

SWIFT - Operations Guide

Version 1.3.17

Contents

About SWIFT	4
Login to the SWIFT dashboard	4
Set password for the admin user	5
Activating the SWIFT installation with a license	5
How to check my existing/free license?	6
How to request and apply a new production license?	8
Option-1 Automated License Wizard.....	8
Option-2 License Administration menu with manual email request.....	13
SWIFT supported container platforms.....	16
SWIFT supported storage-types for source clusters.....	16
SWIFT supported container registries	17
Add more users to the SWIFT	17
Create a new organization.....	17
Create a new user	19
Deleting users from the SWIFT	21
Deleting a user	21
Deleting an organization	23
Add new Kubernetes cluster details to SWIFT	25
Local Cluster.....	25
Oracle OKE Cluster	28
Google GKE Cluster and GCP OpenShift Clusters.....	30
Amazon EKS Cluster and Amazon OpenShift Cluster.....	33
Azure AKS Cluster and Azure OpenShift Cluster	36
IBM Kubernetes Service (IKS) Cluster and IBM OpenShift clusters	40
Akamai Linode Kubernetes Engine (LKE) Cluster	43
Other Common Inputs	44
Add new Image Registry details to SWIFT	44
Amazon Elastic Container Registry (ECR).....	45
Azure Container Registry (ACR)	46
Oracle Cloud Infrastructure Container Registry (OCIR)	47
Google Container Registry (GCR)	48
Docker Hub Container Registry.....	49

Configuring Storage details for SWIFT's use	50
Ceph Storage.....	50
Storage pool Administration	52
Create a Local Storage pool	53
Create a Cloud (Object-Storage) Storage pool.....	54
Modify a Local Storage pool	55
Modify a Cloud Storage pool	56
Delete a Storage pool.....	58
Image-Group Administration	59
Image-Group Create (Clone).....	60
Image-Group Delete.....	60
Change default configurations of managed clusters	61
What is Transient RackWare Agent Image (TRAI) POD?.....	66
Import TRAI image to a private docker registry.....	66
Steps to import a TRAI image.....	66
Making the private registry available to a cluster namespace	67
Configure an image-pull secret within a Kubernetes namespace	67
Configure TRAI details for the cluster under SWIFT	67
Starting a new synchronization or replication	68
Synchronization modes.....	68
A synchronization between Kubernetes clusters	69
Sync Advanced Options.....	75
Pre/Post scripts and YAMLS	75
TRAI Configs	77
Image Registry Mappings.....	78
Service Mappings	79
Volume Sync Config	80
Ingress Config.....	81
Intra-cluster and inter-cluster syncs	82
A synchronization between container registries	83
Configuring DR policies	86
Configure a policy for running or completed application sync.....	86
Configure a policy for running or completed registry sync	87

Configure a policy for application syncs from the BCDR menu	87
Applying the newly created policy to application syncs	92
Applying the newly created policy to registry syncs.....	97
Converting Stage1 Policy to Dynamic-Cluster Provisioning Policy.....	99
DR Policy Administration	102
Unapply a DR policy	102
Pause a DR policy	102
Resume a DR policy.....	103
Delete a DR policy	103
DR Policy failover	104
DR Policy fallback	104
Configuring Backup Policies with SWIFT	105
Restoring a specific Backup with SWIFT	108
Restore through explicit Stage-2 sync	108
Restore through DR policy failover	108
Generating SWIFT operation audit reports	109
Generate all operations audit report.....	109
Generate sync report	111
Known SWIFT operational limitations	113

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.

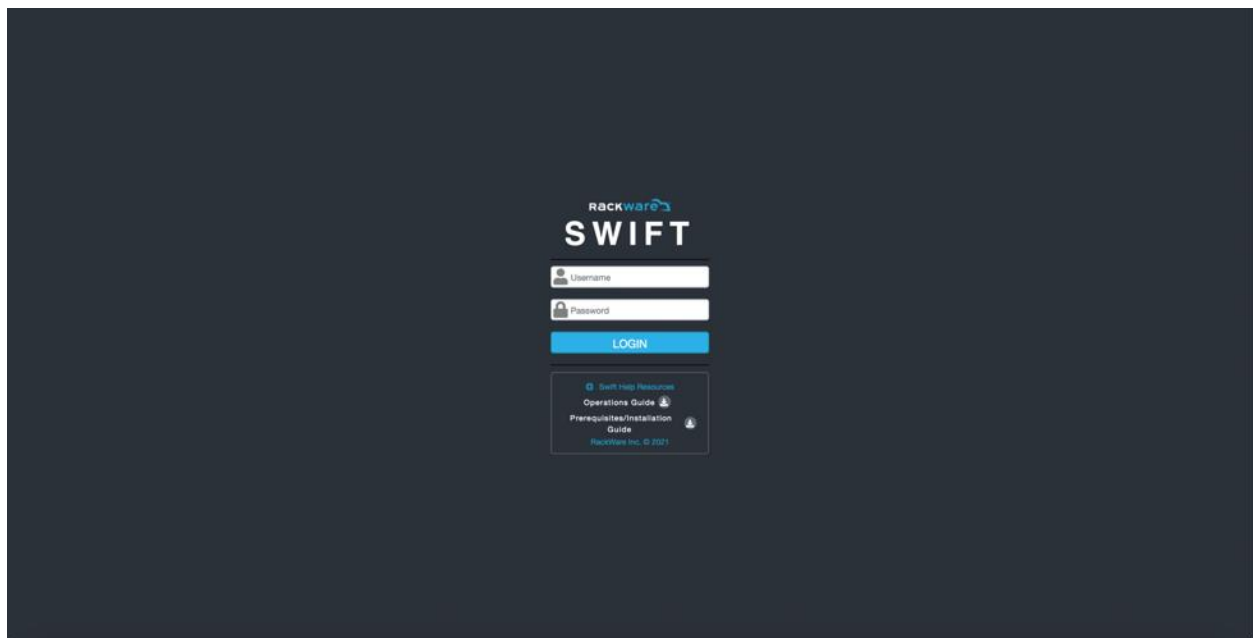
Login to the SWIFT dashboard

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

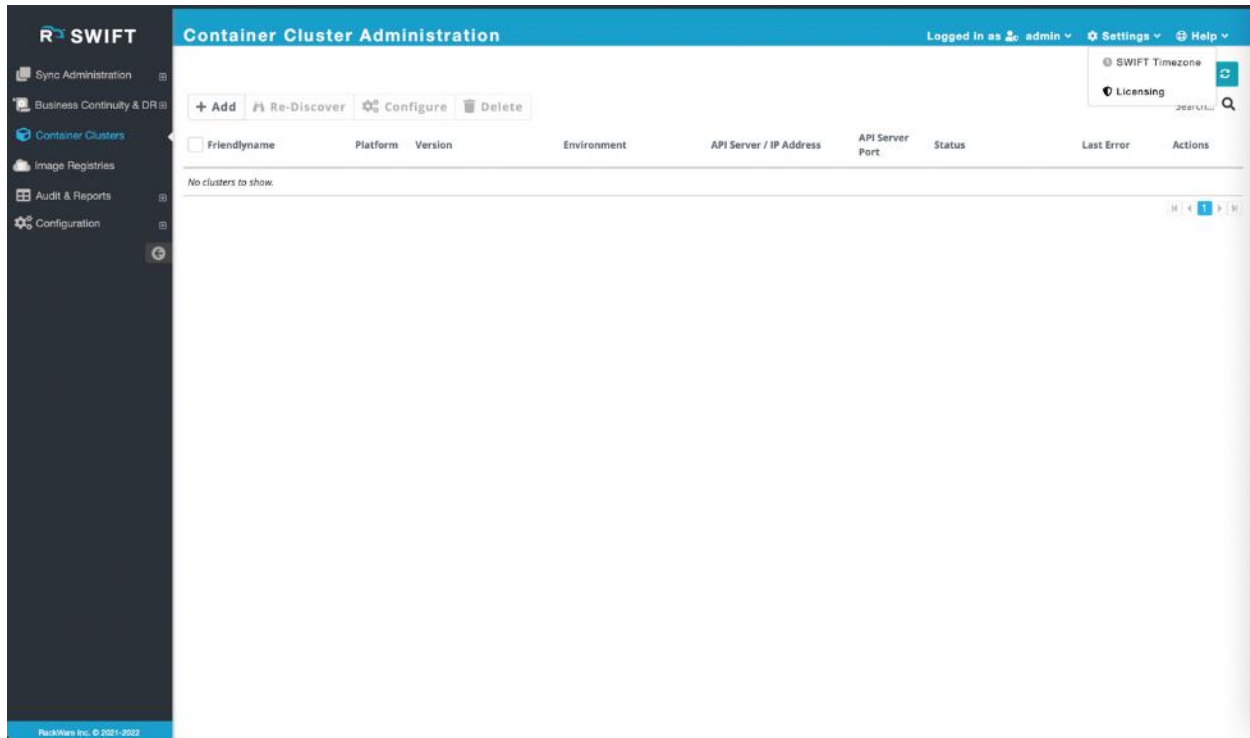
You can then login to the SWIFT dashboard with the 'admin' user and the above set '<yournewpassword>.'

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.



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

Licensing Control Panel

Summary

Operation Statistics

License Administration

SWIFT License Information

License Version : 1.1

Professional Services : No

License Usage Type : One-Time Use

License Type : Production

Latest Functional Type : Backup

Total vCPUs : 2

Freetier License : No

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	<ul style="list-style-type: none"> Registry: 50/50
Base-5 >	OpenShift	Backup	09/16/2022 01:43:02 PM	12/15/2022 01:43:02 PM	Valid	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Cluster: 50/50 Available Service Licenses » Stage1: 50/50

OK

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.

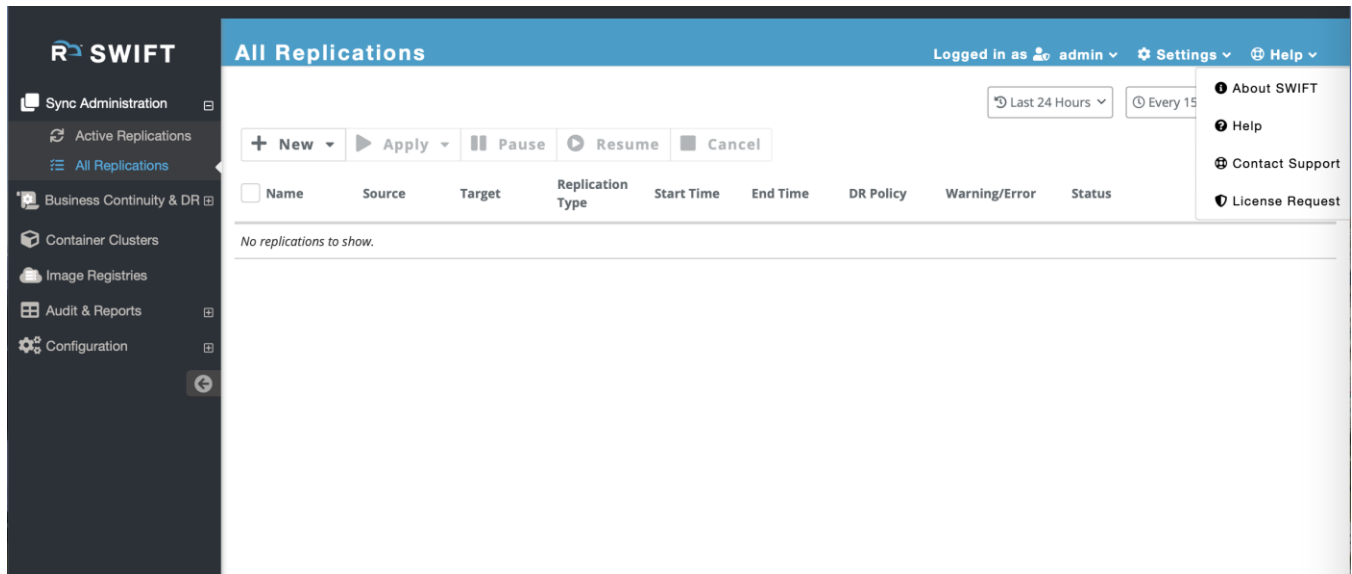
You will start with a 'base' license with absolute validity and counts, and then you would apply for an 'add-on' license in the future to extend the base license validity and/or counts.

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?

Option-1 Automated License Wizard

Login to the SWIFT dashboard and click on the 'Help' menu and then the 'License Request' submenu.



Fill in the required details like Name, email, phone, and license type needed.

License Request

1 Enter Personal Information
2 SWIFT Licensing Model
3 Configure License
4 Confirm & Submit

Please fill out the below form to request a license for your SWIFT server. The SWIFT Licensing Support will review your request. Once approved, you will receive a license file over an email.

Company Details

Company*
Country*

Enter Your Company Name
--Select country--

Company Address*

Enter Company Address

License Procurement Point of Contact Details

First Name*
Last Name*

Enter Your First Name
Enter Your Last Name

Primary Phone*
Alternate Phone

(+1) Enter Primary Phone number
(+1) Enter Alternate Phone number

Email*
Job Title*

Enter Your Email Address
Enter Your Job Title

Add Recipient(s)

--Enter Recipient(s) Emails--

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

SWIFT User Details

User Name: admin
Full Name:

Email:
Phone Number:

Cancel
Next

Once you confirm and submit a license request, you will get a ticket link on the last wizard dialog for the generated license request ticket. The ticket confirmation email is also sent to your official email you specified as part of the wizard inputs. 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).

Once you receive a valid license file (from the previous wizard and ticket step), then upload and apply the license file from the license administration GUI. Login to the SWIFT dashboard and click on the 'Settings' menu and then the 'Licensing' submenu.

Licensing Control Panel

Summary

Operation Statistics

License Administration

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	<ul style="list-style-type: none"> Registry: 50/50
Base-5	OpenShift	Backup	09/16/2022 01:43:02 PM	12/15/2022 01:43:02 PM	Valid	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Cluster: 50/50 Available Service Licenses » Stage1: 50/50

OK

Click on the 'License Administration' tab.

Licensing Control Panel

Summary

Operation Statistics

License Administration

1

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).

Generate Preinstall

2

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.

Download Preinstall

3

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

+ Browse

Drop License file to upload, or Browse

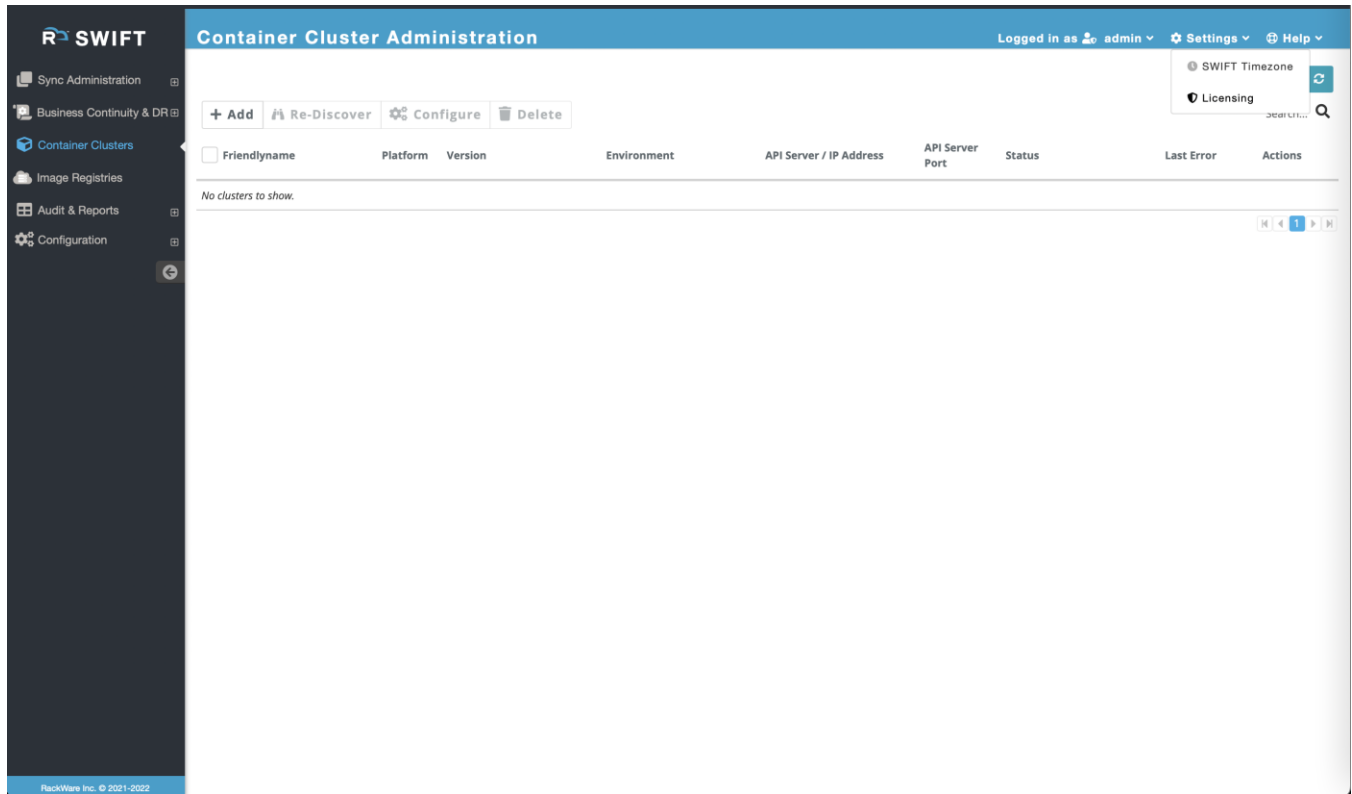
Apply License

OK

Browse and select the received license file and then press the 'Apply License' button. 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.

Option-2 License Administration menu with manual email request

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.

Licensing Control Panel

Summary

Operation Statistics

License Administration

SWIFT License Information

License Version : 1.1

Professional Services : No

License Usage Type : One-Time Use

License Type : Production

Latest Functional Type : Backup

Total vCPUs : 2

Freetier License : No

SWIFT Status : Active

Search...

Batch Name

Platform

Functional Type

Install Date

Expiry Date

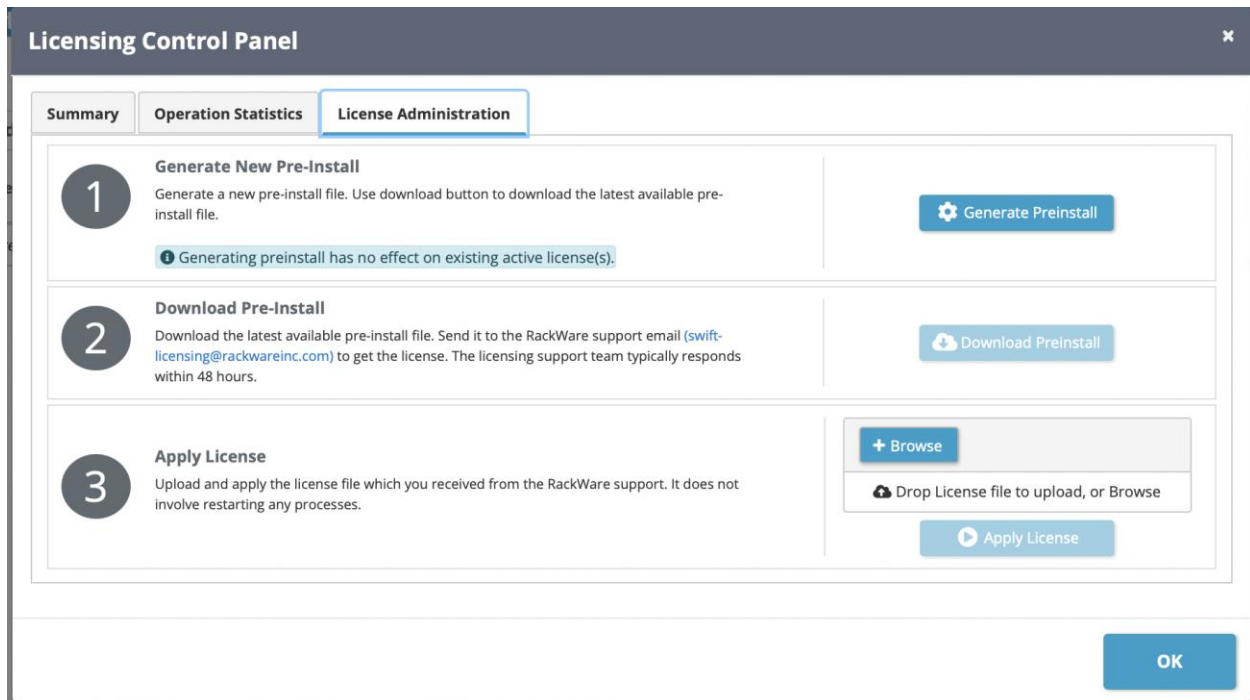
Status

License Summary

Base-6	Imageregistry	Backup	09/16/2022 01:43:02 PM	12/15/2022 01:43:02 PM	Valid	<ul style="list-style-type: none"> Registry: 50/50
Base-5	OpenShift	Backup	09/16/2022 01:43:02 PM	12/15/2022 01:43:02 PM	Valid	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Cluster: 50/50 Available Service Licenses » Stage1: 50/50

OK

Click on the 'License Administration' tab.



