Red Hat OpenShift Service on AWS 4

Setting up clusters and accounts

Getting started with Amazon Managed Red Hat OpenShift 4

Last Updated: 2021-03-24

Red Hat OpenShift Service on AWS 4 Setting up clusters and accounts

Getting started with Amazon Managed Red Hat OpenShift 4

Legal Notice

Abstract

This document provides information on how to get started with Amazon Managed Red Hat OpenShift.

Table of Contents

CHAPTER 1. GETTING STARTED WITH RED HAT OPENSHIFT SERVICE ON AWS

A list of quick start commands for creating a Red Hat OpenShift Service on AWS (ROSA) cluster after the prerequisites have been met.

This chapter walks you through setting up your first Red Hat OpenShift Service on AWS cluster using the **rosa** command-line utility (CLI).

1.1. QUICK START INSTRUCTIONS

If you have already installed the required prerequisites, here are the commands you need to create a cluster.

```
## Configures your AWS account and ensures everything is setup correctly
$ rosa init

## Starts the cluster creation process (~30-40minutes)
$ rosa create cluster --cluster-name <cluster_name>

## Connect your IDP to your cluster
$ rosa create idp --cluster <cluster_name> --interactive

## Promotes a user from your IDP to dedicated-admin level
$ rosa grant user dedicated-admin --user <idp_user_name> --cluster <cluster_name>

## Checks if your install is ready (look for State: Ready),
## and provides your Console URL to login to the web console.
$ rosa describe cluster <cluster_name>
```

CHAPTER 2. PREREQUISITES FOR RED HAT OPENSHIFT SERVICE ON AWS

Red Hat OpenShift Service on AWS (ROSA) provides a model that allows Red Hat to deploy clusters into a customer's existing Amazon Web Service (AWS) account.

2.1. DEPLOYMENT PREREQUISITES

To deploy Red Hat OpenShift Service on AWS (ROSA) into your existing Amazon Web Services (AWS) account, Red Hat requires that several prerequisites are met.

Red Hat recommends the usage of an AWS Organization to manage multiple AWS accounts. The AWS Organization, managed by the customer, hosts multiple AWS accounts. There is a root account in the organization that all accounts will refer to in the account hierarchy.

It is a best practice for the ROSA cluster to be hosted in an AWS account within an AWS Organizational Unit. A Service Control Policy (SCP) is created and applied to the AWS Organizational Unit that manages what services the AWS sub-accounts are permitted to access. The SCP applies only to available permissions within a single AWS account for all AWS sub-accounts within the Organizational Unit. It is also possible to apply a SCP to a single AWS account. All other accounts in the customer's AWS Organization are managed in whatever manner the customer requires. Red Hat Site Reliability Engineers (SRE) will not have any control over SCPs within the AWS Organization.

2.2. CUSTOMER REQUIREMENTS

Red Hat OpenShift Service on AWS (ROSA) clusters must meet several prerequisites before they can be deployed.

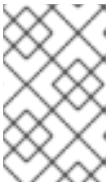
2.2.1. Account

- The customer ensures that the AWS limits are sufficient to support Red Hat OpenShift Service on AWS provisioned within the customer's AWS account.
- The customer's AWS account should be in the customer's AWS Organization with the applicable Service Control Policy (SCP) applied.

NOTE

It is not a requirement that the customer's account be within an AWS Organization or for the SCP to be applied, however Red Hat must be able to perform all the actions listed in the SCP without restriction.

- The customer's AWS account should not be transferable to Red Hat.
- The customer may not impose AWS usage restrictions on Red Hat activities. Imposing restrictions will severely hinder Red Hat's ability to respond to incidents.
- The customer may deploy native AWS services within the same AWS account.



NOTE

Customers are encouraged, but not mandated, to deploy resources in a Virtual Private Cloud (VPC) separate from the VPC hosting Red Hat OpenShift Service on AWS and other Red Hat supported services.

2.2.2. Access requirements

- To appropriately manage the Red Hat OpenShift Service on AWS service, Red Hat must have the **AdministratorAccess** policy applied to the administrator role at all times.

NOTE

This policy only provides Red Hat with permissions and capabilities to change resources in the customer-provided AWS account.

- Red Hat must have AWS console access to the customer-provided AWS account. This access is protected and managed by Red Hat.
- The customer must not utilize the AWS account to elevate their permissions within the Red Hat OpenShift Service on AWS cluster.
- Actions available in the **rosa** CLI utility or OpenShift Cluster Manager (OCM) console must not be directly performed in the customer's AWS account.

2.2.3. Support requirements

- Red Hat recommends that the customer have at least Business Support from AWS.
- Red Hat has authority from the customer to request AWS support on their behalf.
- Red Hat has authority from the customer to request AWS resource limit increases on the customer's account.
- Red Hat manages the restrictions, limitations, expectations, and defaults for all Red Hat OpenShift Service on AWS clusters in the same manner, unless otherwise specified in this requirements section.

2.2.4. Security requirements

- Volume snapshots will remain within the customer's AWS account and customer-specified region.
- Red Hat must have ingress access to EC2 hosts and the API server from allow-listed IP addresses.
- Red Hat must have egress allowed to forward system and audit logs to a Red Hat managed central logging stack.

2.3. REQUIRED CUSTOMER PROCEDURE

Complete these steps before deploying Red Hat OpenShift Service on AWS (ROSA).

Procedure

1. If you, as the customer, are utilizing AWS Organizations, then you must use an AWS account within your organization or create a new one .
2. To ensure that Red Hat can perform necessary actions, you must either create a Service Control Policy (SCP) or ensure that none is applied to the AWS account.
3. Attach the SCP to the AWS account.
4. Follow the ROSA procedures for setting up the environment.

2.3.1. Minimum required Service Control Policy (SCP)

Service Control Policy (SCP) management is the responsibility of the customer. These policies are maintained in the AWS Organization and control what services are available within the attached AWS accounts.

