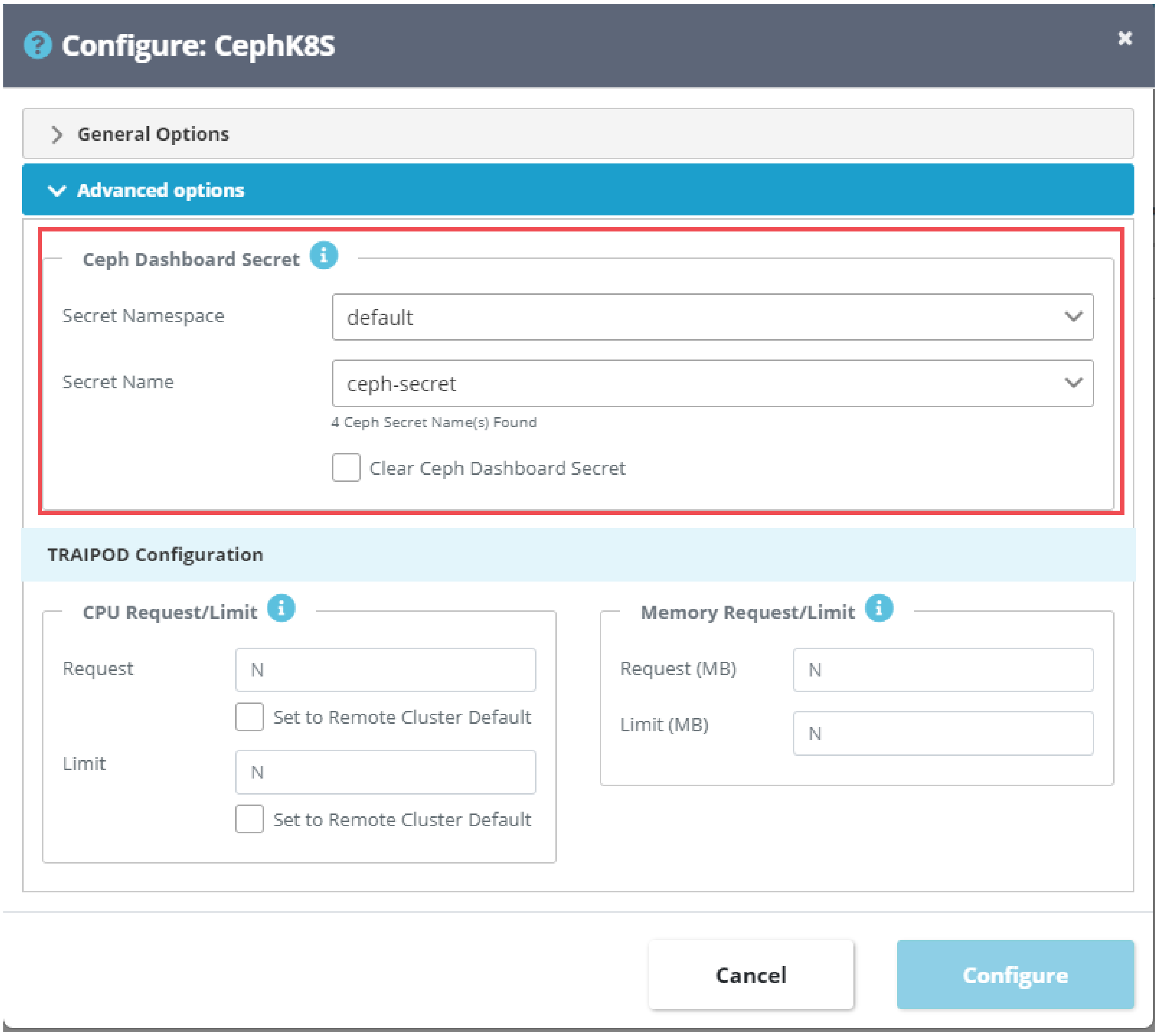
Limit: N

Memory Request/Limit

Request (MB): N

Limit (MB): N

To change the secret name, provide new secret name and namespace during configuring the cluster. Also, already existing Ceph secret can be cleared by ticking on the ‘Clear Ceph Dashboard Secret’ checkbox.



Secret Namespace: default

Secret Name: ceph-secret

4 Ceph Secret Name(s) Found

□ Clear Ceph Dashboard Secret

Request: N

□ Set to Remote Cluster Default

Limit: N

□ Set to Remote Cluster Default

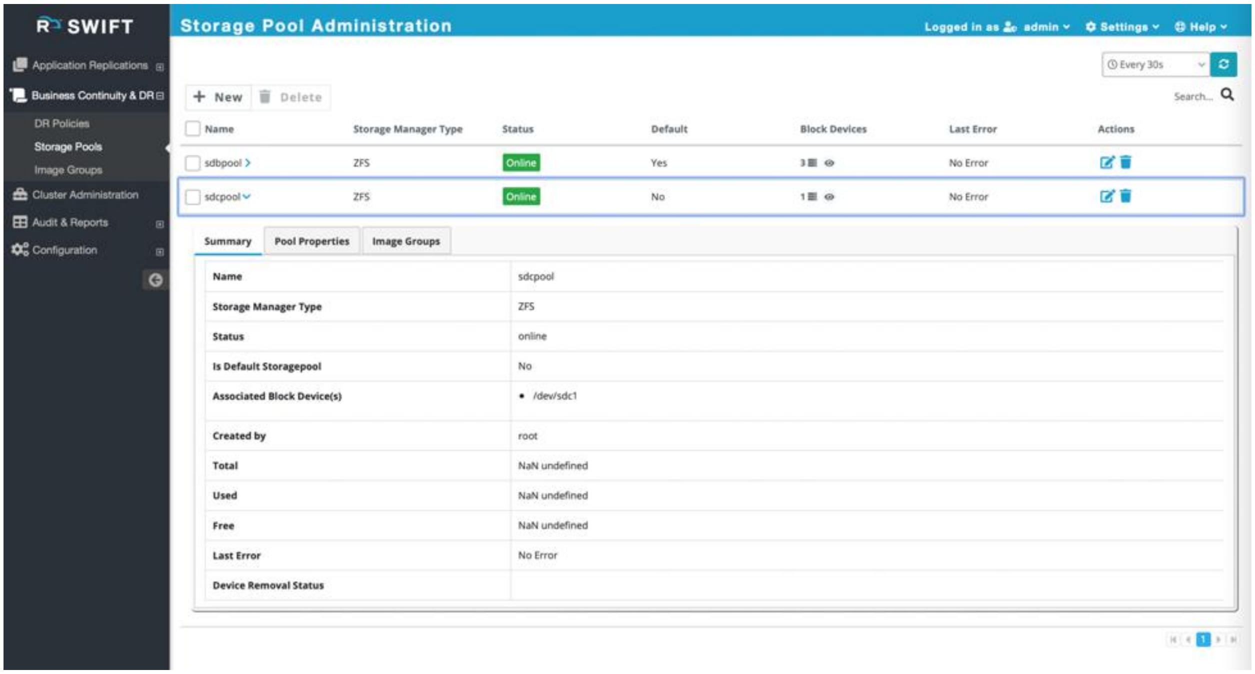
Request (MB): N

Limit (MB): N

Storage pool Administration

A storage pool is a logical storage group created by SWIFT to manage configured block and other storage in SWIFT. SWIFT supports different types of storage pools. Depending on SWIFT release you use, the types of supported storage pools will change. Typically, you will need at least one storage pool created before you can use staged syncs with SWIFT.

You can see storage pool details from the BCDR menu and the Storage pool submenu.



Storage Pool Administration

Name	Storage Manager Type	Status	Default	Block Devices	Last Error	Actions
sdbpool	ZFS	Online	Yes	3	No Error	
sdcpool	ZFS	Online	No	1	No Error	

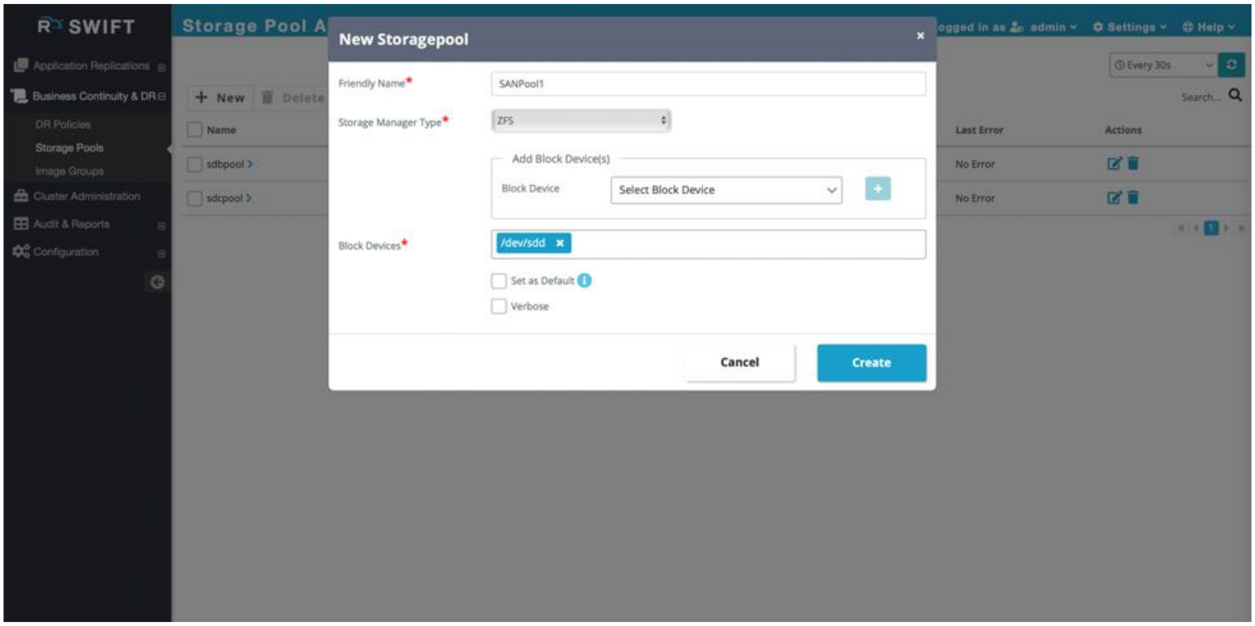
Name	sdcpool
Storage Manager Type	ZFS
Status	online
Is Default Storagepool	No
Associated Block Device(s)	• /dev/sdc1
Created by	root
Total	NaN undefined
Used	NaN undefined
Free	NaN undefined
Last Error	No Error
Device Removal Status	

If you expand a pool, you can see usage summary and other properties, including captured Image-Groups for the pool.

You can do various storage pool operations mentioned in the below sub sections.

Create a Local Storage pool

From the Storage pool administration screen, press the 'New' button. Select 'ZFS' as a type for the pool, which means a local pool.



New Storagepool

Friendly Name* SANPool1

Storage Manager Type* ZFS

Add Block Device(s)

Block Device: Select Block Device

Block Devices* /dev/sdd ×

□ Set as Default

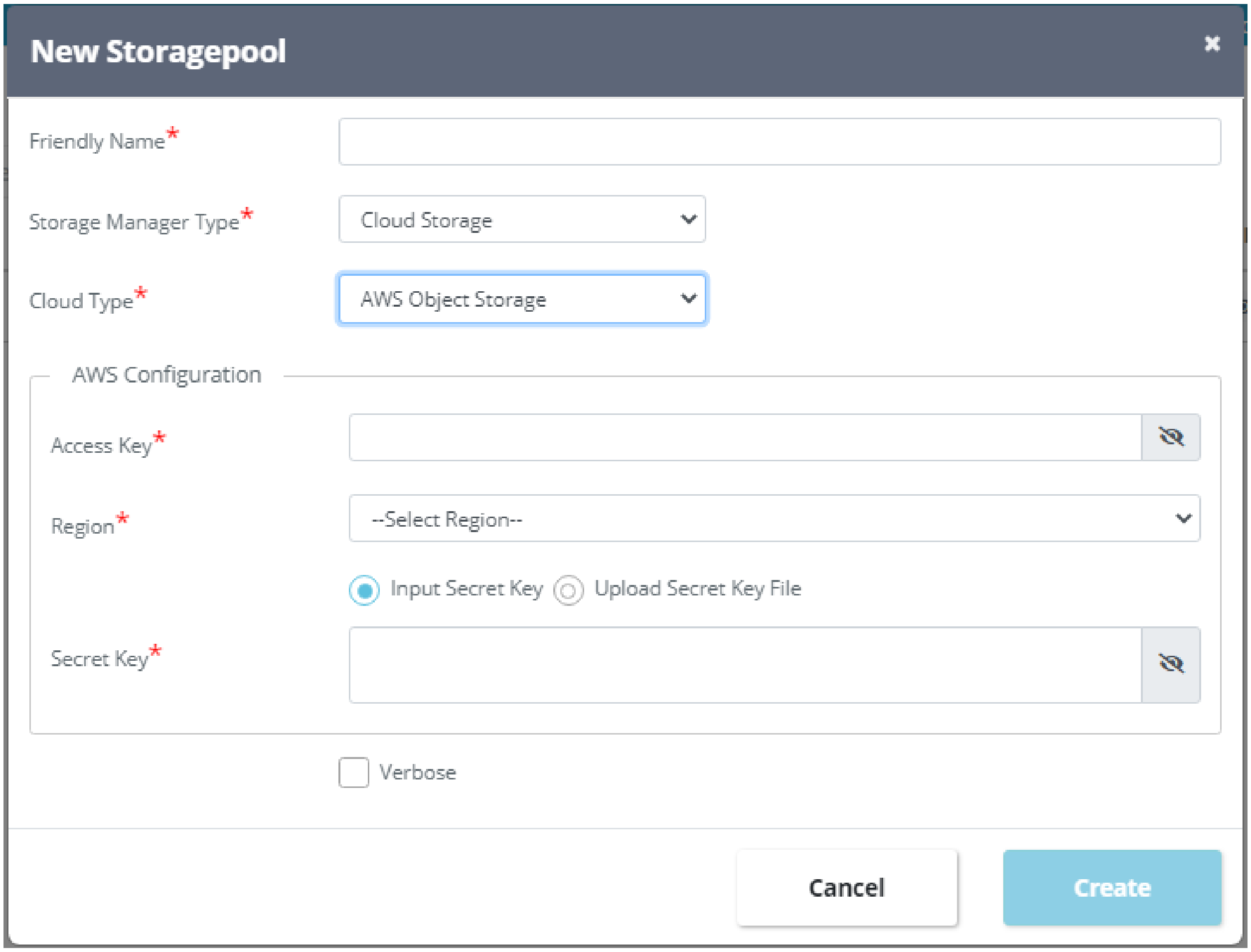
□ Verbose

Give it a name and select devices from the available drop-down for device list. In case your new device is not listed in the drop-down, you can always type the device name or path and add it.

You can also optionally make this new pool being added as the new default pool. Note that first pool added is always marked as the default pool.

Create a Cloud (Object-Storage) Storage pool

From the Storage pool administration screen, press the 'New' button. Select 'Cloud Storage' as a type for the pool.



New Storagepool

Friendly Name*

Storage Manager Type* Cloud Storage

Cloud Type* AWS Object Storage

AWS Configuration

Access Key*

Region* --Select Region--

● Input Secret Key ○ Upload Secret Key File

Secret Key*

□ Verbose

Now select 'Cloud-type' for the pool from the drop-down. Depending on selected cloud type, you will input below details. The cloud credentials you input here can be later modified by modifying the pool in case these credentials change in the future.

Oracle Cloud (OCI)

- User id
- Compartment id
- Tenant id
- Fingerprint
- Private key file

- Region (Selected from a drop-down)

Azure Cloud

- Tenant id
- Subscription id
- Client/App id
- Client/App secret
- Resource group name
- Azure cloud type (Public/Govt/China/etc.)
- Location/Region (Selected from a drop-down)
- Resource group name
- Performance level of storage (Standard/premium/etc.)
- Redundancy of blobs (LRS/GRS/ZRS/GZRS/etc.)
- Access tier (Hot/Cold/etc.)

Amazon Cloud (AWS)

- Access key
- Secret key
- Region (Selected from a drop-down)

Google Cloud (GCP)

- Private key file
- Region (Selected from a drop-down)

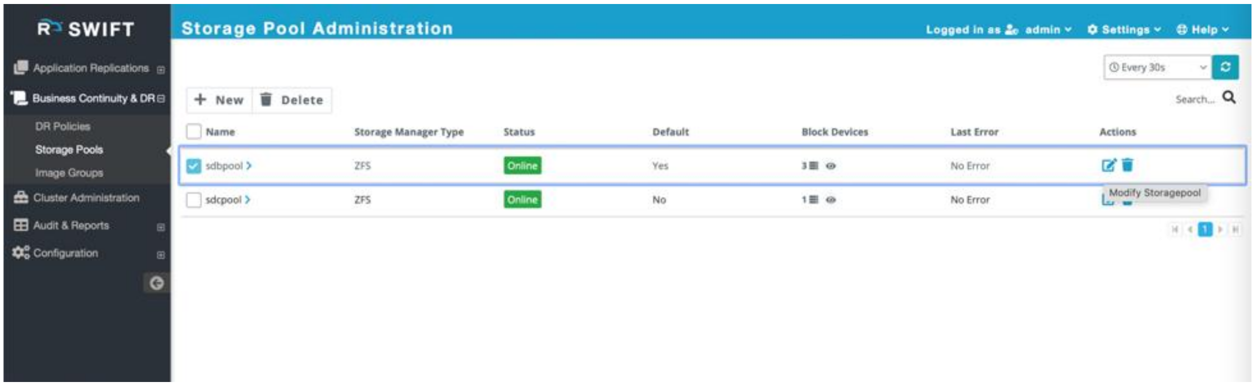
IBM Cloud

- Object-storage service instance name
- API key
- Region (Selected from a drop-down)

It is recommended that you use remote pools for longer-term backups as object storage access is typically slow for frequent backups or regular restore.

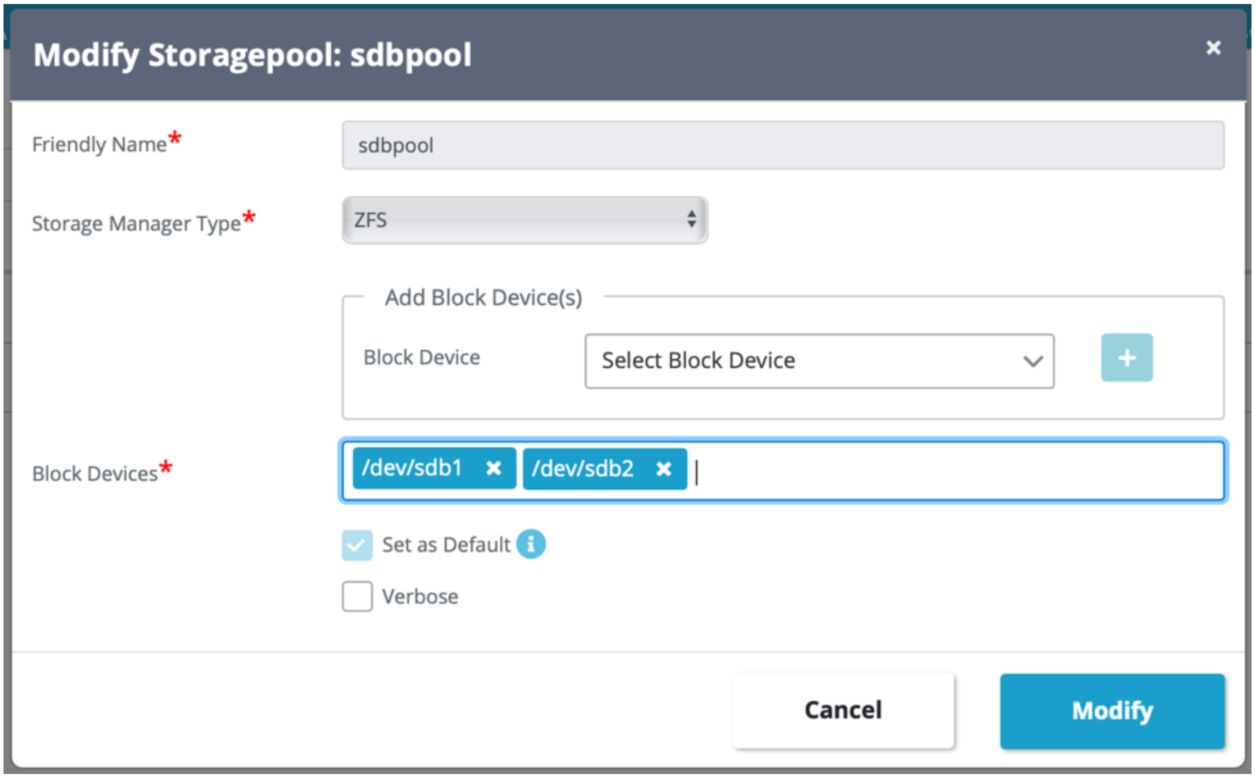
Modify a Local Storage pool

Modify operation for local pool allows you to add or remove devices from the pool. Press the modify icon next to pool entry from the pool administration menu.



☐ Name	Storage Manager Type	Status	Default	Block Devices	Last Error	Actions
☑ sdbpool	ZFS	Online	Yes	3	No Error	
☐ sdcpool	ZFS	Online	No	1	No Error	

You can add or remove devices from this menu and can also change the default property of the pool.



Friendly Name*: sdbpool

Storage Manager Type*: ZFS

Block Device: Select Block Device

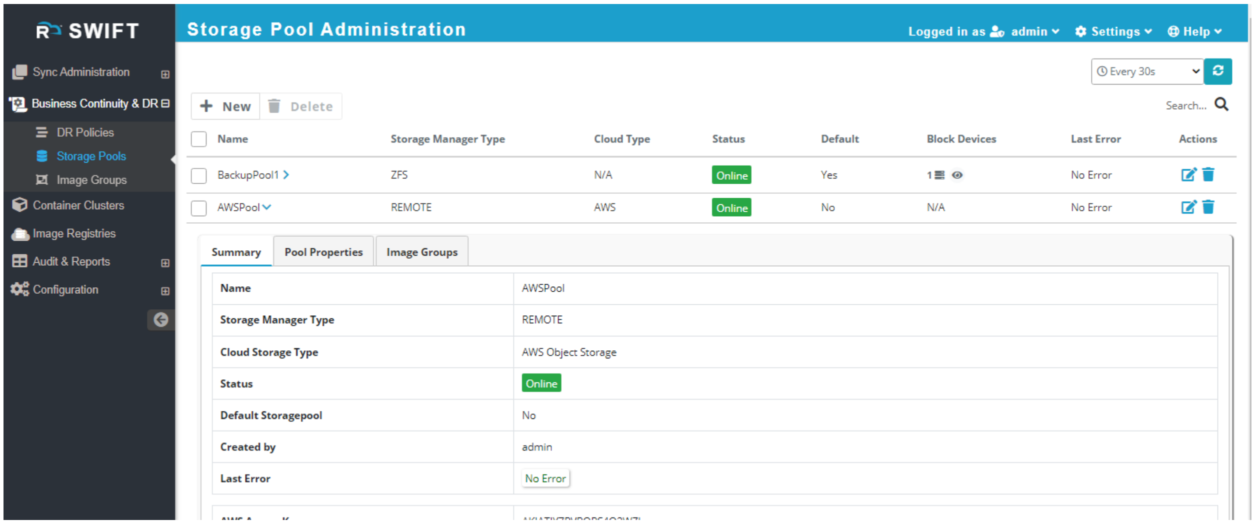
Block Devices*: /dev/sdb1 × /dev/sdb2 ×

☑ Set as Default

☐ Verbose

Modify a Cloud Storage pool

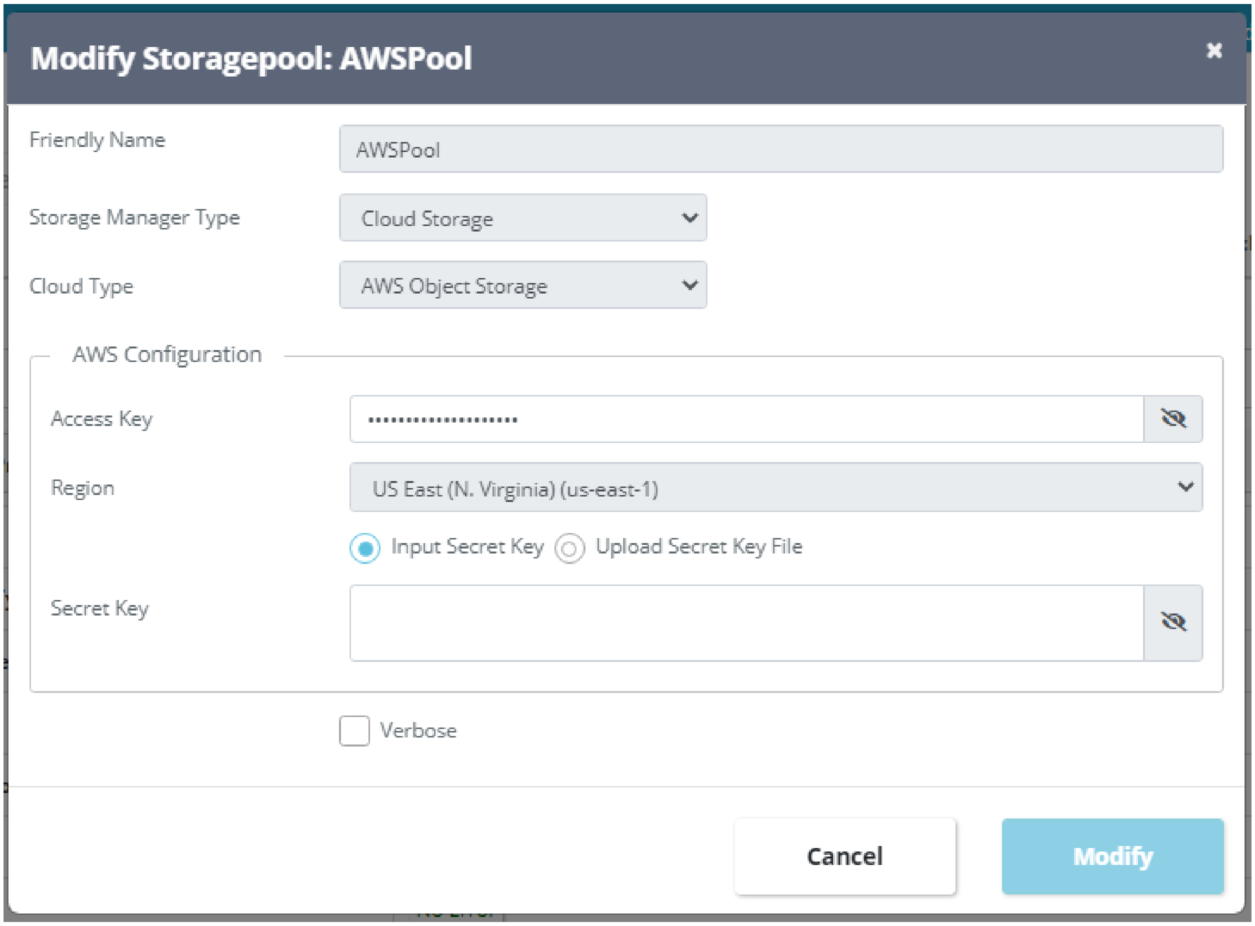
Modify operation for remote cloud storage pool allows you to change object-storage credentials or config for the pool. Press the modify icon next to pool entry from the pool administration menu.



Name	Storage Manager Type	Cloud Type	Status	Default	Block Devices	Last Error	Actions
BackupPool1	ZFS	N/A	Online	Yes	1	No Error	
AWSPool	REMOTE	AWS	Online	No	N/A	No Error	

Name	AWSPool
Storage Manager Type	REMOTE
Cloud Storage Type	AWS Object Storage
Status	Online
Default Storagepool	No
Created by	admin
Last Error	No Error

Modify dialog for remote pool will give you certain credentials and config change options. Once you input new details and press the ‘Modify’ button, SWIFT will validate all new credentials and other inputs. If validation of new inputs or credentials fail, all new config is discarded (Old config is retained as-is).


Modify Storagepool: AWSPool

Friendly Name: AWSPool

Storage Manager Type: Cloud Storage

Cloud Type: AWS Object Storage

AWS Configuration

Access Key: ••••••••••••••••••••

Region: US East (N. Virginia) (us-east-1)

● Input Secret Key ○ Upload Secret Key File

Secret Key:

□ Verbose

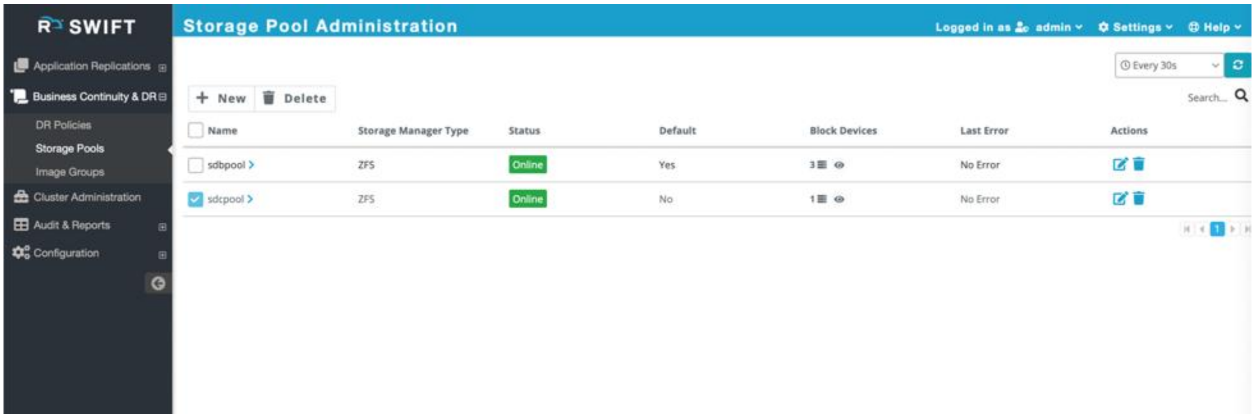
Options allowed for modify include API and access keys, cloud usernames, etc. Only part of the options that you see during cloud storage pool create are allowed for modification.

Note that if you don't see an option on the modify dialog to change one or more parameters that you wanted to change, then you would need to delete the pool and re-add it with new parameter inputs.