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+
Akamai Linode LKE	1.14+
Digital Ocean DOKS	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)
Red Hat OpenShift Data Foundation (OSDF) (CSI and non-CSI storage types)
Ceph Storage (CSI and non-CSI storage types)
Akamai Linode Block Storage (CSI storage)
Digital Ocean Block Storage (CSI storage)
Rancher Longhorn Storage (CSI storage)
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.

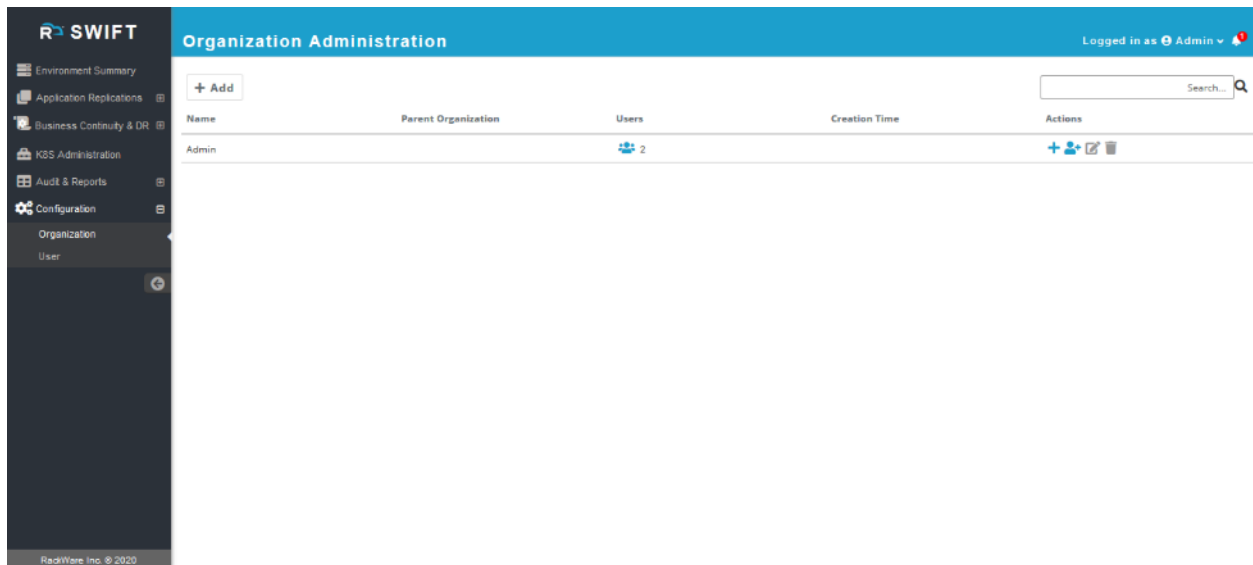
Add more users to the SWIFT

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:

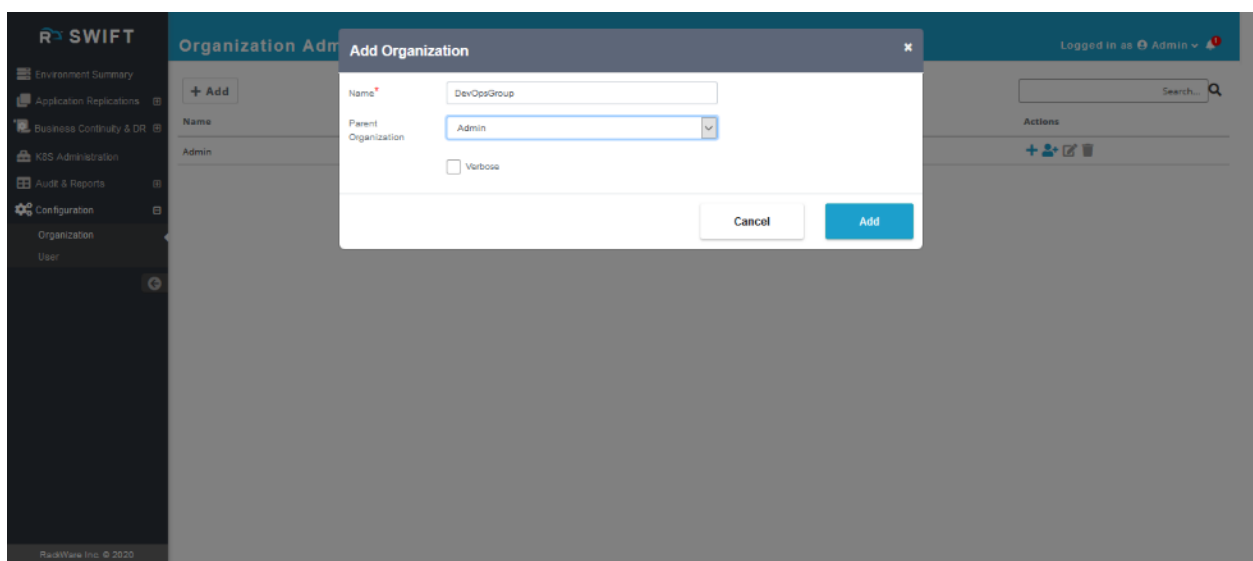
- Admin users
These have full access rights to the corresponding organization as well as all child organizations. Admin users can also add or remove child organizations and users within their organization.
- Operator users
These users can not add or remove any child organization or users from any of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

Create a new organization

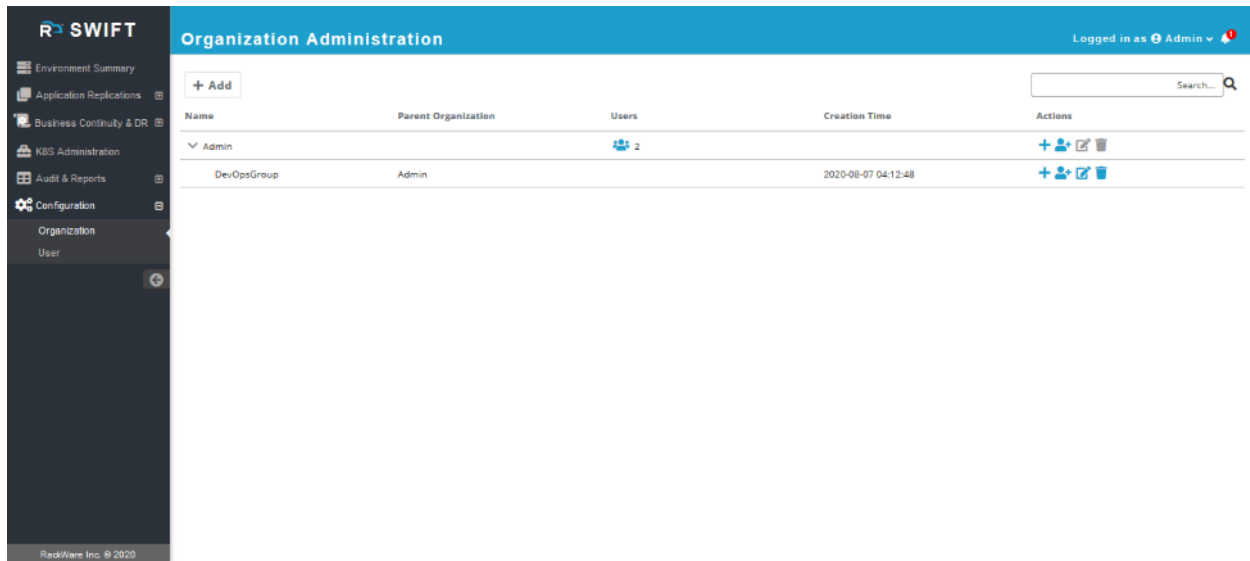
Login to the SWIFT dashboard and navigate to the Configuration menu and Organization sub-menu.



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.



You will see the newly added organization now.



Organization Administration

Logged in as Admin

+ Add

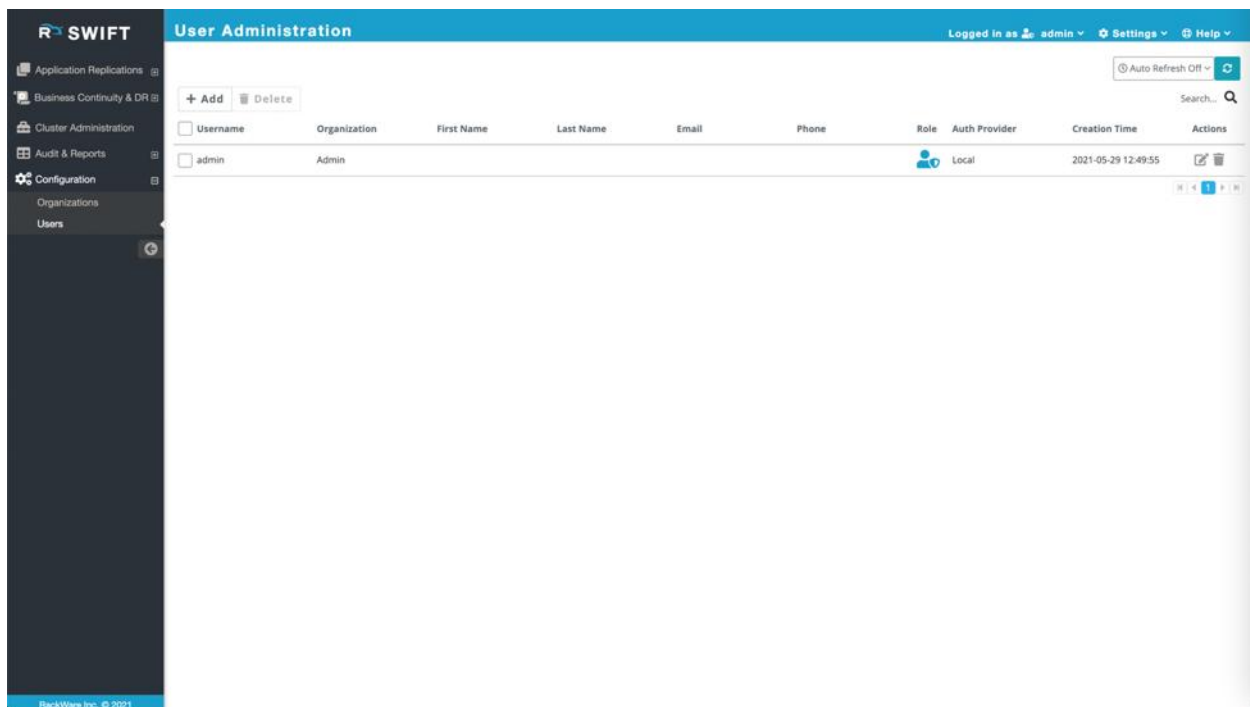
Search...

Name	Parent Organization	Users	Creation Time	Actions
Admin		2		+ [edit] [delete]
DevOpsGroup	Admin		2020-08-07 04:12:48	+ [edit] [delete]

RackWare Inc. © 2020

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

Logged in as admin

+ Add [Delete]

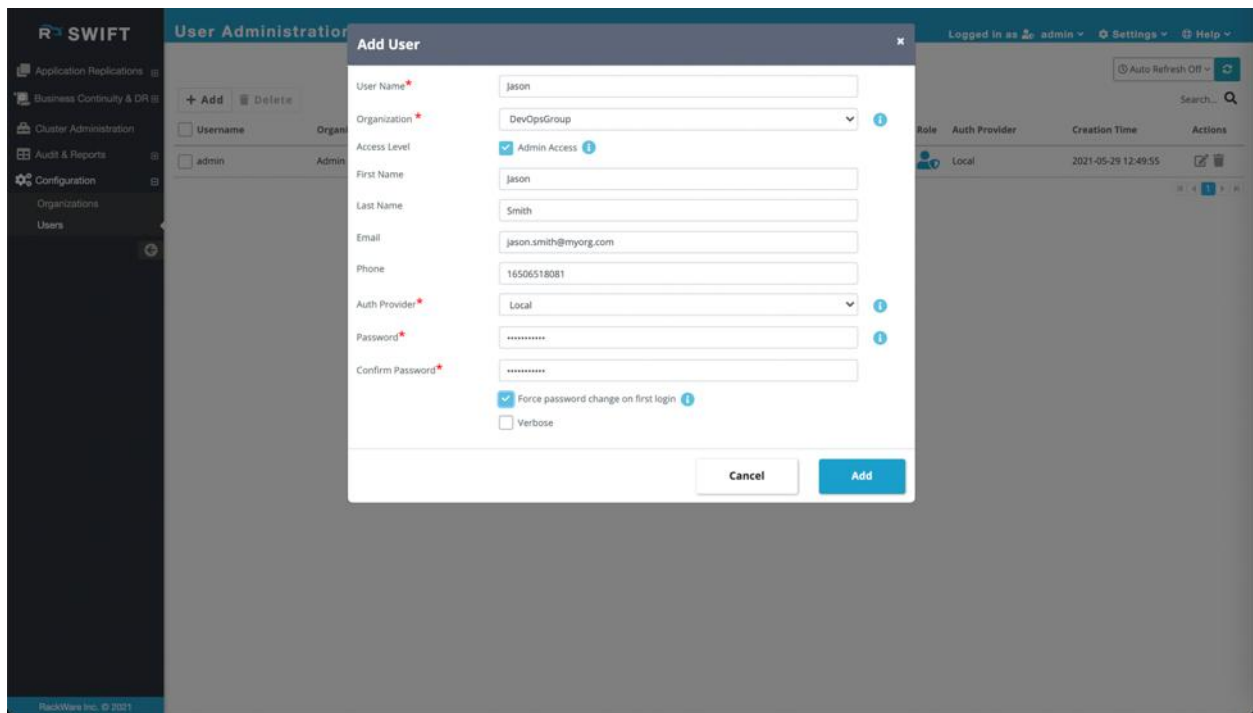
Auto Refresh Off

Search...

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

RackWare Inc. © 2021

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



The screenshot shows the 'Add User' modal form in the Rackware SWIFT User Administration interface. The form is titled 'Add User' and contains the following fields and options:

- User Name***: Text input field with the value 'jason'.
- Organization***: Dropdown menu with the selected value 'DevOpsGroup'.
- Access Level**: Checkboxes for 'Admin Access' (checked) and 'User Access' (unchecked).
- First Name**: Text input field with the value 'jason'.
- Last Name**: Text input field with the value 'Smith'.
- Email**: Text input field with the value 'jason.smith@myorg.com'.
- Phone**: Text input field with the value '16506518081'.
- Auth Provider***: Dropdown menu with the selected value 'Local'.
- Password***: Password input field with masked characters '*****'.
- Confirm Password***: Password input field with masked characters '*****'.
- Force password change on first login**: Checkboxes for 'Force password change on first login' (checked) and 'Verbose' (unchecked).

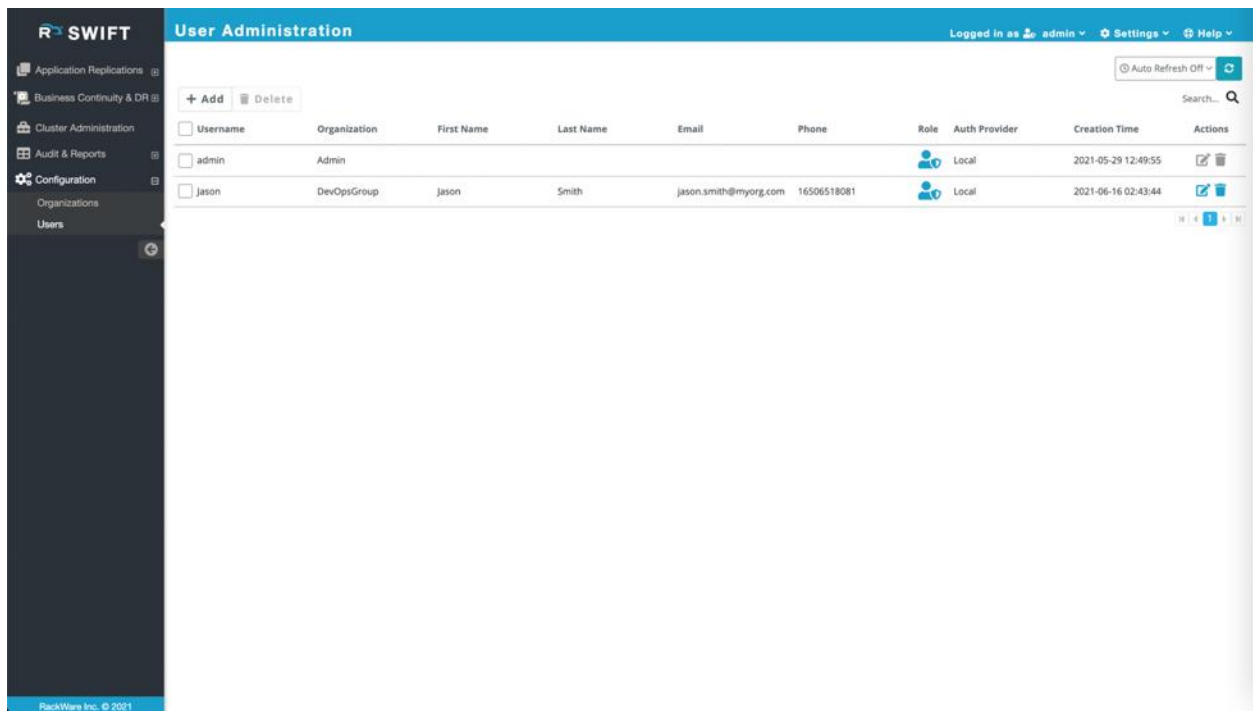
At the bottom of the modal, there are 'Cancel' and 'Add' buttons. The background shows the 'User Administration' page with a table of users, including one named 'admin' with role 'Admin'.

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.



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.