	Service	Actions	Effect
Required	Amazon EC2	All	Allow
	Amazon EC2 Auto Scaling	All	Allow
	Amazon S3	All	Allow
	Identity And Access Management	All	Allow
	Elastic Load Balancing	All	Allow
	Elastic Load Balancing V2	All	Allow
	Amazon CloudWatch	All	Allow
	Amazon CloudWatch Events	All	Allow
	Amazon CloudWatch Logs	All	Allow
	AWS Support	All	Allow
	AWS Key Management Service	All	Allow
	AWS Security Token Service	All	Allow
	AWS Resource Tagging	All	Allow

	Service	Actions	Effect
	AWS Route53 DNS	All	Allow
	AWS Service Quotas	ListServices GetRequestedServiceQuotaChange GetServiceQuota RequestServiceQuotaIncrease ListServiceQuotas	Allow
Optional	AWS Billing	ViewAccount Viewbilling ViewUsage	Allow
	AWS Cost and Usage Report	All	Allow
	AWS Cost Explorer Services	All	Allow

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:*"
```

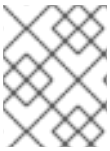
```
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "iam:*"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "elasticloadbalancing:*"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "cloudwatch:*"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "events:*"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "logs:*"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "support:*"
        ],
        "Resource": [
```

```
            "*"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "kms:*"
         ],
         "Resource": [
            "*"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "sts:*"
         ],
         "Resource": [
            "*"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "tag:*"
         ],
         "Resource": [
            "*"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "route53:*"
         ],
         "Resource": [
            "*"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "servicequotas:ListServices",
            "servicequotas:GetRequestedServiceQuotaChange",
            "servicequotas:GetServiceQuota",
            "servicequotas:RequestServiceQuotaIncrease",
            "servicequotas:ListServiceQuotas"
         ],
         "Resource": [
            "*"
         ]
      }
   ]
}
```

2.4. RED HAT MANAGED IAM REFERENCES FOR AWS

Red Hat is responsible for creating and managing the following Amazon Web Services (AWS) resources: IAM policies, IAM users, and IAM roles.

2.4.1. IAM Policies



NOTE

IAM policies are subject to modification as the capabilities of Red Hat OpenShift Service on AWS change.

- The **AdministratorAccess** policy is used by the administration role. This policy provides Red Hat the access necessary to administer the Red Hat OpenShift Service on AWS (ROSA) cluster in the customer's AWS account.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Action": "*",
         "Resource": "*",
         "Effect": "Allow"
      }
   ]
}
```

2.4.2. IAM users

The **osdManagedAdmin** user is created immediately after installing ROSA into the customer's AWS account.

2.5. PROVISIONED AWS INFRASTRUCTURE

This is an overview of the provisioned Amazon Web Services (AWS) components on a deployed Red Hat OpenShift Service on AWS (ROSA) cluster. For a more detailed listing of all provisioned AWS components, see the OpenShift Container Platform documentation.

2.5.1. EC2 instances

AWS EC2 instances are required for deploying the control plane and data plane functions of ROSA in the AWS public cloud.

- Three m5.xlarge minimum (control plane nodes)
- Two m5.xlarge minimum (infrastructure nodes)
- Two m5.xlarge minimum but highly variable (worker nodes)

2.5.2. Elastic Block Storage storage

Amazon EBS block storage is used for both local node storage and persistent volume storage.

Volume requirements for each EC2 instance:

- Control Plane Volume
  - Size: 350GB
  - Type: io1
  - Input/Output Operations Per Second: 1000
- Infrastructure Volume
  - Size: 300GB
  - Type: gp2
  - Input/Output Operations Per Second: 100
- Worker Volume
  - Size: 300GB
  - Type: gp2
  - Input/Output Operations Per Second: 100

2.5.3. Elastic load balancers

Up to two Network Elastic Load Balancers (ELBs) for API and up to two Classic ELBs for application router. For more information, see the ELB documentation for AWS.

2.5.4. S3 storage

The image registry and Elastic Block Store (EBS) volume snapshots are backed by AWS S3 storage. Pruning of resources is performed regularly to optimize S3 usage and cluster performance.

NOTE

Two buckets are required with a typical size of 2TB each.

2.5.5. VPC

Customers should expect to see one VPC per cluster. Additionally, the VPC will need the following configurations:

- **Subnets**: Two subnets for a cluster with a single availability zone, or six subnets for a cluster with multiple availability zones.
- **Router tables**: One router table per private subnet, and one additional table per cluster.
- **Internet gateways**: One Internet Gateway per cluster.
- **NAT gateways**: One NAT Gateway per public subnet.

2.5.6. Security groups

AWS security groups provide security at the protocol and port access level; they are associated with EC2 instances and Elastic Load Balancers. Each security group contains a set of rules that filter traffic coming in and out of an EC2 instance. You must ensure the ports required for the OpenShift installation are open on your network and configured to allow access between hosts.

Group	Type	IP Protocol	Port range
MasterSecurityGroup	**AWS::EC2::SecurityGroup**	**icmp**	**0**
		tcp	**22**
		tcp	**6443**
		tcp	**22623**
WorkerSecurityGroup	**AWS::EC2::SecurityGroup**	**icmp**	**0**
		tcp	**22**
BootstrapSecurityGroup	**AWS::EC2::SecurityGroup**	**tcp**	**22**
		tcp	**19531**

2.6. ADDITIONAL RESOURCES

See Required AWS service quotas for information about required AWS service quotas and how to request increases.

CHAPTER 3. SETTING UP THE ENVIRONMENT

3.1. SETTING UP THE ENVIRONMENT

Complete the following steps to set up your environment before creating your cluster.

Prerequisites

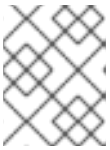
- Review and complete the deployment prerequisites and policies.
- Create a Red Hat account, if you do not already have one. Then, check your email for a verification link. You will need these credentials to install ROSA.

Procedure

1. Select the Amazon Web Services (AWS) account that you want to use.
   It is recommended to use a dedicated AWS account to run production clusters. If you are using AWS Organizations, you can use an AWS account within your organization or create a new one .

   If you are using AWS Organizations and you need to have a Service Control Policy (SCP) applied to the AWS account you plan to use, see the prerequisites for details on the minimum required SCP.

   As part of the cluster creation process, **rosa** creates an **osdCcsAdmin** IAM user. This user uses the IAM credentials you provide when configuring the AWS CLI.