Delete a Storage pool

Deletion steps remain the same irrespective of whether the pool being deleted is local or remote.

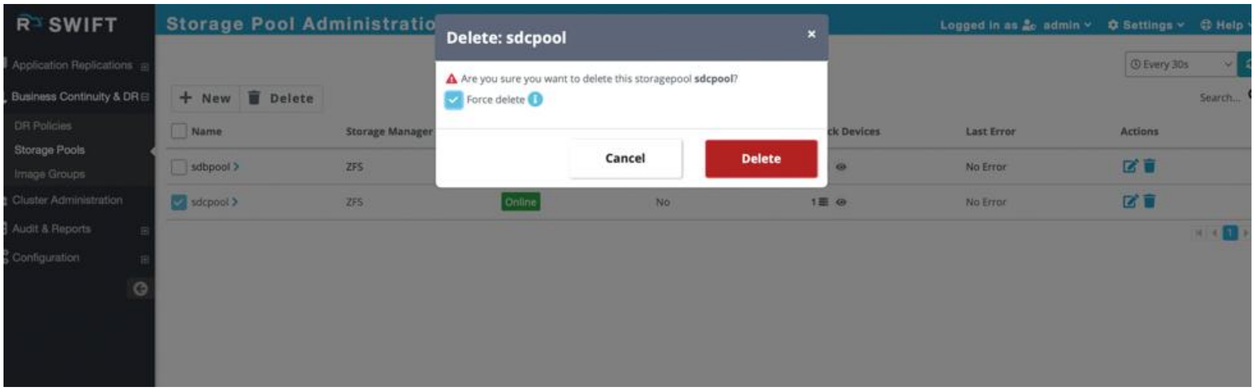
Select one or more pools from the pool administration menu that you want to delete.



Name	Storage Manager Type	Status	Default	Block Devices	Last Error	Actions
□ sdbpool	ZFS	Online	Yes	3	No Error	
☑ sdcpool	ZFS	Online	No	1	No Error	

Press the 'Delete' button to delete the selected pool(s).

If there are any Image-groups or backups captured to the deleted pool, then pool deletion will error on them asking you delete those first.



Delete: sdcpool

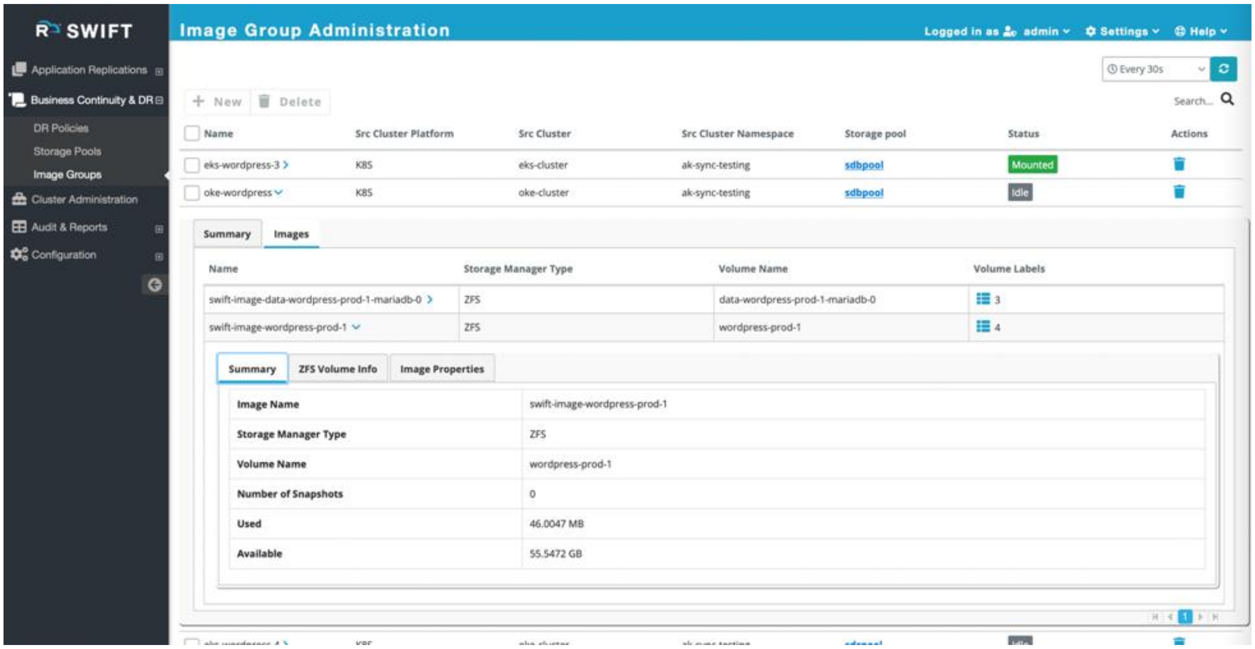
Are you sure you want to delete this storagepool **sdcpool**?

☑ Force delete

Optionally, you can select the 'Force delete' checkbox to forcefully delete the pool along with all underlying captured Image-groups and backups stored in it. You can also use the force deletion if the underlying ZFS pool or object storage location is already deleted explicitly or is in unrecoverable error state. Force deletion in such cases will do a best attempt in the backend to delete such pools first (at the storage or cloud level) and then will clean up the pool entry from the SWIFT CMDB.

Image-Group Administration

An Image-Group in SWIFT represents a logical group of captured Kubernetes or OpenShift volumes. Each image in an Image-Group maps to one Kubernetes/OpenShift volume. You will typically never create an Image-Group manually, as only Stage1 sync can create it, however, you can clone an existing Image-Group to create a new Image-Group.



Name	Src Cluster Platform	Src Cluster	Src Cluster Namespace	Storage pool	Status	Actions
eks-wordpress-3	K8S	eks-cluster	ak-sync-testing	sdbpool	Mounted	
oke-wordpress	K8S	oke-cluster	ak-sync-testing	sdbpool	Idle	

Name	Storage Manager Type	Volume Name	Volume Labels
swift-image-data-wordpress-prod-1-mariadb-0	ZFS	data-wordpress-prod-1-mariadb-0	3
swift-image-wordpress-prod-1	ZFS	wordpress-prod-1	4

Image Name	swift-image-wordpress-prod-1
Storage Manager Type	ZFS
Volume Name	wordpress-prod-1
Number of Snapshots	0
Used	46.0047 MB
Available	55.5472 GB

The Stage1 and Stage2 syncs operate on Image-Groups to capture data to/sync data from it. Every Stage1 and Stage2 sync will input an Image-Group. When you specify an Image-Group for Stage1, it will be created, if it doesn't already exist. Any time recurring Stage1 sync runs, and it finds any volumes as newly added or existing volumes being deleted in the source cluster and selected namespace, then corresponding Image-Group will be updated/modified to reflect the new set of synced volumes by the Stage1 sync. This also means that if you try to reuse an existing Image-Grou,p which was originally captured for a different namespace, then after the next Stage1 sync for the new source namespace, one or more images may be deleted/modified by Stage1 sync to match the new sync source namespace and synced volumes.

The Stage1 sync also intelligently tries to reuse existing images from other Image-Groups (in the same storage pool) by cloning them to the current sync-selected Image-Group, if those other images match the currently synced source volume specification. This smart tactic allows Stage1 sync to save on initial capture of the entire source volume, as now it can instead clone matching existing/previous volume capture and use it as a base for copying data on top. Ultimately, any data changes of cloned volume with the corresponding synced source volume snapshot will still be synced over, but it will still be far quicker than fully capturing volume.

Post-sync backups are also taken at the Image-Group level when you configure a backup policy for your application. Each Image-Group is also linked by the Stage1 sync with synced Kubernetes/OpenShift objects from the source or production cluster, so the same set of objects can be synced then when a Stage2 sync for the Image-Group is triggered. This also means that if you delete any remote clusters which have one or more Image-Groups captured, then cluster objects linked to those Image-Groups will be retained. Image-Group Stage2 syncs can be triggered even if source cluster is not reachable.

The following sections describe Image-Group administration operations in detail.

Image-Group Create (Clone)

The create is technically a clone operation, as you can't create an empty Image-Group. Only Stage1 sync can create a new Image-Group.

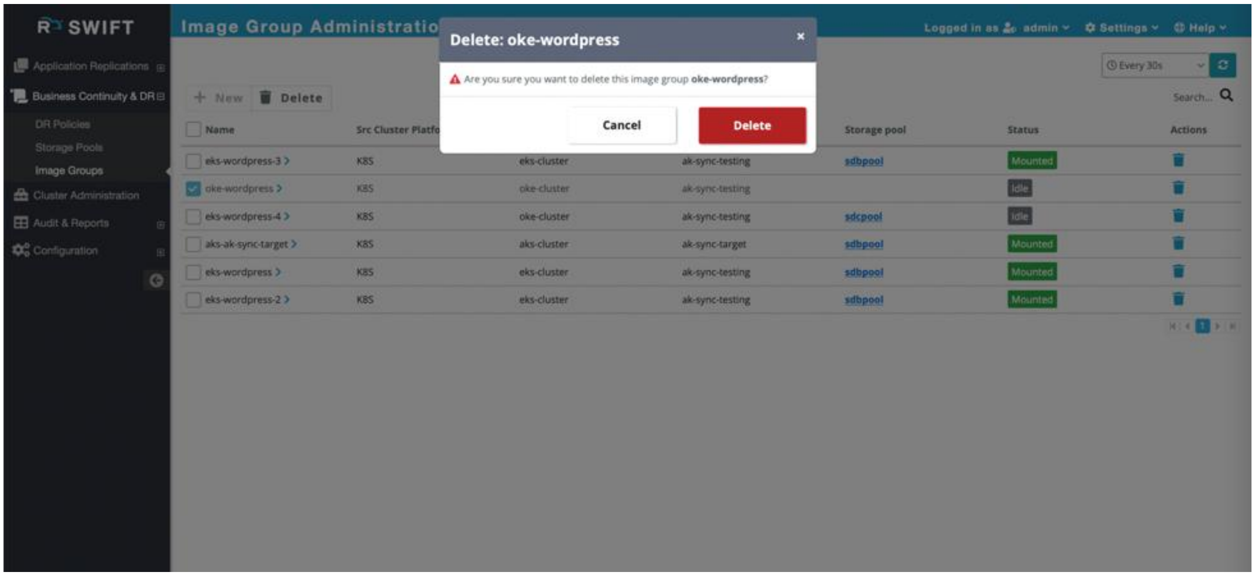
From the Image-Group administration menu, select Image-Group press the 'New' button.

Select the storage pool and source Image-Group to clone, then specify a name for the new Image-Group. Note that Image-Groups can only be cloned within the same storage pool, so you will only see available Image-Groups for cloning in the selected storage pool. Once you click on the Create button, the new Image-Group will be created in the same storage pool as the cloned Image-Group.

The clone operation may take a while depending on the captured data size across images of the source Image-Group, as they will be copied individually for data.

Image-Group Delete

From the Image-Group administration menu, select Image-Group that you want to delete and press the 'Delete' button.



Delete: oke-wordpress

Are you sure you want to delete this image group **oke-wordpress**?

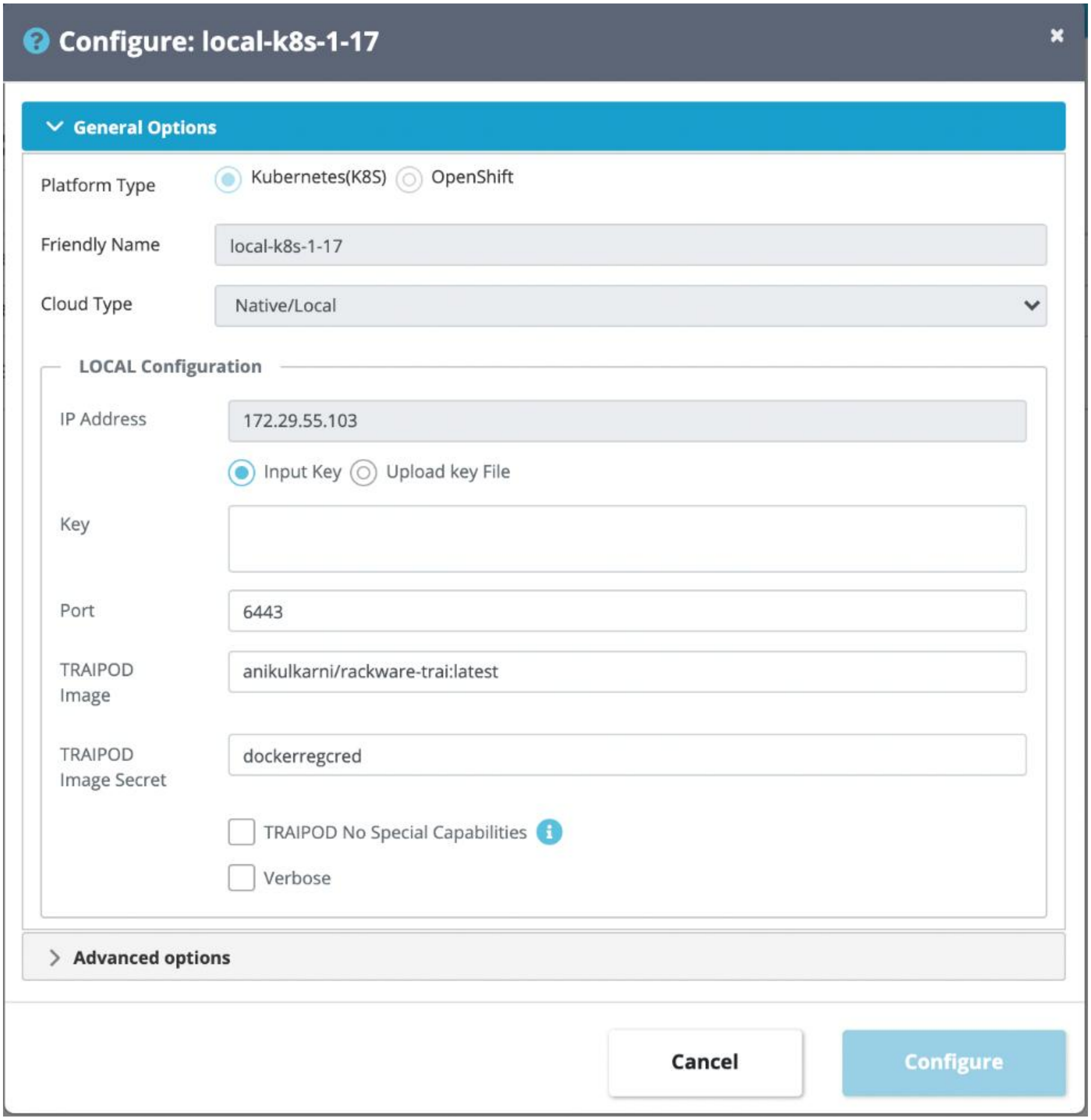
	Name	Src Cluster Platfo			Storage pool	Status	Actions
□	eks-wordpress-3	K8S	eks-cluster	ak-sync-testing	sdbpool	Mounted	
☑	oke-wordpress	K8S	oke-cluster	ak-sync-testing		Idle	
□	eks-wordpress-4	K8S	oke-cluster	ak-sync-testing	sdcpool	Idle	
□	aks-ak-sync-target	K8S	aks-cluster	ak-sync-target	sdbpool	Mounted	
□	eks-wordpress	K8S	eks-cluster	ak-sync-testing	sdbpool	Mounted	
□	eks-wordpress-2	K8S	eks-cluster	ak-sync-testing	sdbpool	Mounted	

The deletion for an Image-Group is prevented if the Image-Group is currently participating in an active sync. Only IDLE Image-Groups can be deleted.

Change default configurations of managed clusters

Once you add a new cluster to the SWIFT, it shows up on the ‘K8S Administration’ page. Currently, you configure these defaults for a cluster from the administration menu:

1. API server port
2. Service account key (which is used to administer the Kubernetes cluster with SWIFT)
3. TRAI image name
4. TRAI image-pull secret
5. TRAI resource configs
6. Image-registry mappings
7. Image pull secret mappings
8. Ceph dashboard config



Configure: local-k8s-1-17

General Options

Platform Type ● Kubernetes(K8S) ○ OpenShift

Friendly Name: local-k8s-1-17

Cloud Type: Native/Local

LOCAL Configuration

IP Address: 172.29.55.103

● Input Key ○ Upload key File

Key:

Port: 6443

TRAIPOD Image: anikulkarni/rackware-trai:latest

TRAIPOD Image Secret: dockerregcred

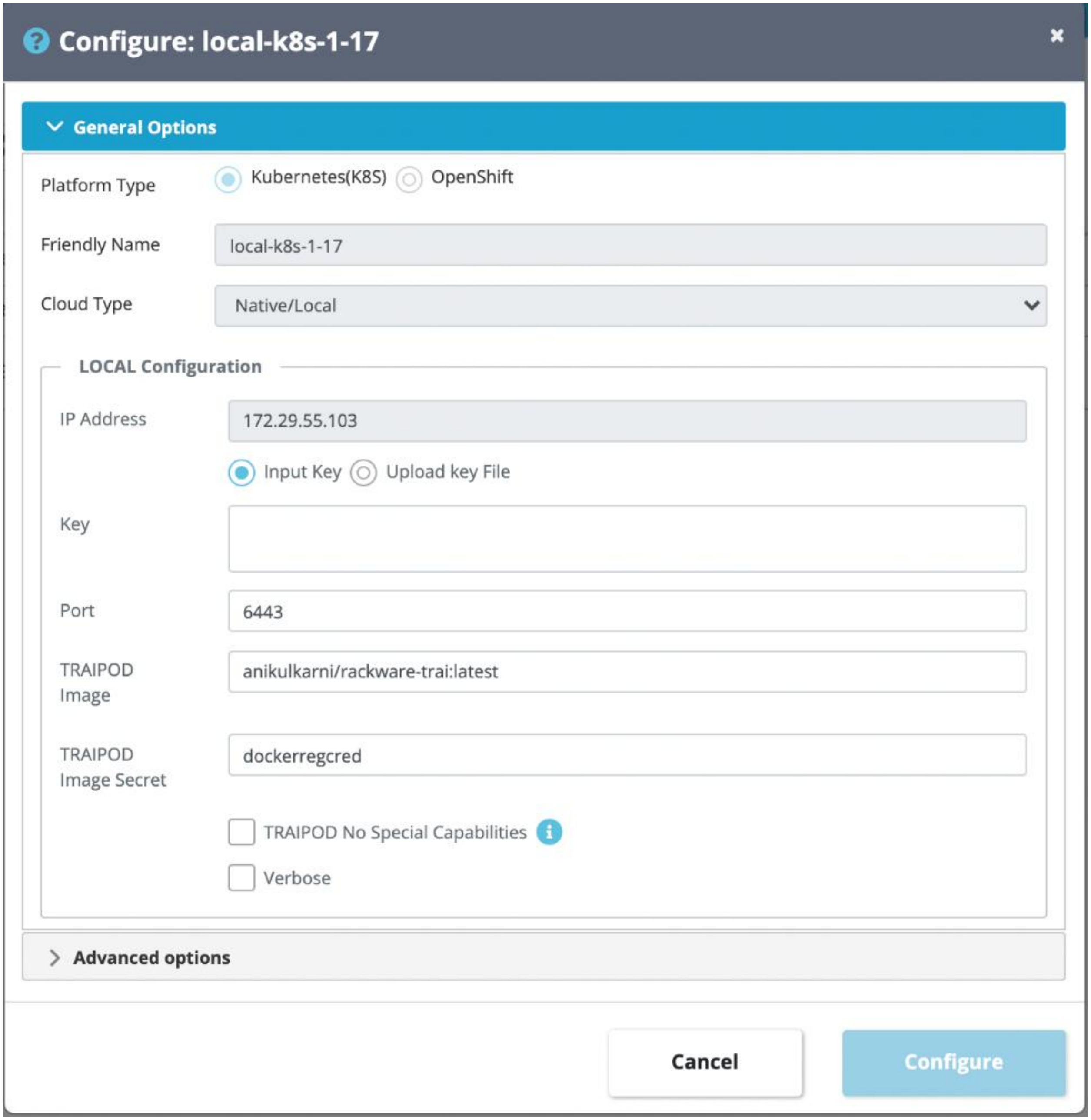
□ TRAIPOD No Special Capabilities

□ Verbose

Additionally, for cloud-based clusters, you can change various cloud-specific credentials and other relevant details.

Transient RackWare Agent Image POD (TRAIPOD) Image and Image Secret defaults are optional inputs. They can be configured for each sync, and the values input for the sync will override the defaults set for the cluster. Please see the TRAI section for more details on what this image is and how it is used during syncs.

For local cluster:



Configure: local-k8s-1-17

General Options

Platform Type ● Kubernetes(K8S) ○ OpenShift

Friendly Name: local-k8s-1-17

Cloud Type: Native/Local

LOCAL Configuration

IP Address: 172.29.55.103

● Input Key ○ Upload key File

Key:

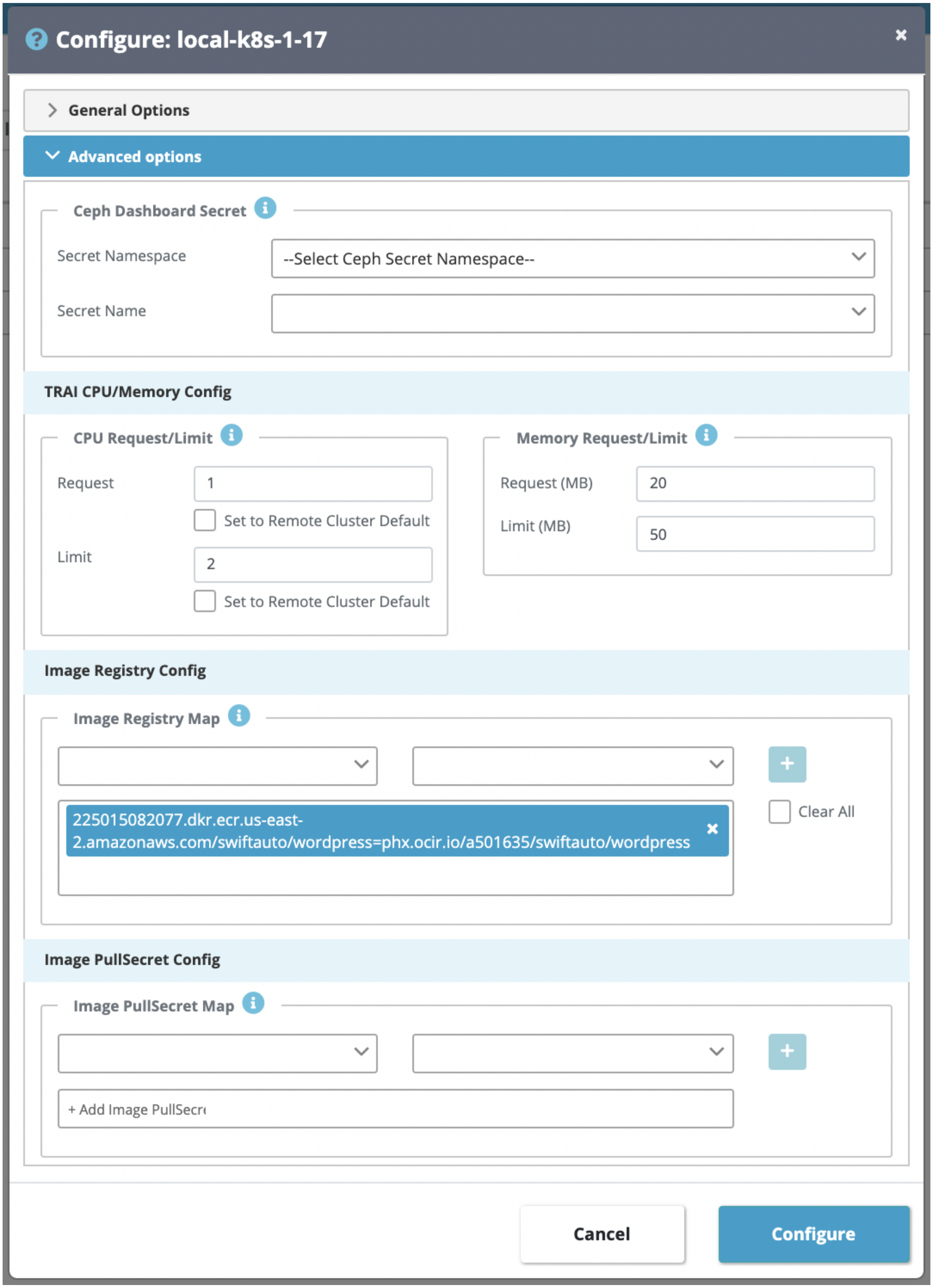
Port: 6443

TRAIPOD Image: anikulkarni/rackware-trai:latest

TRAIPOD Image Secret: dockerregcred

□ TRAIPOD No Special Capabilities

□ Verbose



Configure: local-k8s-1-17

General Options

Advanced options

Ceph Dashboard Secret

Secret Namespace: --Select Ceph Secret Namespace--

Secret Name:

TRAI CPU/Memory Config

CPU Request/Limit

Request: 1

□ Set to Remote Cluster Default

Limit: 2

□ Set to Remote Cluster Default

Memory Request/Limit

Request (MB): 20

Limit (MB): 50

Image Registry Config

Image Registry Map

225015082077.dkr.ecr.us-east-2.amazonaws.com/swiftauto/wordpress=phx.ocir.io/a501635/swiftauto/wordpress

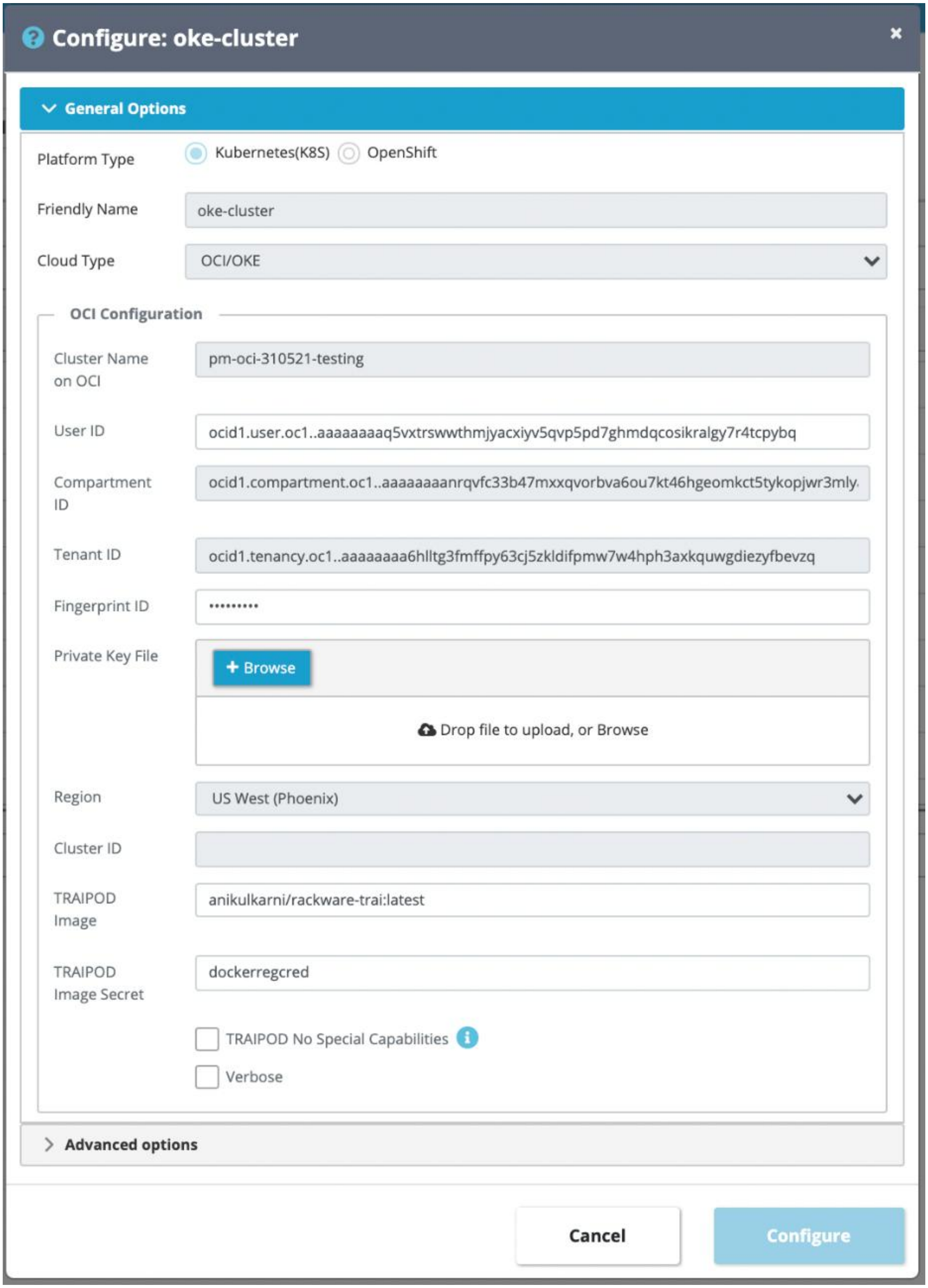
□ Clear All

Image PullSecret Config

Image PullSecret Map

+ Add Image PullSecr

For cloud based cluster:



Platform Type	● Kubernetes(K8S) ○ OpenShift
Friendly Name	oke-cluster
Cloud Type	OCI/OKE

OCI Configuration

Cluster Name on OCI	pm-oci-310521-testing
User ID	ocid1.user.oc1..aaaaaaaaq5vxtrswwthmjyacxiyv5qvp5pd7ghmdqcosikralgy7r4tcpybq
Compartment ID	ocid1.compartment.oc1..aaaaaaaanrqvfc33b47mxxqvorbva6ou7kt46hgeomkct5tykopjwr3mly
Tenant ID	ocid1.tenancy.oc1..aaaaaaaa6hlltg3fmffpy63cj5zkldifpmw7w4hph3axkquwgdiezyfbevzq
Fingerprint ID	•••••••••
Private Key File	+ Browse Drop file to upload, or Browse
Region	US West (Phoenix)
Cluster ID	
TRAIPOD Image	anikulkarni/rackware-trai:latest
TRAIPOD Image Secret	dockerregcred

□ TRAIPOD No Special Capabilities

□ Verbose

The example cluster shown is Oracle OKE based. You can see above that you can configure/change defaults for the User ID as well as for the API key.