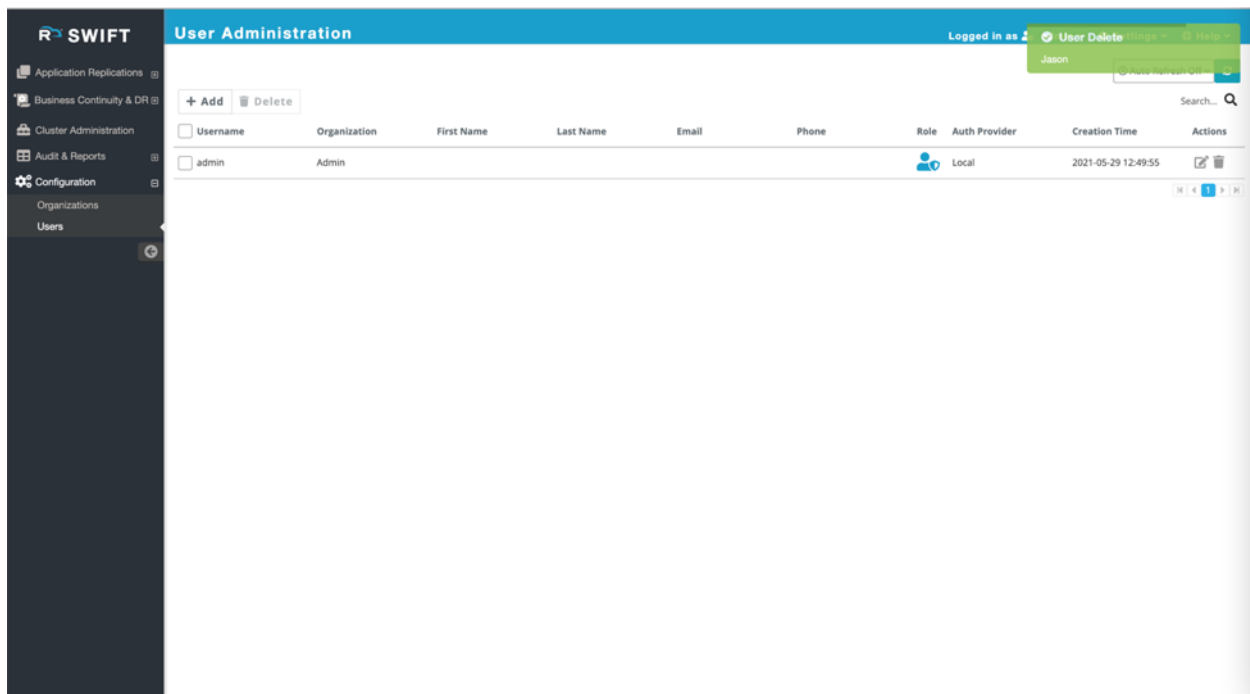
The screenshot shows the 'User Administration' page in the Rackware SWIFT interface. The left sidebar contains navigation links: Application Replications, Business Continuity & DR, Cluster Administration, Audit & Reports, Configuration, Organizations, and Users. The main content area displays a table of users with columns: Username, Organization, First Name, Last Name, Email, Phone, Role, Auth Provider, Creation Time, and Actions. Two users are listed: 'admin' (Admin, Local) and 'jason' (DevOpsGroup, Local). The 'jason' user is selected, and the 'Delete' button is visible in the top right of the table area.

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

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

The screenshot shows the same 'User Administration' page, but with a confirmation dialog box open. The dialog box is titled 'Delete: jason' and contains the text 'Are you sure you want to delete this user jason?'. It has two buttons: 'Cancel' and 'Delete'. The 'jason' user in the background table is now checked, indicating it is the user being targeted for deletion.

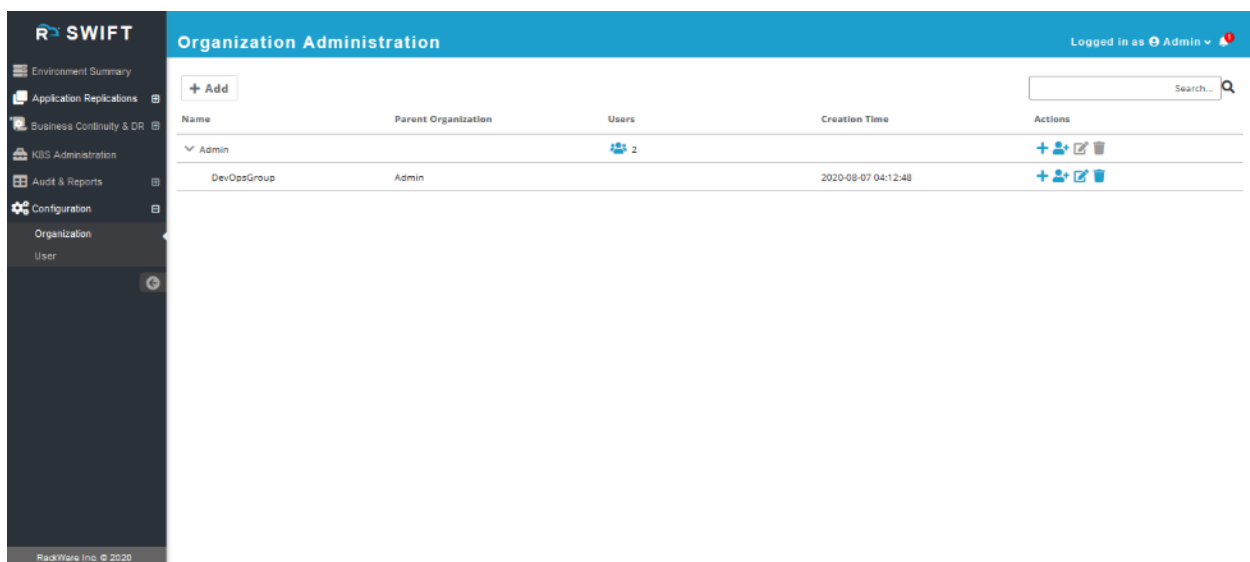
Confirm, and the user will be deleted permanently.



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

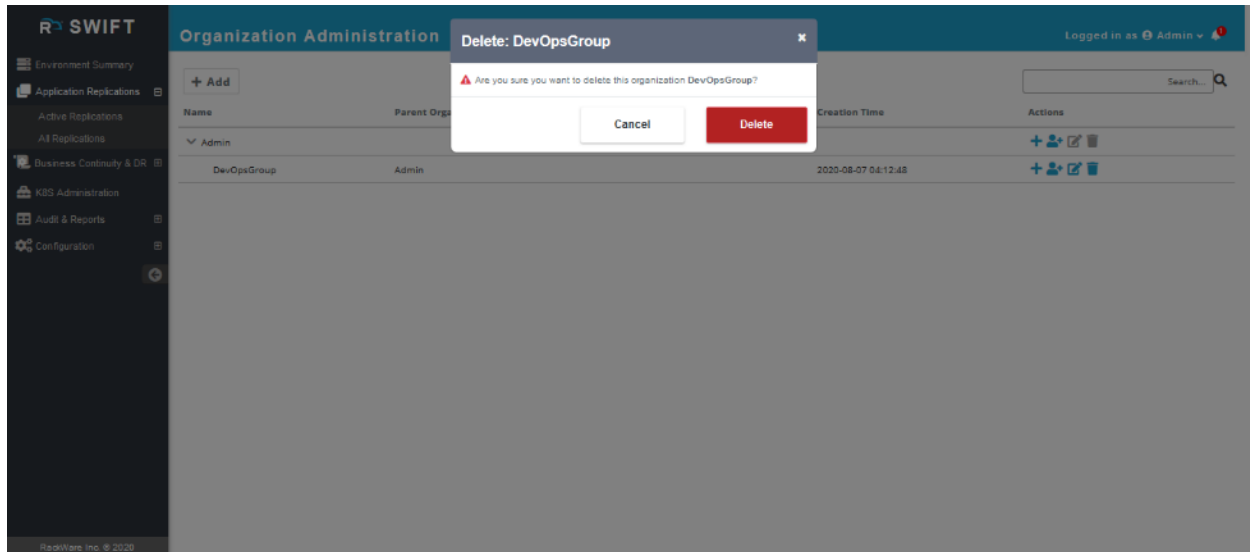
Deleting an organization

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

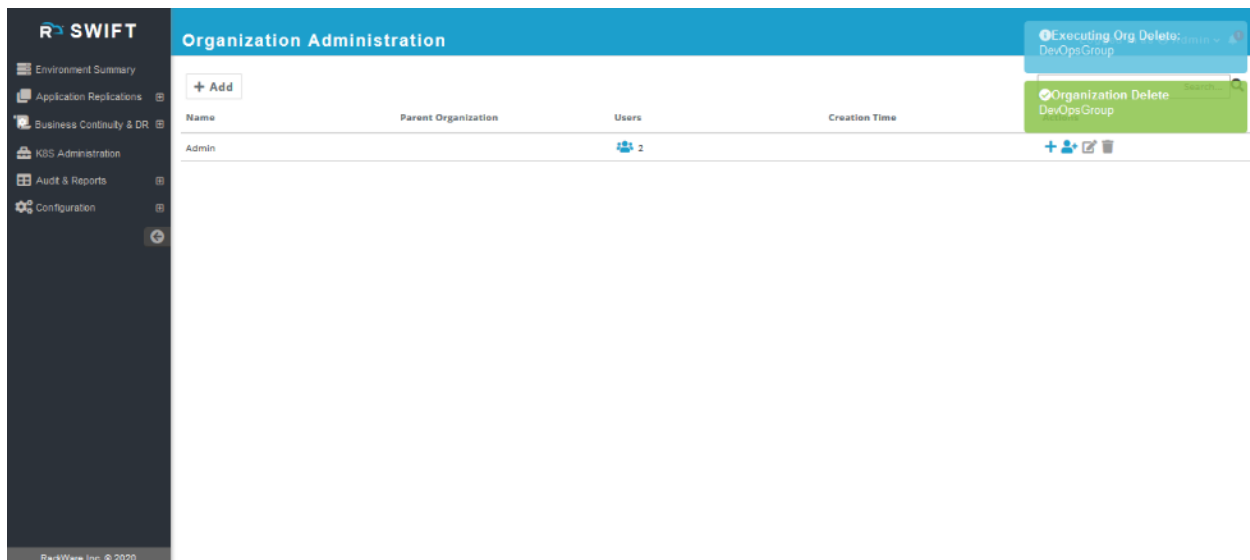


Press the delete button (the one with a dustbin icon) next to the organization which you want to delete. You will get a confirmation dialog and press the 'Delete' button to continue. Note that if you have one or

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.



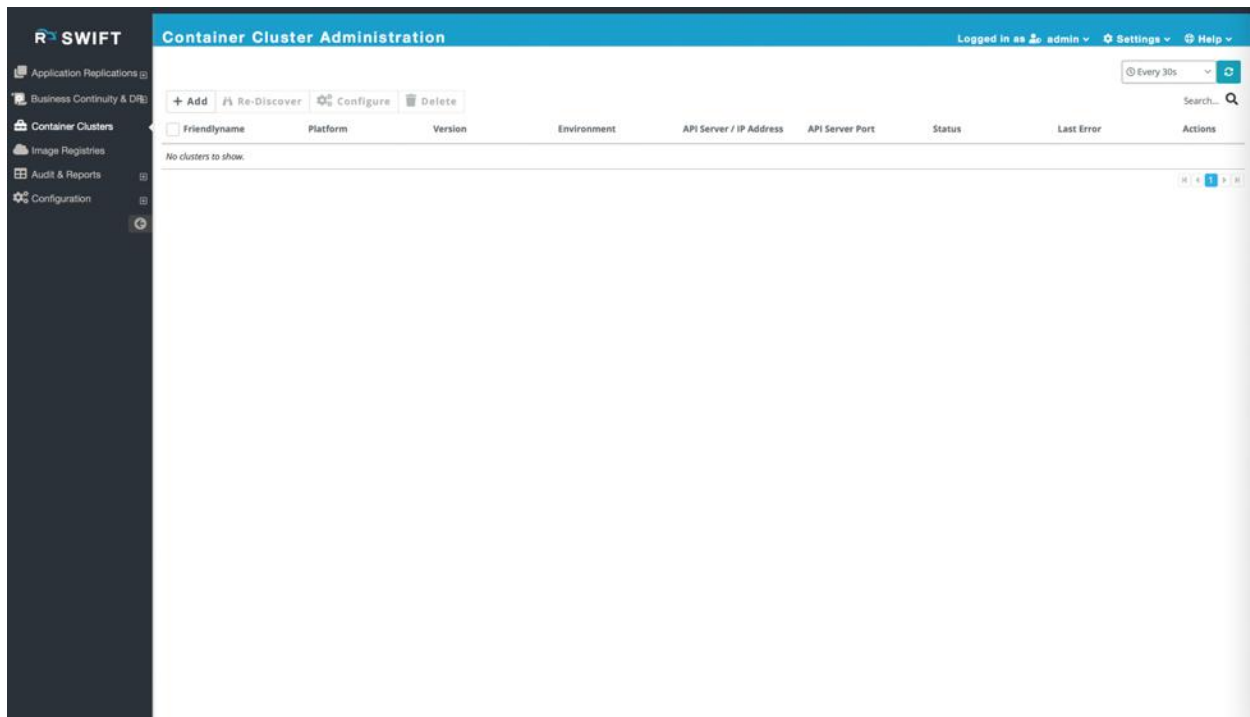
Once deleted, you will see that the organization is no longer on the page.



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.



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

Cloud Type*

LOCAL Configuration

IP Address / DNS Name*

Port*

☒ Input Key
☐ Upload Key File

Key*

Key is Required

TRAIPOD Config

☒ TRAIPOD Image and Secret
☐ TRAIPOD Registry

TRAIPOD Image

TRAIPOD Image Secret

☐ TRAIPOD No Special Capabilities

Cluster Private IP Config

☒ API IP Address
☐ Host Name (FQDN)

Cluster API IP address

☐ Verbose

> Advanced options

Cancel

Add

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.

26 | Page

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

Limit

Memory Request/Limit

Request (MB)

Limit (MB)

Cancel

Add

Very similar input is needed for local OpenShift cluster.

Cluster Add

General Options

Platform Type

☐ Kubernetes(K8S)
☒ OpenShift

Friendly Name

SWIFT autogenerated a friendlyname if left empty

Cloud Type*

Native Local

LOCAL Configuration

IP Address / DNS Name*

Enter API Server's IP Address/DNS Name

Port*

Enter API Server's Port

Input Key

Upload Key File

Key*

TRAIPOD Config

TRAIPOD Image and Secret

TRAIPOD Registry

TRAIPOD Image

rackware-trai:latest

TRAIPOD Image Secret

docker-registry-secret

TRAIPOD No Special Capabilities

Cluster Private IP Config

API IP Address

Host Name (FQDN)

Cluster API IP address

Enter API IP address

Verbose

Advanced options

Cancel

Add

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

28 | Page

Cluster Add

General Options

Platform Type

☒ Kubernetes(K8S)
 ☐ OpenShift

Friendly Name

SWIFT autogenerated a friendlyname if left empty

Cloud Type*

Oracle OCI

OCI Configuration

Cluster Name In OCI cloud*

User ID*

Compartment ID*

Tenant ID*

API Key's Fingerprint*

Region*

--Select Region--

Cluster ID

Private Key File*

+ Browse

Drop file to upload, or Browse

TRAIPOD Config

☒ TRAIPOD Image and Secret
 ☐ TRAIPOD Registry

TRAIPOD Image

rackware-trai:latest

TRAIPOD Image Secret

docker-registry-secret

☐ TRAIPOD No Special Capabilities

Cluster Private IP Config

☒ API IP Address
 ☐ Host Name (FQDN)

Cluster API IP address

Enter API IP address

Port

Enter API Server's Port

☐ Verbose

Advanced options

Cancel



Add


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 autogenerated 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 Config 


☒ TRAIPOD Image and Secret
☐ TRAIPOD Registry


TRAIPOD Image

rackware-trail:latest


TRAIPOD Image Secret


docker-registry-secret


☐ TRAIPOD No Special Capabilities 

Cluster Private IP Config 

☒ API IP Address
☐ Host Name (FQDN)


Cluster API IP address

Enter API IP address


Port

Enter API Server's Port

☐ Verbose




Advanced options


Cancel

Add

31 | Page

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.


Cluster Add



General Options

Platform Type

☐ Kubernetes(K8S)
☒ OpenShift

Friendly Name


SWIFT autogenerated 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*


+ Browse

 Drop file to upload, or Browse


TRAIPOD Config 


☒ TRAIPOD Image and Secret
☐ TRAIPOD Registry


TRAIPOD Image

rackware-trai:latest


TRAIPOD Image Secret


docker-registry-secret


☐ TRAIPOD No Special Capabilities 

Cluster Private IP Config 

☒ API IP Address
☐ Host Name (FQDN)

Cluster API IP address

Enter API IP address


☐ Verbose

> **Advanced options**

Cancel

Add

32 | Page

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

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 Config

☒ TRAIPOD Image and Secret
☐ TRAIPOD Registry

TRAIPD Image

TRAIPD Image Secret

☐ TRAIPOD No Special Capabilities

Cluster Private IP Config

☒ API IP Address
☐ Host Name (FQDN)

Cluster API IP address

Port

☐ Verbose

> Advanced options

Cancel

Add

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 Config

☒ TRAIPOD Image and Secret
☐ TRAIPOD Registry

TRAIPOD Image

rackware-trai:latest

TRAIPOD Image Secret

docker-registry-secret

☐ TRAIPOD No Special Capabilities

Cluster Private IP Config

☒ API IP Address
☐ Host Name (FQDN)

Cluster API IP address

Enter API IP address

☐ Verbose

Advanced options

Cancel



Add


35 | Page

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 Config


☒ TRAIPOD Image and Secret
☐ TRAIPOD Registry

TRAIPOD Image

rackware-trai:latest


TRAIPOD Image Secret


docker-registry-secret


☐ TRAIPOD No Special Capabilities


Cluster Private IP Config

☒ API IP Address
☐ Host Name (FQDN)

Cluster API IP address

Enter API IP address


Port

Enter API Server's Port



☐ Verbose



Advanced options

Cancel

Add

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 autogenerated 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 Config



☒ TRAIPOD Image and Secret
☐ TRAIPOD Registry


TRAIPOD Image

rackware-trai:latest


TRAIPOD Image Secret


docker-registry-secret


☐ TRAIPOD No Special Capabilities



Cluster Private IP Config


☒ API IP Address
☐ Host Name (FQDN)

Cluster API IP address

Enter API IP address


☐ Verbose


Advanced options

Cancel



Add


39 | Page

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


Cloud Type*


IBM Configuration


Cluster Name in IBM cloud*


☒ Input API Key
☐ Upload API Key File


API Key*



TRAIPOD Config


☒ TRAIPOD Image and Secret
☐ TRAIPOD Registry


TRAIPOD Image



TRAIPOD Image Secret


☐ TRAIPOD No Special Capabilities



Cluster Private IP Config


☒ API IP Address
☐ Host Name (FQDN)

Cluster API IP address


Port


☐ Verbose

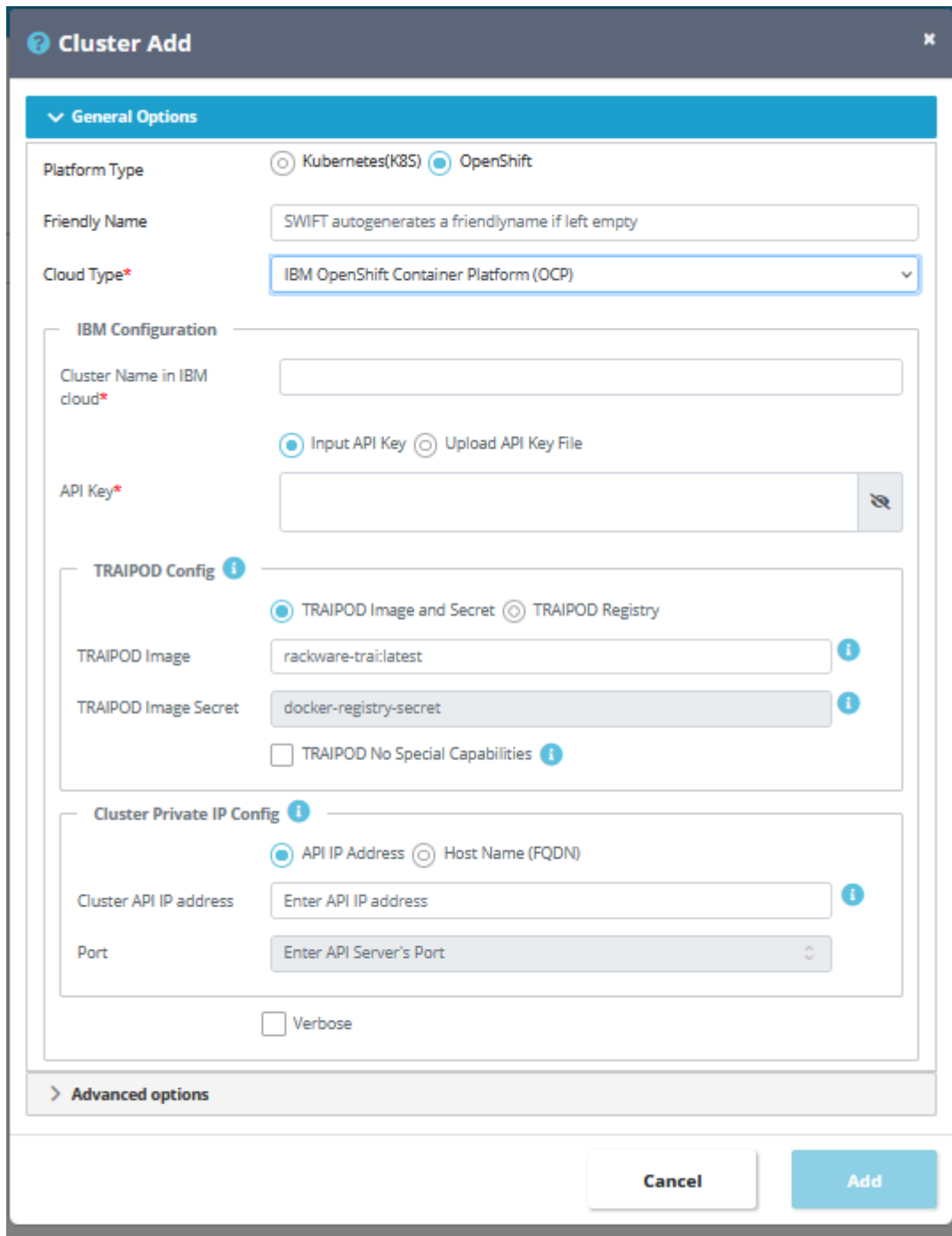

Advanced options

Cancel

Add

40 | Page

A very similar input is needed for IBM OpenShift clusters.



