Red Hat OpenShift Service on AWS 4

Setting up accounts and clusters

Setting up clusters and accounts
Setting up accounts and clusters
Abstract

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

Chapter 1. Getting Started Workflow ................................................. 4  
1.1. Additional Resources ............................................................. 4

Chapter 2. AWS Prerequisites for ROSA ........................................... 5  
2.1. Deployment Prerequisites ......................................................... 5  
2.2. Customer Requirements .......................................................... 5  
2.2.1. Account .............................................................................. 5  
2.2.2. Access requirements ............................................................ 5  
2.2.3. Support requirements ........................................................... 5  
2.2.4. Security requirements ........................................................... 5  
2.3. Required Customer Procedure .................................................. 6  
2.3.1. Minimum required Service Control Policy (SCP) .................... 6  
2.4. Red Hat Managed IAM References for AWS ............................. 11  
2.4.1. IAM Policies ...................................................................... 11  
2.4.2. IAM users ......................................................................... 11  
2.5. Provisioned AWS Infrastructure ............................................... 11  
2.5.1. EC2 instances .................................................................... 11  
2.5.2. Elastic Block Storage storage ............................................ 11  
2.5.3. Elastic load balancers ......................................................... 12  
2.5.4. S3 storage ......................................................................... 12  
2.5.5. VPC .................................................................................. 12  
2.5.6. Security groups ................................................................. 13  
2.6. AWS PrivateLink Firewall Prerequisites ................................... 13  
2.7. Next Steps ............................................................................. 15  
2.8. Additional Resources ............................................................... 16

Chapter 3. Required AWS Service Quotas ....................................... 17  
3.1. Required AWS Service Quotas .................................................. 17  
3.2. Next Steps ............................................................................. 18  
3.3. Additional Resources ............................................................... 18

Chapter 4. Configuring Your AWS Account ..................................... 19  
4.1. Configuring Your AWS Account ............................................... 19  
4.2. Next Steps ............................................................................. 20  
4.3. Additional Resources ............................................................... 20

Chapter 5. Installing ROSA ............................................................... 21  
5.1. Installing ROSA .................................................................... 21  
5.2. Next Steps ............................................................................. 24  
5.3. Additional Resources ............................................................... 24

Chapter 6. Creating A ROSA Cluster ............................................... 25  
6.1. Creating Your Cluster ............................................................... 25  
6.2. Next Steps ............................................................................. 26  
6.3. Additional Resources ............................................................... 26

Chapter 7. Creating An AWS PrivateLink Cluster On ROSA ............... 28  
7.1. Understanding PrivateLink ....................................................... 28  
7.2. Requirements For Using PrivateLink Clusters ......................... 28  
7.3. Creating An AWS PrivateLink Cluster ..................................... 29  
7.4. Next Steps ............................................................................. 30  
7.5. Additional Resources ............................................................... 30
CHAPTER 8. CONFIGURING IDENTITY PROVIDERS
8.1. UNDERSTANDING IDENTITY PROVIDERS
  8.1.1. Supported identity providers
  8.1.2. Identity provider parameters
8.2. CONFIGURING A GITHUB IDENTITY PROVIDER
8.3. CONFIGURING A GITLAB IDENTITY PROVIDER
8.4. CONFIGURING A GOOGLE IDENTITY PROVIDER
8.5. CONFIGURING A LDAP IDENTITY PROVIDER
8.6. CONFIGURING AN OPENID IDENTITY PROVIDER
8.7. NEXT STEPS
8.8. ADDITIONAL RESOURCES

CHAPTER 9. ACCESSING A ROSA CLUSTER
9.1. ACCESSING YOUR CLUSTER QUICKLY
9.2. ACCESSING YOUR CLUSTER WITH AN IDP ACCOUNT
9.3. GRANTING CLUSTER-ADMIN ACCESS
9.4. GRANTING DEDICATED-ADMIN ACCESS
9.5. ADDITIONAL RESOURCES

CHAPTER 10. DELETING ACCESS TO A ROSA CLUSTER
10.1. REVOKING DEDICATED-ADMIN ACCESS
10.2. REVOKING CLUSTER-ADMIN ACCESS

CHAPTER 11. DELETING A ROSA CLUSTER
11.1. DELETING A CLUSTER

CHAPTER 12. COMMAND QUICK REFERENCE FOR CREATING CLUSTERS AND USERS
12.1. COMMAND QUICK REFERENCE LIST
12.2. ADDITIONAL RESOURCES
CHAPTER 1. GETTING STARTED WORKFLOW

Follow this workflow to set up and access Red Hat OpenShift Service on AWS (ROSA) clusters:

1. Perform the AWS prerequisites.
2. Review the required AWS service quotas.
3. Configure your AWS account.
4. Install ROSA.
5. Create a ROSA cluster or Create a ROSA cluster using AWS PrivateLink.
6. Configure identity providers.