What is Transient RackWare Agent Image (TRAI) POD?

TRAI is an exclusive container image deployed with the SWIFT. During syncs, SWIFT will run a TRAI instance as a pod and a service combination. The environment is used for sync staging activities. The TRAIPOD runs for clusters on both sides of passthrough syncs. The TRAIPOD (pod+service) is run under the namespace you are replicating from/to, and only runs for the sync duration.

For Kubernetes and other container platforms, there are various ways to make a container image available to your cluster nodes. You can refer to this document for Kubernetes on different official ways Kubernetes supports for making your container image (from a private registry) available to the cluster nodes. SWIFT supports all modes the respective container platform supports.

The next section highlights how you can register or import a TRAI image (which is deployed with your SWIFT install) to a private docker registry. The TRAI image is docker container format compliant so that it can be run with any of the latest widely known and used container platforms.

Import TRAI image to a private docker registry

The TRAI image for the respective SWIFT version is deployed along with the SWIFT. You can find it at

```
/opt/swift/traipod/rackware-trai-docker.tar.gz
```

on your SWIFT server (where the SWIFT is installed).

Steps to import a TRAI image

1. Copy the TRAI image tar file (mentioned above) from the SWIFT server to a host where you have the 'docker' client installed and configured.
2. Open an SSH shell to the server where the docker client is installed and where you copied the TRAI image tar file in step #1 above.
3. Change to the directory where you have copied the TRAI image tar file (e.g., cd /home/john/swift-files/).
4. Run:

   docker load < rackware-trai-docker.tar.gz

5. The image will be imported with the default tag (which generally maps to the SWIFT version). You can optionally tag the image and then assign it to the registry where you want to push it (for example, we assume the private registry is available at 'myregistry').

   docker image tag rackware-trai:<version> myregistry:latest

   <version> is the default version with which your TRAI image is imported in step #4. You can find this with the 'docker image ls' command.

The above syntax works for docker-hub based registries. Depending on the location and the type of registry you use, you may have to use the alternative syntax below:

docker image tag rackware-trai:<version> myregistry/rackware-trai:latest

6. Push the image to your private registry.
   docker push myregistry:latest

These steps only need to be done after a fresh SWIFT install and after every SWIFT upgrade.

Making the private registry available to a cluster namespace

Once you perform the steps to push the TRAI image to a private registry, the next step is to make the image visible to the required namespaces within the cluster. The steps to configure image-pull credentials within a namespace change from one container platform to another.

Configure an image-pull secret within a Kubernetes namespace

You will have to repeat the steps below for every cluster you are managing with the SWIFT (source as well as target clusters for syncs). The steps also assume that you have a working 'kubectl' utility on a server.

1. Connect to a server where you have working kubectl utility for the required cluster.
2. Create a secret which captures docker registry credentials:

```
kubectl create secret docker-registry regcred --docker-server=<your-registry-server> --docker-username=<your-name> --docker-password=<your-pword> --docker-email=<your-email>
```

It is recommended that you configure docker registry credentials per namespace for better security.

Configure TRAI details for the cluster under SWIFT

Once you register the TRAI image to your private-registry and configured namespace scoped registry secrets for the cluster, you will have to set the details in SWIFT for the respective cluster entry. Typically, you would configure these details for a private registry:

1. Image name and version tag with which the TRAI image was imported to your registry
2. Image-pull secret configured in the namespace

The TRAI image is pulled and used for creating staging POD/containers during sync. You will enter the above details during sync configuration. Alternatively, you can configure defaults, one-time, at the cluster level while adding the cluster entry or using the 'configure' operation for the cluster. Please refer to the respective Operation sections for more details on how to specify the defaults.

If you are syncing container image registry used by Kubernetes or OpenShift cluster, then these above steps of importing TRAI image or configuring image pull secret are optional. You can refer to sync administration section to know more on how to select discovered image registry for automating TRAI upload and usage during syncs.

Starting a new synchronization or replication

The sections below show the detailed steps for starting a synchronization between different container platforms, which are managed by the SWIFT. You can initiate a sync process between any supported and managed container platforms by the SWIFT.

Synchronization modes

SWIFT sync supports four modes:

1. **Passthrough**

   In this mode, sync is run between source and target cluster directly. You do not need direct connectivity between your clusters for this. However, the installed SWIFT must be able to reach both sides of the clusters. The SWIFT will create and use a passthrough data channel between your selected clusters (by connecting to both sides individually). The data (objects and volumes) are replicated directly from the source to the target cluster. The SWIFT will not store any actual volumes' data but will store some metadata about cluster objects on both sides.

   For registry sync in this mode, SWIFT will synchronize selected set of repos/images/tags directly from the source to the target registry. The SWIFT will not store any actual image or tag data apart from metadata about discovered repos.

2. **Stage1**

   In this mode, you sync between your source cluster to the SWIFT. All sync selected cluster volumes are captured in the SWIFT DB. Every volume is captured to its image in the SWIFT, and the set of related volumes (chosen by the sync run) are captured together as an 'image-group.' Cluster objects selected by sync are also captured and stored in the SWIFT DB.

   If you select this mode of the sync, you will configure SWIFT details for the target of the sync (like an 'image-group' name, for example).

   For registry sync in this mode, SWIFT will synchronize selected set of repos/images/tags from the source registry to the storage pool in the SWIFT.

3. **Stage2**

   In this mode, you sync between SWIFT captured volumes (image-group) and objects from Stage1 sync to your target cluster. If there are existing volumes with the same name in the target cluster, they will be delta synced for changes. Any missing volumes for the target cluster will be created afresh and replicated.

   If you select this mode of the sync, you will configure SWIFT details as the source of the sync (like an 'image-group' name, for example) and target cluster details where everything will be synced.

   For registry sync in this mode, SWIFT will synchronize selected set of images/tags from the storage pool to the target registry.

4. **Stage1 and Stage2**

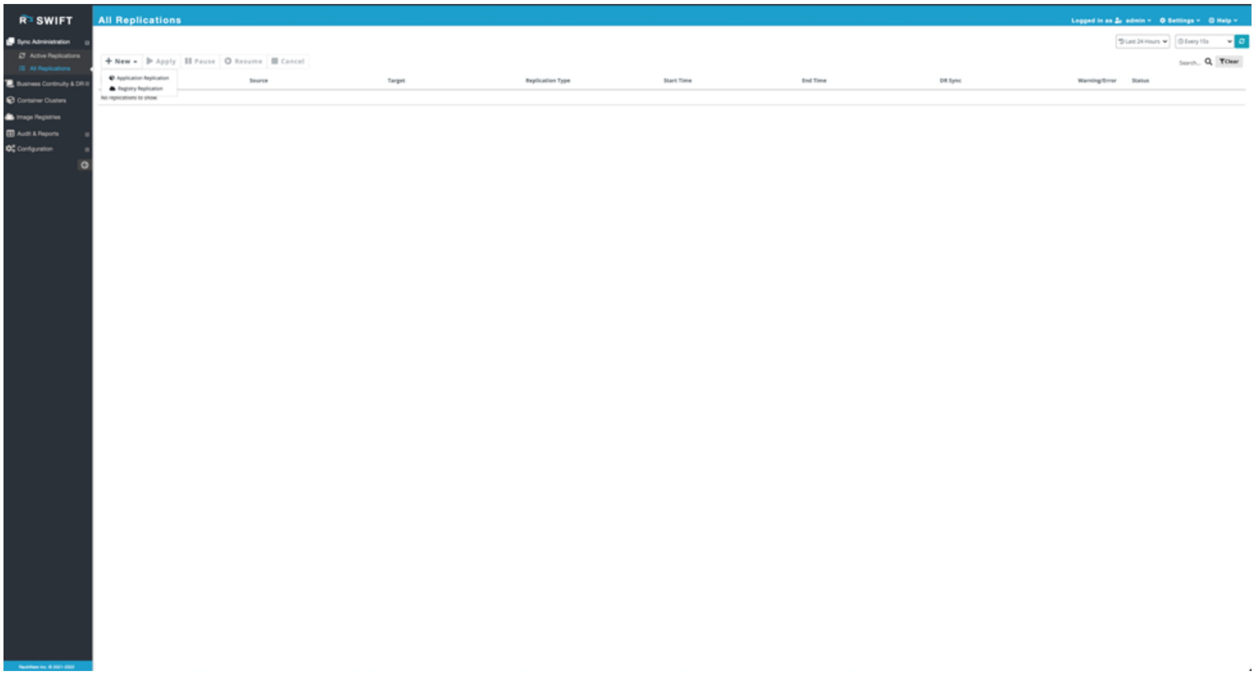
This mode is a mix of stage1 and stage2 syncs together. You will configure details for the source and target cluster as well as the image-group name. Syncs in this mode would run for both stage1 and stage2 parts, and this also remains the same for registry syncs too.

A synchronization between Kubernetes clusters

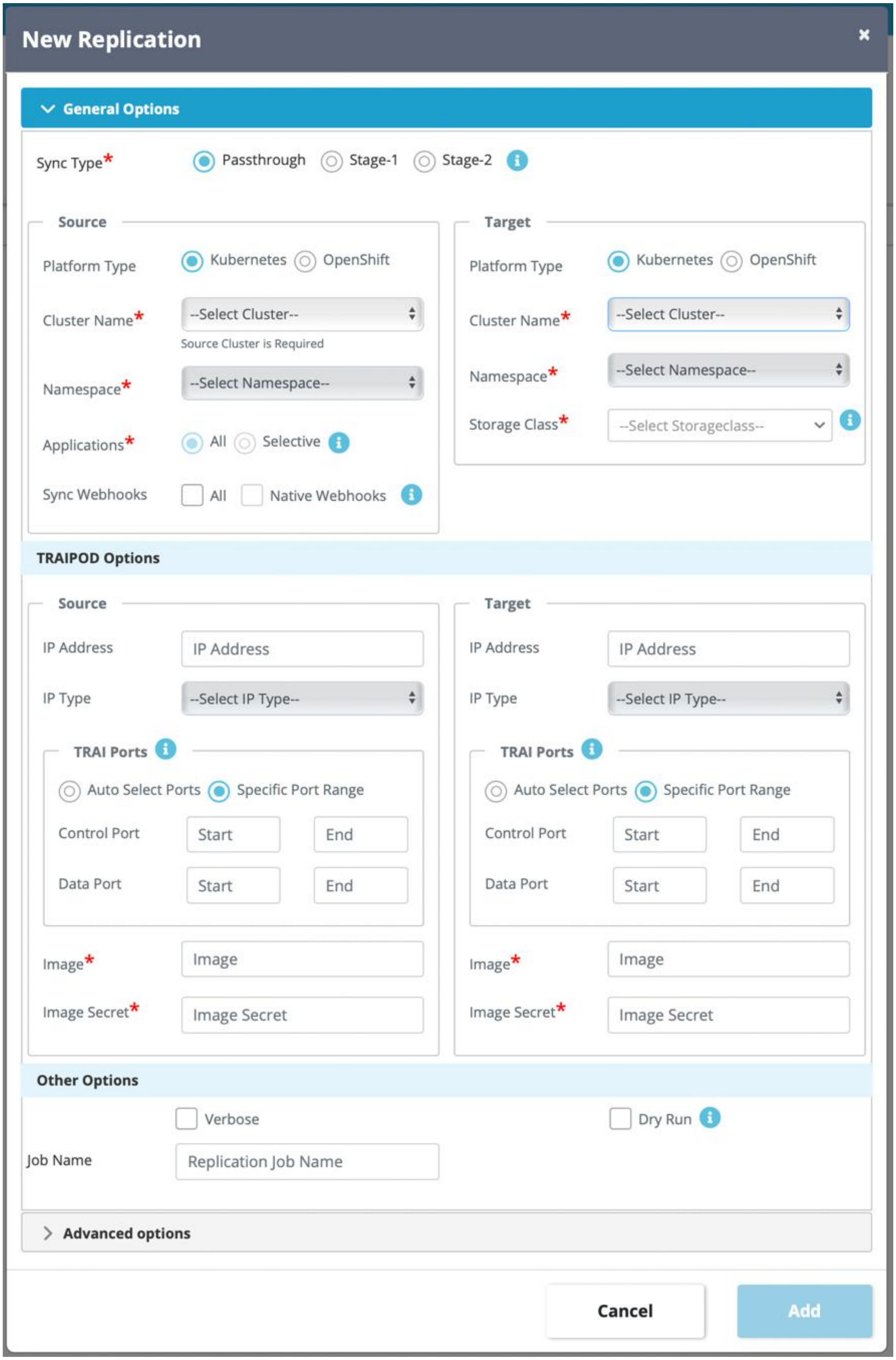
Use the below steps if you want to initiate a sync between two Kubernetes/OpenShift clusters, which are added to the SWIFT. The steps remain the same irrespective of where the managed Kubernetes cluster is located (local vs. in the cloud).

The steps also remain the same for cross-platform syncs.

Connect to the SWIFT dashboard and navigate to the 'Sync Administration' menu and the 'All Replications' submenu.



Press the '+ New' button and then the 'Application Replication' submenu. Select your Kubernetes or OpenShift source and target clusters, and the namespace, and the top-level Kubernetes object for each. If you select 'stage1' or 'stage2' syncs, you will have to configure/select SWIFT image details for captured volumes. Some source and target cluster options will change depending on the selected cluster's type.



New Replication ×

General Options

Sync Type* ● Passthrough ○ Stage-1 ○ Stage-2 ℹ

Source

Platform Type ● Kubernetes ○ OpenShift

Cluster Name* --Select Cluster--

Source Cluster is Required

Namespace* --Select Namespace--

Applications* ● All ○ Selective ℹ

Sync Webhooks □ All □ Native Webhooks ℹ

Target

Platform Type ● Kubernetes ○ OpenShift

Cluster Name* --Select Cluster--

Namespace* --Select Namespace--

Storage Class* --Select Storageclass-- ℹ

TRAIPOD Options

Source

IP Address: IP Address

IP Type: --Select IP Type--

TRAI Ports ℹ

○ Auto Select Ports ● Specific Port Range

Control Port: Start End

Data Port: Start End

Image*: Image

Image Secret*: Image Secret

Target

IP Address: IP Address

IP Type: --Select IP Type--

TRAI Ports ℹ

○ Auto Select Ports ● Specific Port Range

Control Port: Start End

Data Port: Start End

Image*: Image

Image Secret*: Image Secret

Other Options

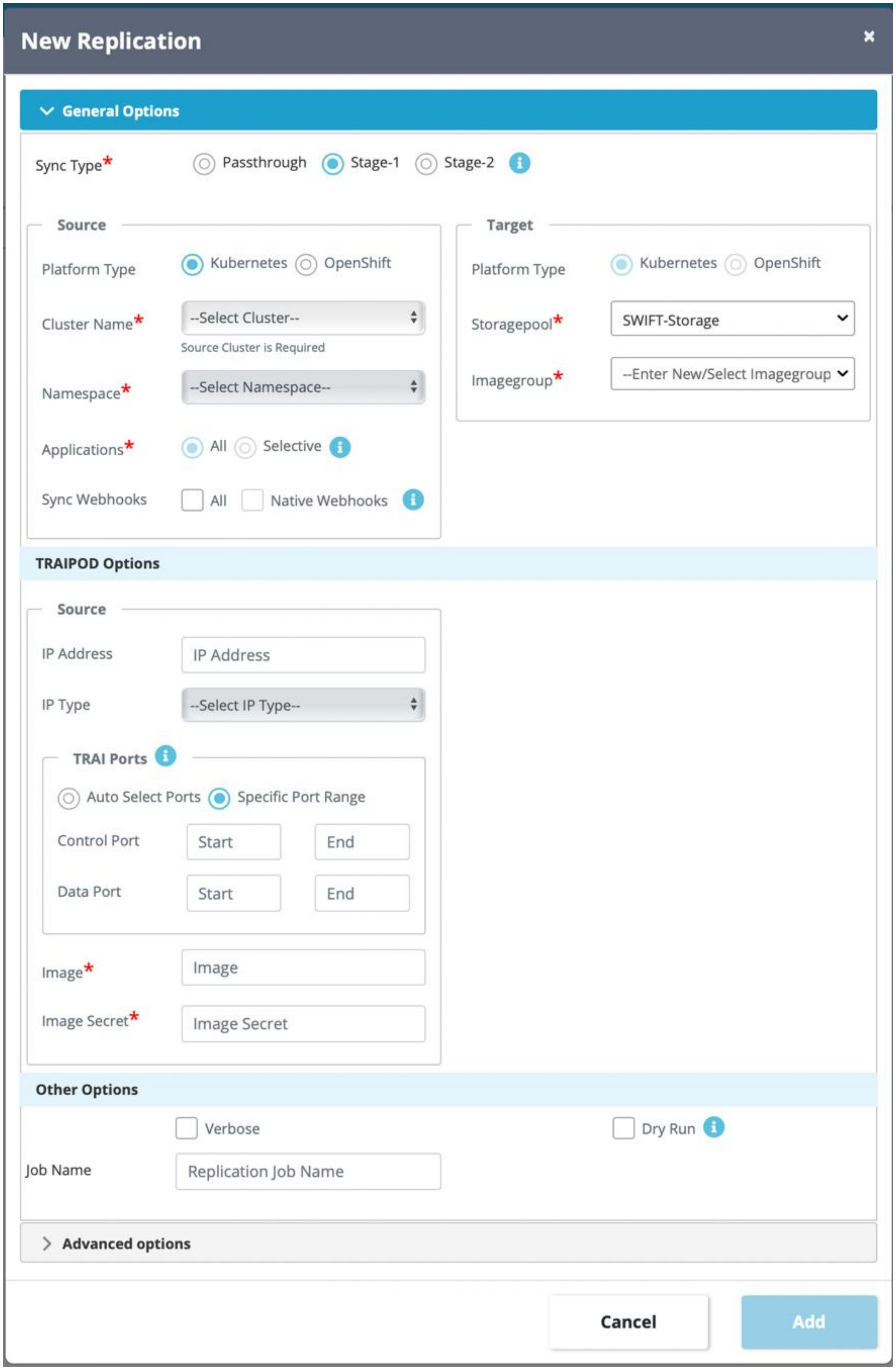
□ Verbose □ Dry Run ℹ

Job Name: Replication Job Name

> Advanced options

The control and data port ranges need to have equal number of ports in both. The number of ports in the input range will determine how may TRAI Pods are run during a sync. If source cluster is multi zonal or regional cluster and if source namespace being synced contains volumes dispersed across regions or zones, then you need to enter number of ports in the range that equals to unique volume regions or zones in the source namespace else sync will fail later highlighting number of ports needed in the input range. Note that even when sync runs with a single TRAI Pod, then it will sync volumes in parallel.

For Stage1 sync, you will specify the source details like passthrough syncs and additionally will also now specify existing storage pool and Image-Group or new Image-Group name to create. If specified Image-Group doesn’t exist, then it will be created by the Stage1 sync.



New Replication ✖

General Options

Sync Type* ○ Passthrough ● Stage-1 ○ Stage-2

Source

Platform Type ● Kubernetes ○ OpenShift

Cluster Name* --Select Cluster--

Source Cluster is Required

Namespace* --Select Namespace--

Applications* ● All ○ Selective

Sync Webhooks □ All □ Native Webhooks

Target

Platform Type ● Kubernetes ○ OpenShift

Storagepool* SWIFT-Storage

Imagegroup* --Enter New/Select Imagegroup

TRAIPOD Options

Source

IP Address: IP Address

IP Type: --Select IP Type--

TRAI Ports

○ Auto Select Ports ● Specific Port Range

Control Port: Start, End

Data Port: Start, End

Image*: Image

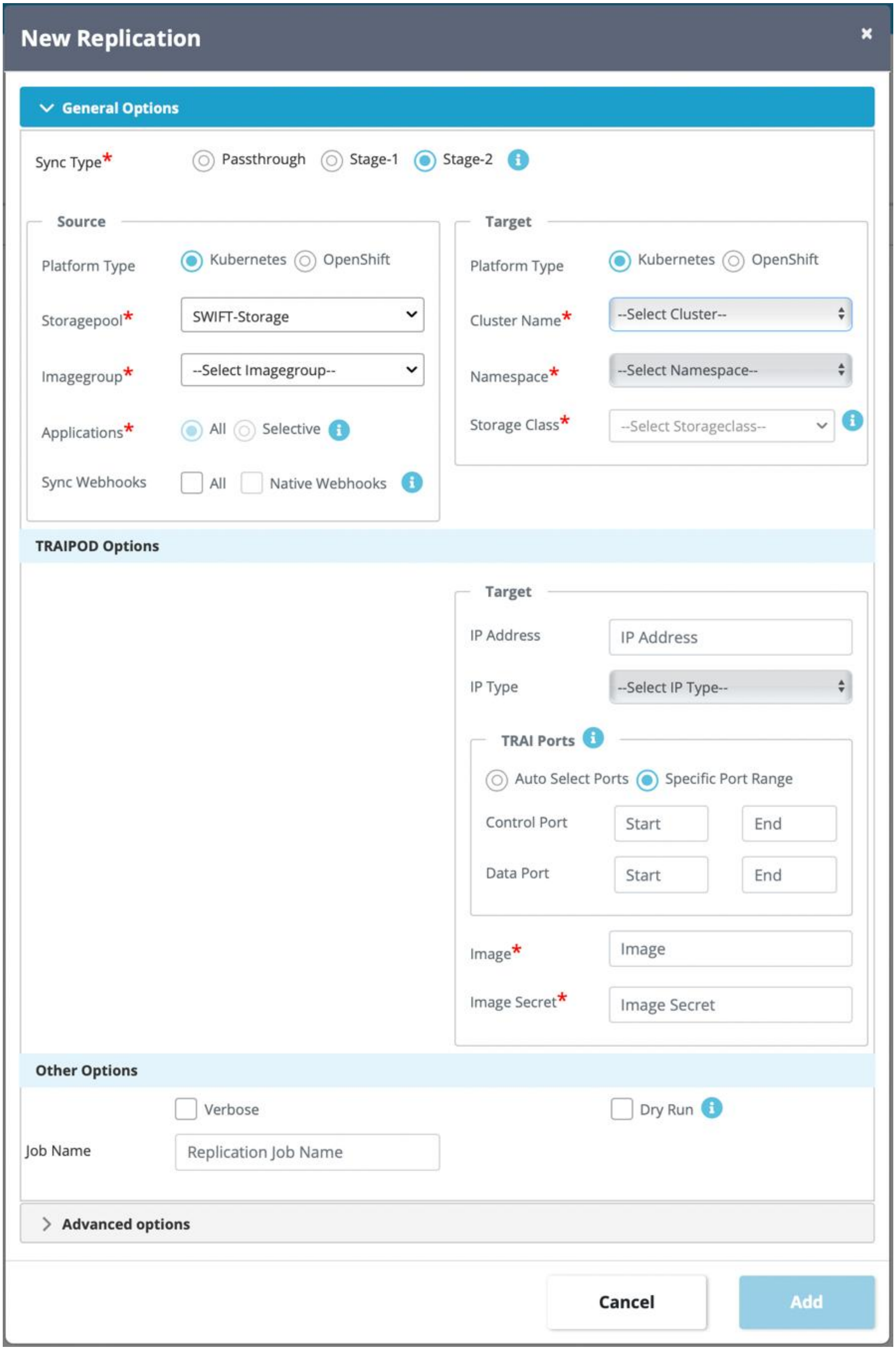
Image Secret*: Image Secret

Other Options

□ Verbose □ Dry Run

Job Name: Replication Job Name

Stage2 syncs will have similar inputs like Stage1 sync and only difference is Storage pool and Image-Group selection is for the source of the sync while cluster and namespace is selected as a target for the sync.



New Replication

General Options

Sync Type* ○ Passthrough ○ Stage-1 ● Stage-2

Source

Platform Type ● Kubernetes ○ OpenShift

Storagepool* SWIFT-Storage

Imagegroup* --Select Imagegroup--

Applications* ● All ○ Selective

Sync Webhooks □ All □ Native Webhooks

Target

Platform Type ● Kubernetes ○ OpenShift

Cluster Name* --Select Cluster--

Namespace* --Select Namespace--

Storage Class* --Select Storageclass--

TRAIPOD Options

Target

IP Address: IP Address

IP Type: --Select IP Type--

TRAI Ports

○ Auto Select Ports ● Specific Port Range

Control Port: Start, End

Data Port: Start, End

Image*: Image

Image Secret*: Image Secret

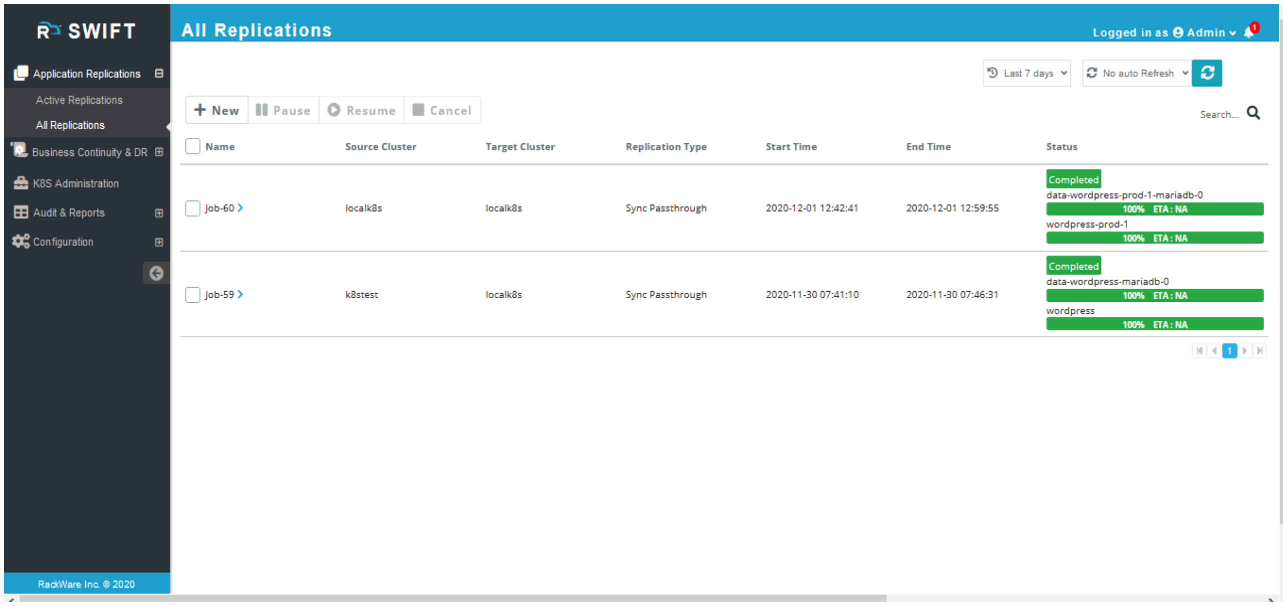
Other Options

□ Verbose □ Dry Run

Job Name: Replication Job Name

Advanced options

Once you have sync configured, press the add button, and sync will start immediately. If you also created a DR policy along with the sync, then the sync will start at the specified start time as per the DR policy schedule. You can monitor all running syncs from the 'Active Replications' as well as the 'All Replications' submenus.



Name	Source Cluster	Target Cluster	Replication Type	Start Time	End Time	Status
Job-60	localk8s	localk8s	Sync Passthrough	2020-12-01 12:42:41	2020-12-01 12:59:55	Completed data-wordpress-prod-1-mariadb-0 100% ETA : NA wordpress-prod-1 100% ETA : NA
Job-59	k8stest	localk8s	Sync Passthrough	2020-11-30 07:41:10	2020-11-30 07:46:31	Completed data-wordpress-mariadb-0 100% ETA : NA wordpress 100% ETA : NA

Sync Advanced Options

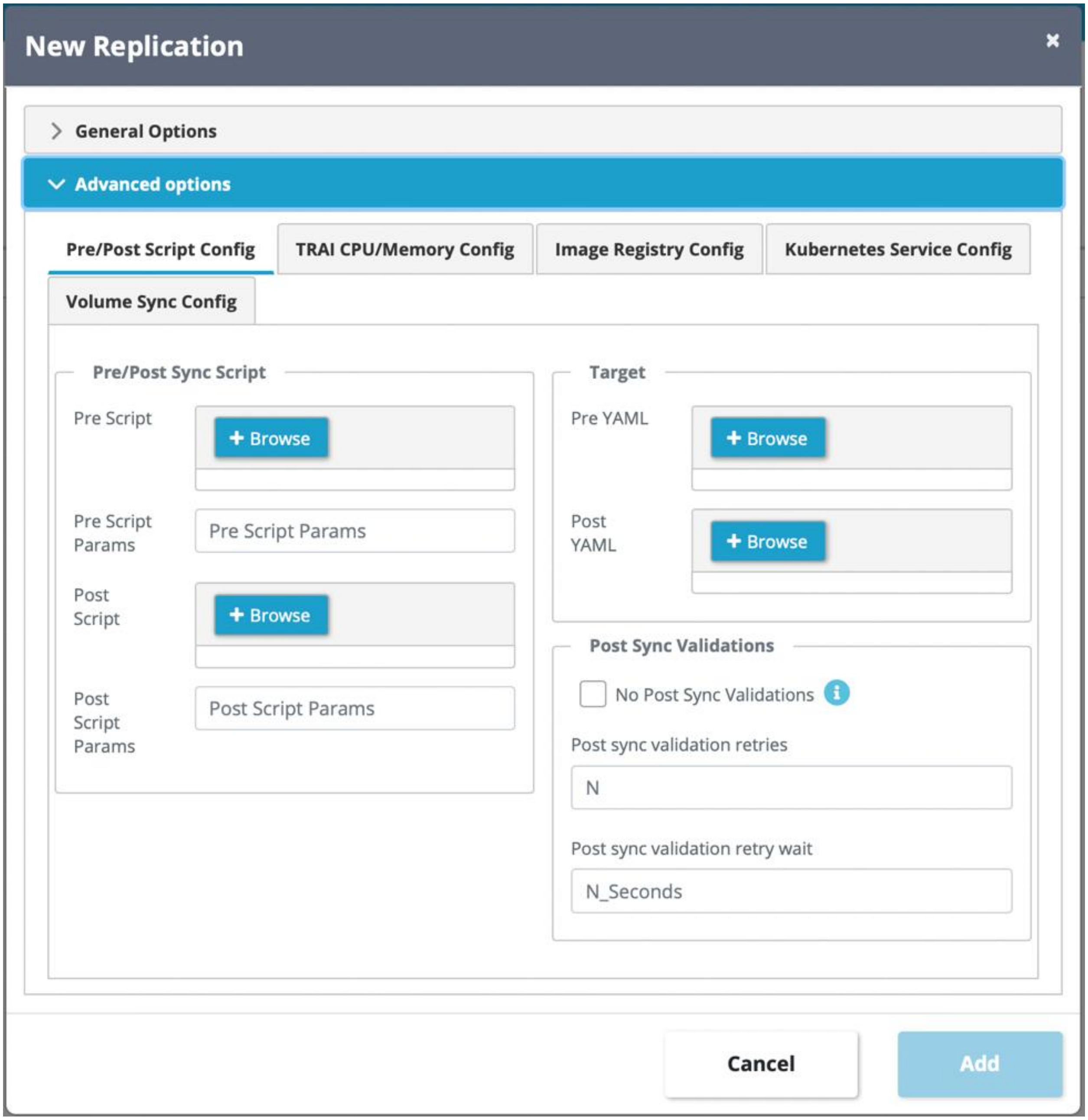
Inputs here in configs allow you to transform the input configuration on the replicated target cluster side. Optionally, you can set all these configurations, one-time, using cluster-administration menu, which will be used by all syncs for that specific cluster. Sync specific configuration will take priority over cluster level defaults.