   NOTE

   This user has **Programmatic** access enabled and the **AdministratorAccess** policy attached to it.

2. Enable the ROSA service in the AWS Marketplace.
   a. Sign in to your AWS account.
   b. To enable ROSA, go to the ROSA service and select **Enable**.
3. Install and configure the AWS CLI.
   a. Follow the AWS command-line interface documentation to install and configure the AWS CLI for your operating system.
   Specify the correct **aws_access_key_id** and **aws_secret_access_key** in the **.aws/credentials** file. See AWS Configuration basics in the AWS documentation.
   b. Optional: Set a default AWS region. **rosa** evaluates regions in the following priority order:
      i. The region specified when running a **rosa** command with the **--region** flag.
      ii. The region set in the **AWS_DEFAULT_REGION** environment variable. See Environment variables to configure the AWS CLI in the AWS documentation.
      iii. The default region set in your AWS configuration file. See Quick configuration with aws configure in the AWS documentation.

c. Optional: Configure your AWS CLI settings and credentials by using an AWS named profile. **rosa** evaluates AWS named profiles in the following priority order:

   i. The profile specified when running a **rosa** command with the **--profile** flag.

   ii. The profile set in the **AWS_PROFILE** environment variable. See Named profiles in the AWS documentation.

d. Verify the AWS CLI is installed and configured correctly by running the following command to query the AWS API:

```
$ aws sts get-caller-identity
```

Example output

```
-----------------------------------------------------------------------------
|                            GetCallerIdentity                              |
+---------------------------------------------------------------------------+
|+----------------------------------+----------------------+----------------+|
||      Account       |              Arn              |      UserID       ||
|+----------------------------------+----------------------+----------------+|
||  <account_name>    | arn:aws:iam<string>:user:name |     <userID>      ||
|+----------------------------------+----------------------+----------------+|
```

4. Install **rosa**, the Red Hat OpenShift Service on AWS command-line interface (CLI).

a. Download the latest release of the **rosa** CLI for your operating system.

b. Optional: Rename the executable file you downloaded to **rosa**. This documentation uses **rosa** to refer to the executable file.

c. Optional: Add **rosa** to your path.

d. Enter the following command to verify your installation:

```
$ rosa
```

Example output

```
Command line tool for ROSA.

Usage:
  rosa [command]

Available Commands:
  completion  Generates bash completion scripts
  create      Create a resource from stdin
  delete      Delete a specific resource
  describe    Show details of a specific resource
  edit        Edit a specific resource
  help        Help about any command
  init        Applies templates to support Managed OpenShift on AWS clusters
  list        List all resources of a specific type
  login       Log in to your Red Hat account
  logout      Log out
```

```
  logs        Show logs of a specific resource
  verify      Verify resources are configured correctly for cluster install
  version     Prints the version of the tool

Flags:
      --debug     Enable debug mode.
  -h, --help      help for rosa
  -v, --v Level   log level for V logs

Use "rosa [command] --help" for more information about a command.
```

e. Optional: You can run the **rosa completion** command to generate a bash completion file.

```
$ rosa completion > /etc/bash_completion.d/rosa
```

Add this file to the correct location for your operating system. For example, on a Linux machine, run the following command to enable **rosa** bash completion:

```
$ source /etc/bash_completion.d/rosa
```

5. Enter the following command to verify that your AWS account has the necessary permissions:

```
$ rosa verify permissions
```

Example output

```
I: Validating SCP policies...
I: AWS SCP policies ok
```

6. Verify that your AWS account has the necessary quota to deploy an Red Hat OpenShift Service on AWS cluster.

```
$ rosa verify quota --region=us-west-2
```

Example output

```
I: Validating AWS quota...
I: AWS quota ok
```

NOTE

Sometimes your AWS quota varies by region. If you receive any errors, try a different region.

If you need to increase your quota, go to your AWS console, and request a quota increase for the service that failed.

After both the permissions and quota checks pass, proceed to the next step.

7. Prepare your AWS account for cluster deployment:

a. Enter the following command to log in to your Red Hat account with **rosa**.

```
$ rosa login
```

Replace **<my_offline_access_token>** with your token.

Example output

```
To login to your Red Hat account, get an offline access token at
https://cloud.redhat.com/openshift/token/rosa
? Copy the token and paste it here: <my-offline-access-token>
```

Example output continued

```
I: Logged in as 'rh-rosa-user' on 'https://api.openshift.com'
```

b. Run the following command to verify your Red Hat and AWS credentials are setup correctly. Check that your AWS Account ID, Default Region and ARN match what you expect. You can safely ignore the rows beginning with OCM for now (OCM stands for OpenShift Cluster Manager).

```
$ rosa whoami
```

Example output

```
AWS Account ID:               000000000000
AWS Default Region:           us-east-2
AWS ARN:                      arn:aws:iam::000000000000:user/hello
OCM API:                      https://api.openshift.com
OCM Account ID:               1DzGIdIhqEWyt8UUXQhSoWaaaaa
OCM Account Name:             Your Name
OCM Account Username:         you@domain.com
OCM Account Email:            you@domain.com
OCM Organization ID:          1HopHfA2hcmhup5gCr2uH5aaaaa
OCM Organization Name:        Red Hat
OCM Organization External ID: 0000000
```

c. Initialize your AWS account. This step runs a CloudFormation template that prepares your AWS account for cluster deployment and management. This step typically takes 1-2 minutes to complete.

```
$ rosa init
```

Example output

```
I: Logged in as 'rh-rosa-user' on 'https://api.openshift.com'
I: Validating AWS credentials...
I: AWS credentials are valid!
I: Validating SCP policies...
I: AWS SCP policies ok
I: Validating AWS quota...
I: AWS quota ok
I: Ensuring cluster administrator user 'osdCcsAdmin'...
I: Admin user 'osdCcsAdmin' created successfully!
```

```
I: Verifying whether OpenShift command-line tool is available...
E: OpenShift command-line tool is not installed.
Run 'rosa download oc' to download the latest version, then add it to your PATH.
```

8. Install the OpenShift CLI (**oc**) from the **rosa** CLI.

   a. Enter this command to download the latest version of the **oc** CLI:

   ```
   $ rosa download oc
   ```

   b. After downloading the **oc** CLI, unzip it and add it to your path.

   c. Enter this command to verify that the **oc** CLI is installed correctly:

   ```
   $ rosa verify oc
   ```

After completing these prerequisite steps, you are ready to create a Red Hat OpenShift Service on AWS cluster.