Cluster Add

General Options

Platform Type: ☐ Kubernetes(K8S) ☒ OpenShift

Friendly Name:

Cloud Type*:

IBM Configuration

Cluster Name in IBM cloud*:

API Key*: ☒ Input API Key ☐ Upload API Key File

TRAIPOD Config

☒ TRAIPOD Image and Secret ☐ TRAIPOD Registry

TRAIPOD Image:

TRAIPOD Image Secret:

☐ TRAIPOD No Special Capabilities

Cluster Private IP Config

☒ API IP Address ☐ Host Name (FQDN)

Cluster API IP address:

Port:



☐ Verbose


> Advanced options

Cancel **Add**

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 autogenerated 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 Config 


☒ TRAIPOD Image and Secret
☐ TRAIPOD Registry


TRAIPOD Image

rackware-trai:latest


TRAIPOD Image Secret


docker-registry-secret


☐ TRAIPOD No Special Capabilities 


Cluster Private IP Config 

☒ API IP Address
☐ Host Name (FQDN)

Cluster API IP address

Enter API IP address


☐ Verbose


Advanced options

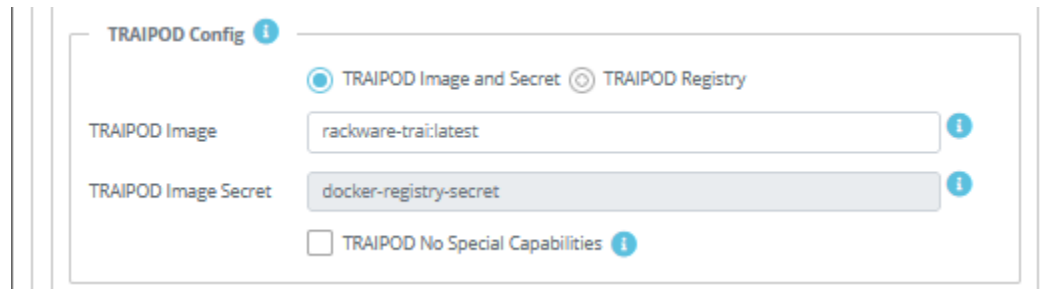
Cancel

Add

42 | Page

Other Common Inputs

TRAI Name and Secret

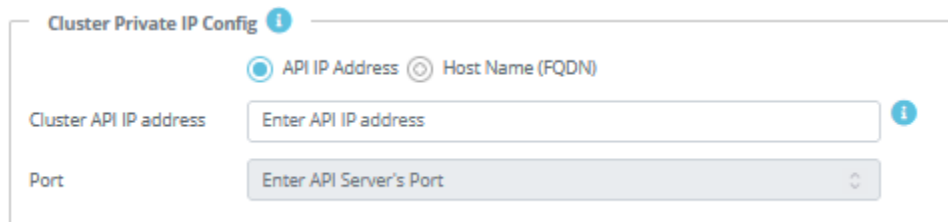


Transient RackWare Agent Image POD (TRAIPOD) Image and Image-Secret defaults are optional inputs. They can be configured for each sync, and the values for the sync will override the defaults set for the cluster. These defaults can be changed at any time using the 'Configure' button from the 'K8S Administration' page. Please see the TRAI section for more details on what this image is and how it is used during syncs.

You can optionally select pre-discovered image registry to use for syncs to/from the cluster where SWIFT will upload TRAI image to the selected registry (for sync time) and delete it once sync finishes.

Selecting the no special capabilities flag disables special kernel capabilities provisioning for the TRAIPOD. You may have to turn this setting ON if one or more of your cluster nodes are running on an older Linux kernel. Set this ON if you see a TRAIPOD deploy failure and error like 'invalid CapAdd: unknown capability' for the TRAIPOD deploy (in the POD events).

Cluster's private IP or FQDN

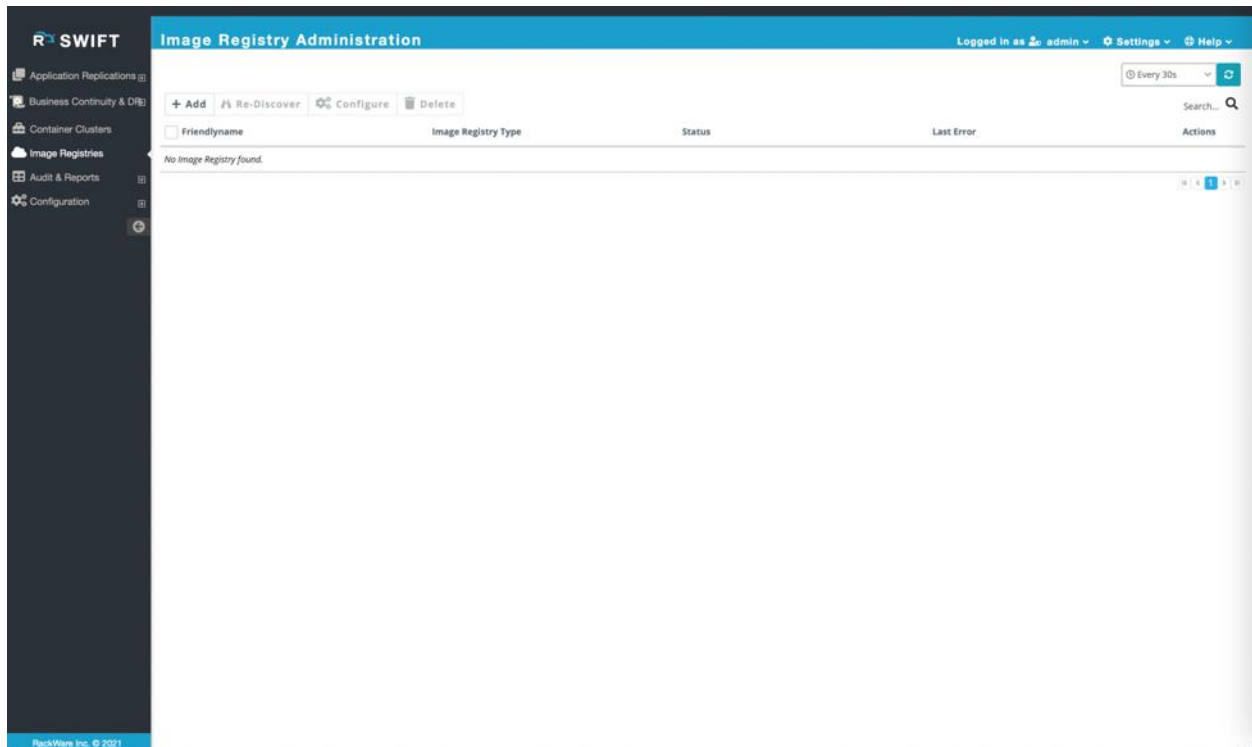


For cloud-based clusters, you will optionally get an option to input Cluster API IP/FQDN and port. These options allow you to override IP/name used by SWIFT to connect to the cluster. Use this option if your cluster is behind private network/firewall and is not reachable publicly.

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.




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

Cancel



Add

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

46 | Page


Image Registry Add


Friendly Name

SWIFT autogenerated 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



Cancel

Add

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*

+ Browse

 Drop file to upload, or Browse

☐ Verbose



Cancel

Add

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


Image Registry Add


Friendly Name

SWIFT autogenerated a friendlyname if left empty

Image Registry Type*

Google GCP

GCP/GCR Configuration

Hostname

--Enter New Hostname / Select Hostname--

Private key File*

+ Browse

Drop file to upload, or Browse

☐ Verbose

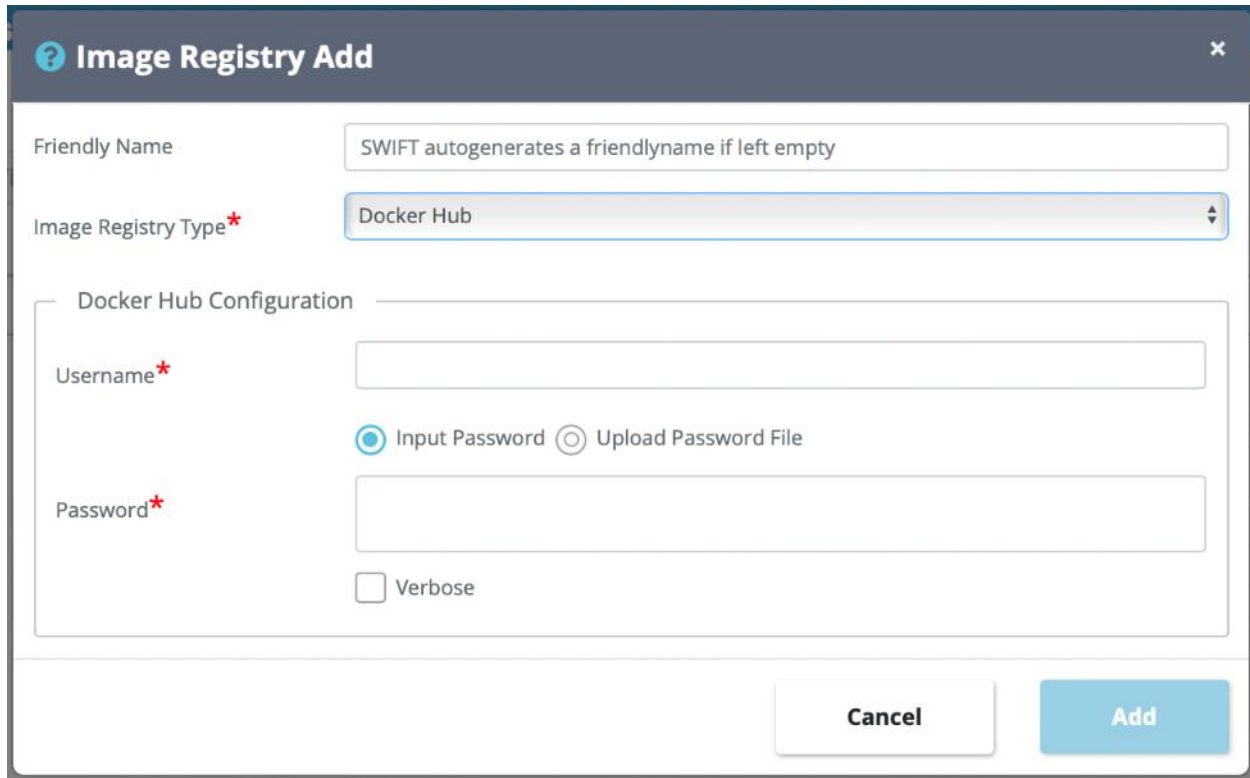
Cancel

Add

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



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

Limit

Memory Request/Limit




Request (MB)

Limit (MB)


Cancel


Add

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.


Configure: CephK8S


> General Options


Advanced options

Ceph Dashboard Secret 

Secret Namespace

default


Secret Name

ceph-secret

4 Ceph Secret Name(s) Found

☐ Clear Ceph Dashboard Secret

TRAIPOD Configuration

CPU Request/Limit 

Request


N

☐ Set to Remote Cluster Default

Limit

N

☐ Set to Remote Cluster Default

Memory Request/Limit 

Request (MB)

N

Limit (MB)

N

Cancel

Configure

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.

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

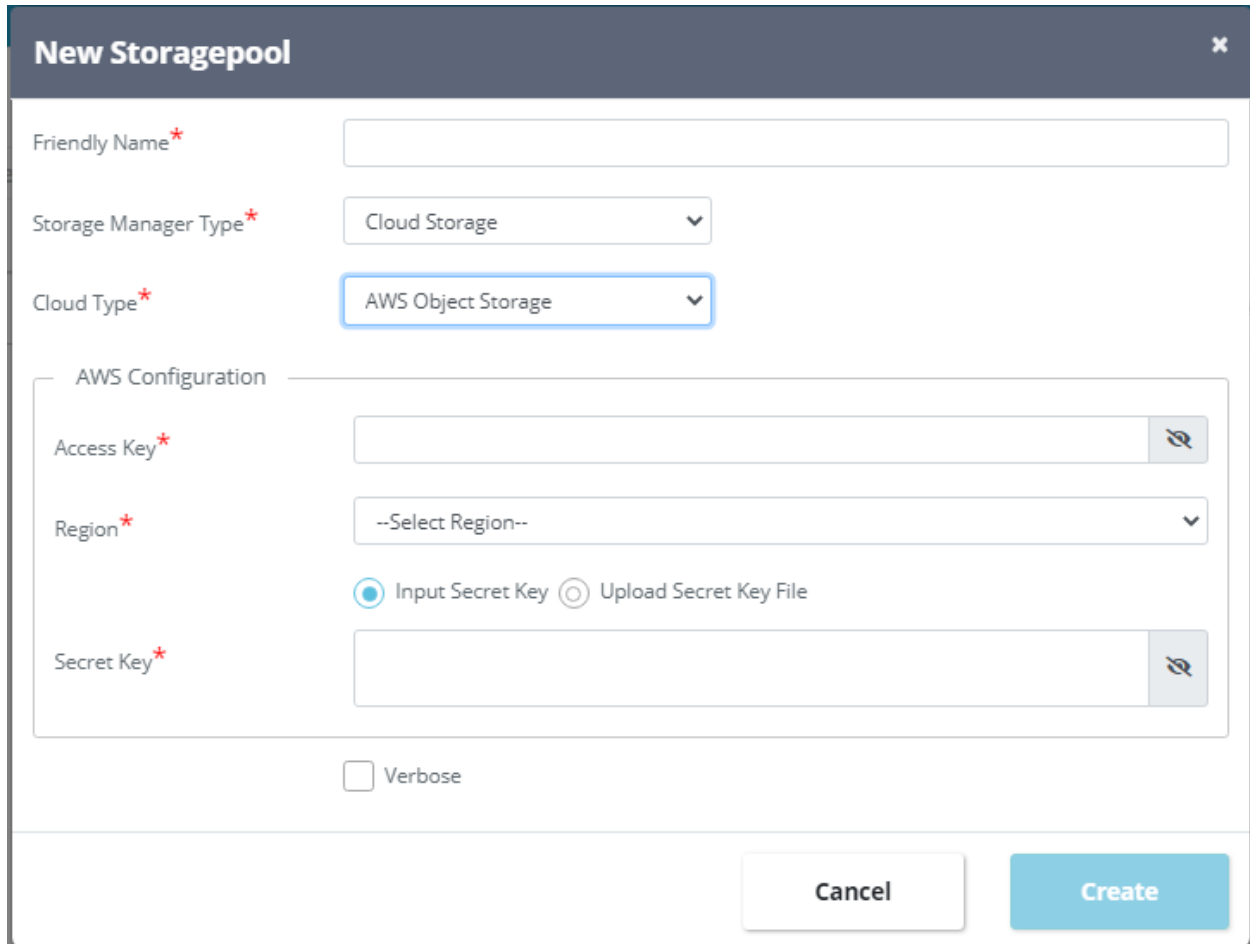
From the Storage pool administration screen, press the 'New' button. Select 'ZFS' as a type for the pool, which means a local pool.

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

Cancel Create

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)

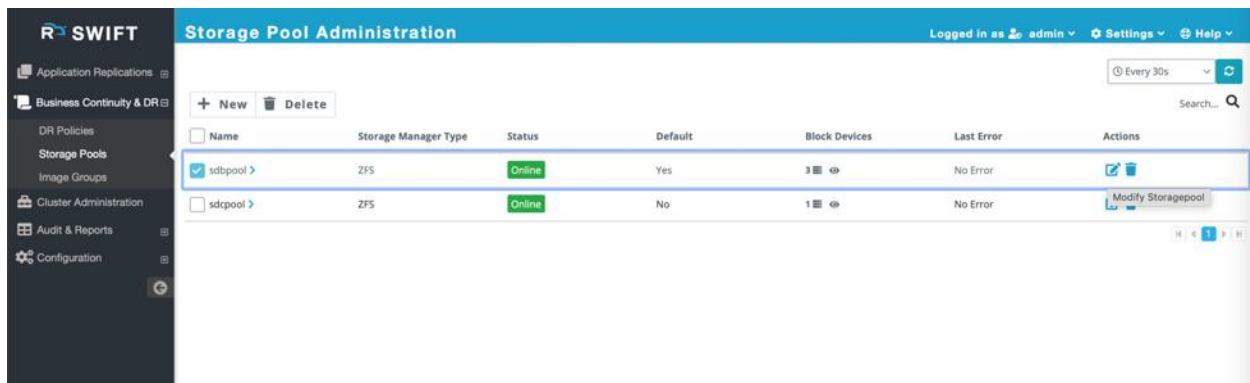
IBM Cloud

- 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.