Depending on the selected sync type, certain options will be unavailable.

Pre/Post scripts and YAMLs

From the Advanced Options menu, you can select pre and post scripts, which are optional. If you configure any of the scripts, the pre-script is run before sync starts, and the post-script is run post sync. In case of the passthrough sync, you can use them to configure clusters on both sides as a pre or post sync event. For Stage1 and Stage2 sync that will be the source and the target cluster respectively.

The scripts need to be uploaded to the SWIFT server and should have appropriate execute permissions (They will be run as a root user on the SWIFT server). If the pre-script fails, the sync will fail, while a failing post-script is just logged as a warning during sync. Output for both the pre and post-script is logged as well as recorded as the sync output (You can always track it as part of sync job).



New Replication

› General Options

⌄ Advanced options

Pre/Post Script Config | TRAI CPU/Memory Config | Image Registry Config | Kubernetes Service Config | Volume Sync Config

Pre/Post Sync Script

Pre Script: + Browse

Pre Script Params: Pre Script Params

Post Script: + Browse

Post Script Params: Post Script Params

Target

Pre YAML: + Browse

Post YAML: + Browse

Post Sync Validations

□ No Post Sync Validations

Post sync validation retries: N

Post sync validation retry wait: N_Seconds

Cancel | Add

Just like the pre and post-script, you can also configure pre and post YAML to apply (patch) for the cluster on either side. You get four interfaces for the YAML apply:

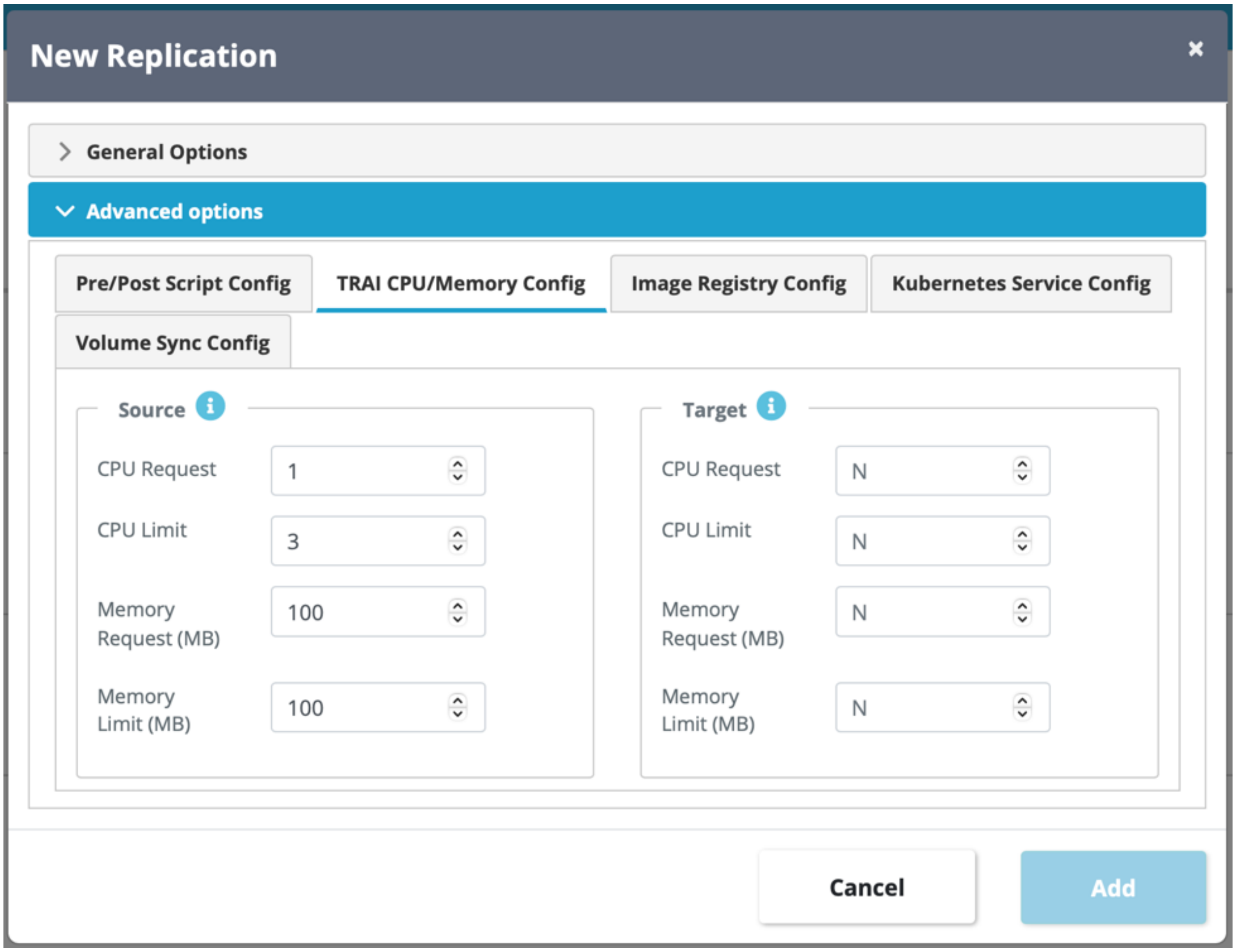
1. Pre-sync on the source cluster
2. Pre-sync on the target cluster
3. Post-sync on the source cluster
4. Post-sync on the target cluster

Like pre/post-script, the failure of pre-sync YAML apply for either side of the clusters is treated as a sync error, while post-sync YAML apply failures would simply result in warnings during sync. Both pre and post

YAML applies for either side of the clusters are tracked as sync progress and so recorded in the sync job too. Input YAML is also upfront validated during the sync for any syntax errors.

TRAI Configs

In addition to pre/post-scripts and YAML, you can also change TRAI resource configs from the advanced sync options menu. Depending on sync type, you can enter TRAI config options for source and/or target cluster.

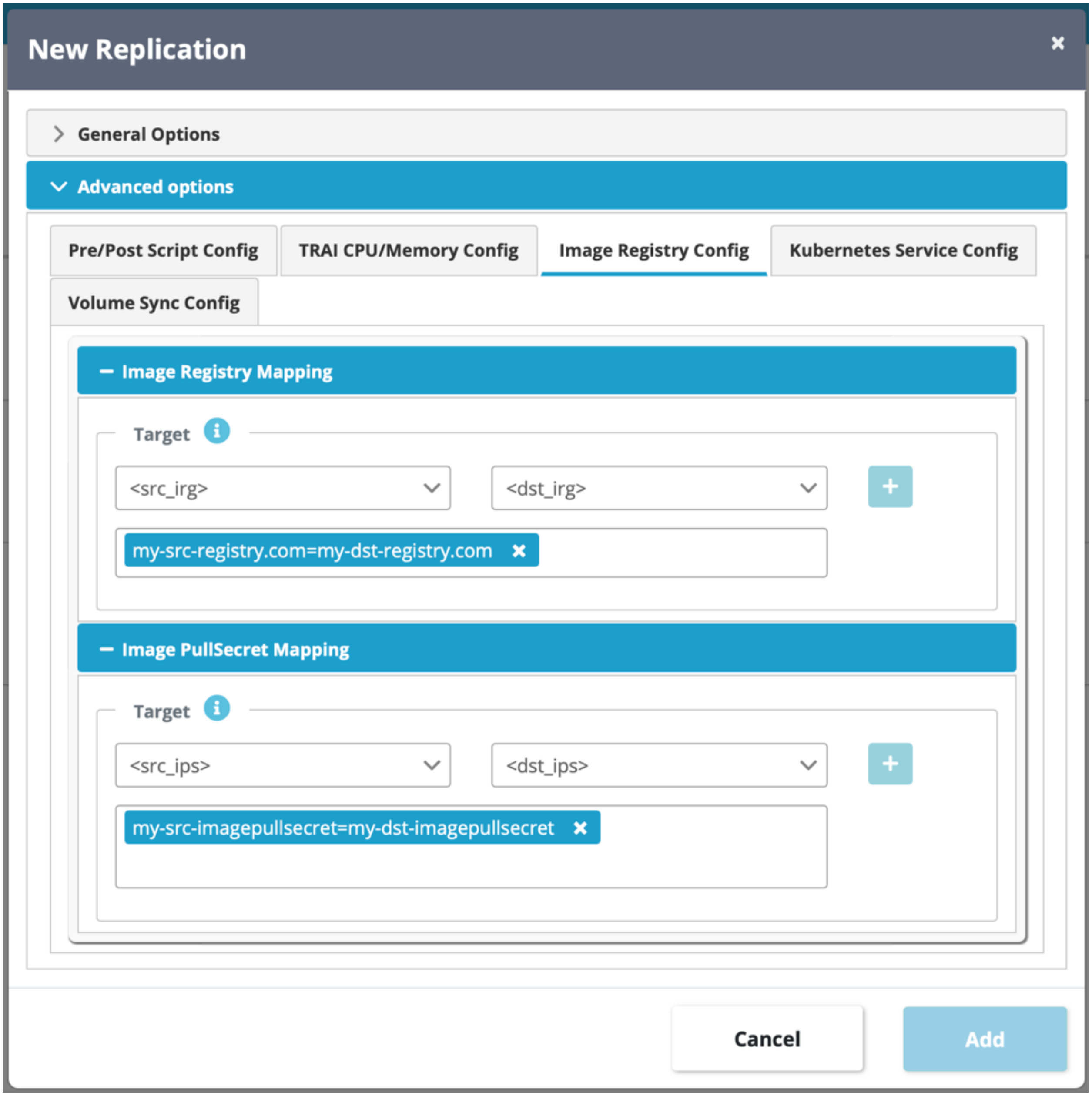


	Source	Target
CPU Request	1	N
CPU Limit	3	N
Memory Request (MB)	100	N
Memory Limit (MB)	100	N

TRAI configs determine how many resources are allocated to SWIFT TRAI Pod, which is a transient Pod run during a sync and used as a staging environment. The request for CPU and Memory determines base requested resource sizes for the TRAI Pod, while limits specify max sizes that can be requested. Depending on the remote Kubernetes or OpenShift cluster condition, TRAI Pod may get allocated anywhere between the requested and limit sizes.

Image Registry Mappings

Configs in this section allow you to specify image-registry string mapping, which is replaced as image-path string for the target replicated Pods. You can either specify part of registry name/path or give full name/path with mapping on the target side.



my-src-registry.com=my-dst-registry.com

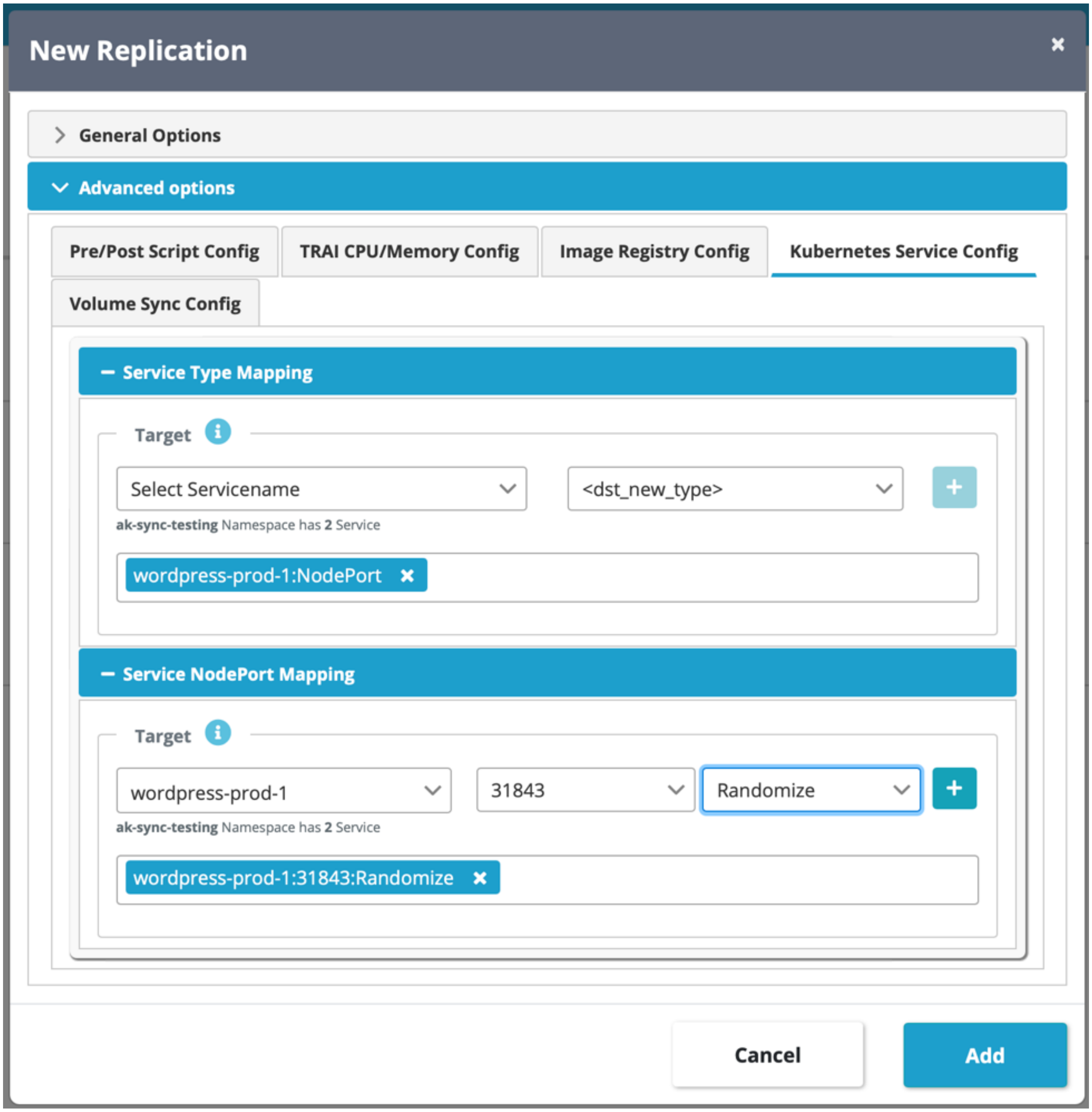
my-src-imagepullsecret=my-dst-imagepullsecret

The pull-secret mapping essentially configures ImagePullSecret config for the target Pods. You can map these individually or configure All/Any to a new pull-secret name with input rules here.

Note that none of these configs do any changes in your source cluster.

Service Mappings

The service mappings or configs allow you to change service types or NodePorts for the service.



New Replication

General Options

Advanced options

Pre/Post Script Config | TRAI CPU/Memory Config | Image Registry Config | Kubernetes Service Config | Volume Sync Config

Service Type Mapping

Target

Select Servicename | <dst_new_type>

ak-sync-testing Namespace has 2 Service

wordpress-prod-1:NodePort

Service NodePort Mapping

Target

wordpress-prod-1 | 31843 | Randomize

ak-sync-testing Namespace has 2 Service

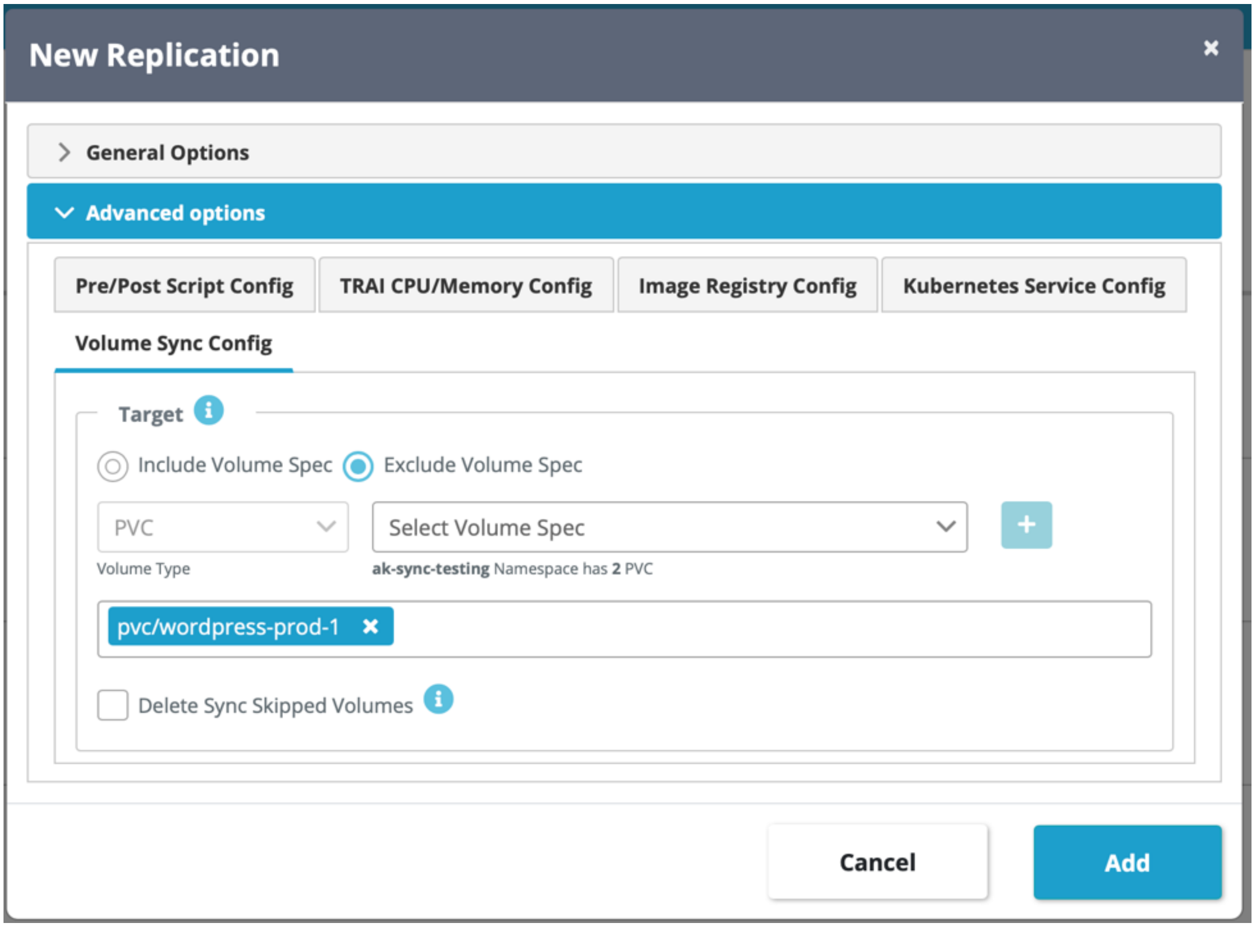
wordpress-prod-1:31843:Randomize

These options are applicable only for passthrough and Stage2 syncs. Depending on sync type and selected sync object and namespace filters, the service and port dropdown will list existing services and ports (with those input object filters). Optionally, you can configure ports explicitly too.

The 'Randomize' option for NodePort will mean service get random NodePort from the service port range of the target cluster.

Volume Sync Config

Options in this section allow you to filter or selectively include persistent volumes (PVs) and claims (PVCs) for the sync. These options are applicable for all sync types.



New Replication ×

○ Include Volume Spec ● Exclude Volume Spec

Volume Type: PVC

Select Volume Spec

ak-sync-testing Namespace has **2** PVC

pvc/wordpress-prod-1 ×

□ Delete Sync Skipped Volumes

These options are quite useful in certain cases. Like, for example, say if you want SWIFT to do initial replication without any filters but then exclude certain static application volumes from the target cluster for recurring DR syncs, then you can configure exclude list. You may want to do that in cases where application volume is static and now it has target specific changes that you don't want recurring syncs to overwrite.

Note that you can only specify include or exclude list for a sync run, as both are mutually exclusive inputs. Include and exclude lists both only work on sync selected source volumes, so if you specify a volume in either list that the current sync with its input object filters will not sync, then the inclusion/exclusion input is ignored for the volume. Inclusion list also technically works as a filter and so excludes volumes, as when specified, only specified volumes will be synced.

The delete checkbox, when selected, will delete volumes from the target cluster which are skipped by the sync, though they were selected with sync input object filters.

Intra-cluster and inter-cluster syncs

You can run multiple concurrent syncs between the same set of clusters (inter-cluster syncs). Make sure, though, that you do not create conflicting syncs where a group of objects or volumes overlap between such concurrent syncs, as SWIFT will not prevent those. The behavior is in line with Kubernetes, where the cluster also doesn't stop you from creating overlapping higher-level objects which use conflicting label selectors.

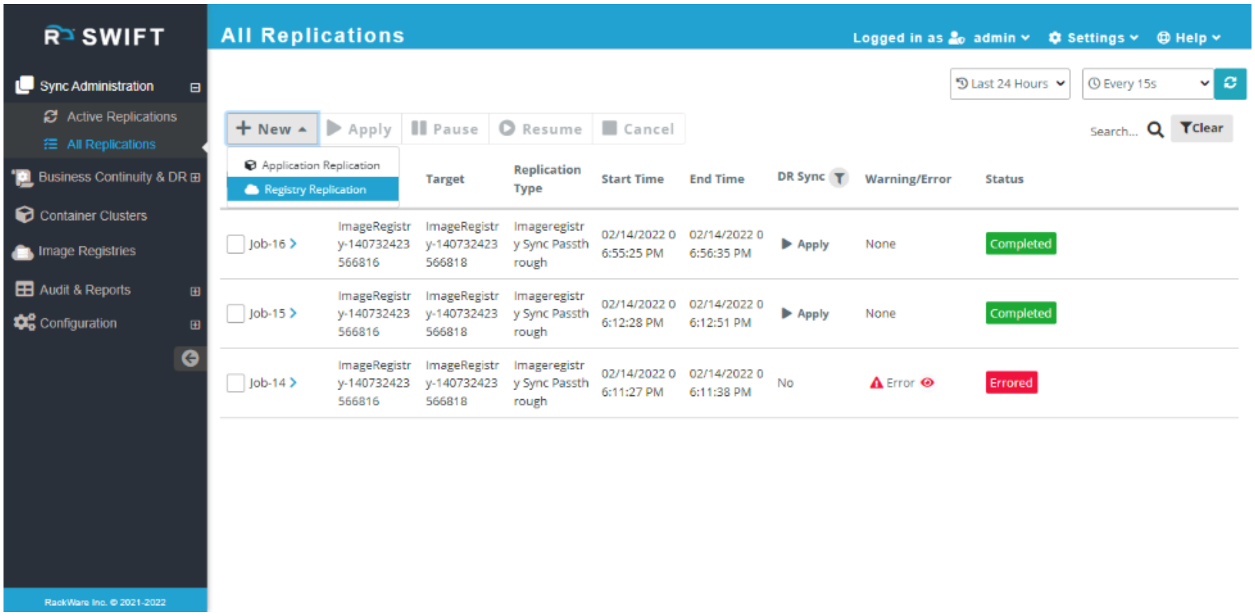
You can also run sync between different namespaces of the same cluster (intra-cluster syncs), and for such cases, you will configure the same cluster as a source and a target of the sync. The SWIFT does not support syncing within the same namespace of the same cluster.

For intra-cluster syncs, make sure you don't sync NodePort or LoadBalancer service between different namespaces of the same cluster without either changing type or explicitly specifying new NodePorts for the service. Please refer to the advanced sync configs from earlier section to know how to specify either of those sync mappings for a service.

A synchronization between container registries

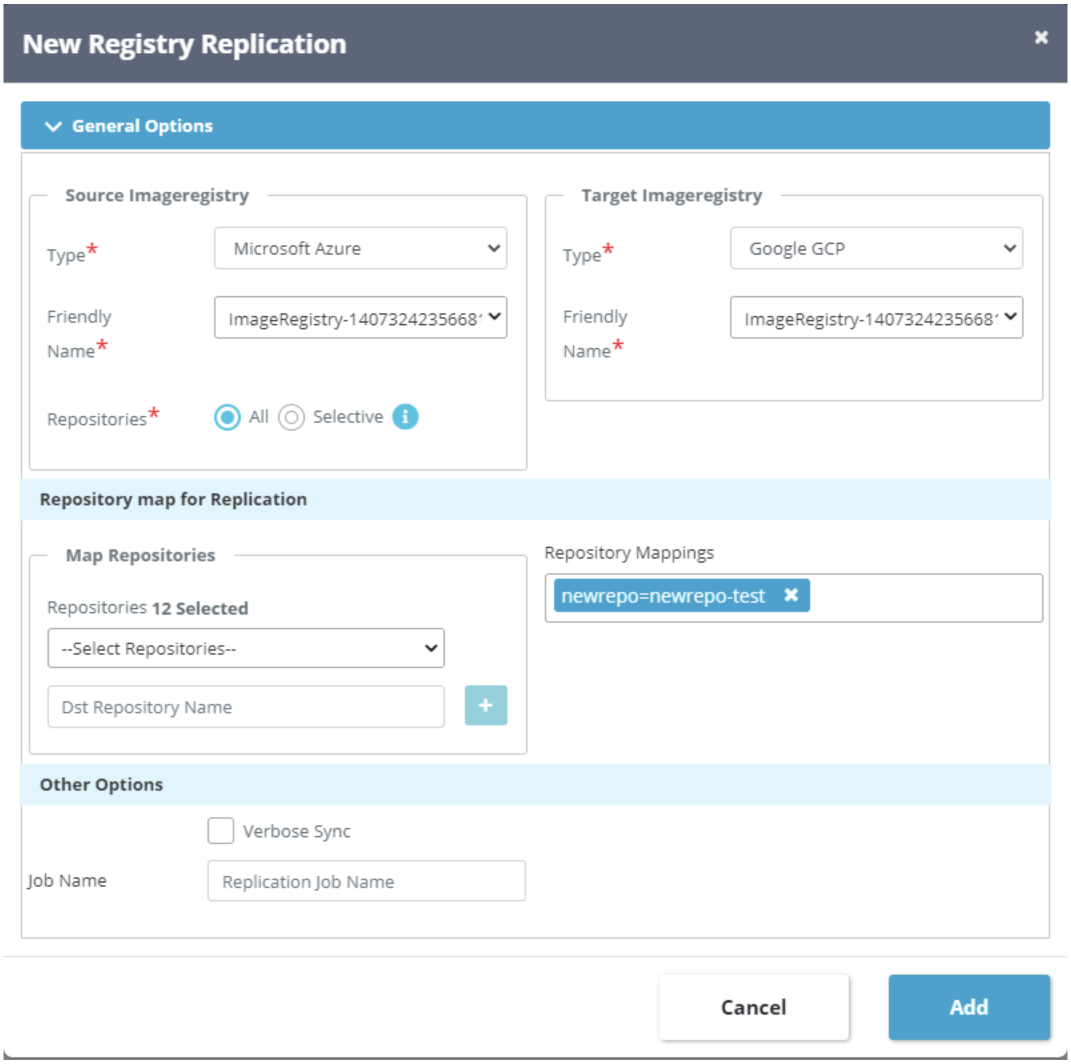
Use the below steps if you want to initiate a sync between two container registries, which are added to the SWIFT. The steps remain the same irrespective of where the managed registry is located (local vs. in the cloud).

Connect to the SWIFT dashboard and navigate to the 'Sync Administration' menu and the 'All Replications' submenu.



		Target	Replication Type	Start Time	End Time	DR Sync	Warning/Error	Status
□ Job-16	ImageRegistry-140732423566816	ImageRegistry-140732423566818	Imageregistry Sync Passthrough	02/14/2022 06:55:25 PM	02/14/2022 06:56:35 PM	▶ Apply	None	Completed
□ Job-15	ImageRegistry-140732423566816	ImageRegistry-140732423566818	Imageregistry Sync Passthrough	02/14/2022 06:12:28 PM	02/14/2022 06:12:51 PM	▶ Apply	None	Completed
□ Job-14	ImageRegistry-140732423566816	ImageRegistry-140732423566818	Imageregistry Sync Passthrough	02/14/2022 06:11:27 PM	02/14/2022 06:11:38 PM	No	⚠ Error	Errored

Press the '+ New' button and the 'Registry Replication' submenu. Select your source and target registries and optionally specific repository and tag and for each registry on the new input dialog. If you select the 'all' repositories option, then all container image repositories will be synced.