1.1. ADDITIONAL RESOURCES

- Deleting a cluster
- Deleting access to a cluster
- Command quick reference for creating clusters and users
CHAPTER 2. AWS PREREQUISITES FOR ROSA

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.

You must ensure that the prerequisites are met before installing ROSA.

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 AWS Organizations to manage multiple AWS accounts. The AWS Organizations, managed by the customer, host 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 Organizations are managed in whatever manner the customer requires. Red Hat Site Reliability Engineers (SRE) will not have any control over SCPs within AWS Organizations.

2.2. CUSTOMER REQUIREMENTS

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

NOTE

In order to create the cluster, the user must be logged in as an IAM user and not an assumed role or STS user.

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 Organizations with the applicable Service Control Policy (SCP) applied.

NOTE

It is not a requirement that the customer’s account be within the AWS Organizations 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.
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 Organizations and control what services are available within the attached AWS accounts.

<table>
<thead>
<tr>
<th>Service</th>
<th>Actions</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Amazon EC2 Auto Scaling</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Amazon S3</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Identity And Access Management</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Elastic Load Balancing</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Elastic Load Balancing V2</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Amazon CloudWatch</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Amazon CloudWatch Events</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Amazon CloudWatch Logs</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>AWS Support</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>AWS Key Management Service</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>AWS Security Token Service</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>AWS Resource Tagging</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Service</td>
<td>Actions</td>
<td>Effect</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>--------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>AWS Route53 DNS</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>AWS Service Quotas</td>
<td>ListServices</td>
<td>Allow</td>
</tr>
<tr>
<td></td>
<td>GetRequestedServiceQuotaChange</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GetServiceQuota</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RequestServiceQuotaInCREASE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ListServiceQuotas</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>AWS Billing</td>
<td>Allow</td>
</tr>
<tr>
<td></td>
<td>ViewAccount</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Viewbilling</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ViewUsage</td>
<td></td>
</tr>
<tr>
<td>AWS Cost and Usage Report</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>AWS Cost Explorer Services</td>
<td>All</td>
<td>Allow</td>
</tr>
</tbody>
</table>