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

Modify Storagepool: sdbpool

Friendly Name*

Storage Manager Type*

ZFS

Add Block Device(s)

Block Device

Select Block Device

+

Block Devices*

/dev/sdb1

/dev/sdb2

☒ Set as Default

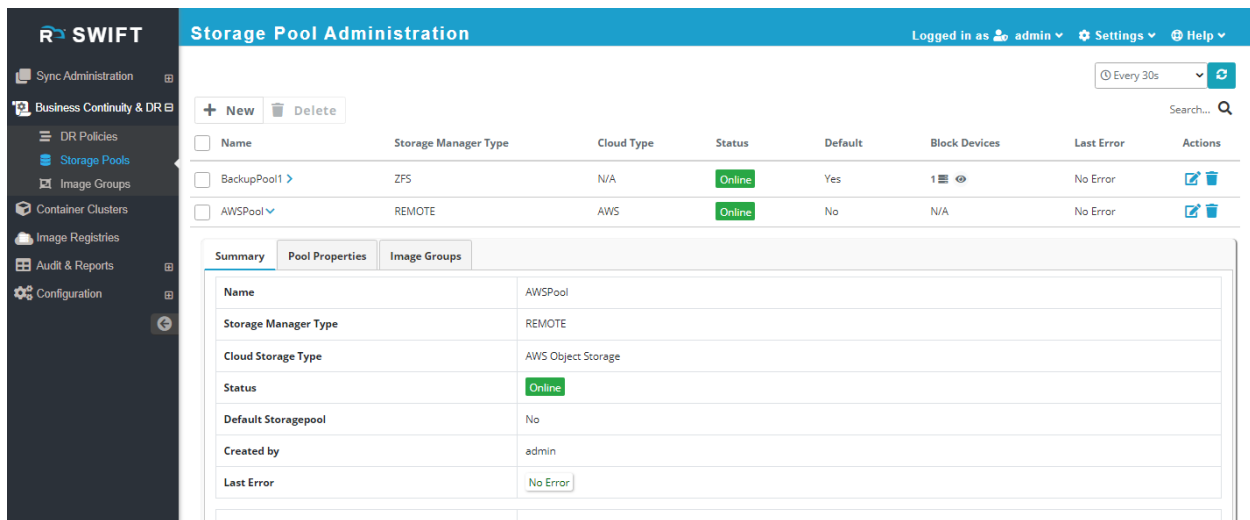
☐ Verbose

Cancel

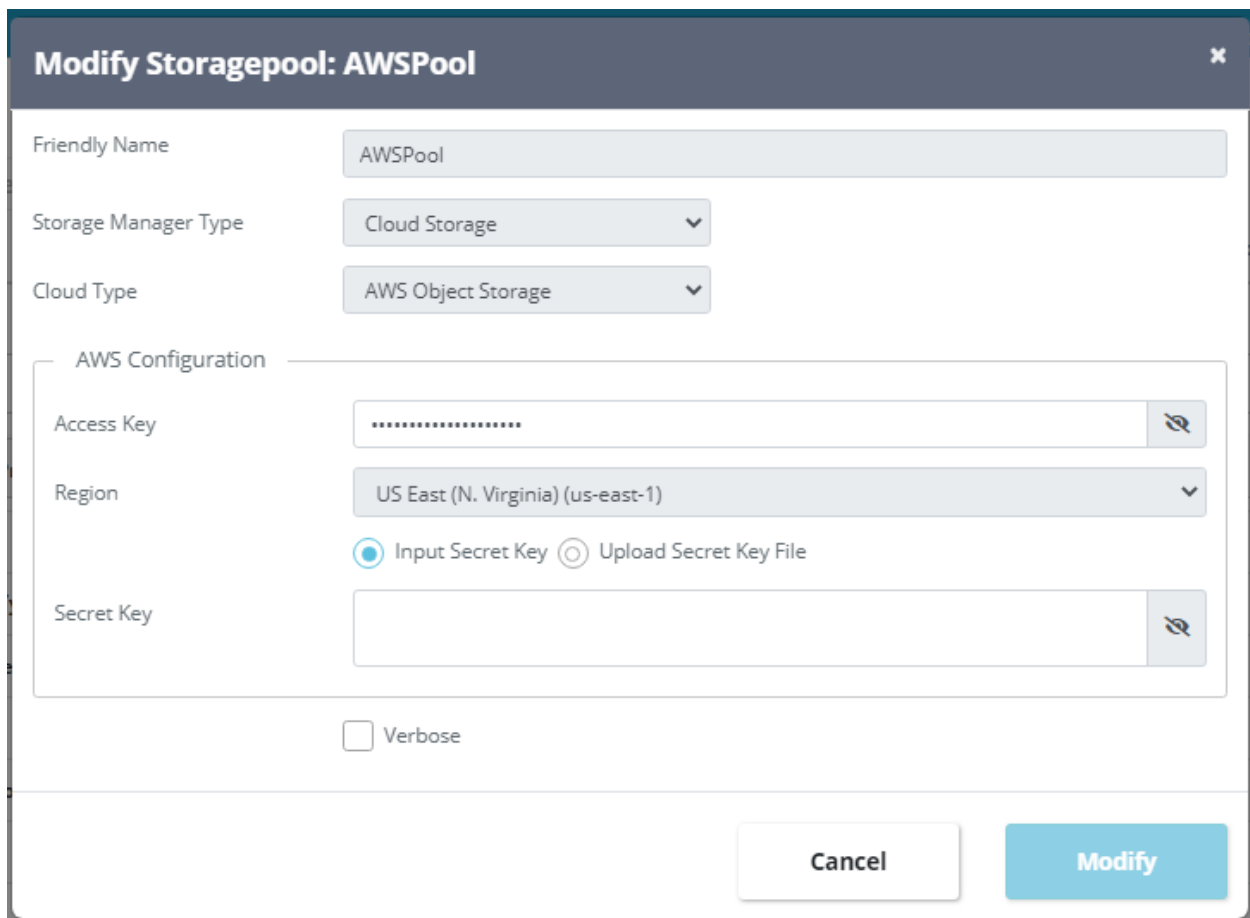
Modify

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.



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).



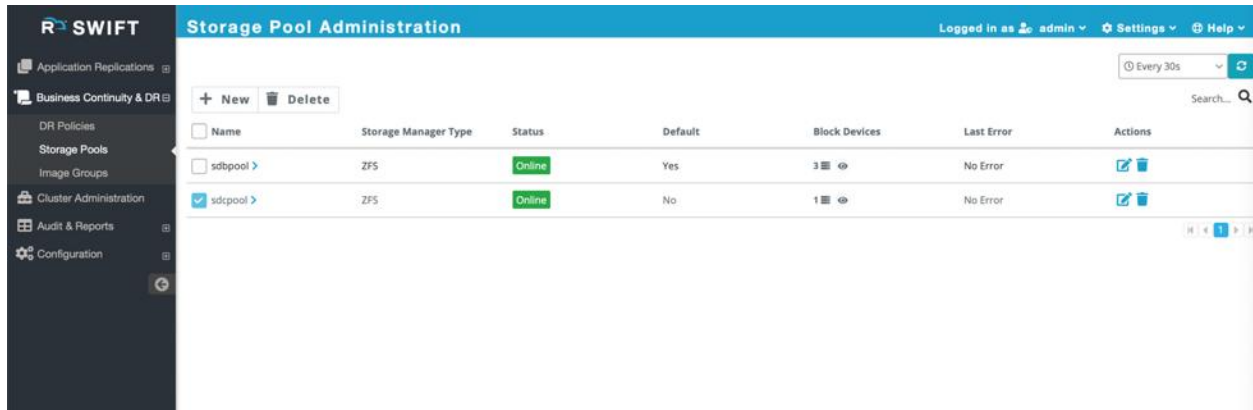
Options allowed for modify include API and access keys, cloud usernames, etc. Only part of the options that you see during cloud storage pool create are allowed for modification.

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

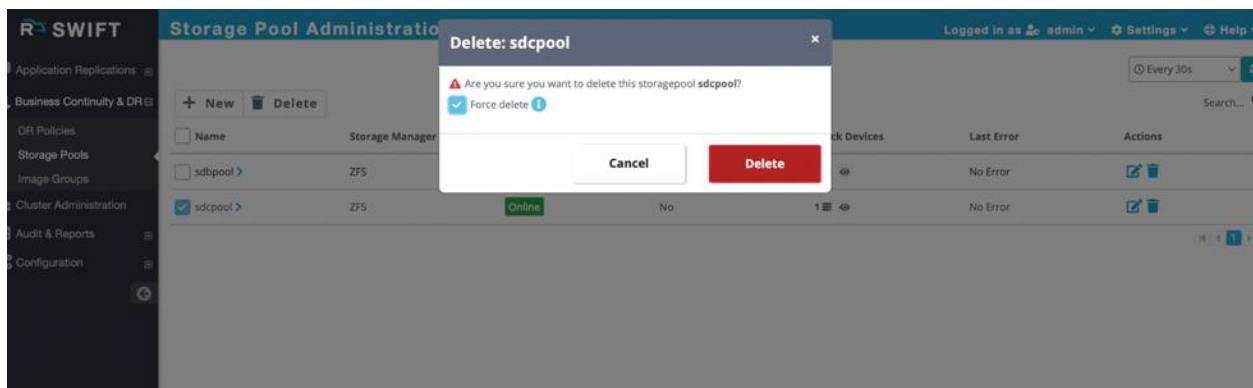
Deletion steps remain the same irrespective of whether the pool being deleted is local or remote.

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



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

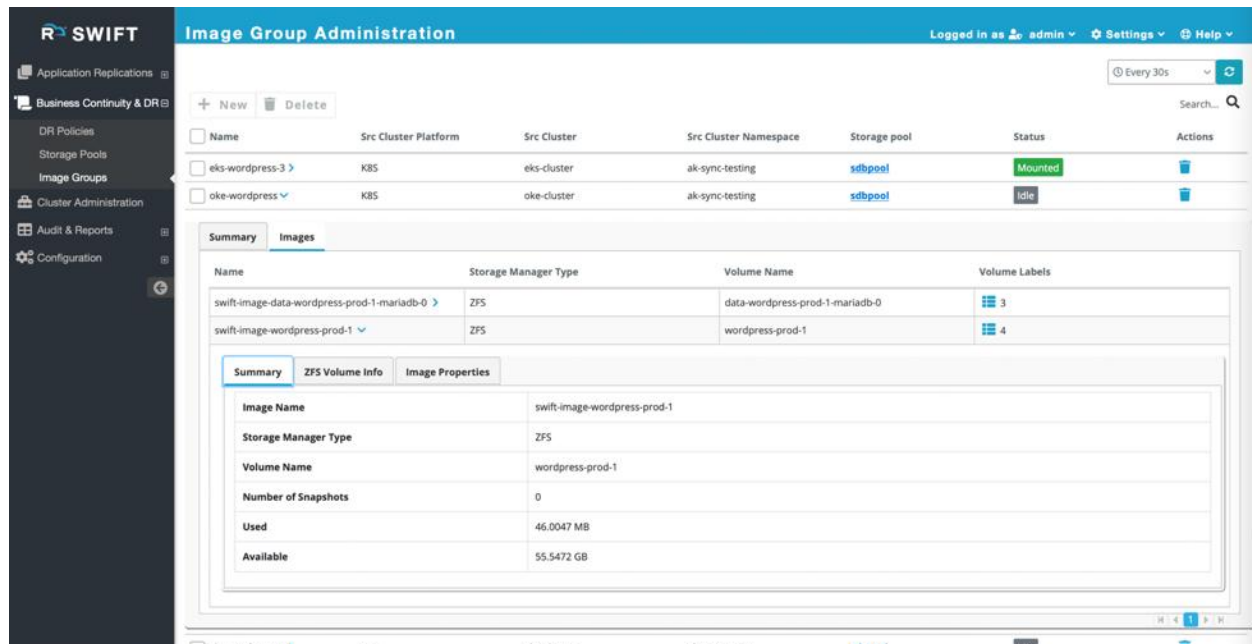
If there are any Image-groups or backups captured to the deleted pool, then pool deletion will error on them asking you delete those first.



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	stlpool	Mounted	
oke-wordpress	K8S	oke-cluster	ak-sync-testing	stlpool	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	Storage Manager Type	Volume Name	Number of Snapshots	Used	Available
swift-image-wordpress-prod-1	ZFS	wordpress-prod-1	0	46.0047 MB	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-Group 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.

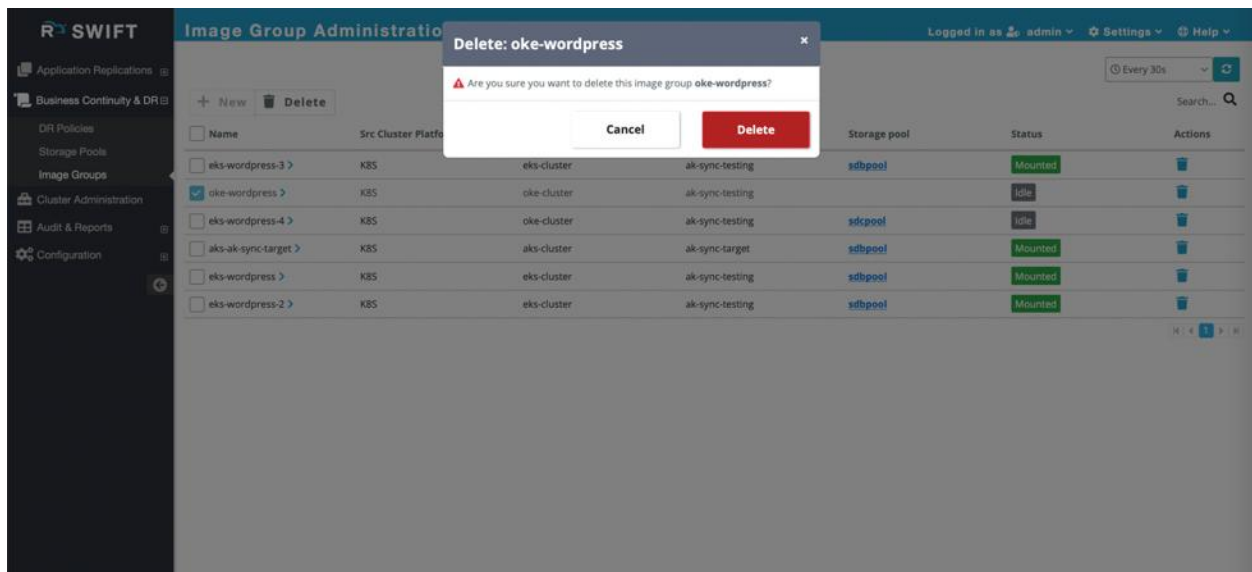
From the Image-Group administration menu, select Image-Group press the 'New' button.

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.






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

Cloud Type


LOCAL Configuration

IP Address


☒ Input Key
☐ Upload key File

Key


Port

TRAIPOD Image

TRAIPOD Image Secret

☐ TRAIPOD No Special Capabilities 

☐ Verbose


Advanced options



Cancel


Configure

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

Cloud Type


LOCAL Configuration

IP Address


☒ Input Key
☐ Upload key File

Key


Port

TRAIPOD Image

TRAIPOD Image Secret

☐ TRAIPOD No Special Capabilities 

☐ Verbose



Advanced options


Cancel

Configure

64 | Page

For cloud based cluster:


Configure: oke-cluster
×


General Options


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..aaaaaaaq5vxtrswwthmjayaciyv5qvp5pd7ghmdqcosikralgy7r4tcpybq

Compartment ID

ocid1.compartment.oc1..aaaaaaaanrqvfc33b47mxxqvorbva6ou7kt46hgeomkct5tykopjwr3mly.


Tenant ID

ocid1.tenancy.oc1..aaaaaaa6hlgt3fmffpy63cj5zklidifpmw7w4hph3axkquwgdiezyfbevzq


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


Advanced options

Cancel

Configure

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 that may bring newer TRAI image.

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

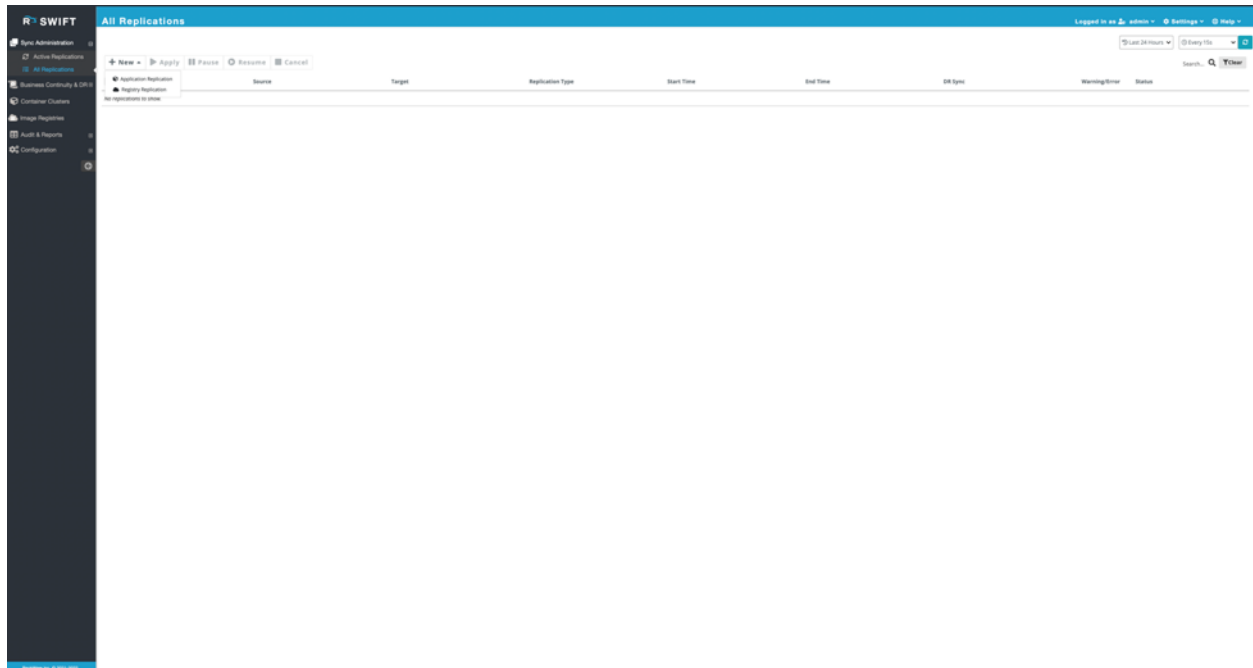
This mode is a mix of stage1 and stage2 syncs together. You will configure details for the source and target cluster as well as the image-group name. Syncs in this mode would run for both stage1 and stage2 parts, and this also remains the same for registry syncs too.

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

Source

Platform Type

☒ Kubernetes
 ☐ OpenShift

Cluster Name*

--Select Cluster--

Source Cluster is Required

Namespace*

--Select Namespace--

Applications*

☒ All
 ☐ Selective

☐ Include K8S Native Objects

Sync Webhooks

☐ All
 ☐ Native Webhooks

☐ Don't Delete Taints

Target

Platform Type

☒ Kubernetes
 ☐ OpenShift

Cluster Name*

--Select Cluster--

Target Cluster is Required

Namespace*

--Select Namespace--

Storage Class*

--Select Storageclass--

Custom Resource Configuration

Choose Custom Resources

CRD Scope

☒ Cluster
 ☐ Namespace

Search by CRD(s) name

☐ CUSTOMRESOURCEDEFINITION

Please Select Source Cluster/Namespace

CR/CRD Object List

Clear All

+ Add CR/CRD Object(s)...

TRAIPOD Options

Source

IP Type

--Select IP Type--

IP Address

☒ Auto Select IP Address
 ☐ Specific IP Address

SWIFT will auto select one of the reachable IPs.

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

Target

IP Type

--Select IP Type--

IP Address

☒ Auto Select IP Address
 ☐ Specific IP Address

SWIFT will auto select one of the reachable IPs.