General Options

Source Imageregistry

Type* Microsoft Azure

Friendly Name* ImageRegistry-1407324235668

Repositories* ● All ○ Selective

Target Imageregistry

Type* Google GCP

Friendly Name* ImageRegistry-1407324235668

Repository map for Replication

Map Repositories

Repositories 12 Selected

--Select Repositories--

Dst Repository Name

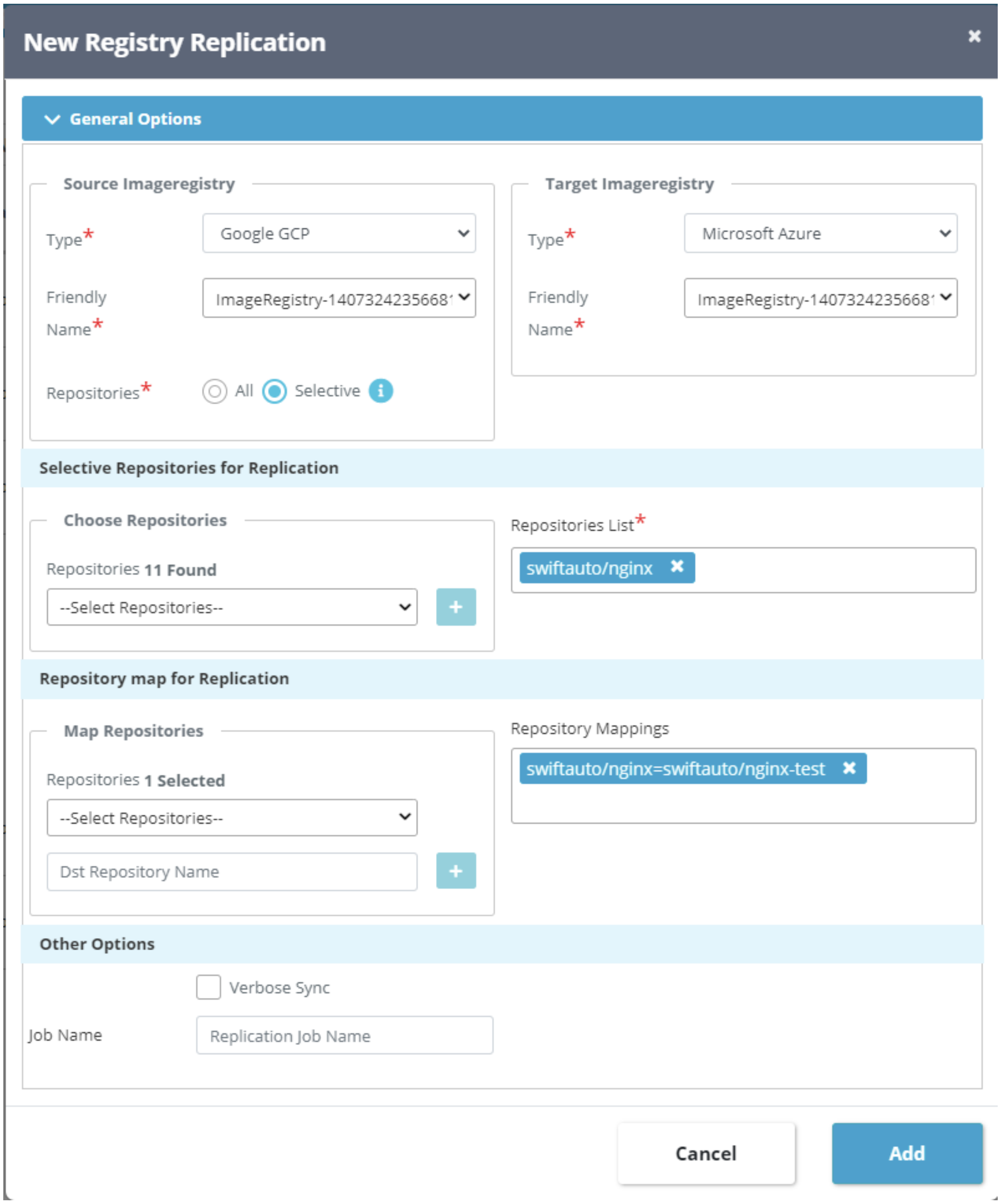
Repository Mappings

newrepo=newrepo-test ×

Other Options

□ Verbose Sync

Job Name: Replication Job Name



New Registry Replication

General Options

Source Imageregistry

Type* Google GCP

Friendly Name* ImageRegistry-1407324235668'

Repositories* ○ All ● Selective

Target Imageregistry

Type* Microsoft Azure

Friendly Name* ImageRegistry-1407324235668'

Selective Repositories for Replication

Choose Repositories

Repositories **11 Found**

--Select Repositories--

Repositories List* swiftauto/nginx ✖

Repository map for Replication

Map Repositories

Repositories **1 Selected**

--Select Repositories--

Dst Repository Name

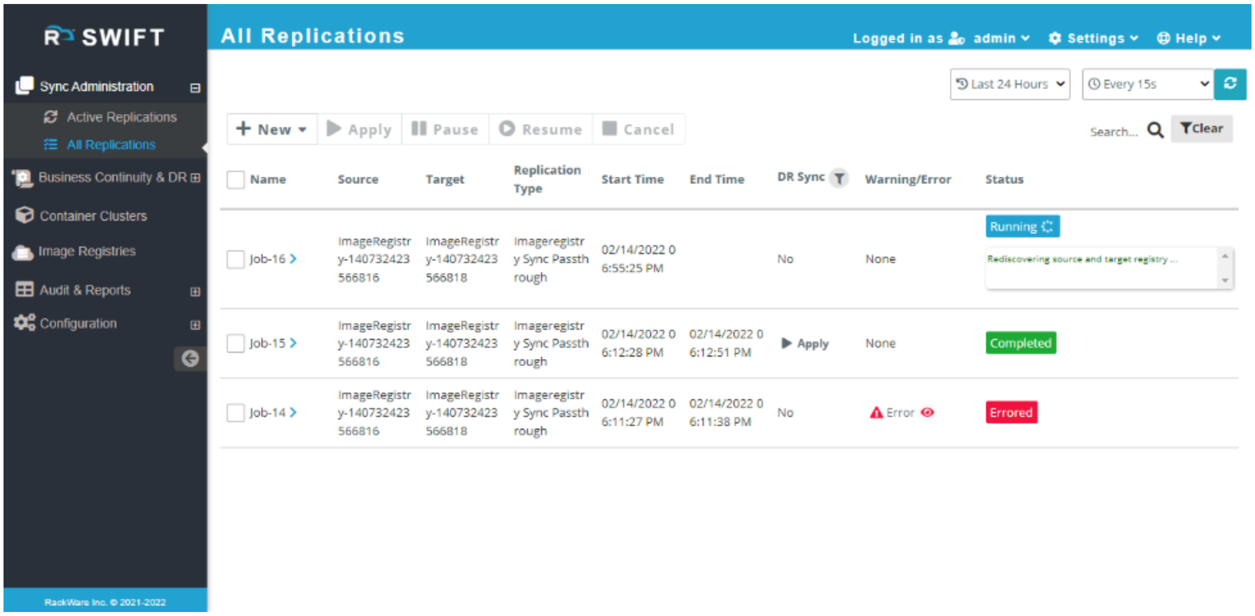
Repository Mappings swiftauto/nginx=swiftauto/nginx-test ✖

Other Options

□ Verbose Sync

Job Name Replication Job Name

Once you have sync configured, press the add button, and sync will start immediately. If you also created a DR policy along with the sync, then the sync will start at the specified start time as per the DR policy schedule. You can monitor all running syncs from the 'Active Replications' as well as the 'All Replications' submenus.



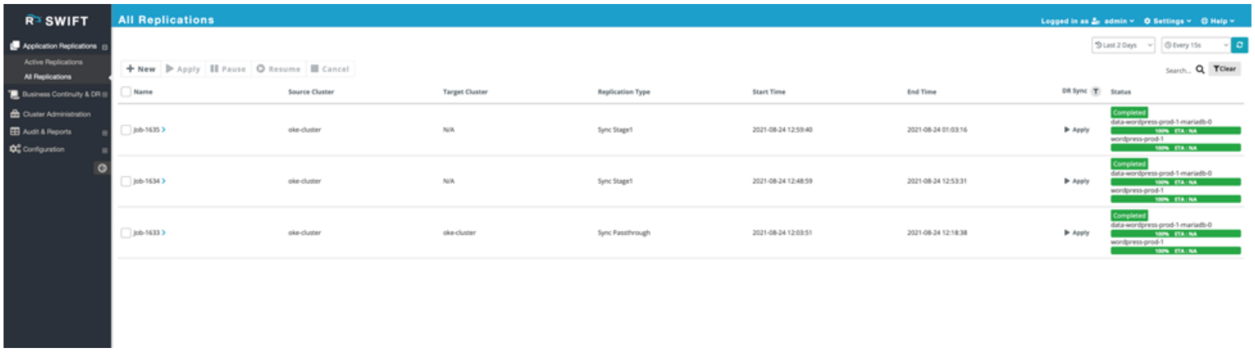
Name	Source	Target	Replication Type	Start Time	End Time	DR Sync	Warning/Error	Status
Job-16	ImageRegistry-140732423566816	ImageRegistry-140732423566818	Imageregistry Sync Passthrough	02/14/2022 06:55:25 PM		No	None	Running Rediscovering source and target registry ...
Job-15	ImageRegistry-140732423566816	ImageRegistry-140732423566818	Imageregistry Sync Passthrough	02/14/2022 06:12:28 PM	02/14/2022 06:12:51 PM	▶ Apply	None	Completed
Job-14	ImageRegistry-140732423566816	ImageRegistry-140732423566818	Imageregistry Sync Passthrough	02/14/2022 06:11:27 PM	02/14/2022 06:11:38 PM	No	Error	Errored

Configuring DR policies

There are two ways you can configure a DR policy in SWIFT. The below sections highlight both ways. The subsequent steps also highlight specific steps for applying policies for image registry syncs.

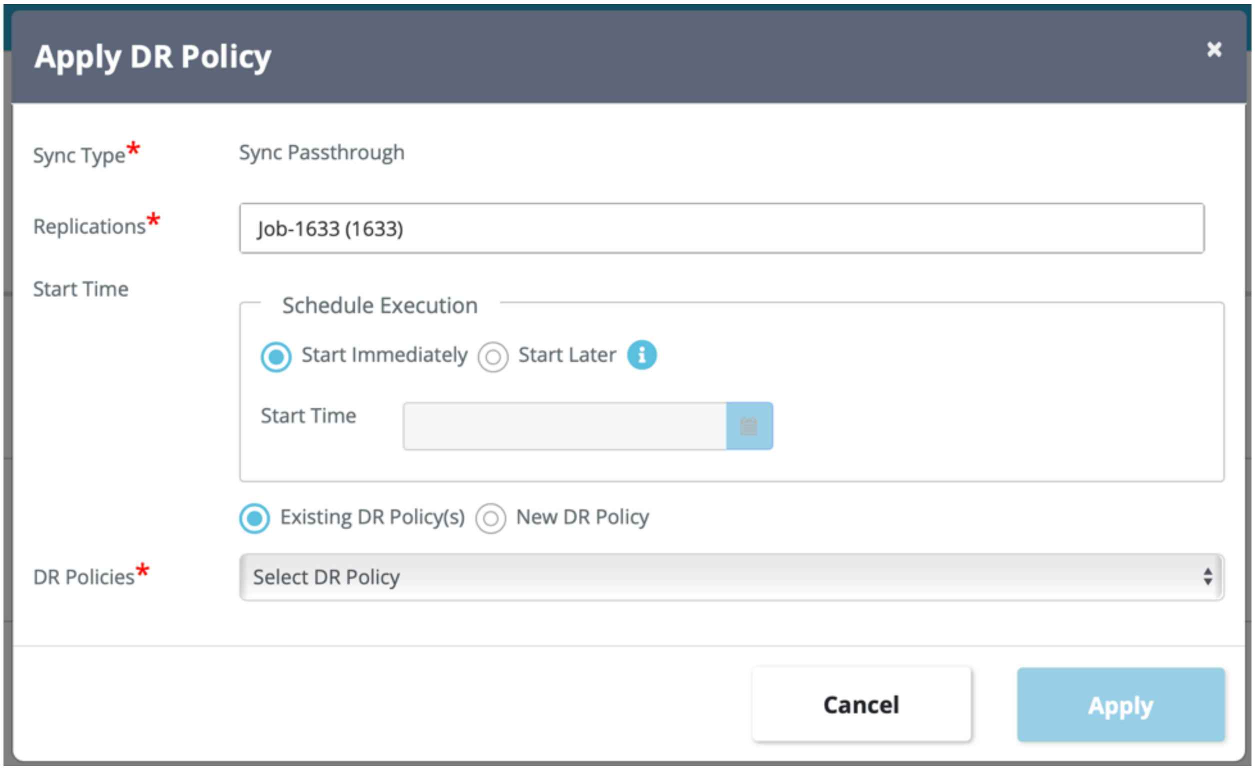
Configure a policy for running or completed application sync

It is very common with SWIFT to run a test sync once to validate all configuration is okay, and only then convert the completed sync job to a DR policy. You can refer to earlier section to learn how to start a fresh sync or replication. Once the test sync completes, you can use below steps to create a DR policy for it.



Name	Source Cluster	Target Cluster	Replication Type	Start Time	End Time	DR Sync	Status
Job-1635	oke-cluster	N/A	Sync Stage1	2021-08-24 12:59:40	2021-08-24 01:03:16	▶ Apply	Completed data-wordpress-prod-1-mariadb-0 100% ETA : NA wordpress-prod-1 100% ETA : NA
Job-1634	oke-cluster	N/A	Sync Stage1	2021-08-24 12:48:59	2021-08-24 12:53:31	▶ Apply	Completed data-wordpress-prod-1-mariadb-0 100% ETA : NA wordpress-prod-1 100% ETA : NA
Job-1633	oke-cluster	oke-cluster	Sync Passthrough	2021-08-24 12:03:51	2021-08-24 12:18:38	▶ Apply	Completed data-wordpress-prod-1-mariadb-0 100% ETA : NA wordpress-prod-1 100% ETA : NA

1. Go to the 'All Replications' menu in the SWIFT dashboard.
2. Find the required sync job that you want to convert to a DR policy. You can optionally use the time and other filters available on the page to locate the required sync job.
3. Select the job and click on the 'Apply' button in its row.
4. Either select the existing policy or create a new one and click on the 'Apply' button.



Apply DR Policy

Sync Type*: Sync Passthrough

Replications*: Job-1633 (1633)

Start Time

Schedule Execution

● Start Immediately ○ Start Later

Start Time

● Existing DR Policy(s) ○ New DR Policy

DR Policies*: Select DR Policy

If you choose to create a new policy in this flow, you will input all details for the new policy, such as frequency/schedule, email alert list, etc. You can refer to all inputs for a new policy creation in the next section.

Note that in this flow, once you create a policy with existing sync job, then the sync configuration is picked up from the selected sync job. You can always apply the policy to more than one sync job of the same type. Applied sync jobs for a policy need not be syncing between the same set of clusters.

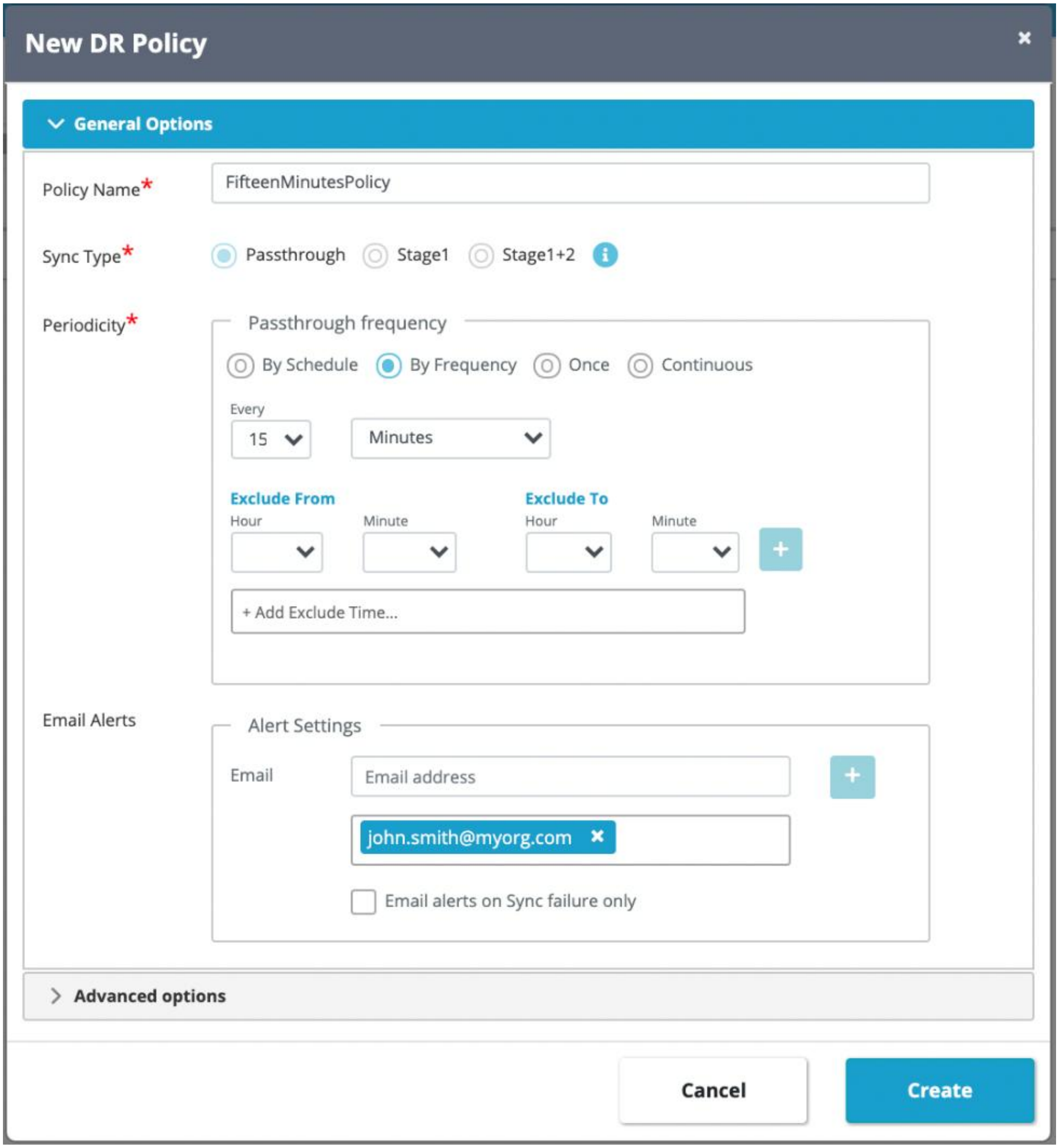
Configure a policy for running or completed registry sync

The steps remain the same as application syncs and policy apply.

1. Go to the ‘All Replications’ menu in the SWIFT dashboard.
2. Find the required registry sync job that you want to convert to a DR policy. You can optionally use the time and other filters available on the page to locate the required sync job.
3. Select the job and click on the ‘Apply’ button in its row.
4. Either select the existing policy or create a new one and click on the ‘Apply’ button.

Configure a policy for application syncs from the BCDR menu

Click on the Business Continuity and Disaster Recovery (BCDR) menu in the SWIFT dashboard. You will see the option to create a ‘New’ policy, click on that.



New DR Policy ×

General Options

Policy Name*: FifteenMinutesPolicy

Sync Type*: ● Passthrough ○ Stage1 ○ Stage1+2

Periodicity*: Passthrough frequency

○ By Schedule ● By Frequency ○ Once ○ Continuous

Every: 15 Minutes

Exclude From — Hour, Minute; Exclude To — Hour, Minute; +

+ Add Exclude Time...

Email Alerts: Alert Settings

Email: Email address +

john.smith@myorg.com ×

□ Email alerts on Sync failure only

Advanced options

Cancel Create

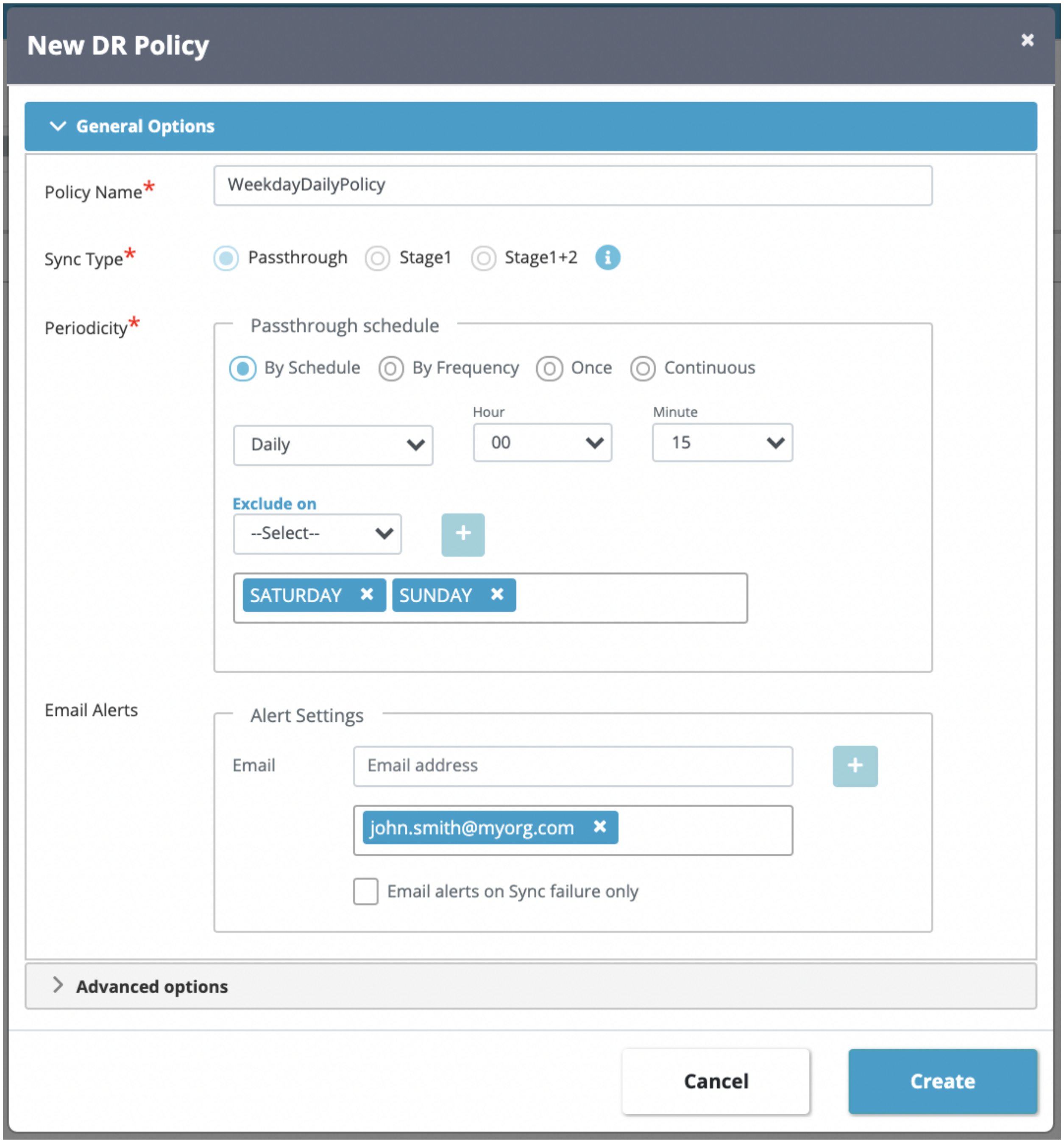
Enter the policy name, policy schedule, and email alert list. The email alert list is optional. You can also configure email alerts only for sync failures. The policy schedule can be one of the:

- By Frequency

- One-time sync

- Custom schedule

- Continuous



New DR Policy

General Options

Policy Name*: WeekdayDailyPolicy

Sync Type*: ● Passthrough ○ Stage1 ○ Stage1+2

Periodicity*: Passthrough schedule

● By Schedule ○ By Frequency ○ Once ○ Continuous

Daily, Hour: 00, Minute: 15

Exclude on: --Select--

SATURDAY ✖ SUNDAY ✖

Email Alerts: Alert Settings

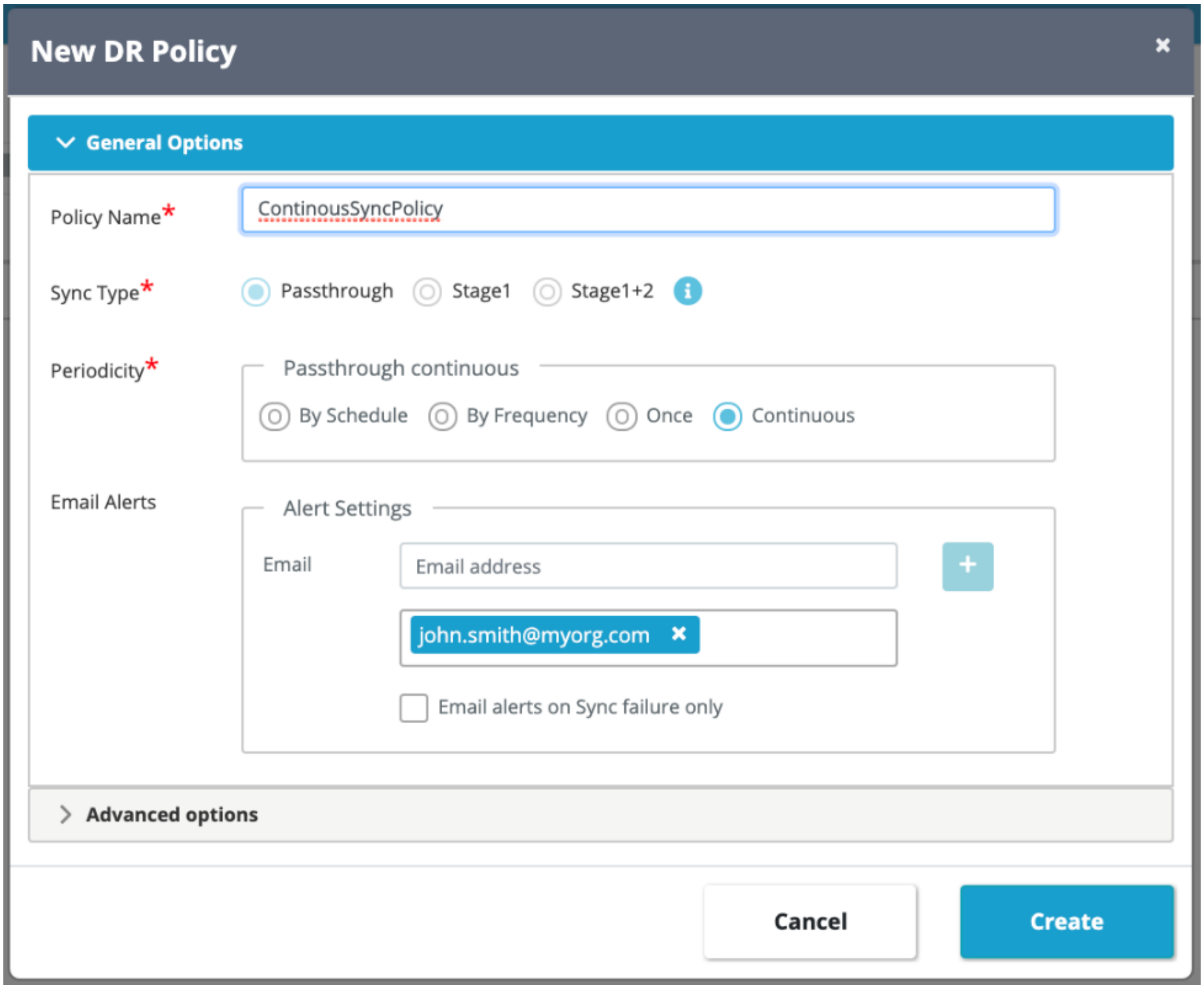
Email: Email address

john.smith@myorg.com ✖

□ Email alerts on Sync failure only

You can also input an exclude or blackout window for syncs on specific dates with weekly schedules, while for frequency-based schedule, the blackout window will be for specific days of the week.

Note that the blackout window configuration is optional.



New DR Policy

General Options

Policy Name* ContinousSyncPolicy

Sync Type* ● Passthrough ○ Stage1 ○ Stage1+2

Periodicity* Passthrough continuous ○ By Schedule ○ By Frequency ○ Once ● Continuous

Email Alerts

Alert Settings

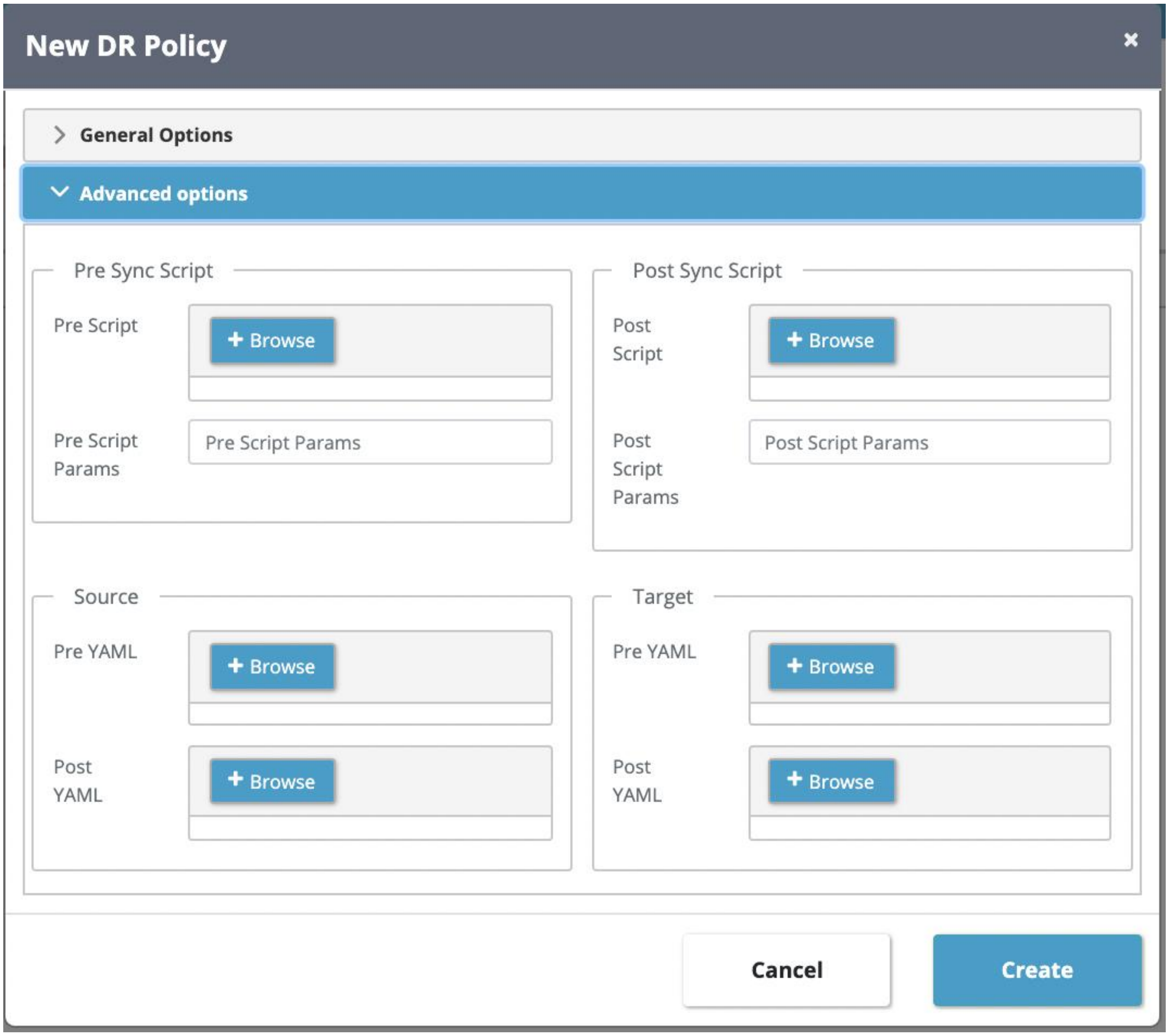
Email: Email address

john.smith@myorg.com

□ Email alerts on Sync failure only

Selecting the continuous sync will do the syncs back-to-back.