3.2. NEXT STEPS

- Create a cluster.

3.3. ADDITIONAL RESOURCES

- See Prerequisites.
- See Required AWS service quotas for information about required AWS service quotas and how to request increases.

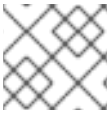
CHAPTER 4. CREATING A ROSA CLUSTER

4.1. CREATING YOUR CLUSTER

You can create an Red Hat OpenShift Service on AWS cluster using **rosa**.

Prerequisites

You have completed the installation prerequisites.



NOTE

AWS Shared VPCs are not currently supported for ROSA installs.

Procedure

1. Enter the following command to create your cluster with the default cluster settings.

```
$ rosa create cluster --cluster-name=rh-rosa-test-cluster1
```

NOTE

Multiple availability zones (AZ) are recommended for production workloads. The default is a single availability zone. Use **--help** for an example of how to set this option manually or use interactive mode to be prompted for this setting.

To view other options when creating a cluster, enter **rosa create cluster --help**.

To follow a set of interactive prompts, enter **rosa create cluster --interactive**.

Example output

```
I: Creating cluster with identifier '1de87g7c30g75qechgh7l5b2bha6r04e' and name 'rh-rosa-test-cluster1'
I: To view list of clusters and their status, run `rosa list clusters`
I: Cluster 'rh-rosa-test-cluster1' has been created.
I: Once the cluster is 'Ready' you will need to add an Identity Provider and define the list of cluster administrators. See `rosa create idp --help` and `rosa create user --help` for more information.
I: To determine when your cluster is Ready, run `rosa describe cluster rh-rosa-test-cluster1`.
```

NOTE

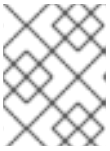
Creating a cluster can take up to 40 minutes.

2. Enter the following command to check the status of your cluster. During cluster creation the **State** field from the output will transition from **pending** to **installing**, and finally to **ready**.

```
$ rosa describe cluster rh-rosa-test-cluster1
```

Example output

```
Name: rh-rosa-test-cluster1
OpenShift Version: 4.6.8
DNS: *.example.com
ID: uniqueidnumber
External ID: uniqueexternalidnumber
AWS Account: 123456789101
API URL: https://api.rh-rosa-test-cluster1.example.org:6443
Console URL: https://console-openshift-console.apps.rh-rosa-test-cluster1.example.or
Nodes: Master: 3, Infra: 2, Compute: 2
Region: us-west-2
Multi-AZ: false
State: ready
Channel Group: stable
Private: No
Created: Jan 15 2021 16:30:55 UTC
Details Page: https://cloud.redhat.com/examplename/details/idnumber
```



NOTE

If installation fails or the **State** field does not change to **ready** after 40 minutes, check the installation troubleshooting documentation for more details.

3. Enter the following command to follow the OpenShift installer logs to track the progress of your cluster:

```
$ rosa logs install rh-rosa-test-cluster1 --watch
```

CHAPTER 5. CONFIGURING IDENTITY PROVIDERS

After your Red Hat OpenShift Service on AWS (ROSA) cluster is created, you must configure identity providers to determine how users log in to access the cluster.

5.1. UNDERSTANDING IDENTITY PROVIDERS

Red Hat OpenShift Service on AWS includes a built-in OAuth server. Developers and administrators obtain OAuth access tokens to authenticate themselves to the API. As an administrator, you can configure OAuth to specify an identity provider after you install your cluster. Configuring identity providers allows users to log in and access the cluster.

5.1.1. Supported identity providers

You can configure the following types of identity providers:

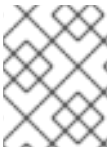
Identity provider	Description
GitHub or GitHub Enterprise	Configure a **github** identity provider to validate user names and passwords against GitHub or GitHub Enterprise's OAuth authentication server.
GitLab	Configure a **gitlab** identity provider to use GitLab.com or any other GitLab instance as an identity provider.
Google	Configure a **google** identity provider using Google's OpenID Connect integration.
LDAP	Configure the **ldap** identity provider to validate user names and passwords against an LDAPv3 server, using simple bind authentication.
OpenID Connect	Configure an **oidc** identity provider to integrate with an OpenID Connect identity provider using an Authorization Code Flow.

5.1.2. Identity provider parameters

The following parameters are common to all identity providers:

Parameter	Description
name	The provider name is prefixed to provider user names to form an identity name.

Parameter	Description
mappingMethod	Defines how new identities are mapped to users when they log in. Enter one of the following values: **claim** The default value. Provisions a user with the identity's preferred user name. Fails if a user with that user name is already mapped to another identity. **lookup** Looks up an existing identity, user identity mapping, and user, but does not automatically provision users or identities. This allows cluster administrators to set up identities and users manually, or using an external process. Using this method requires you to manually provision users. **generate** Provisions a user with the identity's preferred user name. If a user with the preferred user name is already mapped to an existing identity, a unique user name is generated. For example, **myuser2**. This method should not be used in combination with external processes that require exact matches between Red Hat OpenShift Service on AWS user names and identity provider user names, such as LDAP group sync. **add** Provisions a user with the identity's preferred user name. If a user with that user name already exists, the identity is mapped to the existing user, adding to any existing identity mappings for the user. Required when multiple identity providers are configured that identify the same set of users and map to the same user names.



NOTE

When adding or changing identity providers, you can map identities from the new provider to existing users by setting the **mappingMethod** parameter to **add**.

5.2. CONFIGURING A GITHUB IDENTITY PROVIDER

Configure a GitHub identity provider to validate user names and passwords against GitHub or GitHub Enterprise's OAuth authentication server and access your Red Hat OpenShift Service on AWS cluster. OAuth facilitates a token exchange flow between Red Hat OpenShift Service on AWS and GitHub or GitHub Enterprise.