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

☐ Verbose
 ☐ Dry Run

Job Name

Replication Job Name

Advanced options

Cancel

Add

70 | Page

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 many 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

No storagepool found to configure a replication.

To Add + Storagepool(s), please navigate to the [Storagepool Administration](#) page.

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

☐ Include K8S Native Objects

Sync Webhooks

☐ All
☐ Native Webhooks

☐ Don't Delete Taints

Target

Platform Type

☒ Kubernetes
☐ OpenShift

Storagepool*

--Select Storagepool--

Imagegroup*

☒ New
☐ Existing

Enter Imagegroup Name

Custom Resource Configuration

Choose Custom Resources

CRD Scope

☒ Cluster
☐ Namespace

Search by CRD(s) name

☐ CUSTOMRESOURCEDEFINITION

Please Select Source Cluster/Namespac

CR/CRD Object List

+ Add CR/CRD Object(s)...

TRAIPOD Options

Source

IP Type

--Select IP Type--

IP Address

☒ Auto Select IP Address
☐ Specific IP Address

SWIFT will auto select one of the reachable IPs.

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

☐ Verbose
☐ Dry Run

Job Name

Replication Job Name

Advanced options

Backup options

Cancel

Add

72 | Page

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

No storagepool found to configure a replication.

To Add + Storagepool(s), please navigate to the [Storagepool Administration](#) page.

Sync Type*

☐ Passthrough
☐ Stage-1
☒ Stage-2

Source

Platform Type

☒ Kubernetes
☐ OpenShift

Storagepool*

--Select Storagepool--

Imagegroup*

--Select Imagegroup--

No Imagegroups found associated to the kubernetes clusters.

Backup Name

--Select Backup--

Applications*

☒ All
☐ Selective

☐ Include K8S Native Objects

Sync Webhooks

☐ All
☐ Native Webhooks

☐ Don't Delete Taints

Target

Platform Type

☒ Kubernetes
☐ OpenShift

Cluster Name*

--Select Cluster--

Target Cluster is Required

Namespace*

--Select Namespace--

Storage Class*

--Select Storageclass--

Custom Resource Configuration

Choose Custom Resources

CRD Scope

☒ Cluster
☐ Namespace

Search by CRD(s) name

☐ CUSTOMRESOURCEDEFINITION

Please Select Source Imagegroup

CR/CRD Object List

+ Add CR/CRD Object(s)...

TRAIPOD Options

Target

IP Type

--Select IP Type--

IP Address

☒ Auto Select IP Address
☐ Specific IP Address

SWIFT will auto select one of the reachable IPs.

TRAIP 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

☐ Verbose
☐ Dry Run

Job Name

Replication Job Name

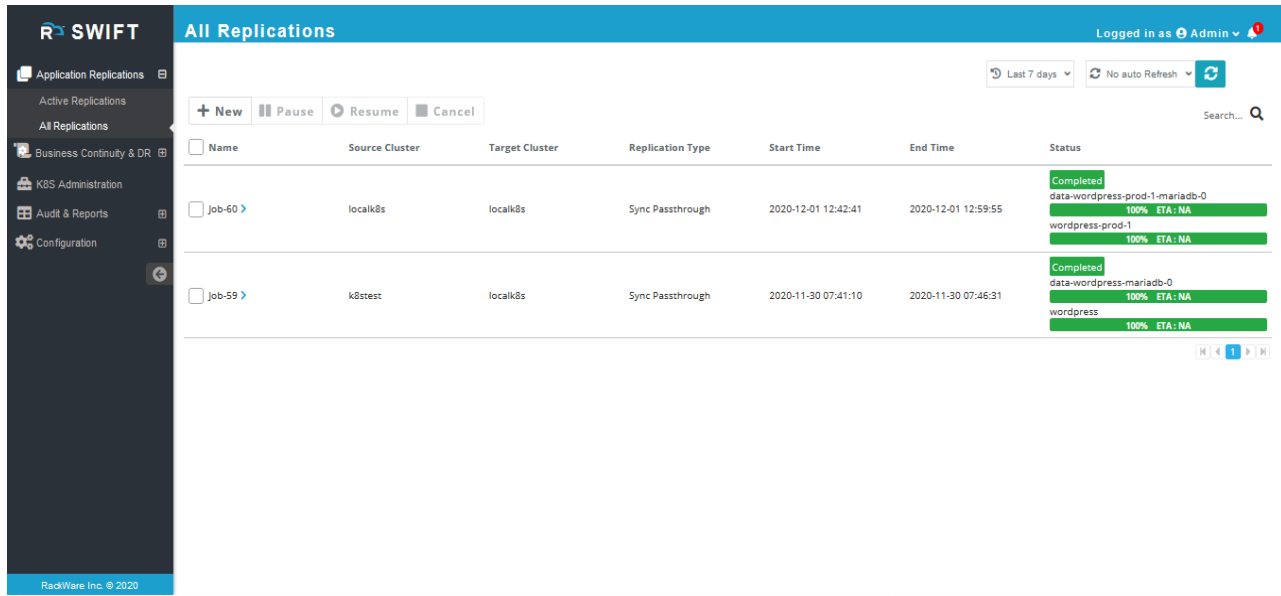
Advanced options

Cancel

Add

74 | Page

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.



The screenshot shows the 'All Replications' page in the R SWIFT interface. The page has a sidebar with navigation options: Application Replications, Active Replications, All Replications, Business Continuity & DR, K8S Administration, Audit & Reports, and Configuration. The main content area displays a table of replication jobs. The table has columns for Name, Source Cluster, Target Cluster, Replication Type, Start Time, End Time, and Status. There are two jobs listed, both with a status of 'Completed'.

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

Pre/Post Script Config

TRAI Config

Image Registry Config

Kubernetes Service Config

Volume Sync Config

Kubernetes Ingress Config

Source

Pre YAML

+ Browse

Post YAML

+ Browse

Target

Pre YAML

+ Browse

Post YAML

+ Browse

Pre/Post Sync Script

Pre Script

+ Browse

Pre Script Params

Pre Script Params

Post Script

+ Browse

Post Script Params

Post Script Params

Post Sync Validations

☐ No Post Sync Validations

Post sync validation retries

N

Post sync validation retry wait

N_Seconds

Cancel

Add

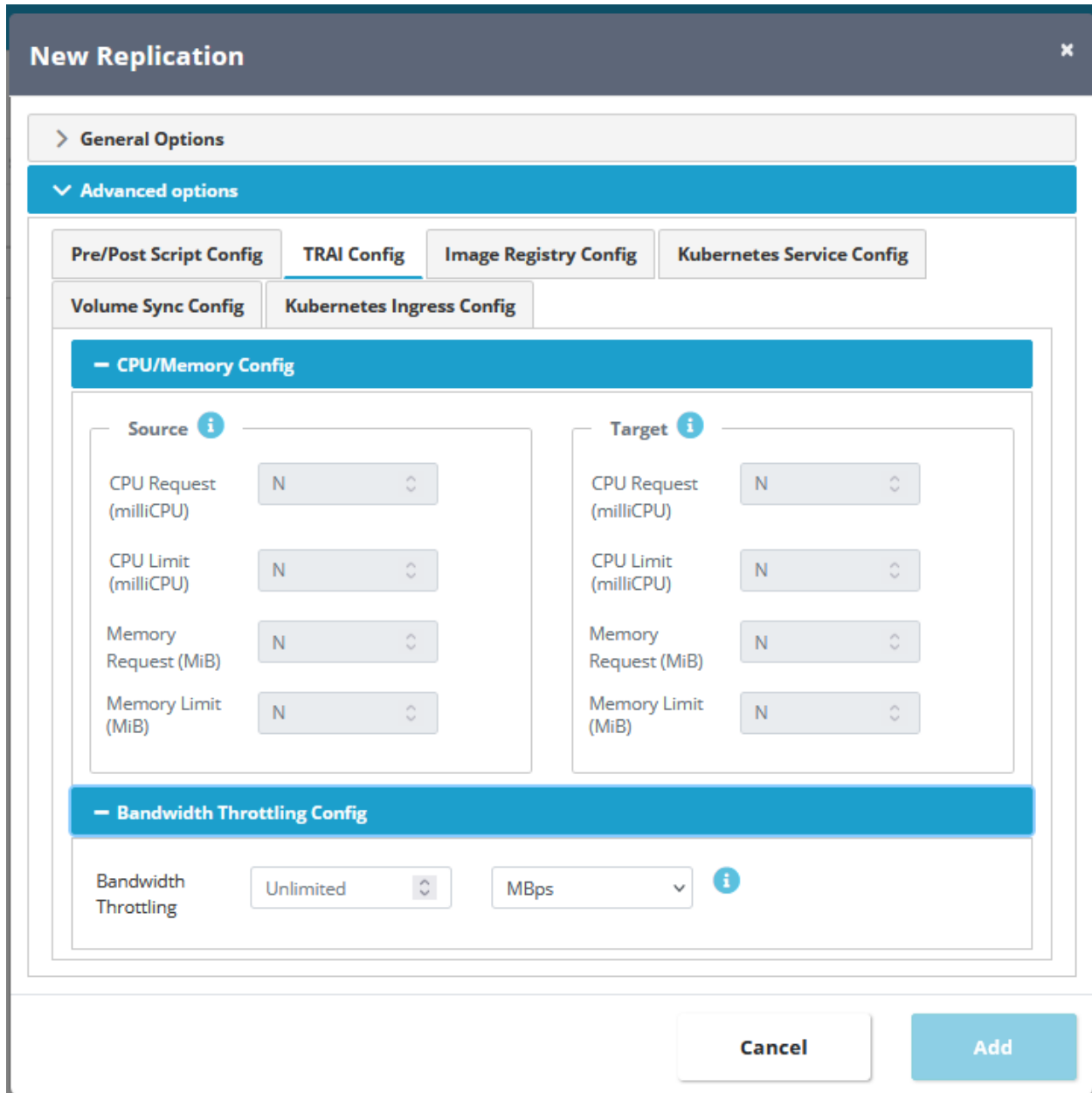
Just like the pre and post-script, you can also configure pre and post YAML to apply (patch) for the cluster on either side. You get four interfaces for the YAML apply:

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.



New Replication

> General Options

▼ Advanced options

Pre/Post Script Config **TRAI Config** Image Registry Config Kubernetes Service Config

Volume Sync Config Kubernetes Ingress Config

— CPU/Memory Config

Source	Target
CPU Request (milliCPU)	CPU Request (milliCPU)
CPU Limit (milliCPU)	CPU Limit (milliCPU)
Memory Request (MiB)	Memory Request (MiB)
Memory Limit (MiB)	Memory Limit (MiB)

— Bandwidth Throttling Config

Bandwidth Throttling: Unlimited MBps

Cancel Add

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.

The sync bandwidth throttling option allows you to set maximum bandwidth that would be used for data replications.

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.

New Replication

General Options

Advanced options

Pre/Post Script Config

TRAJ Config

Image Registry Config

Kubernetes Service Config

Volume Sync Config

Kubernetes Ingress Config

Image Registry Mapping

Target

<src_irc>

<dst_irc>

+

+ Add Image Registry Mapping(s)...

Image PullSecret Mapping

Target

<src_ips>

<dst_ips>

+

+ Add Image PullSecret Mapping(s)...

Cancel

Add

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

Cancel

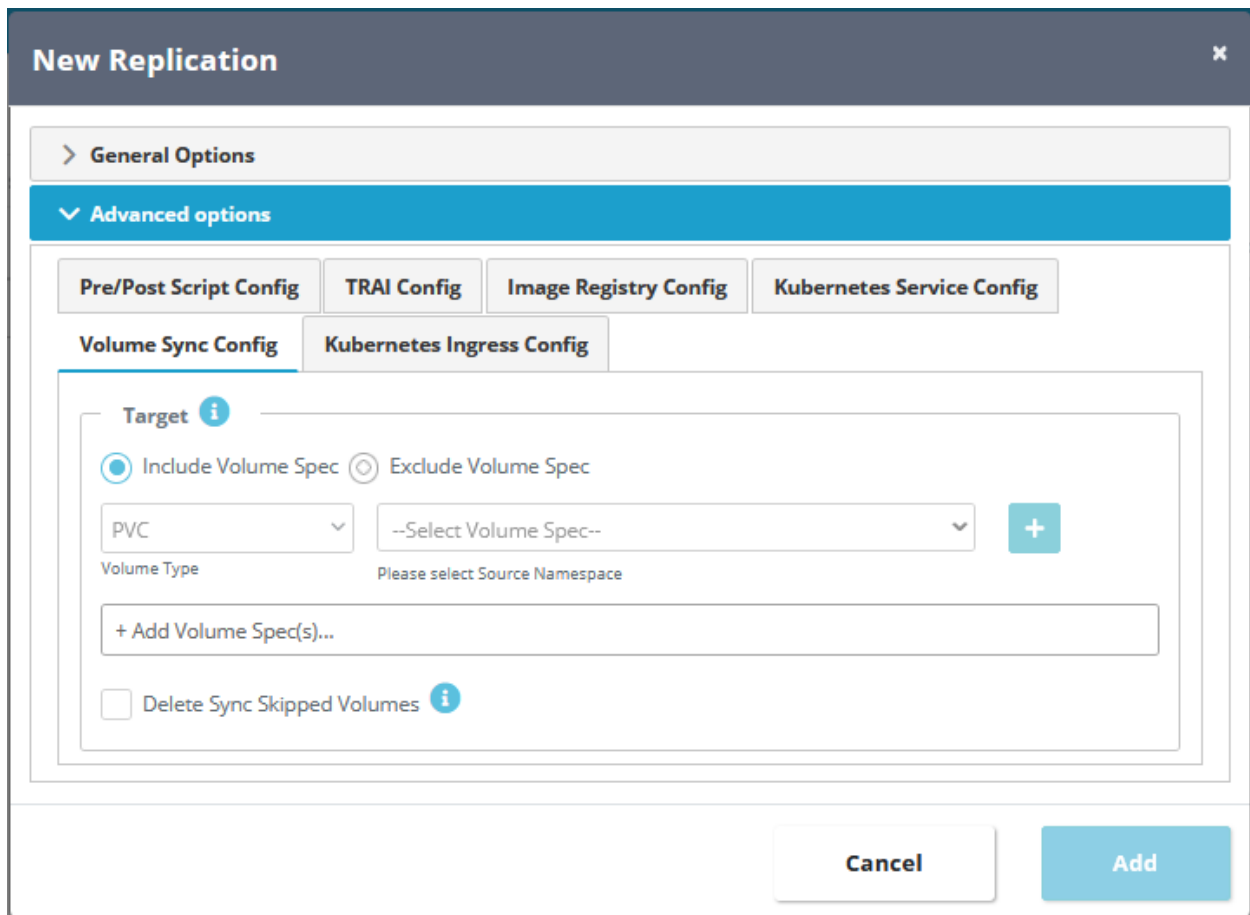
Add

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.



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.

Ingress Config

Options in this section allow you to map ingress class for ingress replication. You can map ingress class directly for each selected ingress from the replicated namespace to specify a new class. Optionally, you can also specify new annotations for each selected ingress from the replicated namespace that indirectly selects ingress class post replication. Annotations can also be used to specify any optional configuration options for the target ingress class and controller that is mapped for each ingress.

If no mapping is done for class, SWIFT will remove ingress class related annotations and class name from each replicated ingress so those use the default class on the target cluster post replication.

New Replication

General Options

Advanced options

Pre/Post Script Config

TRAI Config

Image Registry Config

Kubernetes Service Config

Volume Sync Config

Kubernetes Ingress Config

Ingress Class Mapping

Target

--Select Ingress name--

<dst_ingress_class>

+

Please select Source Namespace

+ Add Ingress Class Mapping(s)...

Ingress Annotation Mapping

Target

--Select Ingressname--

Annotation key

Annotation value

+

Please select Source Namespace

+ Add Ingress Annotation Mapping(s)

Cancel

Add

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.

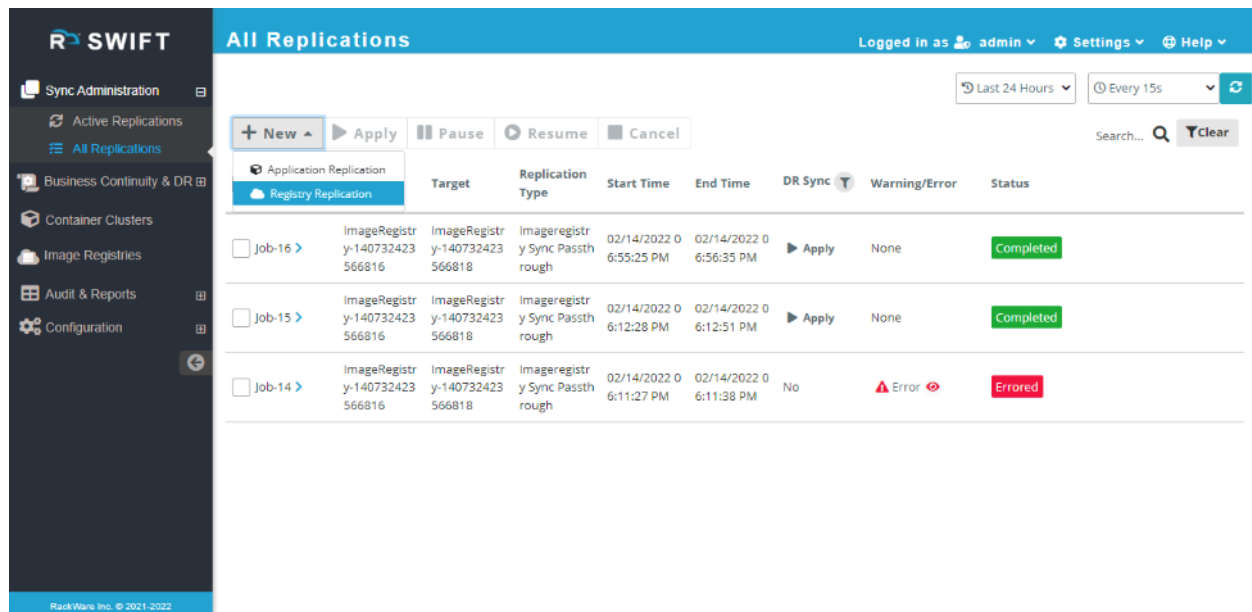
You can also run sync between different namespaces of the same cluster (intra-cluster syncs), and for such cases, you will configure the same cluster as a source and a target of the sync. The SWIFT does not support syncing within the same namespace of the same cluster.

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.



The screenshot shows the 'All Replications' page in the SWIFT dashboard. The page has a sidebar on the left with navigation links: Sync Administration, Active Replications, All Replications (selected), Business Continuity & DR, Container Clusters, Image Registries, Audit & Reports, and Configuration. The main content area has a header with 'All Replications' and user information 'Logged in as admin'. Below the header are filters for 'Last 24 Hours' and 'Every 15s'. A toolbar contains buttons for '+ New', 'Apply', 'Pause', 'Resume', and 'Cancel'. A search bar is also present. The main table lists replication jobs with columns: Job, Target, Replication Type, Start Time, End Time, DR Sync, Warning/Error, and Status.

Job	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.

New Registry Replication ✕

General Options

Source Imageregistry

Type*

Friendly Name*

Repositories* ☒ All ☐ Selective ?

Target Imageregistry

Type*

Friendly Name*

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

Cancel

Add

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*

swiftdauto/nginx

×

Repository map for Replication

Map Repositories

Repositories 1 Selected

--Select Repositories--

Dst Repository Name

+

Repository Mappings

swiftdauto/nginx=swiftdauto/nginx-test

×

Other Options

☐ Verbose Sync

Job NameReplication Job Name

Cancel

Add

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.