You can also apply YAML or run a script for pre/post DR policy failover and failback operation event by selecting those configs through the ‘Advanced Options’ config on the policy create dialog.

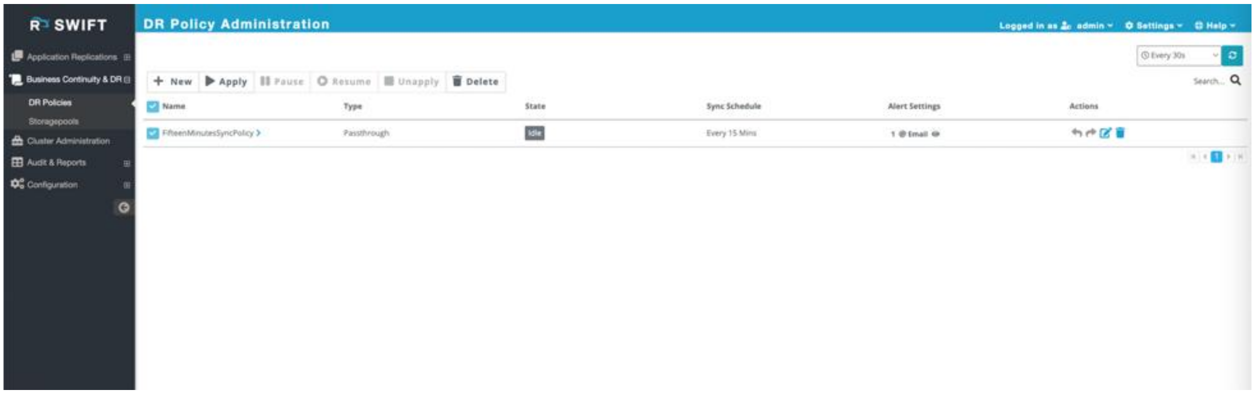


The selected YAML or script file will be run from the SWIFT server and needs to be present at the specified path all the time. The uploaded files are stored on the SWIFT server. The script will be run as the root user. This YAML pre/post config is ignored if the policy is applied to a registry sync.

If a pre-script or YAML apply fails, then the DR policy failover or failback operation will fail, while failures for post operation are only logged in the operation logs.

The input pre/post config here is run as part of the policy failover and fallback operations. Optionally, you can configure pre/post scripts and YAMLs as part of each sync operation from the policy too.

Once you are done with a policy configuration, press the ‘Create’ button and the new policy will be created. At this stage, policy will be in the IDLE state as it is not applied to any sync configuration yet. Refer to the next section for exact steps to apply the policy to a sync job.



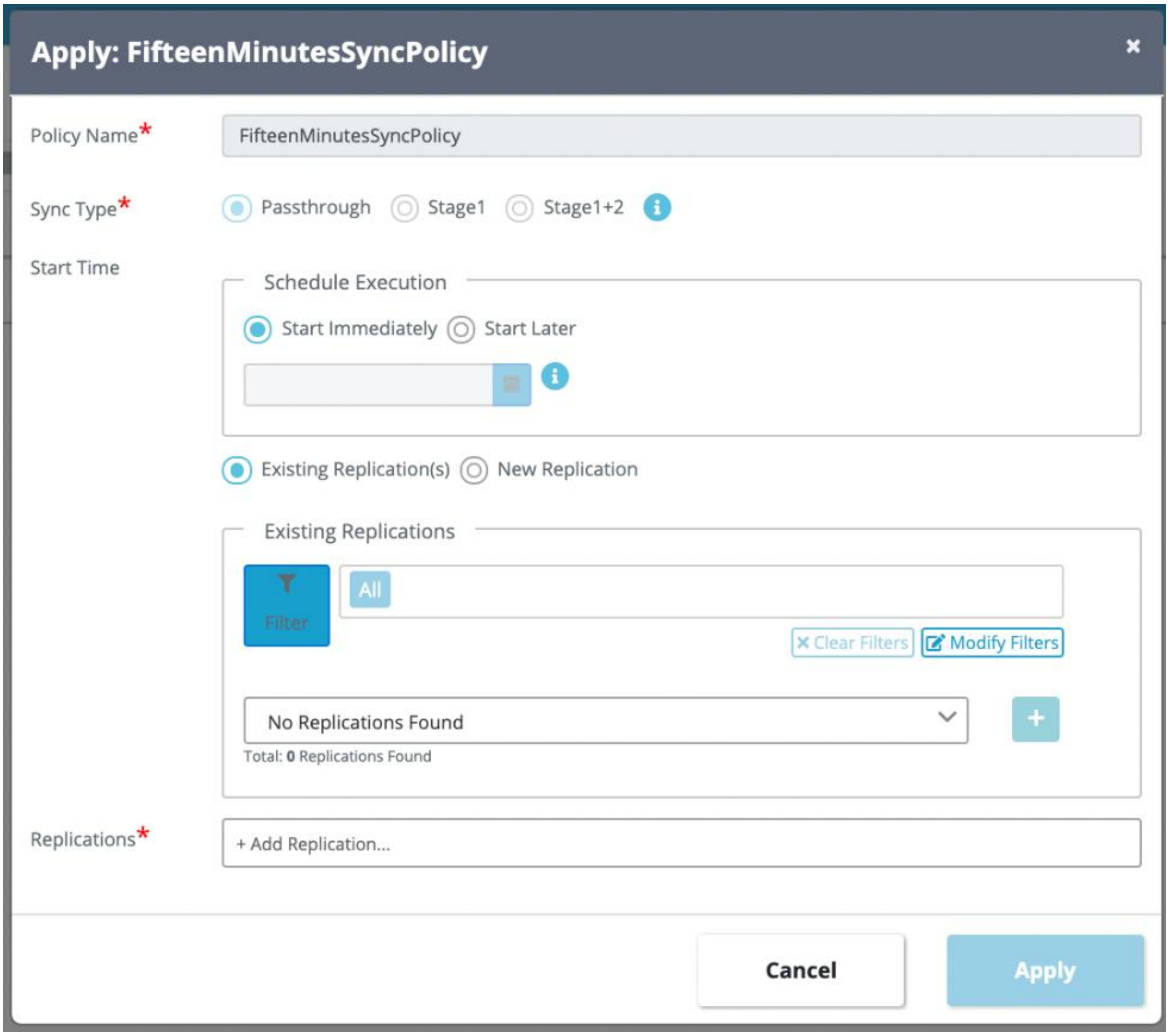
Name	Type	State	Sync Schedule	Alert Settings	Actions
FifteenMinutesSyncPolicy	Passthrough	Idle	Every 15 Mins	1 Email	

Applying the newly created policy to application syncs

Go the Business Continuity and Disaster Recovery (BCDR) menu. You will see list of all policies on this page. Select the policy you want to apply and click on the 'Apply' button.

You will have a choice of applying policy to

Existing application replication



Apply: FifteenMinutesSyncPolicy

Policy Name* FifteenMinutesSyncPolicy

Sync Type* ● Passthrough ○ Stage1 ○ Stage1+2

Start Time

Schedule Execution

● Start Immediately ○ Start Later

● Existing Replication(s) ○ New Replication

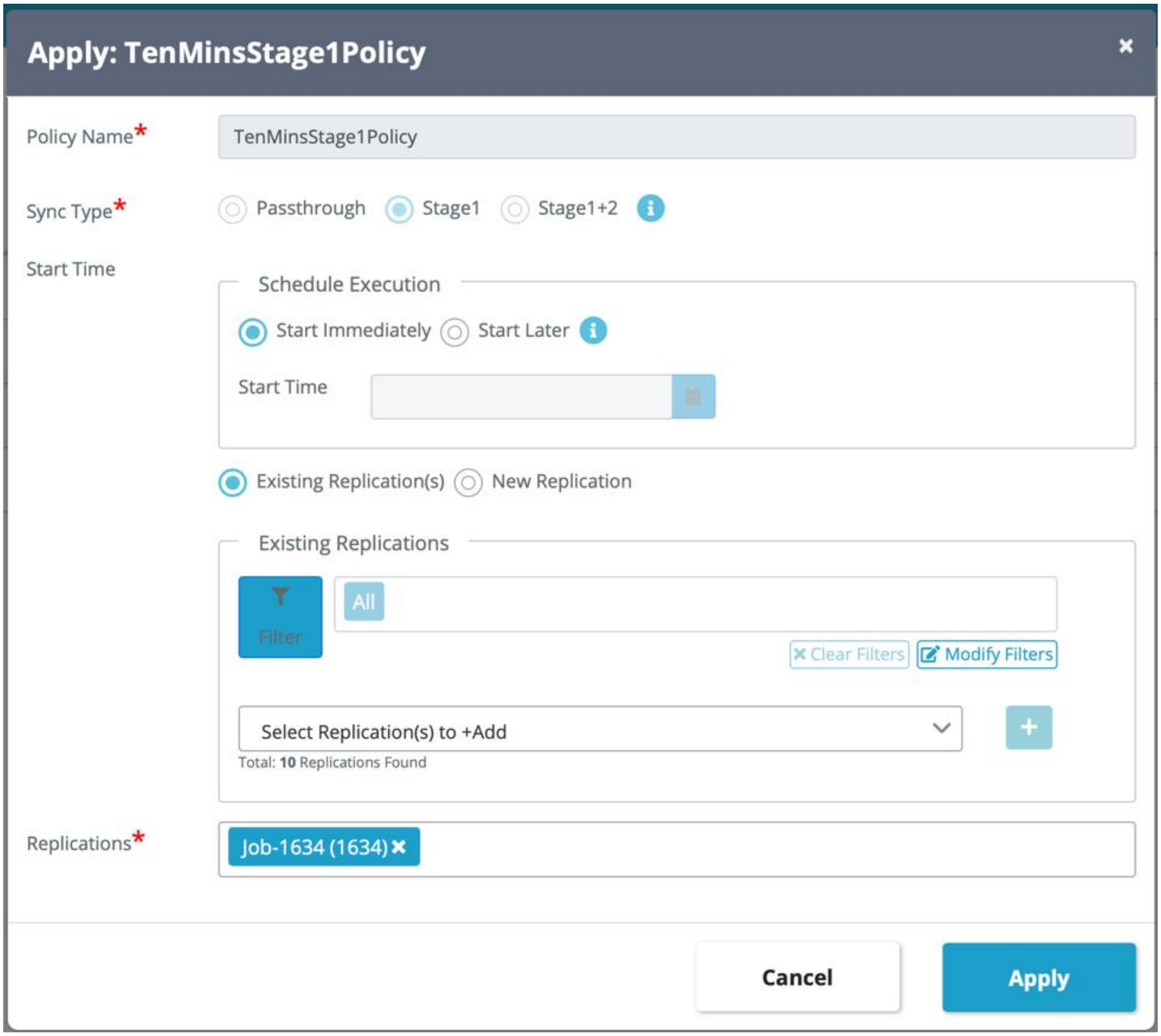
Existing Replications

No Replications Found

Total: 0 Replications Found

Replications* + Add Replication...

If you select existing replication, then you will get an option to select the existing sync jobs as a drop-down list. Note that only those jobs will be listed which match the policy-type. You can select more than one jobs to apply and press the 'Apply' button. The policy will schedule running exact replications which were done under those sync jobs.



Apply: TenMinsStage1Policy

Policy Name* TenMinsStage1Policy

Sync Type* ○ Passthrough ● Stage1 ○ Stage1+2

Start Time

Schedule Execution

● Start Immediately ○ Start Later

Start Time

● Existing Replication(s) ○ New Replication

Existing Replications

All

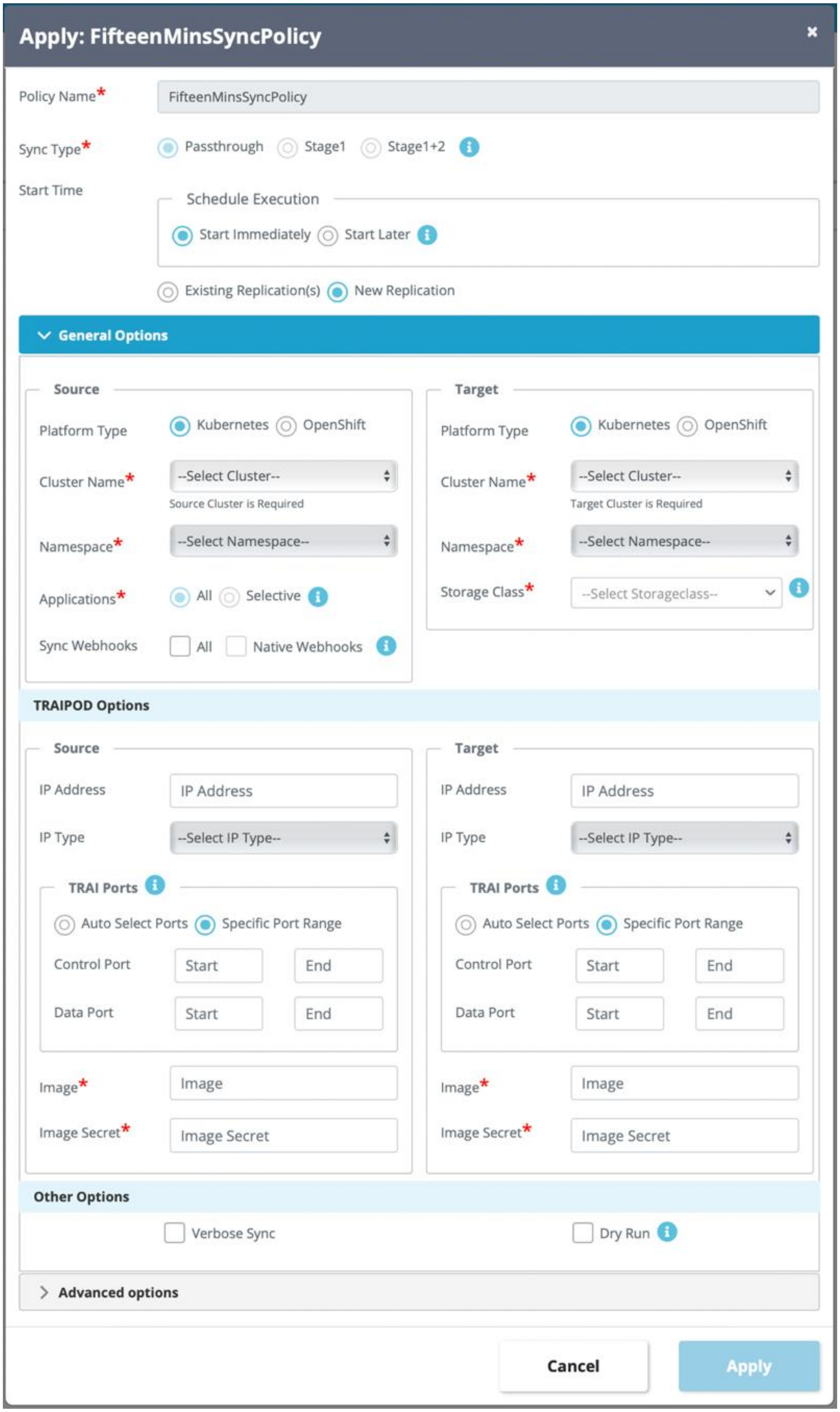
Select Replication(s) to +Add

Total: **10** Replications Found

Replications* Job-1634 (1634) ✖

New application replication

The new replication addition under policy will give exact same options as starting a new replication from the 'All Replications' menu. You will input source and target for the sync and all relevant sync options. Please refer to the sync administration section in this document for more details on new sync options.



Apply: FifteenMinsSyncPolicy ×

Policy Name* FifteenMinsSyncPolicy

Sync Type* ● Passthrough ○ Stage1 ○ Stage1+2

Start Time

Schedule Execution

● Start Immediately ○ Start Later

○ Existing Replication(s) ● New Replication

General Options

Source

Platform Type ● Kubernetes ○ OpenShift

Cluster Name* --Select Cluster--

Source Cluster is Required

Namespace* --Select Namespace--

Applications* ● All ○ Selective

Sync Webhooks □ All □ Native Webhooks

Target

Platform Type ● Kubernetes ○ OpenShift

Cluster Name* --Select Cluster--

Target Cluster is Required

Namespace* --Select Namespace--

Storage Class* --Select Storageclass--

TRAIPOD Options

Source

IP Address: IP Address

IP Type: --Select IP Type--

TRAI Ports

○ Auto Select Ports ● Specific Port Range

Control Port: Start End

Data Port: Start End

Image*: Image

Image Secret*: Image Secret

Target

IP Address: IP Address

IP Type: --Select IP Type--

TRAI Ports

○ Auto Select Ports ● Specific Port Range

Control Port: Start End

Data Port: Start End

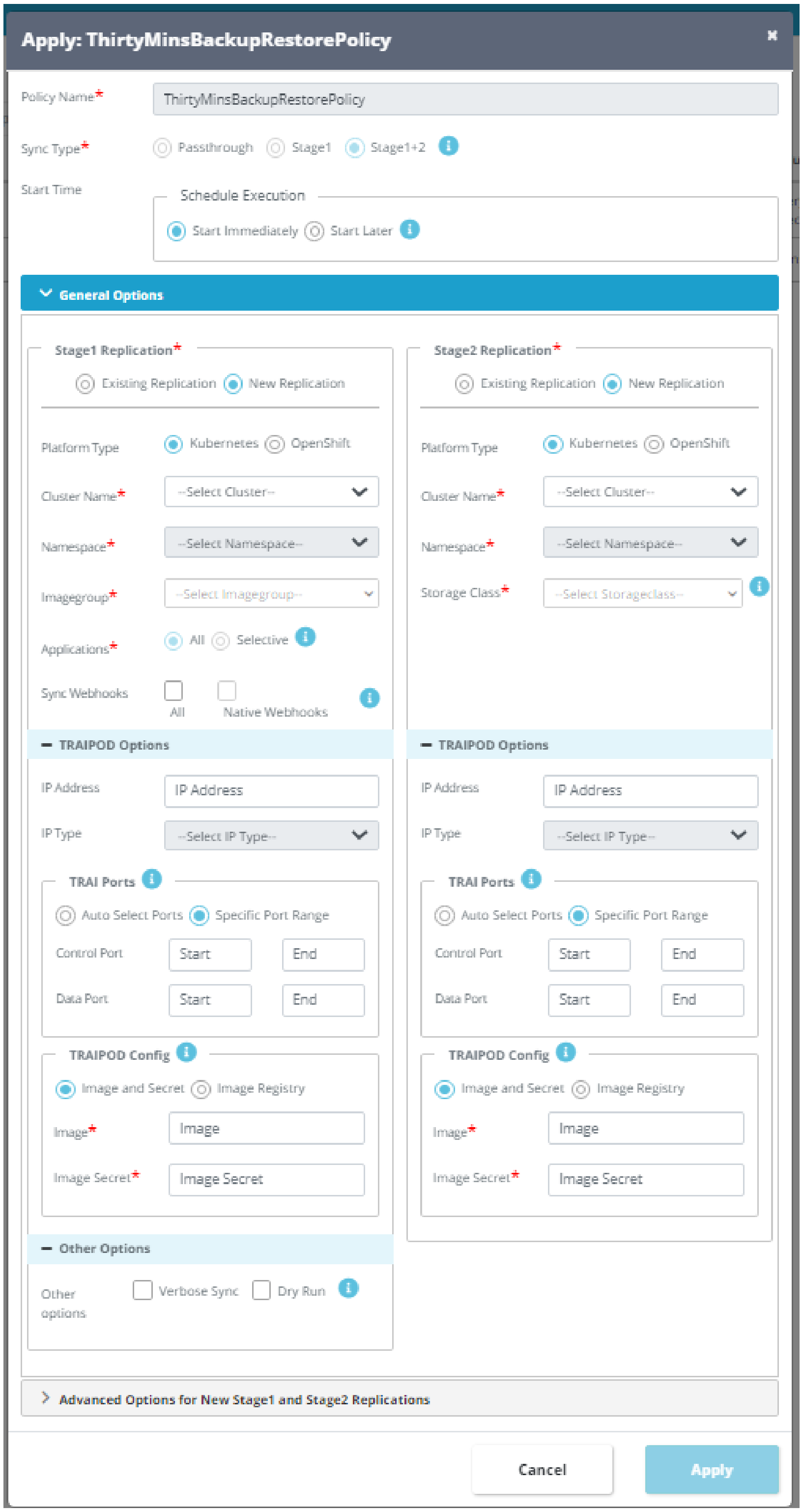
Image*: Image

Image Secret*: Image Secret

Other Options

□ Verbose Sync □ Dry Run

Advanced options



Apply: ThirtyMinsBackupRestorePolicy

Policy Name* ThirtyMinsBackupRestorePolicy

Sync Type* ○ Passthrough ○ Stage1 ● Stage1+2

Start Time — Schedule Execution: ● Start Immediately ○ Start Later

General Options

Stage1 Replication* ○ Existing Replication ● New Replication

Platform Type ● Kubernetes ○ OpenShift

Cluster Name* --Select Cluster--

Namespace* --Select Namespace--

Imagegroup* --Select Imagegroup--

Applications* ● All ○ Selective

Sync Webhooks □ All □ Native Webhooks

TRAIPOD Options

IP Address: IP Address

IP Type: --Select IP Type--

TRAI Ports: ○ Auto Select Ports ● Specific Port Range

Control Port: Start End

Data Port: Start End

TRAIPOD Config: ● Image and Secret ○ Image Registry

Image*: Image

Image Secret*: Image Secret

Other Options

Other options □ Verbose Sync □ Dry Run

Stage2 Replication* ○ Existing Replication ● New Replication

Platform Type ● Kubernetes ○ OpenShift

Cluster Name* --Select Cluster--

Namespace* --Select Namespace--

Storage Class* --Select Storageclass--

TRAIPOD Options

IP Address: IP Address

IP Type: --Select IP Type--

TRAI Ports: ○ Auto Select Ports ● Specific Port Range

Control Port: Start End

Data Port: Start End

TRAIPOD Config: ● Image and Secret ○ Image Registry

Image*: Image

Image Secret*: Image Secret

Advanced Options for New Stage1 and Stage2 Replications

Once the policy is applied, the selected replications are run at the specified date and time and reported according to alert configuration. You can always find DR policy health and average sync times from the BCDR menu and by selecting the specific policy.

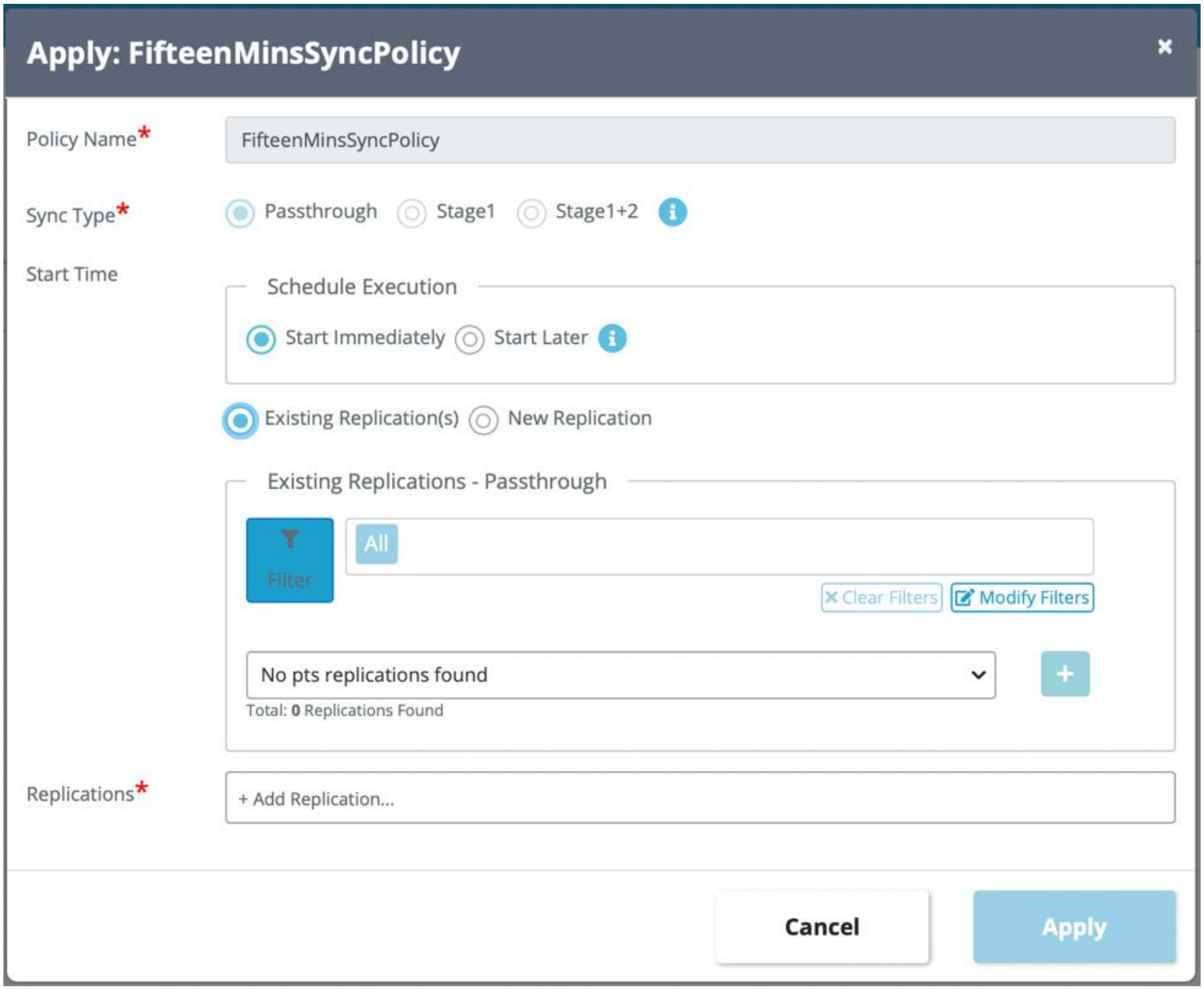
Applying the newly created policy to registry syncs

Go the Business Continuity and Disaster Recovery (BCDR) menu. You will see list of all policies on this page. Select the policy you want to apply and click on the 'Apply' button.

You will have a choice of applying policy to

Existing registry replication

If you select existing replication, then you will get an option to select the existing sync jobs as a drop-down list. Note that only those jobs will be listed which match the policy-type. You can select more than one jobs to apply and press the 'Apply' button. Select registry sync jobs here and policy will be applied for registry sync path. The policy will schedule running exact replications which were done under those selected registry sync jobs.



Apply: FifteenMinsSyncPolicy

Policy Name*: FifteenMinsSyncPolicy

Sync Type*: ● Passthrough ○ Stage1 ○ Stage1+2

Start Time: Schedule Execution — ● Start Immediately ○ Start Later

● Existing Replication(s) ○ New Replication

Existing Replications - Passthrough

Filter: All

No pts replications found

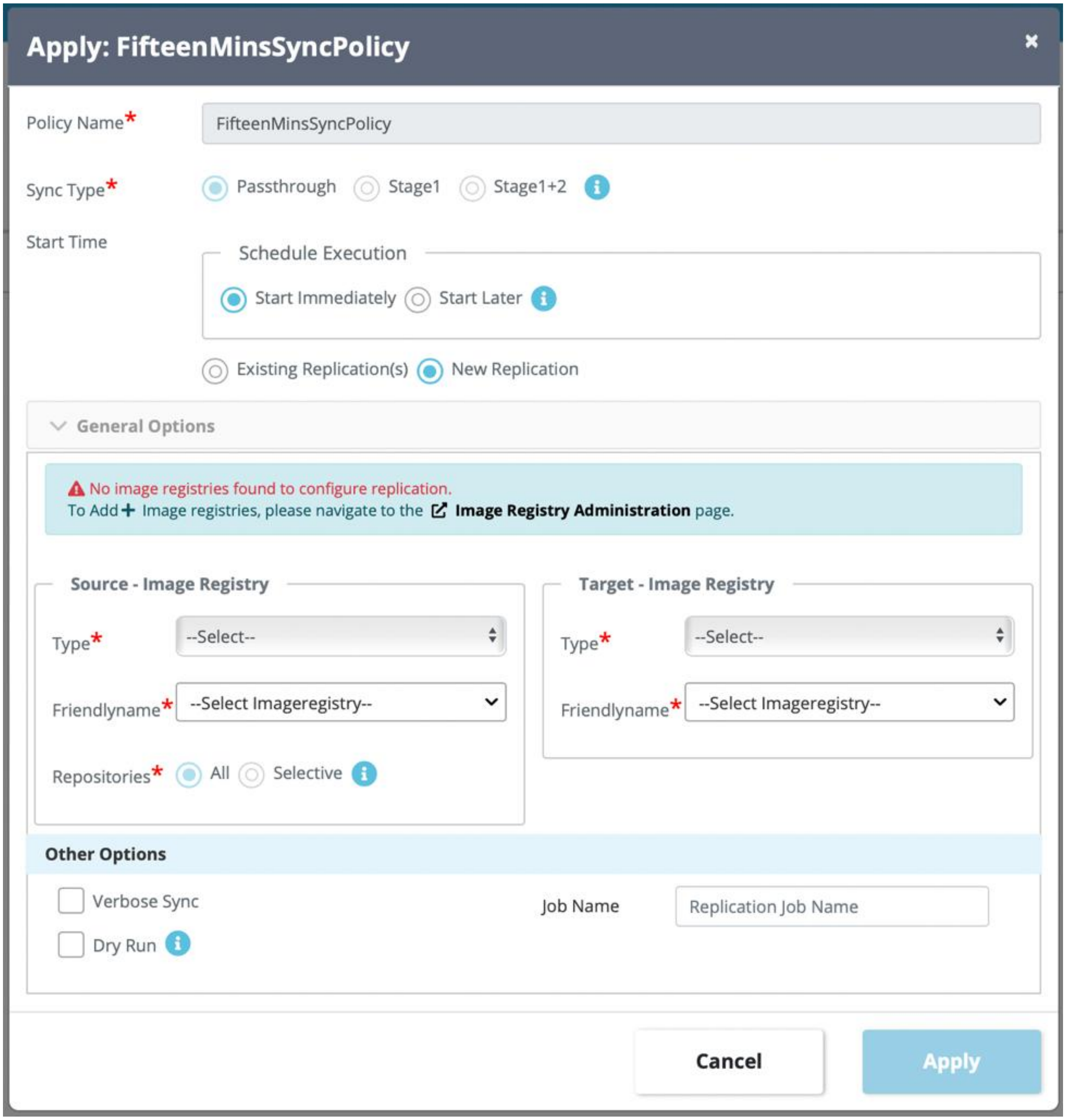
Total: 0 Replications Found

Replications*: + Add Replication...