WARNING

Configuring GitHub authentication allows users to log in to Red Hat OpenShift Service on AWS with their GitHub credentials. To prevent anyone with any GitHub user ID from logging in to your Red Hat OpenShift Service on AWS cluster, you must restrict access to only those in specific GitHub organizations or teams.

Prerequisites

- The OAuth application must be created directly within the GitHub organization settings by the GitHub organization administrator.
- GitHub organizations or teams are set up in your GitHub account.

Procedure

1. Navigate to the **Clusters** page and select the cluster you need to configure identity providers for.
2. Click the **Access control** tab.
3. Click **Add identity provider**.

NOTE

You can also click the **Add Oauth configuration** link in the warning message displayed after cluster creation to configure your identity providers.

4. Select **GitHub** from the drop-down menu.
5. Enter a unique name for the identity provider. This name cannot be changed later.
   - An **OAuth callback URL** is automatically generated in the provided field. You will use this to register the GitHub application.

     ```
     https://oauth-openshift.apps.<cluster-name>.<cluster-domain>/oauth2callback/<idp-provider-name>
     ```

     For example:

     ```
     https://oauth-openshift.apps.example-openshift-cluster.com/oauth2callback/github/
     ```

6. Register an application on GitHub.
7. Return to Red Hat OpenShift Service on AWS and select a mapping method from the drop-down menu. **Claim** is recommended in most cases.
8. Enter the **Client ID** and **Client secret** provided by GitHub.
9. Enter a **hostname**. A hostname must be entered when using a hosted instance of GitHub Enterprise.
10. Optional: You can use a certificate authority (CA) file to validate server certificates for the configured GitHub Enterprise URL. Click **Browse** to locate and attach a **CA file** to the identity provider.
11. Select **Use organizations** or **Use teams** to restrict access to a particular GitHub organization or a GitHub team.
12. Enter the name of the organization or team you would like to restrict access to. Click **Add more** to specify multiple organizations or teams that users can be a member of.
13. Click **Confirm**.

Verification

The configured identity provider is now visible on the **Access control** tab of the **Clusters** page.

5.3. CONFIGURING A GITLAB IDENTITY PROVIDER

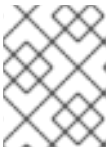
Configure a GitLab identity provider to use GitLab.com or any other GitLab instance as an identity provider.

Prerequisite

- If you use GitLab version 7.7.0 to 11.0, you connect using the OAuth integration. If you use GitLab version 11.1 or later, you can use OpenID Connect (OIDC) to connect instead of OAuth.

Procedure

1. Navigate to the **Clusters** page and select the cluster you need to configure identity providers for.
2. Click the **Access control** tab.
3. Click **Add identity provider**.



NOTE

You can also click the **Add Oauth configuration** link in the warning message displayed after cluster creation to configure your identity providers.

4. Select **GitLab** from the drop-down menu.
5. Enter a unique name for the identity provider. This name cannot be changed later.
   - An **OAuth callback URL** is automatically generated in the provided field. You will provide this URL to GitLab.

     ```
     https://oauth-openshift.apps.<cluster-name>.<cluster-domain>/oauth2callback/<idp-provider-name>
     ```

     For example:

     ```
     https://oauth-openshift.apps.example-openshift-cluster.com/oauth2callback/gitlab/
     ```

6. Add a new application in GitLab .
7. Return to Red Hat OpenShift Service on AWS and select a mapping method from the drop-down menu. **Claim** is recommended in most cases.
8. Enter the **Client ID** and **Client secret** provided by GitLab.
9. Enter the **URL** of your GitLab provider.
10. Optional: You can use a certificate authority (CA) file to validate server certificates for the configured GitLab URL. Click **Browse** to locate and attach a **CA file** to the identity provider.
11. Click **Confirm**.

Verification

The configured identity provider is now visible on the **Access control** tab of the **Clusters** page.

5.4. CONFIGURING A GOOGLE IDENTITY PROVIDER

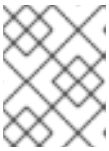
Configure a Google identity provider to allow users to authenticate with their Google credentials.

WARNING

Using Google as an identity provider allows any Google user to authenticate to your server. You can limit authentication to members of a specific hosted domain with the **hostedDomain** configuration attribute.

Procedure

1. Navigate to the **Clusters** page and select the cluster you need to configure identity providers for.
2. Click the **Access control** tab.
3. Click **Add identity provider**.



NOTE

You can also click the **Add Oauth configuration** link in the warning message displayed after cluster creation to configure your identity providers.

4. Select **Google** from the drop-down menu.
5. Enter a unique name for the identity provider. This name cannot be changed later.
   - An **OAuth callback URL** is automatically generated in the provided field. You will provide this URL to Google.

     ```
     https://oauth-openshift.apps.<cluster-name>.<cluster-domain>/oauth2callback/<idp-provider-name>
     ```

     For example:

     ```
     https://oauth-openshift.apps.example-openshift-cluster.com/oauth2callback/github/
     ```

6. Configure a Google identity provider using Google's OpenID Connect integration.
7. Return to Red Hat OpenShift Service on AWS and select a mapping method from the drop-down menu. **Claim** is recommended in most cases.
8. Enter the **Client ID** of a registered Google project and the **Client secret** issued by Google.
9. Enter a hosted domain to restrict users to a Google Apps domain.

10. Click **Confirm**.

Verification

The configured identity provider is now visible on the **Access control** tab of the **Clusters** page.

5.5. CONFIGURING A LDAP IDENTITY PROVIDER

Configure the LDAP identity provider to validate user names and passwords against an LDAPv3 server, using simple bind authentication.