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.

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

Cancel

Apply

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

i

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

x

☐ 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

88 | Page

- 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

x

☐ Email alerts on Sync failure only

Advanced options

Cancel

Create

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

i

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

Advanced options

Cancel

Create

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.

90 | Page

New DR Policy

General Options

Advanced options

Pre Sync Script

Pre Script

+ Browse

Pre Script Params

Post Sync Script

Post Script

+ Browse

Post Script Params

Source

Pre YAML

+ Browse

Post YAML

+ Browse

Target

Pre YAML

+ Browse

Post YAML

+ Browse

Cancel

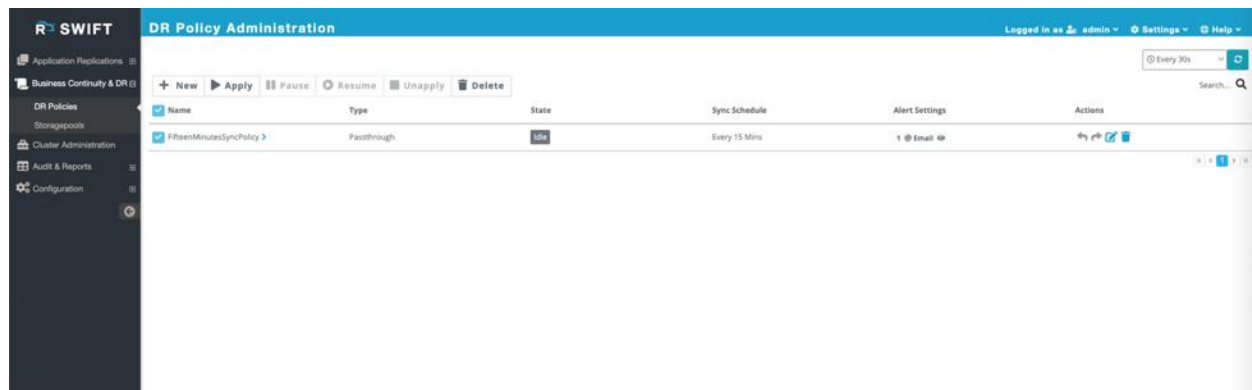
Create

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 fallback 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.



Applying the newly created policy to application syncs

Go to the Business Continuity and Disaster Recovery (BCDR) menu. You will see a 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

i

Start Time

Schedule Execution

☒ Start Immediately
 ☐ Start Later

i

☒ Existing Replication(s)
 ☐ New Replication

Existing Replications

Filter

All

×

 Clear Filters

✎

 Modify Filters

No Replications Found

▼

+

Total: 0 Replications Found

Replications*

+ Add Replication...

Cancel

Apply

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.

93 | Page

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

Filter

All

×

 Clear Filters

✎

 Modify Filters

Select Replication(s) to +Add

▼

+

Total: 10 Replications Found

Replications*

Job-1634 (1634) ✕

Cancel

Apply

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: FifteenMinsPTPolicy

Policy Name*

FifteenMinsPTPolicy

Sync Type*

☒ Passthrough
☐ Stage1
☐ Stage1+2

Start Time

Schedule Execution

☒ Start Immediately
☐ Start Later

☒ New Replication
☐ Existing Replication

General Options

Source

Platform Type

☒ Kubernetes
☐ OpenShift

Cluster Name*

--Select Cluster--

Source Cluster is Required

Namespace*

--Select Namespace--

Applications*

☒ All
☐ Selective

☐ Include K8S Native Objects

Sync Webhooks

☐ All
☐ Native Webhooks

☐ Don't Delete Taints

Target

Platform Type

☒ Kubernetes
☐ OpenShift

Cluster Name*

--Select Cluster--

Target Cluster is Required

Namespace*

--Select Namespace--

Storage Class*

--Select Storageclass--

Custom Resource Configuration

Choose Custom Resources

CRD Scope

☒ Cluster
☐ Namespace

Search by CRD(s) name

☐ CUSTOMRESOURCEDEFINITION

CR/CRD Object List

+ Add CR/CRD Object(s)...

TRAIPOD Options

Source

IP Type

--Select IP Type--

IP Address

☒ Auto Select IP Address
☐ Specific IP Address

SWIFT will auto select one of the reachable IPs.

TRAIP 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

Target

IP Type

--Select IP Type--

IP Address

☒ Auto Select IP Address
☐ Specific IP Address

SWIFT will auto select one of the reachable IPs.

TRAIP 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

☐ Verbose Sync
☐ Dry Run

Advanced options

Cancel

Apply

95 | Page

Apply: ThirtyMinsWeeklyBackupPolicy

Policy Name*

ThirtyMinsWeeklyBackupPolicy

Sync Type*

☐ Passthrough
☐ Stage1
☒ Stage1+2

Start Time

Schedule Execution

☒ Start Immediately
☐ Start Later

General Options

No storagepool found to configure a Stage1+2 replication.

To Add + Storagepool(s), please navigate to the [Storagepool Administration](#) page.

Stage1 Replication*

☒ New Replication
☐ Existing Replication

Platform Type

☒ Kubernetes
☐ OpenShift

Cluster Name*

--Select Cluster--

Source Cluster is Required

Namespace*

--Select Namespace--

Storagepool*

--Select Storagepool--

Imagegroup*

☒ New
☐ Existing

Enter Imagegroup Name

Applications*

☒ All
☐ Selective

☐ Include K8S Native Objects

Sync Webhooks

☐ All
☐ Native Webhooks

☐ Don't Delete Taints

Stage2 Replication*

☒ New Replication
☐ Existing Replication

Platform Type

☒ Kubernetes
☐ OpenShift

Cluster Name*

--Select Cluster--

Target Cluster is Required

Namespace*

--Select Namespace--

Storage Class*

--Select Storageclass--

Custom Resource Configuration

Choose Custom Resources

CRD Scope

☒ Cluster
☐ Namespace

Search by CRD(s) name

CUSTOMRESOURCEDEFINITION

Please Select Source Cluster/Namespac

CR/CRD Object List

+ Add CR/CRD Object(s)...

TRAIPOD Options

Source

IP Type

--Select IP Type--

IP Address

☒ Auto Select IP Address
☐ Specific IP Address

SWIFT will auto select one of the reachable IPs.

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

Target

IP Type

--Select IP Type--

IP Address

☒ Auto Select IP Address
☐ Specific IP Address

SWIFT will auto select one of the reachable IPs.

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

☐ Verbose Sync
☐ Dry Run

Advanced Options for New Stage1 and Stage2 Replications

Cancel

Apply

96 | Page

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

×

 Clear Filters

✓

 Modify Filters

No pts replications found

▼

+

Total: 0 Replications Found

Replications*

+ Add Replication...

Cancel

Apply

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

i

Start Time

Schedule Execution

☒ Start Immediately
 ☐ Start Later

i

☐ 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

i

Target - Image Registry

Type*

--Select--

Friendlyname*

--Select Imageregistry--

Other Options

☐ Verbose Sync

Job Name

Replication Job Name

☐ Dry Run

i

Cancel

Apply

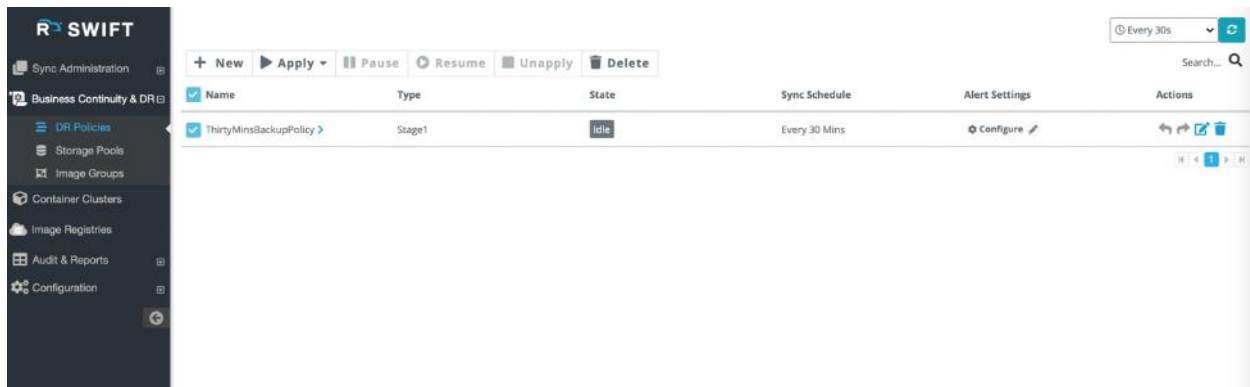
Converting Stage1 Policy to Dynamic-Cluster Provisioning Policy

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.

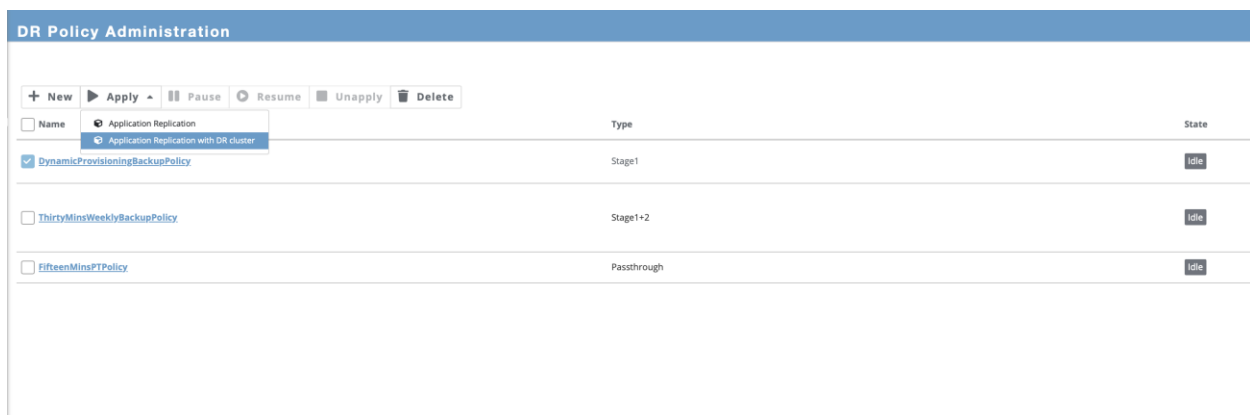
Typically, when you create a Stage1-only policy type, it can only be used to do periodic backups for one or more namespace of the applied source cluster. There is no DR cluster config selected in these policies by default so these policies can't be failed over by default. Only way to restore latest or 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.



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



Select the cloud type for DR cluster provisioning.

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. The below section highlights cloud-specific input parameters for DR cluster. Note that not all parameters are mandatory.

Oracle Cloud (OCI)

- 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)

Amazon Cloud (AWS)

- Access key
- Region
- Availability zones
- Instance type
- Secret key

Google Cloud (GCP)

- Region
- Zone
- Machine type
- Private key

Microsoft Azure Cloud

- Subscription id
- Tenant id
- Client id
- Resource group
- Cloud type (Public/Government/China)
- Region
- Zone
- Pricing Tier (Standard/Free)
- Network config (Kubenet/Azure CNI)
- Node size
- Client secret

IBM Cloud

- Cluster type (VPC/Classic)
- Metro
- Zone
- Resource group
- Flavor
- API Key

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 optionally cleaned up as part of a subsequent fallback for the policy operation.

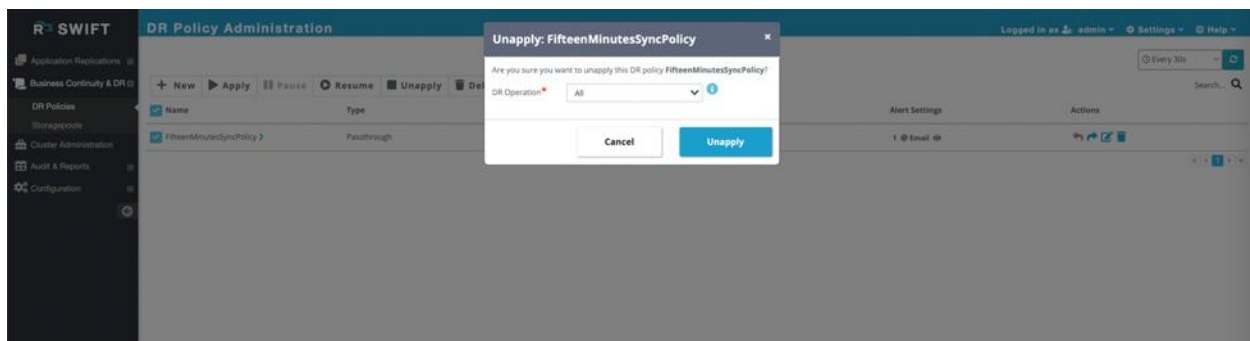
DR Policy Administration

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.



Pause a DR policy

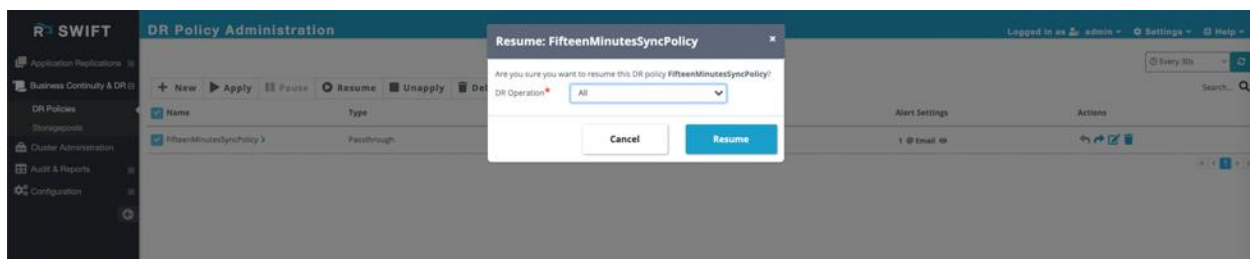
From the BCDR menu, select the policy you want to pause and click on the 'Pause' button. You can select more than one policy to pause in a batch.



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

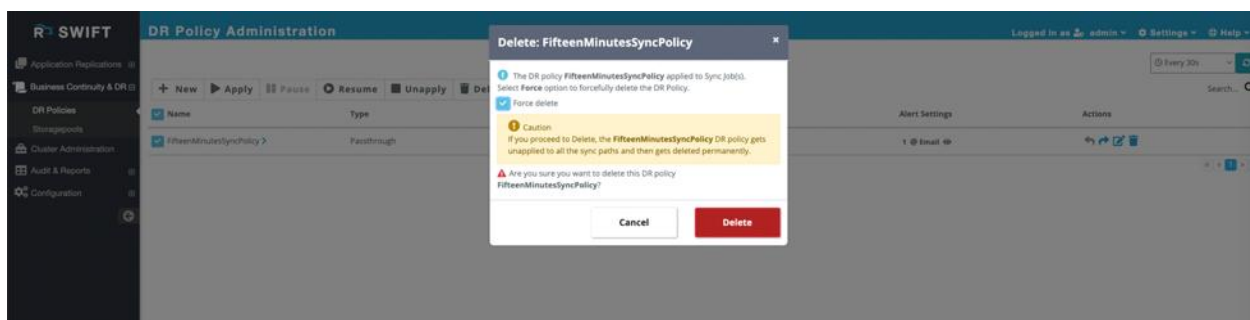
From the BCDR menu, select the policy you want to resume and click on the 'Resume' button. You can select more than one policy to resume in a batch. Note that you can only select policies which are in the PAUSED or PARTIALLY_ACTIVE state.



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.

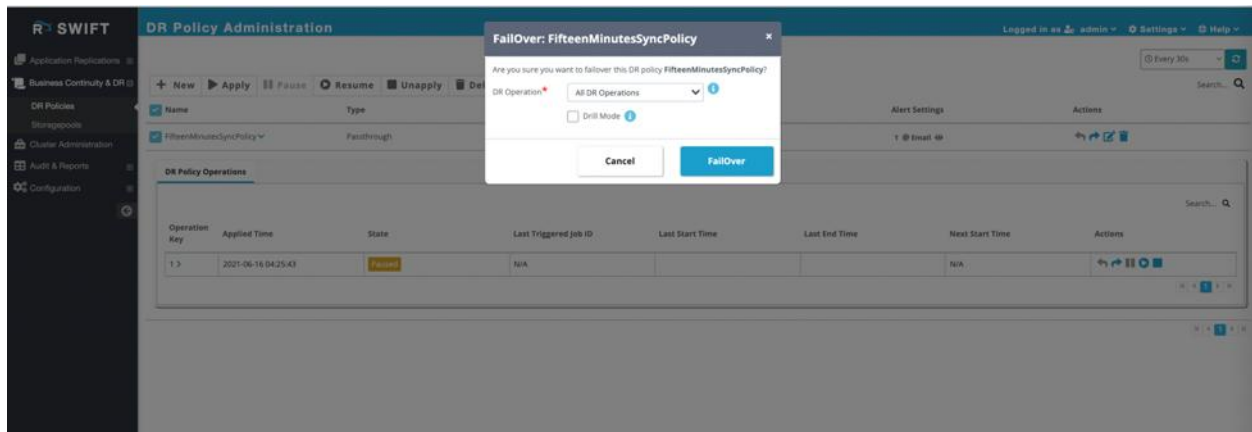


By default, policies which are in the ACTIVE or PARTIALLY_ACTIVE state will not be allowed for deletion. You can only delete the IDLE policies. Selecting the 'Force delete' option will delete any policy. The force deletion will also first internally unapply the policy.

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



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.

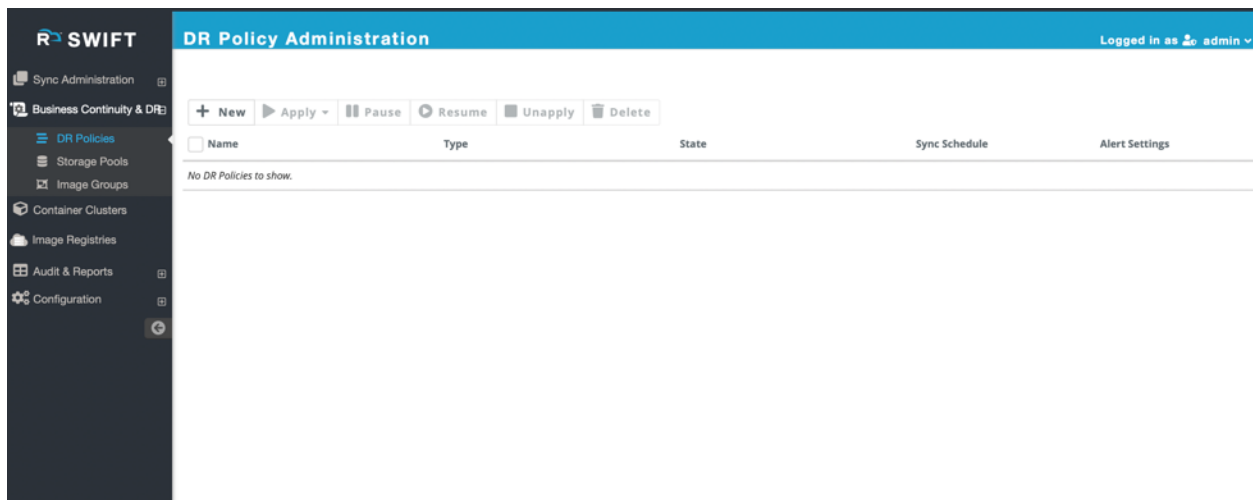
If a policy was failed over without the drill mode, then by default fallback will do a reverse sync (i.e., sync from your DR side to the original production from before the failover), while policies failed over with drill mode by default will not do a reverse sync as part of fallback. You can optionally 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.