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

```json
{
  "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.

Instance types can vary for control plane and infrastructure nodes, depending on the worker node count.

- Three m5.xlarge minimum (control plane nodes)
- Two r5.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.

<table>
<thead>
<tr>
<th>Group</th>
<th>Type</th>
<th>IP Protocol</th>
<th>Port range</th>
</tr>
</thead>
<tbody>
<tr>
<td>MasterSecurityGroup</td>
<td>AWS::EC2::Security Group</td>
<td>icmp</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tcp</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tcp</td>
<td>6443</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tcp</td>
<td>22623</td>
</tr>
<tr>
<td>WorkerSecurityGroup</td>
<td>AWS::EC2::Security Group</td>
<td>icmp</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tcp</td>
<td>22</td>
</tr>
<tr>
<td>BootstrapSecurityGroup</td>
<td>AWS::EC2::Security Group</td>
<td>tcp</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tcp</td>
<td>19531</td>
</tr>
</tbody>
</table>

### 2.6. AWS PRIVATELINK FIREWALL PREREQUISITES

If you are using a firewall and want to create your cluster using AWS PrivateLink, you must configure your firewall to grant access to the sites that Red Hat OpenShift Service on AWS requires.

See [Configuring your firewall for OpenShift Container Platform](#) for information about granting access to sites that OCP requires.

#### Procedure

1. Allowlist the following URLs that are used to install and download packages and tools:

<table>
<thead>
<tr>
<th>URL</th>
<th>Function</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>registry.redhat.io</td>
<td>Provides core components such as dev tools and operator based add-ons, and Red Hat provided container images including middleware and Universal Base Image.</td>
<td>Required</td>
</tr>
<tr>
<td>quay.io</td>
<td>Used by the cluster to download the platform container images</td>
<td>Required</td>
</tr>
</tbody>
</table>
The https://cloud.redhat.com/openshift site uses authentication from sso.redhat.com to download the pull secret and use Red Hat SaaS solutions to facilitate monitoring of your subscriptions, cluster inventory, chargeback reporting, and so on.

pull.q1w2.quay.rhcloud.com Provides a fallback registry used by the cluster when quay.io is not available. Recommended

When you add a site such as quay.io to your allowlist, do not add a wildcard entry such as *.quay.io to your denylist. In most cases, image registries use a content delivery network (CDN) to serve images. If a firewall blocks access, then image downloads are denied when the initial download request is redirected to a host name such as cdn01.quay.io.

2. Managed clusters require Telemetry to be enabled to allow Red Hat to react more quickly to problems and better support our customers, as well as better understand how product upgrades impact clusters. See About remote health monitoring for more information about how remote health monitoring data is used by Red Hat.

https://cloud.redhat.com Used by the cluster for the insights operator that integrates with the SaaS Red Hat Insights Required

3. For Amazon Web Services (AWS), you must grant access to the URLs that provide the AWS API and DNS:

*.amazonaws.com Required to access AWS services and resources. Review the AWS Service Endpoints in the AWS documentation to determine the exact endpoints to allow for the regions that you use. Required

4. Allowlist the following URLs:

mirror.openshift.com Required to access mirrored installation content and images. This site is also a source of release image signatures, although the Cluster Version Operator needs only a single functioning source. Required

storage.googleapis.com/openshift-release Alternative site to download platform release signatures that are used by the cluster to know what images to pull from quay.io. Recommended
### URL | Function | Requirement
---|---|---
**quay-registry.s3.amazonaws.com** | Used to access Quay image content in AWS. | Required

**api.openshift.com** | Required to check if updates are available for the cluster. | Required

**art-rhcos-ci.s3.amazonaws.com** | Specifies the {op-system-first} images to download. | Required

**cloud.redhat.com/openshift** | Required for your cluster token. | Required

**registry.access.redhat.com** | Access to the odo CLI tool that helps developers build on OpenShift and Kubernetes. | Required

5. Allowlist the following OpenShift Dedicated URLs:

| URL | Function | Requirement |
---|---|---|
**api.pagerduty.com** and **events.pagerduty.com** | This alerting service is used by the in-cluster alertmanager to send alerts notifying SREs of an event to take action on. | Required

**api.deadmanssnitch.com** and **nosnch.in** | Alerting service used by OpenShift Dedicated to send periodic pings that indicate whether the cluster is available and running. | Required

**sftp.access.redhat.com** | FTP server used by must-gather-operator to upload diagnostic logs to help troubleshoot issues with the cluster. | Recommended

***.osdsecuritylogs.splunkcloud.com** | Used by the splunk-forwarder-operator as a logging forwarding endpoint to be used by the SRE Platform Team for log-based alerting. | Required

**observatorium.api.openshift.com** | Used for Managed OpenShift-specific telemetry. | Required

6. Allowlist any site that provides resources for a language or framework that your builds require.

7. Allowlist any outbound URLs that depend on the languages and frameworks used in OpenShift. See **OpenShift Outbound URLs to Allow** for a list of recommended URLs to be allowed on the firewall or proxy.

### 2.7. NEXT STEPS
Review the required AWS service quotas

2.8. ADDITIONAL RESOURCES

- See [SRE access to all Red Hat OpenShift Service on AWS clusters](#) for information about how Red Hat site reliability engineering accesses ROSA clusters.

- [Getting started workflow](#)
CHAPTER 3. REQUIRED AWS SERVICE QUOTAS

Review this list of the required Amazon Web Service (AWS) service quotas that are required to run an Red Hat OpenShift Service on AWS cluster.

3.1. 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.

NOTE

The AWS SDK allows ROSA to check quotas, but the AWS SDK calculation does not include your existing usage. Therefore, it is possible that the quota check can pass in the AWS SDK yet the cluster creation can fail. To fix this issue, increase your quota.

If you need to modify or increase a specific quota, see Amazon’s documentation on requesting a quota increase.

<table>
<thead>
<tr>
<th>Quota name</th>
<th>Service code</th>
<th>Quota code</th>
<th>Minimum required value</th>
<th>Recommended value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of EIPs - VPC EIPs</td>
<td>ec2</td>
<td>L-0263D0A3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances</td>
<td>ec2</td>
<td>L-1216C47A</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>VPCs per Region</td>
<td>vpc</td>
<td>L-F678F1CE</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Internet gateways per Region</td>
<td>vpc</td>
<td>L-A4707A72</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Network interfaces per Region</td>
<td>vpc</td>
<td>L-DF5E4CA3</td>
<td>5,000</td>
<td>5,000</td>
</tr>
<tr>
<td>General Purpose SSD (gp2) volume storage</td>
<td>ebs</td>
<td>L-D18FCD1D</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Number of EBS snapshots</td>
<td>ebs</td>
<td>L-309BACF6</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Provisioned IOPS SSD (io1) volume storage</td>
<td>ebs</td>
<td>L-B3A130E6</td>
<td>300,000</td>
<td>300,000</td>
</tr>
<tr>
<td>Provisioned IOPS SSD (io1) volume storage</td>
<td>ebs</td>
<td>L-FD252861</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Quota name</td>
<td>Service code</td>
<td>Quota code</td>
<td>Minimum required value</td>
<td>Recommended value</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------</td>
<td>------------</td>
<td>------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Application Load Balancers per Region</td>
<td>elasticloadbalancing</td>
<td>L-53DA6B97</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Classic Load Balancers per Region</td>
<td>elasticloadbalancing</td>
<td>L-E9E9831D</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

### 3.2. NEXT STEPS
- Configure your AWS account

### 3.3. ADDITIONAL RESOURCES
- Getting started workflow
CHAPTER 4. CONFIGURING YOUR AWS ACCOUNT

After you complete the AWS prerequisites, configure your AWS account and enable the Red Hat OpenShift Service on AWS (ROSA) service.

4.1. CONFIGURING YOUR AWS ACCOUNT

To configure your AWS account to use the ROSA service, complete the following steps.

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. Log in to the Amazon Web Services (AWS) account that you want to use.
   A dedicated AWS account is recommended 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 AWS Prerequisites for details on the minimum required SCP.
   As part of the cluster creation process, rosa establishes 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 Console.
   a. Sign in to your AWS account.
   b. To enable ROSA, go to the ROSA service and select Enable OpenShift.

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. Set a default AWS region.

   **NOTE**
   It is recommended to set the default AWS region by using the environment variable.
The ROSA service 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> |
```