Prerequisite

- When configuring a LDAP identity provider, you will need to enter a configured **LDAP URL**. The configured URL is an RFC 2255 URL, which specifies the LDAP host and search parameters to use. The syntax of the URL is:

```
ldap://host:port/basedn?attribute?scope?filter
```

URL component	Description
ldap	For regular LDAP, use the string **ldap**. For secure LDAP (LDAPS), use **ldaps** instead.
host:port	The name and port of the LDAP server. Defaults to **localhost:389** for ldap and **localhost:636** for LDAPS.
basedn	The DN of the branch of the directory where all searches should start from. At the very least, this must be the top of your directory tree, but it could also specify a subtree in the directory.
attribute	The attribute to search for. Although RFC 2255 allows a comma-separated list of attributes, only the first attribute will be used, no matter how many are provided. If no attributes are provided, the default is to use **uid**. It is recommended to choose an attribute that will be unique across all entries in the subtree you will be using.
scope	The scope of the search. Can be either **one** or **sub**. If the scope is not provided, the default is to use a scope of **sub**.
filter	A valid LDAP search filter. If not provided, defaults to **(objectClass=*)**

When doing searches, the attribute, filter, and provided user name are combined to create a search filter that looks like:

```
(&(<filter>)(<attribute>=<username>))
```

IMPORTANT

If the LDAP directory requires authentication to search, specify a **bindDN** and **bindPassword** to use to perform the entry search.

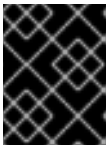
Procedure

1. Navigate to the **Clusters** page and select the cluster you need to configure identity providers for.
2. Click the **Access control** tab.
3. Click **Add identity provider**.

NOTE

You can also click the **Add Oauth configuration** link in the warning message displayed after cluster creation to configure your identity providers.

4. Select **LDAP** from the drop-down menu.
5. Enter a unique name for the identity provider. This name cannot be changed later.
6. Select a mapping method from the drop-down menu. **Claim** is recommended in most cases.
7. Enter a **LDAP URL** to specify the LDAP search parameters to use.
8. Optional: Enter a **Bind DN** and **Bind password**.
9. Enter the attributes that will map LDAP attributes to identities.
    - Enter an **ID** attribute whose value should be used as the user ID. Click **Add more** to add multiple ID attributes.
    - Optional: Enter a **Preferred username** attribute whose value should be used as the display name. Click **Add more** to add multiple preferred username attributes.
    - Optional: Enter an **Email** attribute whose value should be used as the email address. Click **Add more** to add multiple email attributes.
10. Optional: Click **Show advanced Options** to add a certificate authority (CA) file to your LDAP identity provider to validate server certificates for the configured URL. Click **Browse** to locate and attach a **CA file** to the identity provider.
11. Optional: Under the advanced options, you can choose to make the LDAP provider **Insecure**. If you select this option, a CA file cannot be used.



IMPORTANT

If you are using an insecure LDAP connection (ldap:// or port 389), then you must check the **Insecure** option in the configuration wizard.

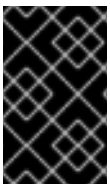
12. Click **Confirm**.

Verification

The configured identity provider is now visible on the **Access control** tab of the **Clusters** page.

5.6. CONFIGURING AN OPENID IDENTITY PROVIDER

Configure an OpenID identity provider to integrate with an OpenID Connect identity provider using an Authorization Code Flow.



IMPORTANT

The Authentication Operator in Red Hat OpenShift Service on AWS requires that the configured OpenID Connect identity provider implements the OpenID Connect Discovery specification.

Claims are read from the JWT **id_token** returned from the OpenID identity provider and, if specified, from the JSON returned by the Issuer URL.

At least one claim must be configured to use as the user's identity.

You can also indicate which claims to use as the user's preferred user name, display name, and email address. If multiple claims are specified, the first one with a non-empty value is used. The standard claims are:

preferred_username	The preferred user name when provisioning a user. A shorthand name that the user wants to be referred to as, such as **janedoe**. Typically a value that corresponding to the user's login or username in the authentication system, such as username or email.
email	Email address.
name	Display name.

See the OpenID claims documentation for more information.

Prerequisite

- Before you configure OpenID Connect, check the installation prerequisites for any Red Hat product or service you want to use with your Red Hat OpenShift Service on AWS cluster.

Procedure

1. Navigate to the **Clusters** page and select the cluster you need to configure identity providers for.
2. Click the **Access control** tab.
3. Click **Add identity provider**.

NOTE

You can also click the **Add Oauth configuration** link in the warning message displayed after cluster creation to configure your identity providers.

4. Select **OpenID** from the drop-down menu.
5. Enter a unique name for the identity provider. This name cannot be changed later.
   - An **OAuth callback URL** is automatically generated in the provided field.
     -

> https://oauth-openshift.apps.<cluster-name>.<cluster-domain>/oauth2callback/<idp-provider-name>

For example:

> https://oauth-openshift.apps.example-openshift-cluster.com/oauth2callback/openid/

6. Create an authorization request using an Authorization Code Flow .
7. Return to Red Hat OpenShift Service on AWS and select a mapping method from the drop-down menu. **Claim** is recommended in most cases.
8. Enter a **Client ID** and **Client secret** provided from OpenID.
9. Enter an **Issuer URL**. This is the URL that the OpenID provider asserts as the Issuer Identifier. It must use the https scheme with no URL query parameters or fragments.
10. Enter an **Email** attribute whose value should be used as the email address. Click **Add more** to add multiple email attributes.
11. Enter a **Name** attribute whose value should be used as the preferred username. Click **Add more** to add multiple preferred usernames.
12. Enter a **Preferred username** attribute whose value should be used as the display name. Click **Add more** to add multiple display names.
13. Optional: Click **Show advanced Options** to add a certificate authority (CA) file to your OpenID identity provider.
14. Optional: Under the advanced options, you can add **Additional scopes**. By default, the **OpenID** scope is requested.
15. Click **Confirm**.

Verification

The configured identity provider is now visible on the **Access control** tab of the **Clusters** page.

5.7. NEXT STEPS

Accessing a cluster

CHAPTER 6. ACCESSING A ROSA CLUSTER

As a best practice, access your Red Hat OpenShift Service on AWS (ROSA) cluster using an identity provider (IDP) account. However, the cluster administrator who created the cluster can access it using the quick access procedure.

6.1. ACCESSING YOUR CLUSTER

To log in to your cluster, you can use this quick access procedure.

NOTE

As a best practice, access your cluster with an IDP account instead.