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

i

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--

i

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

☐ Email alerts on Sync failure only

Advanced Options

Backup Options

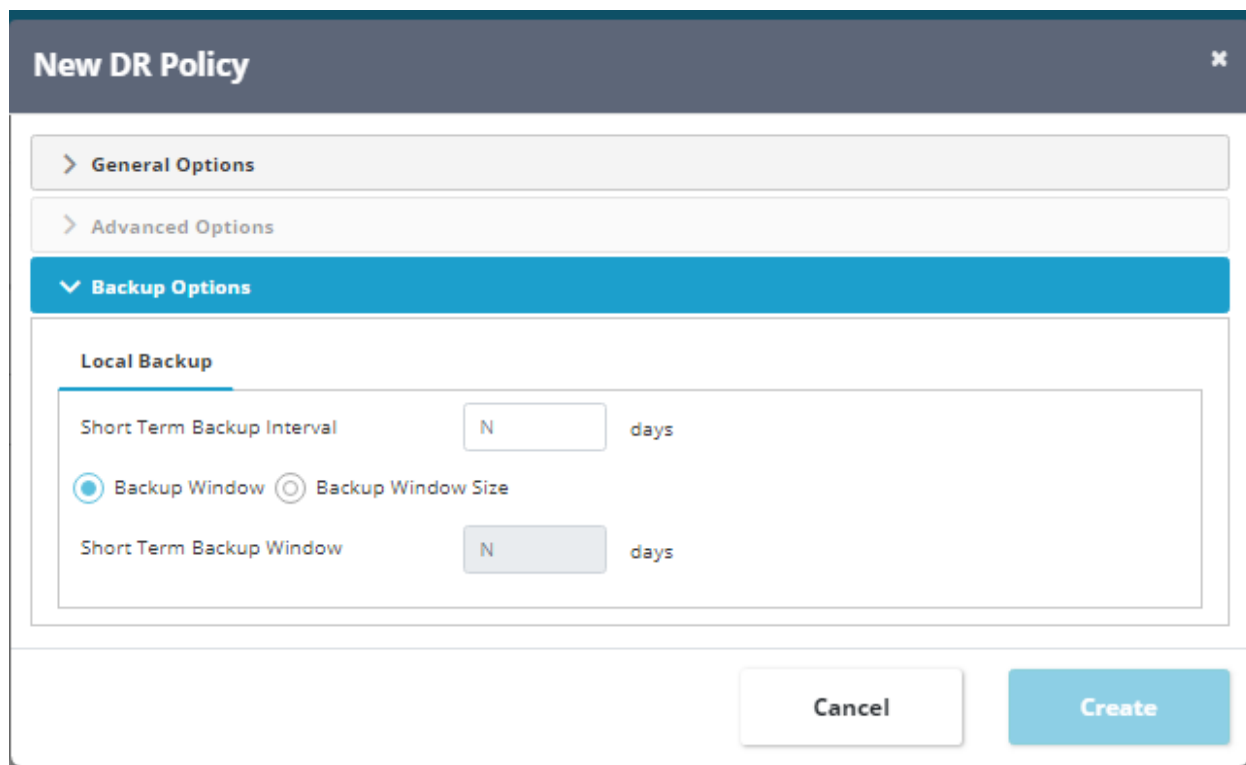
Cancel

Create

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

106 | Page



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.

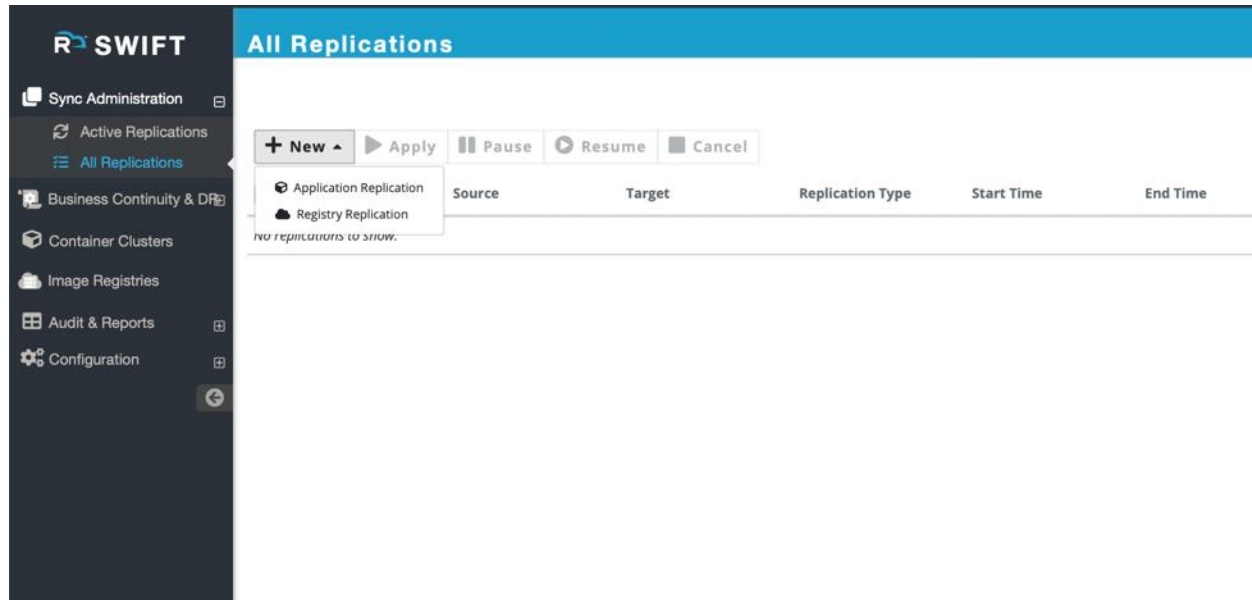
The created local and remote backups can be seen under respective Image-Groups once the new backup DR policy is applied. You can optionally also traverse to those through DR policy config and then Image-Group details.

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.

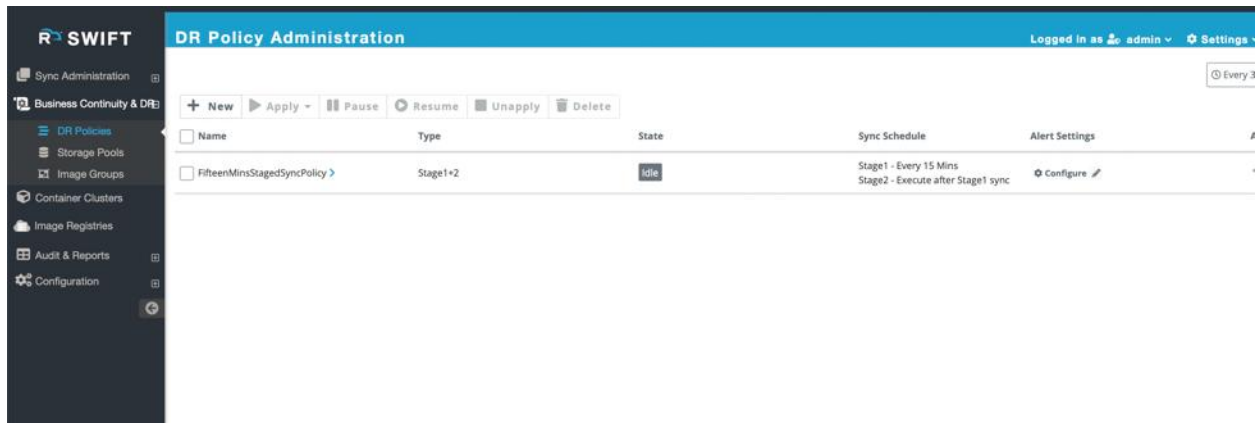


Select the 'Stage-2' sync type on the 'New Replication' dialog and then select the Image-Group that you want to restore. You can select either Kubernetes (K8S) or OpenShift as the target platform.

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.



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.

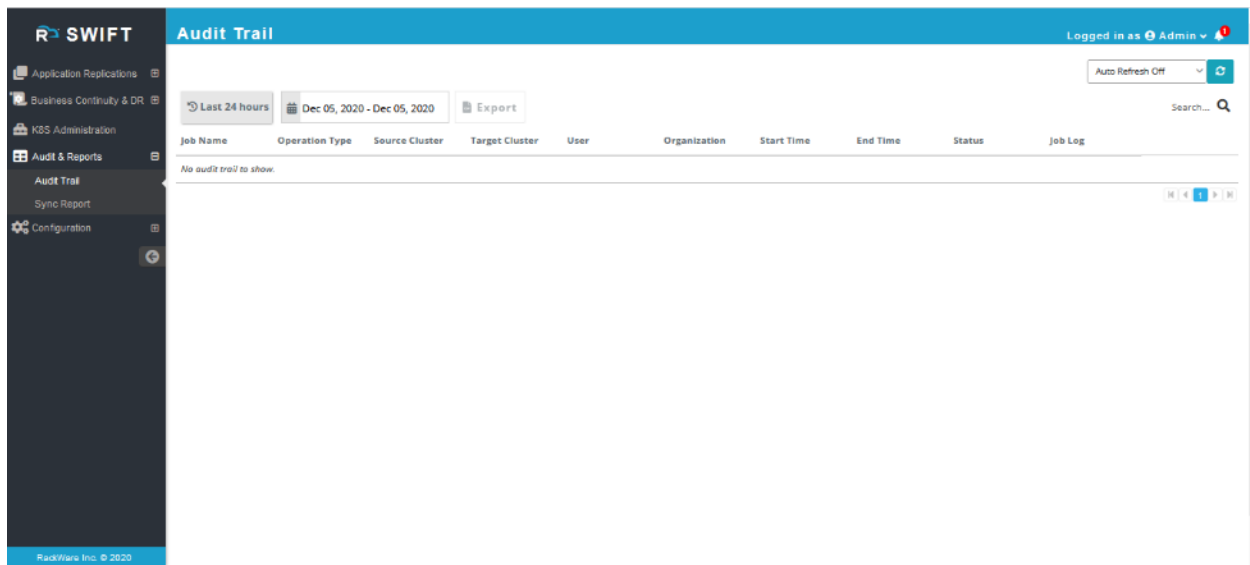
Generating SWIFT operation audit reports

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.

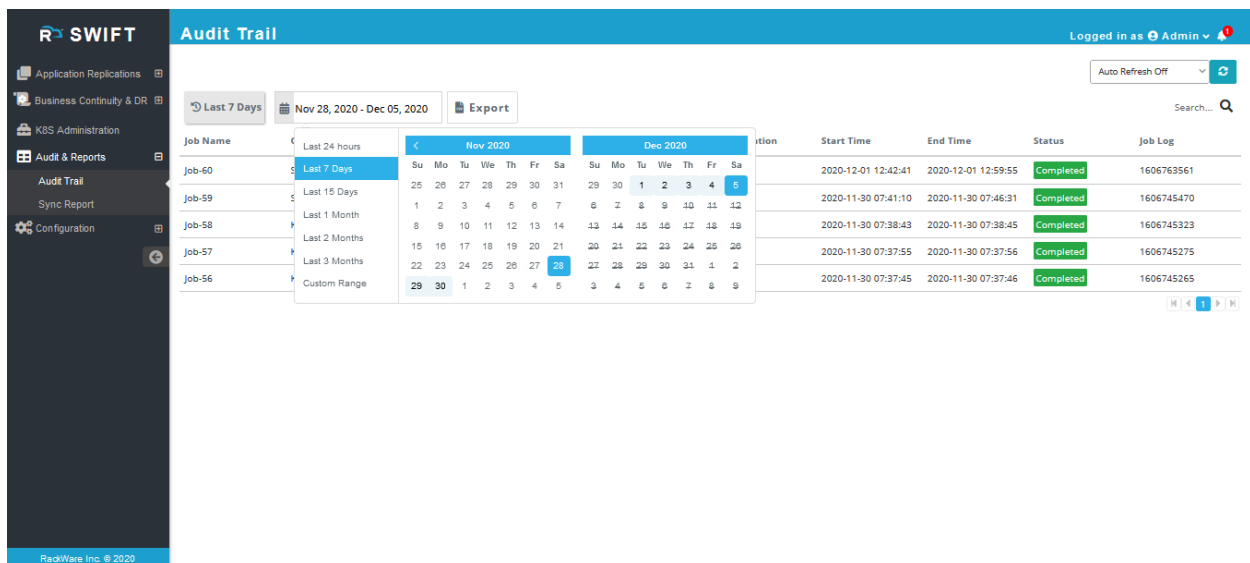
Note that these reports are only available from the SWIFT dashboard or GUI.

Generate all operations audit report

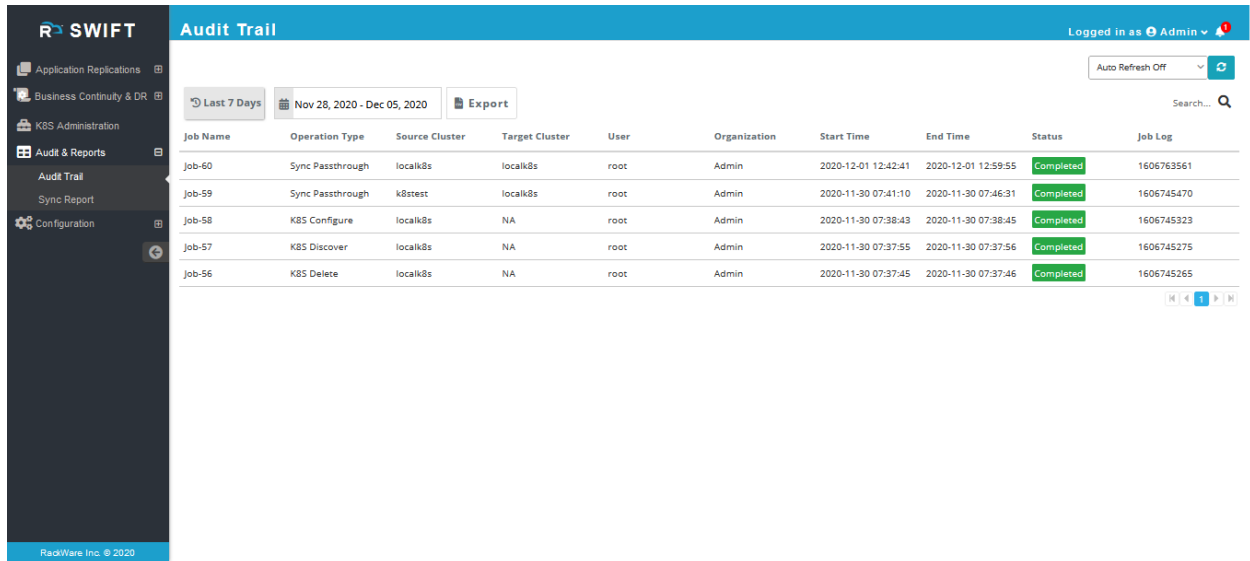
Login to the SWIFT dashboard and navigate to the 'Audit & Reports' menu and 'Audit Trail' submenu.



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.



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



SWIFT Audit Trail

Logged in as Admin

Auto Refresh Off

Last 7 Days | Nov 28, 2020 - Dec 05, 2020 | Export

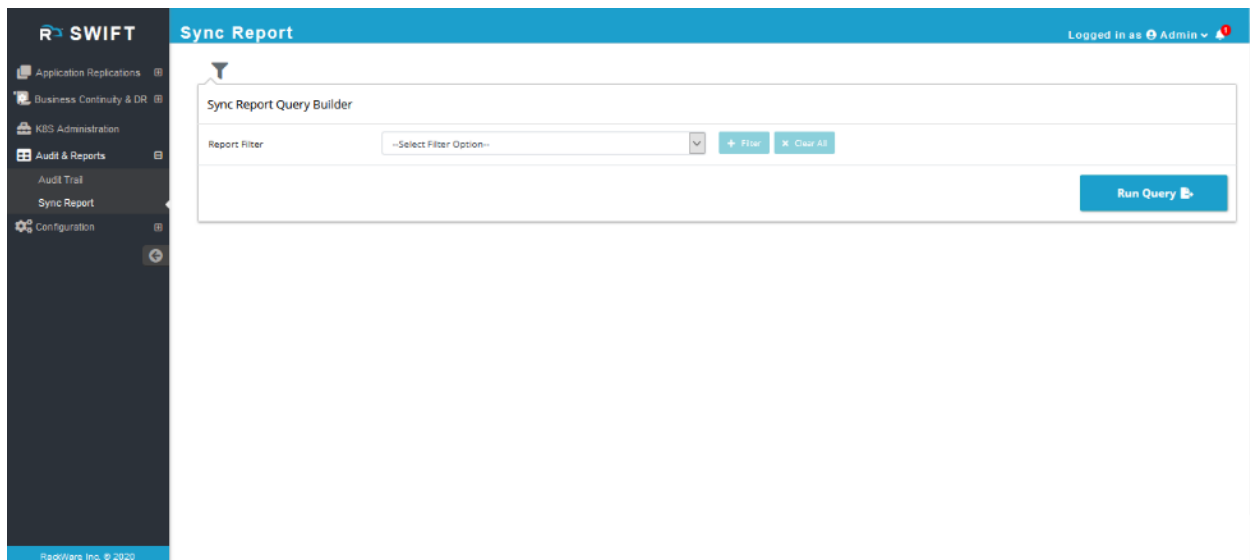
Search...

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

RackWare Inc. © 2020

Generate sync report

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



SWIFT Sync Report

Logged in as Admin

Sync Report Query Builder

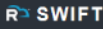
Report Filter: --Select Filter Option--

+ Filter - Clear All

Run Query

RackWare Inc. © 2020

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



- Application Replications
- Business Continuity & DR
- K8S Administration
- Audit & Reports
 - Audit Trail
 - Sync Report
- Configuration

Sync Report

Logged in as Admin

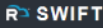
Sync Report Query Builder

Report Filter

Source Cluster
--Select Filter Option--
Source Cluster
Target Cluster
Source Namespace
Target Namespace
Object Type
Organization
User
Period
Job Status

+ Filter
Clear All

Run Query



- Application Replications
- Business Continuity & DR
- K8S Administration
- Audit & Reports
 - Audit Trail
 - Sync Report
- Configuration

Sync Report

Logged in as Admin

dot_friendname: localids
job_status: COMPLETED

Export

Clear Filters
Modify Filters

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	localids	localids	Sync Passthrough	Completed
root	Admin	2020-11-30 07:41:10	2020-11-30 07:46:31	k8snew	localids	Sync Passthrough	Completed
admin	Admin	2020-11-27 05:10:06	2020-11-27 05:20:19	okeclus1	localids	Sync Passthrough	Completed

1

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.

Sync fails with error similar to this:

Sync action failed. [ERROR: Failed to upload the object: <Object Kind> [name : <Object Name>]ERROR: Object create request failed: POST request to the remote host failed [HTTP-Code: 500]: Server returned error response: {"kind":"Status","apiVersion":"v1","metadata":{},"status":"Failure","message":"Internal error occurred: failed calling webhook: failed to call webhook: Post \"https://<Service Name>.<Namespace>

Name>.svc:443/<group>/<version>/<ObjectKind>?timeout=10s\": dial tcp 10.100.73.33:443: connect: connection refused" }

This error arises when the necessary webhook dependencies are not ready. If the sync was initiated using the sync webhook option, verify the sync job for any warnings indicating the unreadiness of the workload object. If present, and if the migrated workload logs show an error indicating absence of a particular service, split the sync into two syncs.

The first sync should target the namespace specified in the error message as the source namespace, using the sync webhook option. The objective is to ensure that all requisite services are available in the namespace required by the webhook dependencies.

Perform the second sync with the original namespace, excluding the sync webhook option. This sync will proceed after the first sync ensures the readiness of the webhook dependencies.