New registry replication

The new replication addition under policy will give exact same options as starting a new replication from the 'All Replications' menu. You will input source and target registry/repository/tag for the sync and all other relevant sync options. Please refer to the sync administration section in this document for more details on new sync options.

Once the policy is applied, the selected registry replications are run at the specified date and time and reported according to alert configuration. You can always find DR policy health and average sync times from the BCDR menu and by selecting the specific policy.



Apply: FifteenMinsSyncPolicy ×

Policy Name* FifteenMinsSyncPolicy

Sync Type* ● Passthrough ○ Stage1 ○ Stage1+2

Start Time — Schedule Execution: ● Start Immediately ○ Start Later

○ Existing Replication(s) ● New Replication

General Options

No image registries found to configure replication.
To Add + Image registries, please navigate to the **Image Registry Administration** page.

Source - Image Registry
- Type* --Select--
- Friendlyname* --Select Imageregistry--
- Repositories* ● All ○ Selective

Target - Image Registry
- Type* --Select--
- Friendlyname* --Select Imageregistry--

Other Options
- □ Verbose Sync
- □ Dry Run
- Job Name: Replication Job Name

Cancel | Apply

Converting Stage1 Policy to Dynamic-Cluster Provisioning Policy

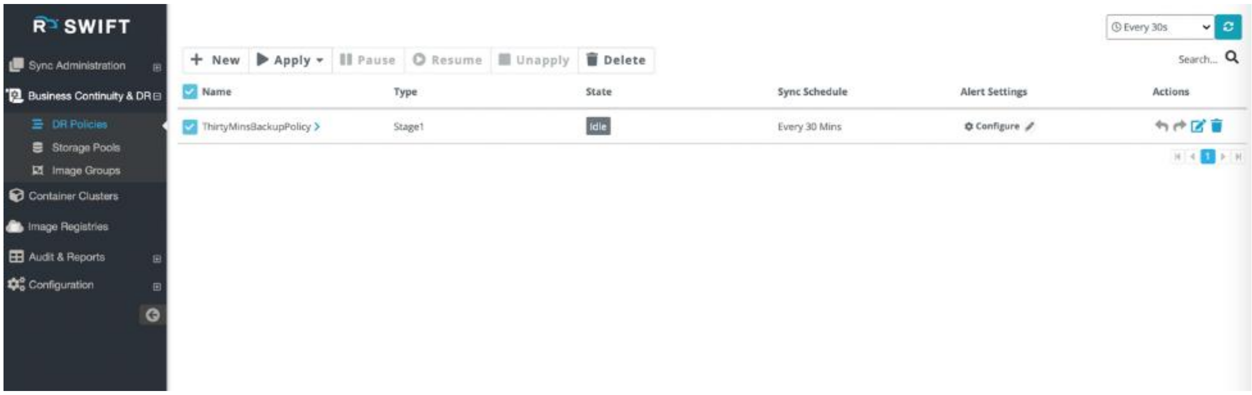
Dynamic cluster provisioning policy is a special Stage1-only policy that allows failover as well as fallback operations. Such policies, when failed over, will first create cloud-based DR clusters (which are configured as part of policy operations) and then restore the selected backup to the newly provisioned cluster.

Selecting a backup name is optional step during policy failover, and by default it will restore the latest application backup to the DR cluster.

Typically, when you create a Stage1-only policy type, it can only be used to do periodic backups for one or more namespace of the applied source cluster. There is no DR cluster config selected in these policies by default so these policies can't be failed over by default. Only way to restore latest or any specific backups done as part of the policy is to explicitly start a stage2 sync job that syncs required backup for required Image-Group to the pre-existing DR cluster.

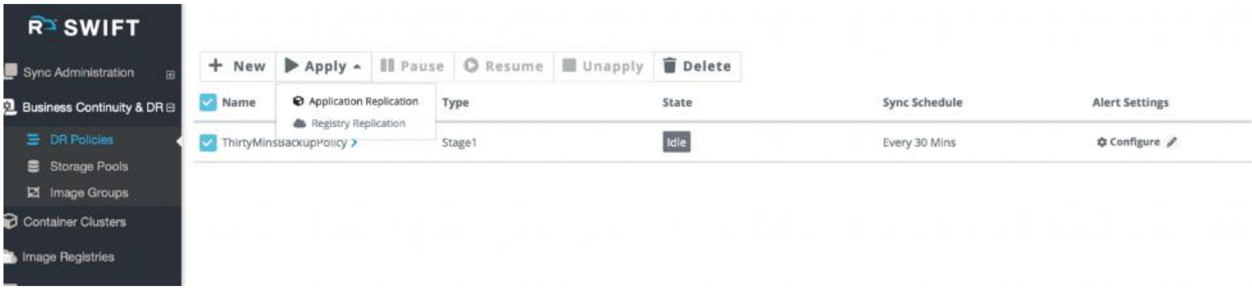
To enable dynamic provisioning for Stage1-only policies (and so failover/fallback capabilities), you need to follow the steps mentioned below.

Select the 'Business Continuity and Disaster Recovery (BCDR)' menu and 'DR Policies' sub-menu.



Name	Type	State	Sync Schedule	Alert Settings	Actions
☑ ThirtyMinsBackupPolicy	Stage1	Idle	Every 30 Mins	Configure	

Select the required Stage1 policy and press the 'Apply' button. Select the 'Application Replication' submenu.



Name	Type	State	Sync Schedule	Alert Settings
☑ ThirtyMinsBackupPolicy	Stage1	Idle	Every 30 Mins	Configure

Now, on the apply dialog, you will see 'DR Cluster' options. Select a cloud-type and then depending on cloud-type, you will input one or more parameters for the respective cloud. For now, only Oracle OCI is supported as a type, which means you can dynamically provision only Oracle Kubernetes Engine (OKE) clusters. The below section highlights cloud-specific input parameters for DR cluster. Note that not all parameters are mandatory.

Oracle Cloud (OCI)

- Cluster name
- Node shape

- Number of nodes
- Kubernetes version
- User id
- Tenant id
- Fingerprint (for API key)
- Private key
- Region
- Compartment name
- Network type
- Availability domain (AD)

The selected cloud is where your DR cluster will be provisioned as part of a failover for the policy operation (including ‘drill’ failover). The DR cluster is also cleaned up as part of a subsequent fallback for the policy operation.

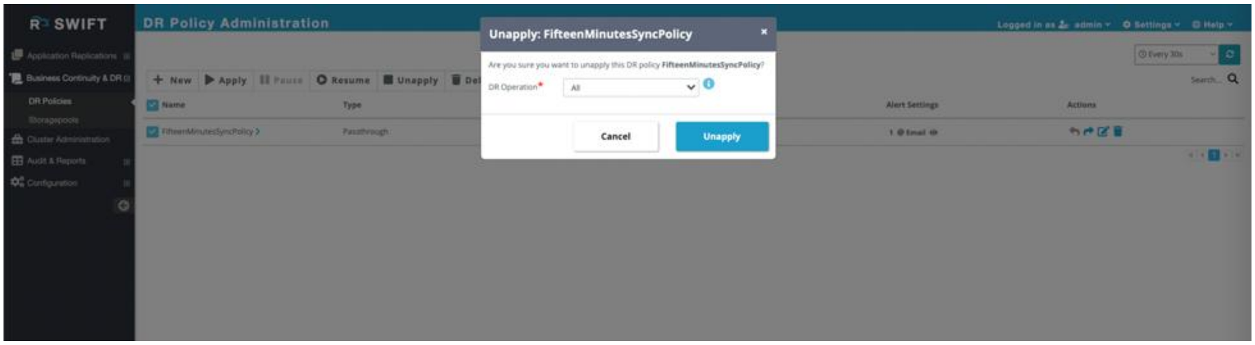
DR Policy Administration

In the previous section, we saw how you can create a new policy and apply it to the existing sync job or create a new sync operation under it. In this section, we will see additional operations that you can do on the created policy.

Unapply a DR policy

From the BCDR menu, select the policy you want to unapply and click on the ‘Unapply’ button.

You will get options to select the policy instances which you want to remove/unapply. The selected policy instances will be removed, and corresponding scheduled syncs will stop running immediately. Note that any of the ongoing policy triggered syncs will not be cancelled.



Unapply: FifteenMinutesSyncPolicy

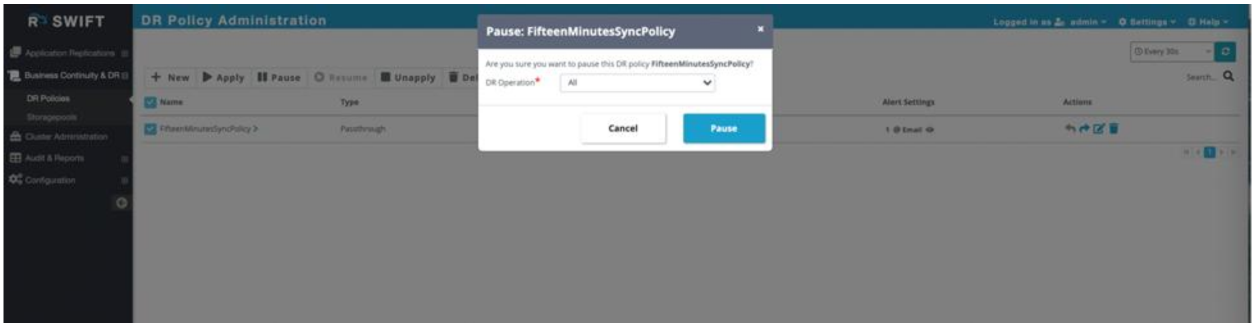
Are you sure you want to unapply this DR policy **FifteenMinutesSyncPolicy**?

DR Operation* All

Name	Type	Alert Settings	Actions
☑ FifteenMinutesSyncPolicy	Passthrough	1 Email	

Pause a DR policy

From the BCDR menu, select the policy you want to pause and click on the ‘Pause’ button. You can select more than one policy to pause in a batch.



Pause: FifteenMinutesSyncPolicy

Are you sure you want to pause this DR policy **FifteenMinutesSyncPolicy**?

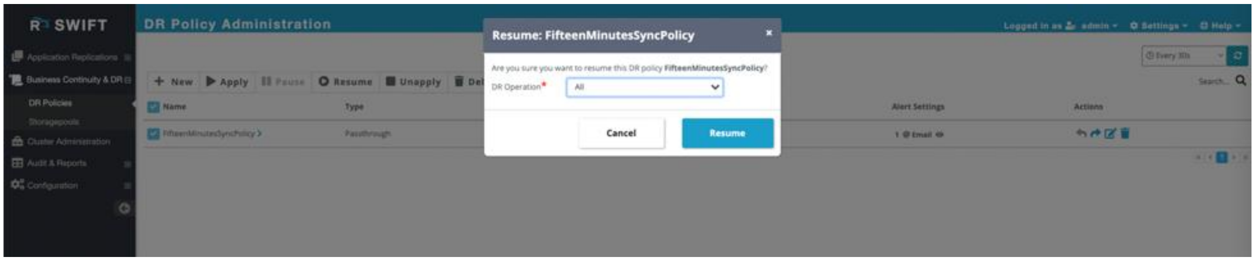
DR Operation*: All

Cancel | Pause

Once you pause a policy, all the syncs or policy instances will be paused, and the policy will go into the ‘PAUSED’ state.

Resume a DR policy

From the BCDR menu, select the policy you want to resume and click on the ‘Resume’ button. You can select more than one policy to resume in a batch. Note that you can only select policies which are in the PAUSED or PARTIALLY_ACTIVE state.



Resume: FifteenMinutesSyncPolicy

Are you sure you want to resume this DR policy **FifteenMinutesSyncPolicy**?

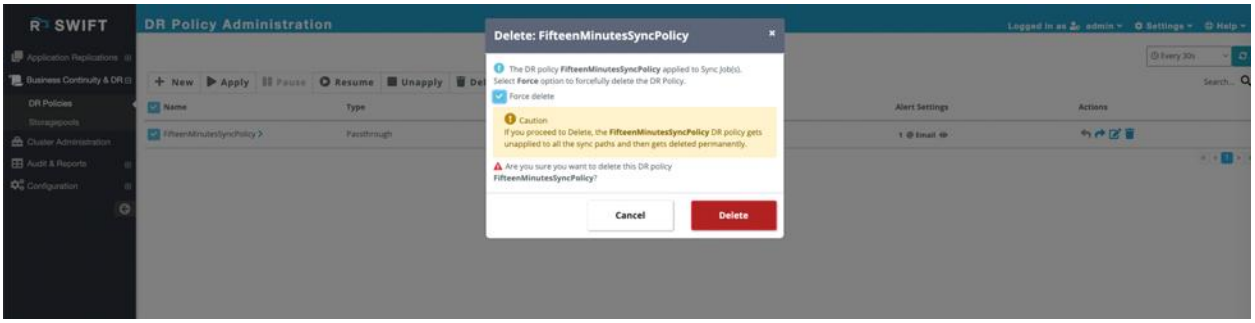
DR Operation*: All

Cancel | Resume

Once you resume a policy, all the policy syncs or policy instances will start running by their configured frequency or schedule. The resumed policy will also go into ACTIVE state.

Delete a DR policy

From the BCDR menu, select the policy you want to delete and click on the ‘Delete’ button.



Delete: FifteenMinutesSyncPolicy

The DR policy **FifteenMinutesSyncPolicy** applied to Sync Job(s). Select **Force** option to forcefully delete the DR Policy.

☑ Force delete

Caution
If you proceed to Delete, the **FifteenMinutesSyncPolicy** DR policy gets unapplied to all the sync paths and then gets deleted permanently.

Are you sure you want to delete this DR policy **FifteenMinutesSyncPolicy**?

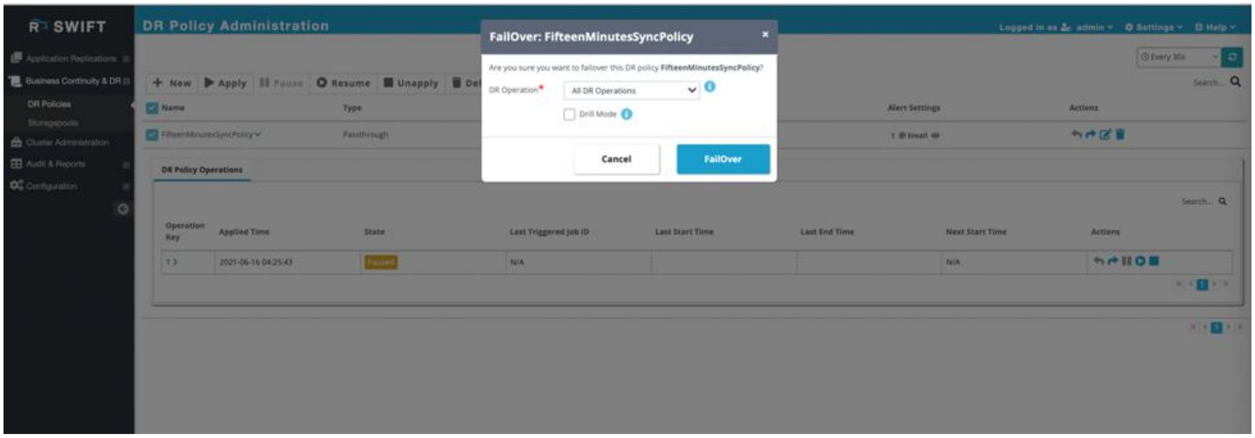
Cancel | Delete

By default, policies which are in the ACTIVE or PARTIALLY_ACTIVE state will not be allowed for deletion. You can only delete the IDLE policies. Selecting the ‘Force delete’ option will delete any policy. The force deletion will also first internally unapply the policy.

DR Policy failover

You can select a policy from the BCDR menu and initiate a failover. Two types of failovers are supported with the policies:

1. Test/Drill failover
2. Real/Non-Drill failover



FailOver: FifteenMinutesSyncPolicy

Are you sure you want to failover this DR policy **FifteenMinutesSyncPolicy**?

DR Operation* All DR Operations

□ Drill Mode

Cancel | FailOver

Name	Type	Alert Settings	Actions
FifteenMinutesSyncPolicy	Passthrough	1 Email	

DR Policy Operations

Operation Key	Applied Time	State	Last Triggered Job ID	Last Start Time	Last End Time	Next Start Time	Actions
1	2021-06-16 04:25:43	Paused	N/A			N/A	

The drill failover will failover the policy by doing one forward sync end to end. The drill mode is assumed to be used for testing of your actual DR drills. In case of a real failover, the forward sync is done as a best attempt (though stage-2 syncs will always be performed if policy is staged sync type). The drill mode is marked with a special policy state.

You can optionally specify one or more operation keys for the failover. If selected, then only the specified operations (or replications tracked under those operations) are run for forward path and policy will go in partially failed over state. If no operation key is specified, then the full policy fails over with performing all forward syncs tracked by all policy operations.

Along with Stage12 policies, Stage1-only policies with DR cluster configuration in them (that allows dynamically provisioning DR cluster as part of a failover) are supported for failover operation. For Stage1-only policy, only policy operations that have DR cluster configuration in them are allowed for a failover.

When Stage1-only policy operation that has a DR cluster configuration in it fails over, SWIFT will first dynamically provision the DR cluster (with pre-configuration that is stored in the policy operation) and then initiate a stage2 sync for the failover.

If DR cluster pre-exists with the required config that is stored in a Stage1-only policy operation, then SWIFT will re-use the existing DR cluster and restore or failover to it.

DR Policy fallback

You can select a policy from the BCDR menu which is in failed over state (full or partial) and then initiate a fallback.

If policy was failed over in the drill mode, then by default fallback will not do a reverse sync (i.e., sync from your DR side to the original production from before the failover). You can optionally configure SWIFT to do a reverse sync during fallback operation for policies that were failed over in the drill mode by selecting a confirmation checkbox for reverse sync on the fallback confirmation dialog.

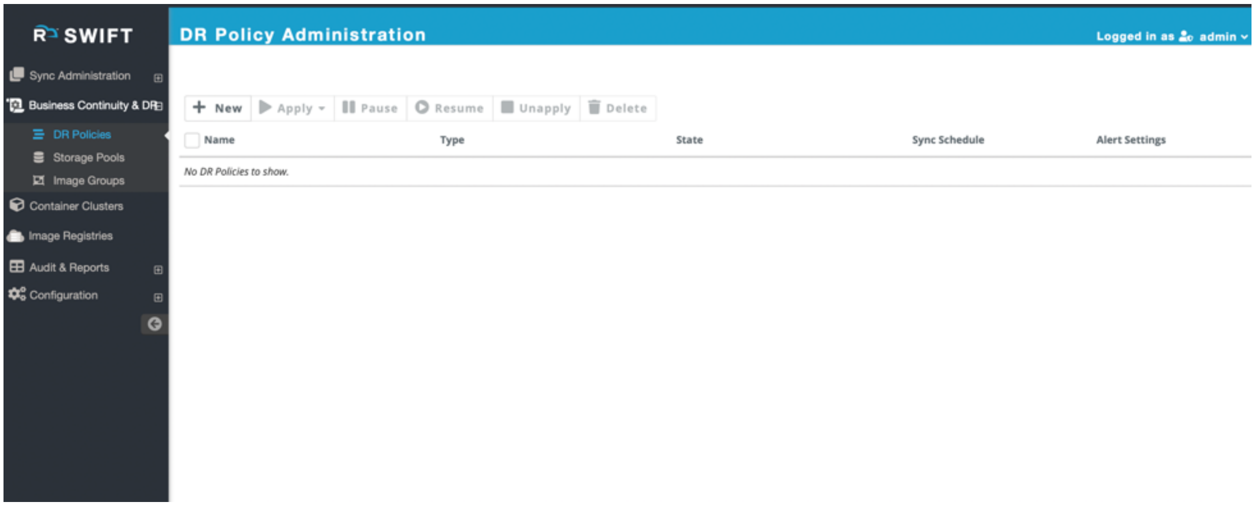
If a policy was failed over without the drill mode, then by default fallback will do a reverse sync (i.e., sync from your DR side to the original production from before the failover), while policies failed over with drill mode by default will not do a reverse sync as part of fallback. You can optionally configure SWIFT to not do a reverse sync during fallback operation for policies that were failed over in non-drill mode by selecting a confirmation checkbox for no-reverse-sync on the fallback confirmation dialog.

Along with Stage12 policies, Stage1-only policies with DR cluster configuration in them (that allows dynamically provisioning DR cluster as part of a failover) are supported for fallback operation. For Stage1-only policy, only policy operations that have DR cluster configuration in them are allowed for a fallback.

When Stage1-only policy operation that has a DR cluster configuration in it falls back, SWIFT will first reverse sync from dynamically provisioned DR cluster to an image-group (that is stored in the DR policy operation) and then initiate a decommissioning of the DR cluster. You can choose to skip decommissioning of the DR cluster step as part of fallback inputs. If the required DR cluster doesn't exist for Stage1-only policy operation's fallback and if you select reverse sync for drill fallback (or if it is real/non-drill fallback), then SWIFT will error for fallback in such cases, while such errors will be treated as a warning if reverse sync is not required for the fallback.

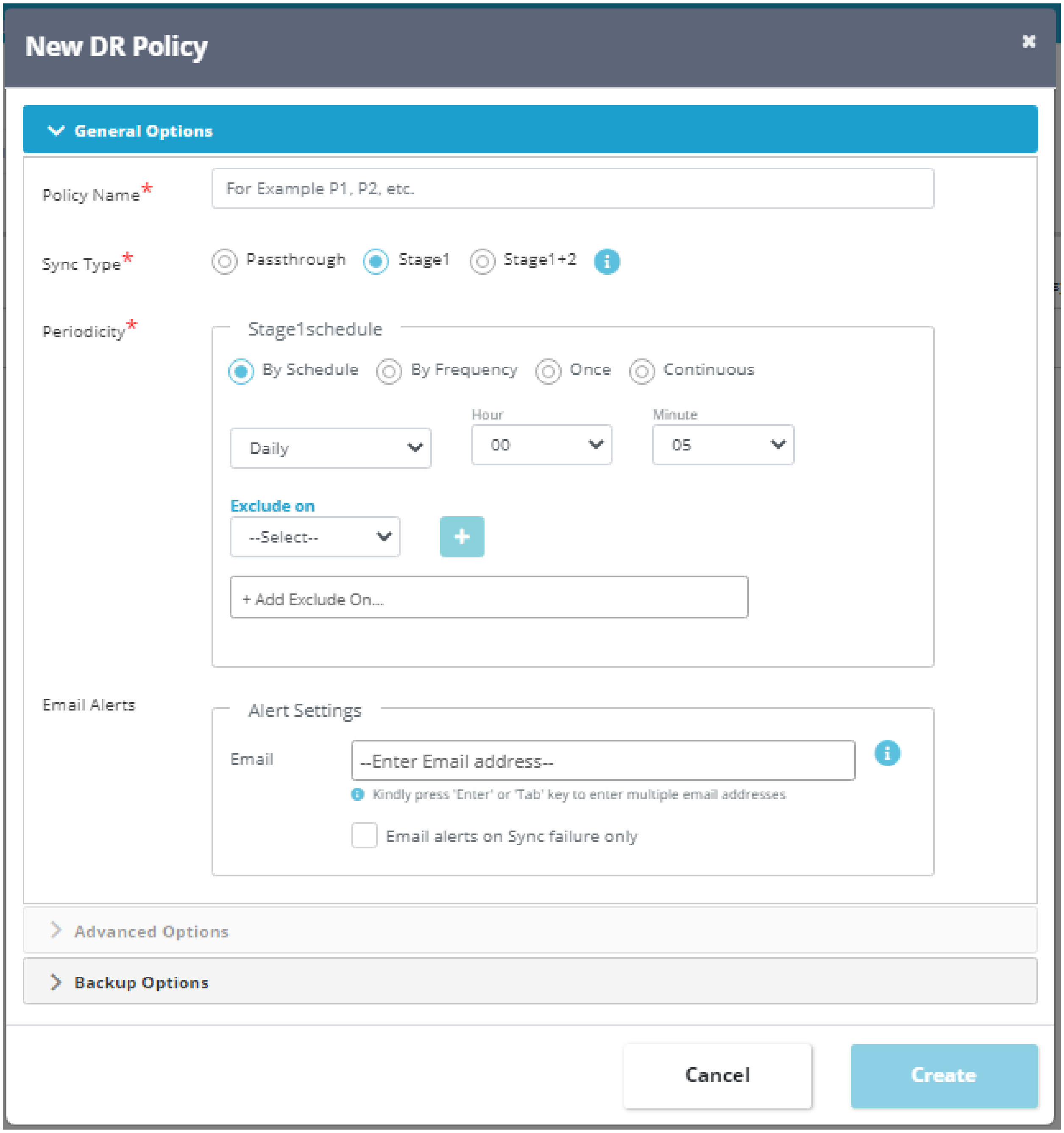
Configuring Backup Policies with SWIFT

You can create backup policies with SWIFT using DR policy administration. Go to the 'Business Continuity & DR' menu in the dashboard and then to the 'DR Policies' submenu.



□ Name	Type	State	Sync Schedule	Alert Settings
No DR Policies to show.				

Press the 'New' button to create a new policy. Select either Stage1 or Stage12 policy type.



New DR Policy

General Options

Policy Name* For Example P1, P2, etc.

Sync Type* ○ Passthrough ● Stage1 ○ Stage1+2

Periodicity*

Stage1schedule

● By Schedule ○ By Frequency ○ Once ○ Continuous

Daily | Hour 00 | Minute 05

Exclude on --Select--

+ Add Exclude On...

Email Alerts

Alert Settings

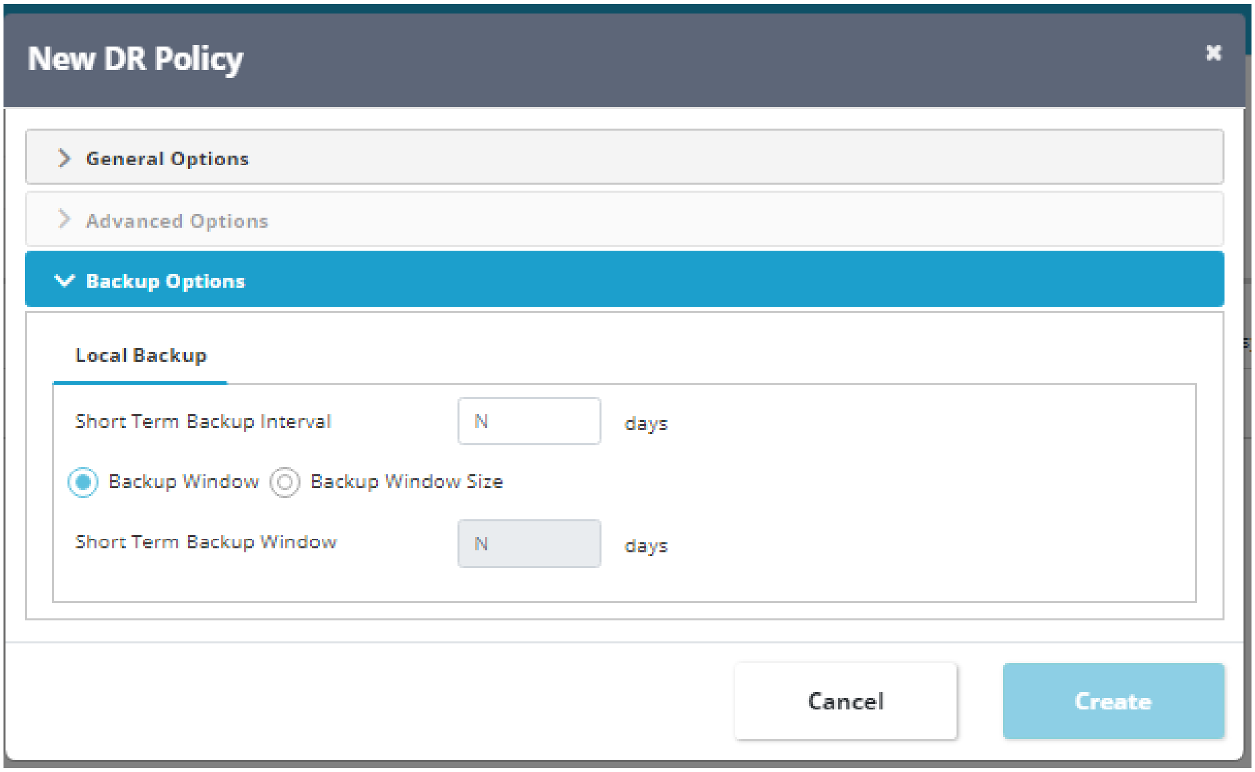
Email --Enter Email address--

Kindly press 'Enter' or 'Tab' key to enter multiple email addresses

□ Email alerts on Sync failure only

During DR policy create, you will get an option to configure following backup schedule options:

1. Short-term backup interval
2. Short-term backup window
3. Short-term backup window size
4. Long-term backup interval
5. Long-term backup window
6. Long-term backup window size



New DR Policy ✖

> General Options

> Advanced Options

⌄ Backup Options

Local Backup

Short Term Backup Interval N days

● Backup Window ○ Backup Window Size

Short Term Backup Window N days

The short-term options help you configure local backups while long-term options are used to specify remote backup schedule. The local backups go in the ZFS based storage pools created in SWIFT, so to locally attached disks-based storage available with SWIFT, while remote backups always go to cloud object-storage based storage pools created in SWIFT.