Procedure

To access your cluster:

1. Enter the following command:

```
$ rosa create admin -c <cluster_name>
```

Example output

```
W: It is recommended to add an identity provider to login to this cluster. See 'rosa create idp --help' for more information.
I: Admin account has been added to cluster 'cluster_name'. It may take up to a minute for the account to become active.
I: To login, run the following command:
oc login https://api.cluster-name.t6k4.i1.oragnization.org:6443 \
--username cluster-admin \
--password FWGYL-2mkJI-3ZTTZ-rINns
```

2. Enter the **oc login** command, username, and password from the output of the previous command:

Example output

```
$ oc login https://api.cluster_name.t6k4.i1.oragnization.org:6443 \
>  --username cluster-admin \
>  --password FWGYL-2mkJI-3ZTTZ-rINns
Login successful.
You have access to 77 projects, the list has been suppressed. You can list all projects with ' projects'
```

3. Using the default project, enter this **oc** command to verify that the cluster administrator access is created:

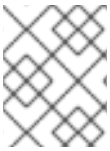
```
$ oc whoami
```

Example output

```
cluster-admin
```

6.2. ACCESSING YOUR CLUSTER WITH AN IDP ACCOUNT

To log in to your cluster, you can configure an identity provider (IDP). This procedure uses GitHub as an example IDP. To view other supported IDPs, run the **rosa create idp --help** command.



NOTE

Alternatively, as the user who created the cluster, you can use the quick access procedure.

Procedure

To access your cluster using an IDP account:

1. Add an IDP.
   a. The following command creates an IDP backed by GitHub. After running the command, follow the interactive prompts from the output to access your GitHub developer settings and configure a new OAuth application.

```
$ rosa create idp --cluster=rh-rosa-test-cluster --interactive
```

   b. Enter the following values:

   - Type of identity provider: **github**
   - Restrict to members of: **organizations** (if you do not have a GitHub Organization, you can create one now.)
   - GitHub organizations: **rh-test-org** (enter the name of your org)

Example output

```
I: Interactive mode enabled.
Any optional fields can be left empty and a default will be selected.
? Type of identity provider: github
? Restrict to members of: organizations
? GitHub organizations: rh-test-org
? To use GitHub as an identity provider, you must first register the application:
  - Open the following URL:
    https://github.com/organizations/rh-rosa-test-cluster/settings/applications/new?
oauth_application%5Bcallback_url%5D=https%3A%2F%2Foauth-openshift.apps.rh-
rosa-test-cluster.z7v0.s1.devshift.org%2Foauth2callback%2Fgithub-
1&oauth_application%5Bname%5D=rh-rosa-test-cluster-
stage&oauth_application%5Burl%5D=https%3A%2F%2Fconsole-openshift-
console.apps.rh-rosa-test-cluster.z7v0.s1.devshift.org
  - Click on 'Register application'
...
```

   c. Follow the URL from the output. This creates a new OAuth application in the GitHub organization you specified.

d. Click **Register application** to access your client ID and client secret.

e. Use the information from the GitHub application you created and continue the prompts. Enter the following values:

- Client ID: **<my_github_client_id>**
- Client Secret: [? for help] **<my_github_client_secret>**
- Hostname: (optional, you can leave it blank for now)
- Mapping method: **claim**

Continued example output

```
...
? Client ID: <my_github_client_id>
? Client Secret: [? for help] <my_github_client_secret>
? Hostname:
? Mapping method: claim
I: Configuring IDP for cluster 'rh_rosa_test_cluster'
I: Identity Provider 'github-1' has been created. You need to ensure that there is a list of cluster administrators defined. See 'rosa create user --help' for more information. To login into the console, open https://console-openshift-console.apps.rh-test-org.z7v0.s1.devshift.org and click on github-1
```

The IDP can take 1–2 minutes to be configured within your cluster.

f. Enter the following command to verify that your IDP has been configured correctly:

```
$ rosa list idps --cluster rh-rosa-test-cluster1
```

Example output

```
NAME        TYPE      AUTH URL
github-1    GitHub    https://oauth-openshift.apps.rh-rosa-test-cluster1.j9n4.s1.devshift.org/oauth2callback/github-1
```

2. Log in to your cluster.

a. Enter the following command to get the **Console URL** of your cluster:

```
$ rosa describe cluster rh-rosa-test-cluster1
```

Example output

```
Name:        rh-rosa-test-cluster1
ID:          1de87g7c30g75qechgh7l5b2bha6r04e
External ID: 34322be7-b2a7-45c2-af39-2c684ce624e1
API URL:     https://api.rh-rosa-test-cluster1.j9n4.s1.devshift.org:6443
Console URL: https://console-openshift-console.apps.rh-rosa-test-cluster1.j9n4.s1.devshift.org
Nodes:       Master: 3, Infra: 3, Compute: 4
```

```
Region:      us-east-2
State:       ready
Created:     May 27, 2020
```

b. Navigate to the **Console URL**, and log in using your Github credentials.

c. In the top right of the OpenShift console, click your name and click **Copy Login Command**.

d. Select the name of the IDP you added (in our case **github-1**), and click **Display Token**.

e. Copy and paste the **oc** login command into your terminal.

```
$ oc login --token=z3sgOGVDk0k4vbqo_wFqBQQTnT-nA-nQLb8XEmWnw4X --
server=https://api.rh-rosa-test-cluster1.j9n4.s1.devshift.org:6443
```

Example output

```
Logged into "https://api.rh-rosa-cluster1.j9n4.s1.devshift.org:6443" as "rh-rosa-test-user"
using the token provided.

You have access to 67 projects, the list has been suppressed. You can list all projects
with 'oc projects'

Using project "default".
```

f. Enter a simple **oc** command to verify everything is setup properly and that you are logged in.

```
$ oc version
```

Example output

```
Client Version: 4.4.0-202005231254-4a4cd75
Server Version: 4.3.18
Kubernetes Version: v1.16.2
```

6.3. GRANTING CLUSTER-ADMIN ACCESS

As the user who created the cluster, add the **cluster-admin** user role to your account to have the maximum administrator privileges. These privileges are not automatically assigned to your user account when you create the cluster.

Additionally, only the user who created the cluster can grant cluster access to other **cluster-admin** or **dedicated-admin** users. Users with **dedicated-admin** access have fewer privileges. As a best practice, limit the number of **cluster-admin** users to as few as possible.

Prerequisites

- You have added an identity provider (IDP) to your cluster.
- You have the IDP user name for the user you are creating.
- You are logged in to the cluster.

Procedure

1. Give your user **cluster-admin** privileges:

```
$ rosa grant user cluster-admin --user <idp_user_name> --cluster <cluster_name>
```

2. Verify your user is listed as a cluster administrator:

```
$ rosa list users --cluster <cluster_name>
```

Example output

```
GROUP            NAME
cluster-admins   rh-rosa-test-user
dedicated-admins rh-rosa-test-user
```

3. Enter the following command to verify that your user now has **cluster-admin** access. A cluster administrator can run this command without errors, but a dedicated administrator cannot.

```
$ oc get all -n openshift-apiserver
```

Example output

```
NAME                READY   STATUS    RESTARTS   AGE
pod/apiserver-6ndg2   1/1     Running   0          17h
pod/apiserver-lrmxs   1/1     Running   0          17h
pod/apiserver-tsqhz   1/1     Running   0          17h
NAME          TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
service/api   ClusterIP   172.30.23.241   <none>       443/TCP   18h
NAME                   DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR                  AGE
daemonset.apps/apiserver   3         3         3       3            3           node-
role.kubernetes.io/master=   18h
```

6.4. GRANTING DEDICATED-ADMIN ACCESS

Only the user who created the cluster can grant cluster access to other **cluster-admin** or **dedicated-admin** users. Users with **dedicated-admin** access have fewer privileges. As a best practice, grant **dedicated-admin** access to most of your administrators.

Prerequisites

- You have added an identity provider (IDP) to your cluster.
- You have the IDP user name for the user you are creating.
- You are logged in to the cluster.

Procedure

1. Enter the following command to promote your user to a **dedicated-admin**:

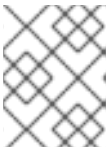
```
$ rosa grant user dedicated-admin --user <idp_user_name> --cluster <cluster_name>
```

2. Enter the following command to verify that your user now has **dedicated-admin** access:

```
$ oc get groups dedicated-admins
```

Example output

```
NAME              USERS
dedicated-admins  rh-rosa-test-user
```



NOTE

A **Forbidden** error displays if user without **dedicated-admin** privileges runs this command.

CHAPTER 7. DELETING ACCESS TO A ROSA CLUSTER

Delete access to a Red Hat OpenShift Service on AWS (ROSA) cluster using the **rosa** command-line.

7.1. REVOKING DEDICATED-ADMIN ACCESS

Only the user who created the cluster can revoke access for a **dedicated-admin** users.

Prerequisites

- You have added an Identity Provider (IDP) to your cluster.
- You have the IDP user name for the user whose privileges you are revoking.
- You are logged in to the cluster.

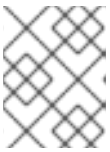
Procedure

1. Enter the following command to revoke access for a **dedicated-admin**:

```
$ rosa revoke user dedicated-admin --user <idp_user_name> --cluster <cluster_name>
```

2. Enter the following command to verify that your user no longer has **dedicated-admin** access. The user will not be listed in the output.

```
$ oc get groups dedicated-admins
```



NOTE

A **Forbidden** error displays if user without **dedicated-admin** privileges runs this command.

7.2. REVOKING CLUSTER-ADMIN ACCESS

Only the user who created the cluster can revoke access for **cluster-admin** users.

Prerequisites

- You have added an Identity Provider (IDP) to your cluster.
- You have the IDP user name for the user whose privileges you are revoking.
- You are logged in to the cluster.

Procedure

1. Revoke the user **cluster-admin** privileges:

```
$ rosa revoke user --cluster <cluster_name> --cluster-admins <idp_user_name>
```

2. Verify your user is no longer listed as a **cluster-admin**:

```
$ rosa list users --cluster <cluster_name>
```

CHAPTER 8. DELETING A ROSA CLUSTER

Delete a Red Hat OpenShift Service on AWS (ROSA) cluster using the **rosa** command-line.

8.1. DELETING A CLUSTER

You can delete an Red Hat OpenShift Service on AWS cluster using the **rosa** CLI.

If add-ons are installed, the deletion takes longer because add-ons are uninstalled before the cluster is deleted. The amount of time depends on the number and size of the add-ons.

Procedure

1. Enter the following command to delete a cluster and watch the logs, replacing **<cluster_name>** with the name or ID of your cluster:

```
$ rosa delete cluster --cluster=<cluster_name> --watch
```

2. To clean up your CloudFormation stack, enter the following command:

```
$ rosa init --delete-stack
```

CHAPTER 9. REQUIRED AWS SERVICE QUOTAS

A list of Amazon Web Service (AWS) service quotas required to run an Red Hat OpenShift Service on AWS (ROSA) cluster.

This section lists the required Amazon Web Service (AWS) service quotas to run an Red Hat OpenShift Service on AWS cluster.

9.1. REQUIRED AWS SERVICE QUOTAS

The table below describes the AWS service quotas and levels required to create and run an Red Hat OpenShift Service on AWS cluster.

If you need to modify or increase a specific quota, please refer to Amazon's documentation on requesting a quota inscrease .

Quota name	Service code	Quota code	Minimum required value	Recommended value
Number of EIPs - VPC EIPs	ec2	L-0263D0A3	5	5
Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances	ec2	L-1216C47A	100	100
VPCs per Region	vpc	L-F678F1CE	5	5
Internet gateways per Region	vpc	L-A4707A72	5	5
Network interfaces per Region	vpc	L-DF5E4CA3	5,000	5,000
General Purpose SSD (gp2) volume storage	ebs	L-D18FCD1D	300	300
Number of EBS snapshots	ebs	L-309BACF6	300	300
Provisioned IOPS	ebs	L-B3A130E6	300,000	300,000
Provisioned IOPS SSD (io1) volume storage	ebs	L-FD252861	300	300
Application Load Balancers per Region	elasticloadbalancing	L-53DA6B97	50	50

Quota name	Service code	Quota code	Minimum required value	Recommended value
Classic Load Balancers per Region	elasticloadbalancing	L-E9E9831D	20	20