The backup interval is expected to be greater than or equal to stage-1 interval you are specifying for the policy. The window and window-size are mutually exclusive options. The window can be in minutes/hours/days/weeks/months/etc. while window-size is always numeric value. The window allows you to specify maximum backups stored in respective pool in terms of time scale, while the window-size allows you to specify numeric value highlighting upper cap on how many total backups SWIFT will store in the respective pool for the Image-Group. Once window is reached for local or remote backups, the oldest backup for the Image-Group will be deleted first before creating a new local/remote backup in the respective SWIFT storage pool.

Both local and remote backup configs are optional inputs. You can specify one of the two or both, and both work independently with their own backup frequency.

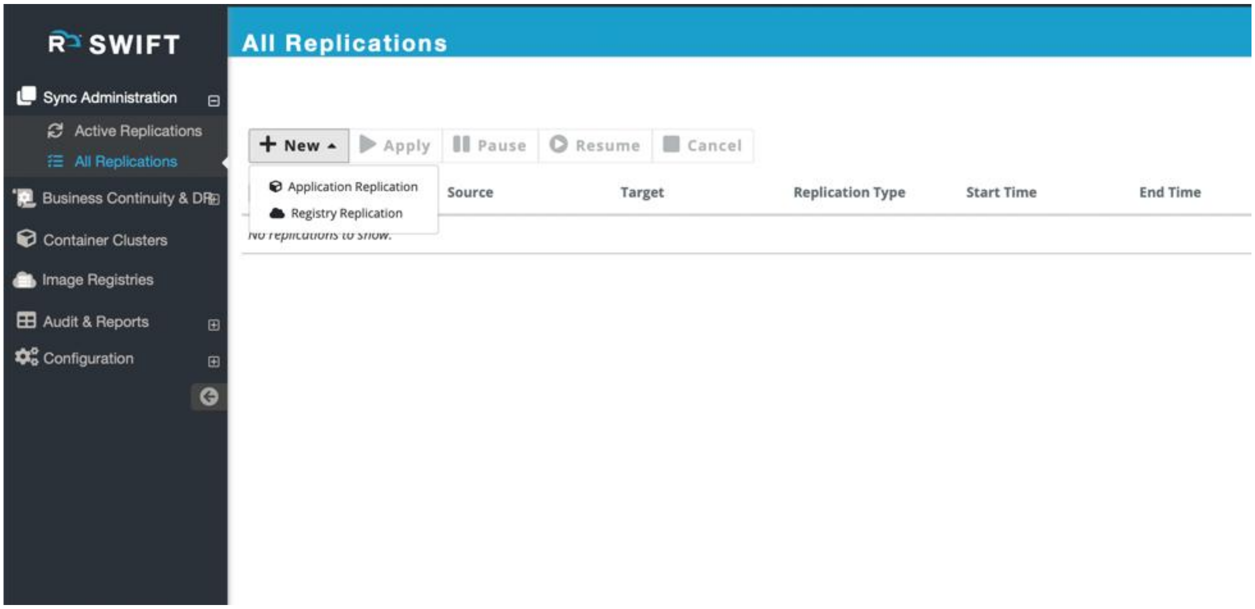
The created local and remote backups can be seen under respective Image-Groups once the new backup DR policy is applied. You can optionally also traverse to those through DR policy config and then Image-Group details.

Restoring a specific Backup with SWIFT

There are two ways you can restore a specific backup for any Image-Group to your target or DR cluster.

Restore through explicit Stage-2 sync

Go to the ‘Sync Administration’ and then ‘All Replications’ menu in dashboard. Press the ‘New’ button and then ‘Application Replication’ option.

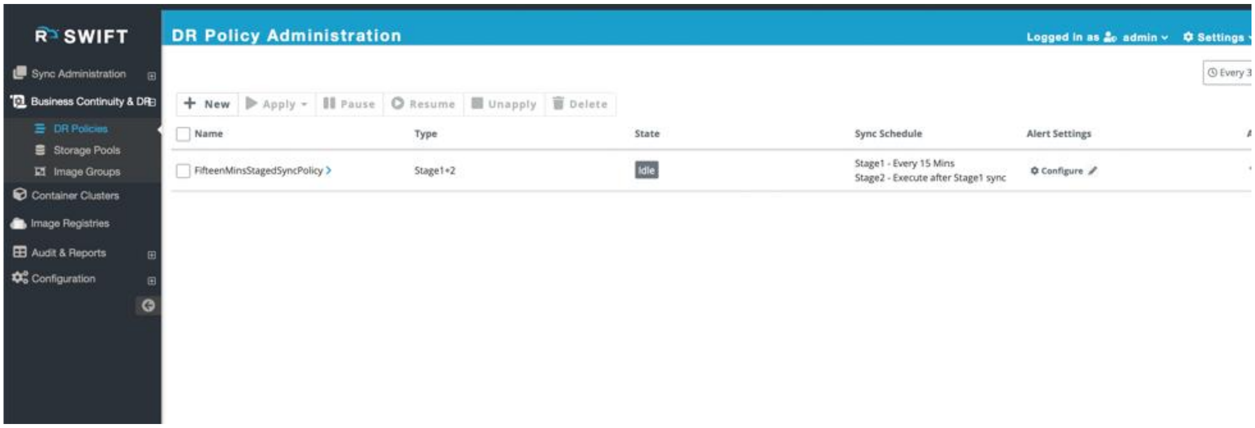


Select the ‘Stage-2’ sync type on the ‘New Replication’ dialog and then select the Image-Group that you want to restore. You can select either Kubernetes (K8S) or OpenShift as the target platform.

You can now see all backups available for the Image-Group on left side in a drop-down. Select the backup that you want to restore to the selected DR or target cluster and start a Stage-2 sync. You can select either local or remote storage pool-based backup. SWIFT will internally first restore the selected backup to the local Image-Group and then sync the Image-Group to your target or DR cluster.

Restore through DR policy failover

Restore through DR policy failover is straightforward. Go to the ‘Business Continuity & DR’ menu and to the ‘DR Policies’ submenu.



Name	Type	State	Sync Schedule	Alert Settings
□ FifteenMinsStagedSyncPolicy	Stage1+2	Idle	Stage1 - Every 15 Mins Stage2 - Execute after Stage1 sync	Configure

Select the required staged sync DR policy that is currently backing up your application. Then press the 'Failover' button.

On the Failover dialog, once you select the specific DR policy operation that is backing up your production application, you will see a backup listing drop-down for the Image-Group that is used by the policy operation. Select the required local or remote storage pool backup of your Image-Group and continue with the failover as usual. Now behind the scenes, SWIFT will first restore that selected backup to the Image-Group and then sync the Image-Group over to the target or DR cluster. After the failover completes, the target application will run with point-in-time copy of data and Kubernetes/OpenShift objects from the backup time.

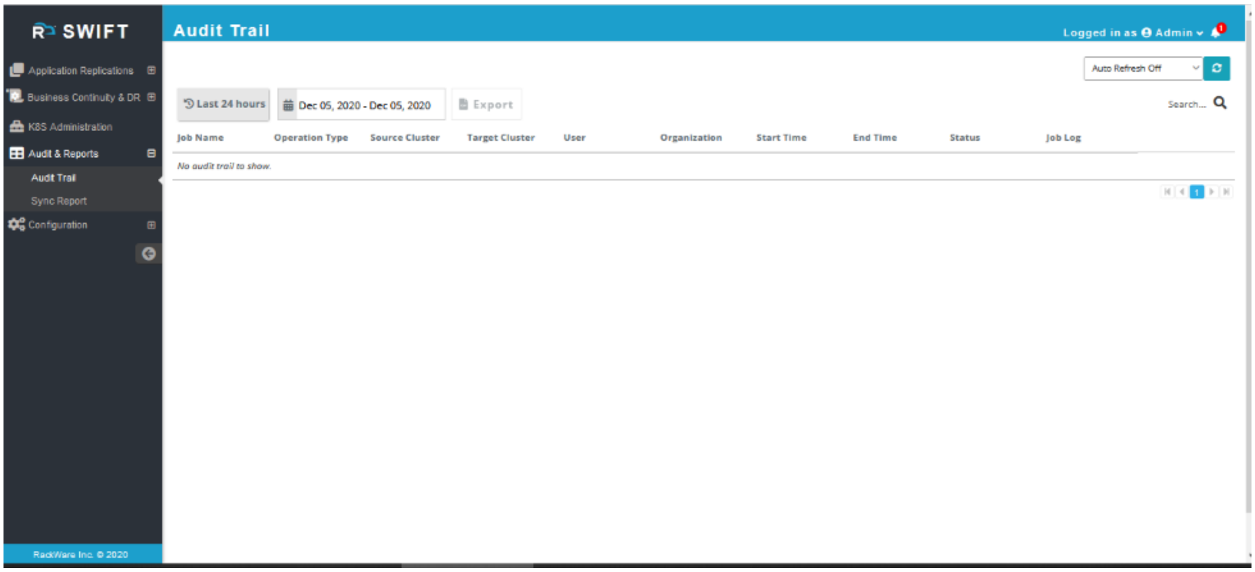
Generating SWIFT operation audit reports

Often you will need the SWIFT operation events log or report and want to save it. This section highlights the steps to generate such synchronization reports or other SWIFT operations' reports.

Note that these reports are only available from the SWIFT dashboard or GUI.

Generate all operations audit report

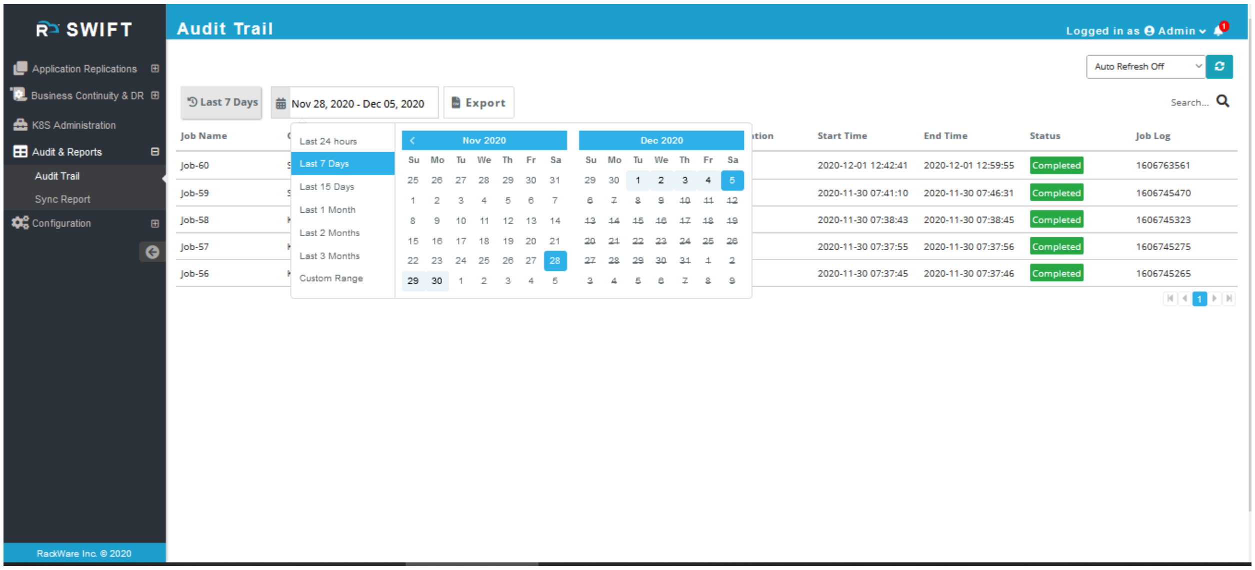
Login to the SWIFT dashboard and navigate to the 'Audit & Reports' menu and 'Audit Trail' submenu.



Dec 05, 2020 - Dec 05, 2020

Job Name	Operation Type	Source Cluster	Target Cluster	User	Organization	Start Time	End Time	Status	Job Log
No audit trail to show.									

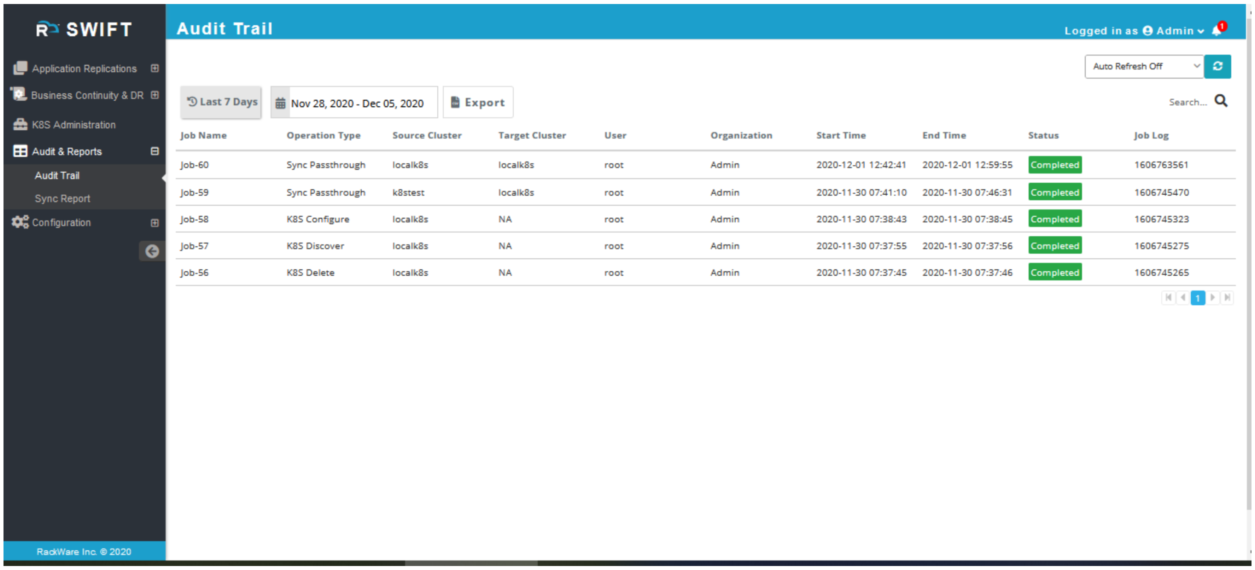
By default, the page would show you all operations within the last 24-hours. You can use filters in the search box, like a particular username, for example. Additionally, you can also pick operations for a specific fixed timeline by selecting a date range from the date-picker widget at the top left.



Nov 28, 2020 - Dec 05, 2020

Job Name	Start Time	End Time	Status	Job Log
Job-60	2020-12-01 12:42:41	2020-12-01 12:59:55	Completed	1606763561
Job-59	2020-11-30 07:41:10	2020-11-30 07:46:31	Completed	1606745470
Job-58	2020-11-30 07:38:43	2020-11-30 07:38:45	Completed	1606745323
Job-57	2020-11-30 07:37:55	2020-11-30 07:37:56	Completed	1606745275
Job-56	2020-11-30 07:37:45	2020-11-30 07:37:46	Completed	1606745265

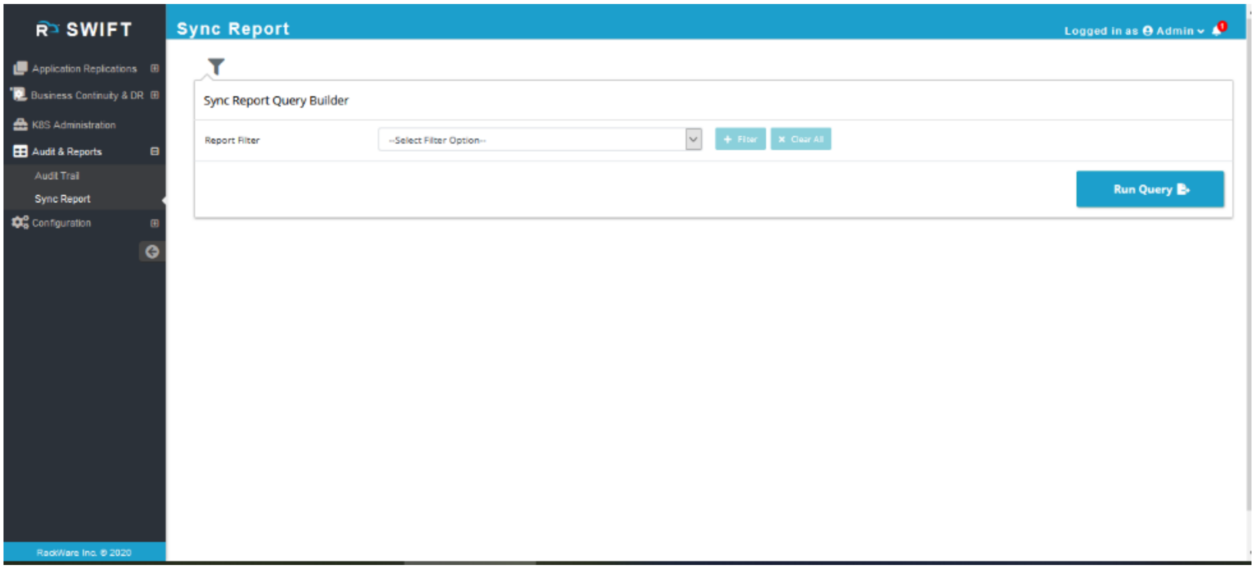
Once you have all the necessary filters applied, you can export the generate audit trail report as a CSV with the ‘Export’ button.



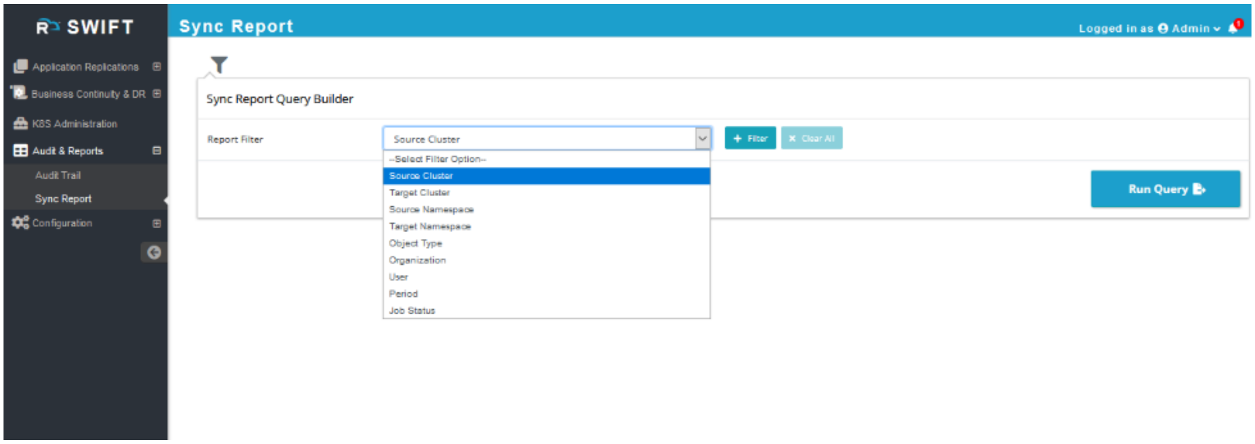
Job Name	Operation Type	Source Cluster	Target Cluster	User	Organization	Start Time	End Time	Status	Job Log
Job-60	Sync Passthrough	localk8s	localk8s	root	Admin	2020-12-01 12:42:41	2020-12-01 12:59:55	Completed	1606763561
Job-59	Sync Passthrough	k8stest	localk8s	root	Admin	2020-11-30 07:41:10	2020-11-30 07:46:31	Completed	1606745470
Job-58	K8S Configure	localk8s	NA	root	Admin	2020-11-30 07:38:43	2020-11-30 07:38:45	Completed	1606745323
Job-57	K8S Discover	localk8s	NA	root	Admin	2020-11-30 07:37:55	2020-11-30 07:37:56	Completed	1606745275
Job-56	K8S Delete	localk8s	NA	root	Admin	2020-11-30 07:37:45	2020-11-30 07:37:46	Completed	1606745265

Generate sync report

Login to the SWIFT dashboard and navigate to the ‘Audit & Reports’ menu and ‘Sync Report’ submenu.

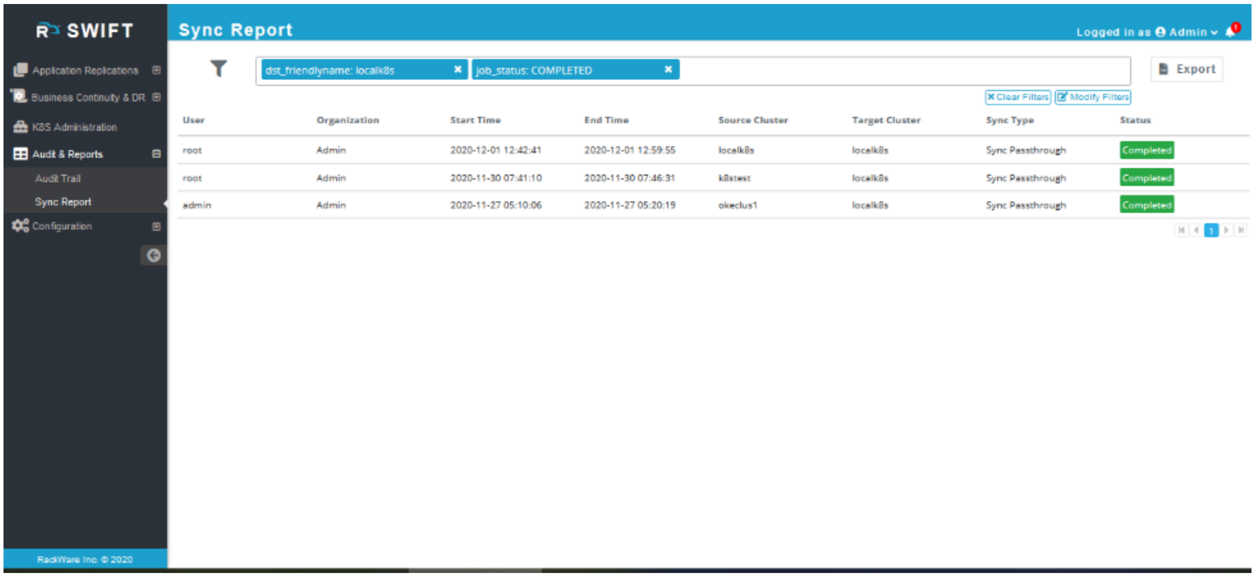


Once you have all the appropriate filters selected, then press the ‘Generate’ button to create a report.



Sync Report Query Builder

Report Filter: Source Cluster



dst_friendlyname: localk8s ✖ job_status: COMPLETED ✖

User	Organization	Start Time	End Time	Source Cluster	Target Cluster	Sync Type	Status
root	Admin	2020-12-01 12:42:41	2020-12-01 12:59:55	localk8s	localk8s	Sync Passthrough	Completed
root	Admin	2020-11-30 07:41:10	2020-11-30 07:46:31	k8stest	localk8s	Sync Passthrough	Completed
admin	Admin	2020-11-27 05:10:06	2020-11-27 05:20:19	okeclus1	localk8s	Sync Passthrough	Completed

You can export the generated report as a CSV by pressing the ‘Export’ button on the page.

Known SWIFT operational limitations

For the current SWIFT release (v1.3.0.x), below are the known operational limitations.

SWIFT won't allow creating users and organizations with the same name under different organization hierarchy

For example, you may have 'Engineering' and 'Support' organizations created under the SWIFT. If you try to create the same child organization or a user under both these organizations with the same name, then you will receive an error of already existing user/organization with the same name.

A simple workaround here, for now, is to change the friendlyname of the corresponding user/organization to make it unique and remove the conflict.

SWIFT needs two unique ports per sync even if parallel syncs are running between the same clusters and namespaces

Currently, if you start two or more parallel syncs between two clusters A and B, and both are syncing the same namespace 'mynamespace' from cluster A to B (but syncing different objects between namespaces), then you would still need two unique ports for each sync. The limitation stems from the fact that all these parallel syncs will launch their pair of TRAI POD and service instance in the namespace on both sides of clusters.

A future release of the SWIFT will add support for a proxy TRAI service instance, which will allow sharing of the data channel, and so two ports between two clusters across parallel syncs between the two clusters.

If any of the synced Kubernetes clusters are in the cloud, and if using the LoadBalancer service type for RackWare TRAI service, then Control/Data port inputs are mandatory for sync from/to the cloud.

If any of your synced Kubernetes clusters (i.e., any of the source or target clusters for a sync) are in the cloud, and if the SWIFT is located remote, and using LoadBalancer service type for the RackWare TRAI service, then you will have to configure Control (HTTP) and Data (SSH) ports for the sync. These port inputs, if not specified, then are picked automatically from the service-port range for the Kubernetes cluster. Most cloud firewalls don't automatically open those randomly selected ports of Kubernetes LoadBalancer services, while explicitly specifying the ports for sync would open them up as part of the RackWare TRAI service creation. Note that the ports opened by the transient RackWare TRAI service are only used for the sync duration.

If you already have explicitly whitelisted the entire Kubernetes service-port range in the corresponding cloud firewall, then this limitation does not apply.

SWIFT only supports 'Local' as an identity provider

By default, the SWIFT install is enabled and configured with the 'Local' identity provider. What it means is the SWIFT admin user, as well as any more users and organizations you add, will be created within the SWIFT CMDB. As of the latest release v1.1.x, the SWIFT only supports SWIFT CMDB hosted users and organizations.

The SWIFT backend is pluggable and will support more IAM providers (including cloud IAM providers) in a future release. When it is supported, you would be able to configure your IAM provider details in the SWIFT and extend your existing users and groups to the SWIFT for login and access control.

Running CLI as 'root' user gives unrestricted access

When you run the swiftcli from the 'root' user's shell, you will get unrestricted admin access to SWIFT operations. This is by design that it will not ask you for any interactive login for these user shells.

SWIFT treats 'root' user as a superuser, so any CLI it runs is already coming from the pre-authenticated shell, so CLI will not expect any authentication. If you run CLI from any other user's shell, then it would expect you to do the necessary authentication. Also, the 'root' user credentials will not work for SWIFT dashboard access.

Sync fails during TRAIPOD deploy step and you get an error similar to this: Failed to deploy the TRAIPOD for the XXXX cluster <cluster_friendlyname> [ERROR: TRAIPOD did not get ready for the K8S <cluster_friendlyname>. ERROR: TRAI POD is waiting: Reason:CreateContainerError Message: Error response from daemon: invalid CapAdd: unknown capability: "CAP_AUDIT_YYYY"]

The XXXX string will be either 'source' or 'target,' and <cluster_friendlyname> will be friendlyname of your one of the managed clusters from SWIFT. The YYYY string will be one of the capabilities like READ or WRITE.

It is not a bug in SWIFT but Linux kernel issue. The error happens when one or more worker nodes from the cluster are running with an older version of the Linux kernel and SWIFT tries to use special container capabilities that are not supported by the kernel. To fix the issue, open the SWIFT dashboard and go to the 'K8S Administration' screen. Locate the cluster which was highlighted in the error message. Select the cluster and press the 'Configure' button on the page. Make sure to select the 'TRAIPOD No Special Capabilities' checkbox on the configuration dialog, and then press the 'Configure' button. Retry the failed sync and it will work now.

SWIFT doesn't have access to OS or any other info for Kubernetes or OpenShift cluster nodes outside of what Kubernetes APIs provide. The kernel version information for cluster nodes is not tracked by Kubernetes today, so SWIFT doesn't have access to this, which is why this is a configuration input currently.

SWIFT Sync doesn't support CSI storage classes with Immediate binding mode for multi-zonal/multi-regional cloud clusters

SWIFT launches TRAIPOD after source snapshot or target volumes are created in the cluster, and it tries to provision source snapshot or target volumes in required/one region where TRAIPOD would be launched. In case of CSI storage classes in clouds like Azure AKS, SWIFT's ability to control provisioned volume's region or zone is limited. With Immediate binding mode for CSI storage class in the cluster, volumes may get provisioned in different region/zone where the cluster spans with its nodes causing TRAIPOD deploy to fail later, as Pod and volumes will run in different regions/zones.

Due to this, the SWIFT will currently detect such CSI storage class and multi-region/zone configuration for the source and target clusters, and it will fail sync upfront asking you to reconfigure your CSI storageclass to the WaitForFirstConsumer binding mode. The change in binding mode for a storageclass is non-intrusive and non-disruptive operation for even a production cluster. It is also a generally recommended mode for a storage classes, as it will do lazy volume provisioning when Pod needing it starts running. Once you change the CSI storage class binding mode in respective cluster, the sync failing earlier would be allowed.

SWIFT Staged sync and ImageGroup operations fails if EFI Secure boot is enabled on the server and ZFS storage pool is used

SWIFT ImageGroup create, modify, and clone operations as well as Staged syncs fail if EFI Secure boot is enabled for the SWIFT server boot, and you are using ZFS storage pools. This happens because ZFS module used by SWIFT is not signed (by the ZFS community) and EFI Secure boot environment only allows signed modules to load. Since ZFS module or kernel driver is not loaded in such a context, no ZFS pool operations can be performed correctly.

Soon, SWIFT will sign ZFS modules with its own valid public key. But for now, the only workaround for the issue is to disable the Secure boot for the SWIFT server for the SWIFT to work correctly for all Image-Group and storage pool operations.

Fallback for multi zonal or regional source cloud clusters may fail if sync selected cluster volumes are in different regions or zones within the same namespace

SWIFT will launch a single TRAI POD today for destination or DR cluster. In case of fallback sync, the original production cluster is used as a target for syncing everything over. If this original production cluster has existing volumes in the namespace that is being synced and the volumes are in different regions or zones, then fallback sync will fail. A simple workaround is to delete the namespace and recreate it or simply delete the existing volumes for the original production cluster (which is the target cluster for fallback sync).

This is not an issue most times as original production cluster will have been rebuilt during DR event so may not have any volumes at all. Even if you delete volumes from such a cluster, SWIFT will recreate those volume correctly and repopulate them with data as part of its fallback sync.

This limitation will go away in future releases of SWIFT when it starts supporting multiple TRAI PODs for the target cluster.