After completing these steps, install ROSA.

4.2. NEXT STEPS

- Install ROSA

4.3. ADDITIONAL RESOURCES

- AWS prerequisites
- Required AWS service quotas and requesting increases
- Getting started workflow
CHAPTER 5. INSTALLING ROSA

After you configure your AWS account, install Red Hat OpenShift Service on AWS (ROSA).

5.1. INSTALLING ROSA

Complete the following steps to install ROSA before creating a cluster.

Prerequisites

- Review and complete the AWS prerequisites and ROSA 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.
- Configure your AWS account and enable the ROSA service in your AWS account.

Procedure

1. 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.

   **Example**

   ```
   $ mv rosa /usr/local/bin/rosa
   ```

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
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
```

2. 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
```

3. Log in to your Red Hat account with `rosa`.

   a. Enter the following command.

   ```
   $ rosa login
   ```

   b. 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'
   ```

4. 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.

5. Prepare your AWS account for cluster deployment:

a. 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:               1DzGldIhqEWyt8UUXQhSoWaaaaa
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
```

b. 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
```
6. Install the OpenShift CLI (oc) from the rosa CLI.
   a. Enter this command to download the latest version of the oc CLI:
      
      ```bash
      $ 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:
      
      ```bash
      $ rosa verify oc
      ```

   After installing ROSA, you are ready to create a cluster.

5.2. NEXT STEPS

- Create a ROSA cluster or Create an AWS PrivateLink cluster on ROSA.

5.3. ADDITIONAL RESOURCES

- AWS Prerequisites

- Required AWS service quotas and requesting increases

- Getting started workflow
CHAPTER 6. CREATING A ROSA CLUSTER

After you set up your environment and install Red Hat OpenShift Service on AWS (ROSA), create a cluster.

This document describes how to set up a ROSA cluster. Alternatively, you can create a ROSA cluster with AWS PrivateLink.

6.1. CREATING YOUR CLUSTER

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

**Prerequisites**

You have installed {product-title}.

**NOTE**

AWS Shared VPCs are not currently supported for ROSA installs.

**Procedure**

1. You can create a cluster using the default settings or by specifying custom settings using the interactive mode. To view other options when creating a cluster, enter `rosa create cluster --help`.

Creating a cluster can take up to 40 minutes.

**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 create your cluster with the default cluster settings:

  ```
  $ rosa create cluster --cluster-name=<cluster_name>
  ```

**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`
```

- To create a cluster using interactive prompts:

  ```
  $ rosa create cluster --interactive
  ```
To configure your networking IP ranges, you can use the following default ranges. For more information when using manual mode, use `rosa create cluster --help | grep cidr`. In interactive mode, you are prompted for the settings.

- Node CIDR: 10.0.0.0/16
- Service CIDR: 172.30.0.0/16
- Pod CIDR: 10.128.0.0/14

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`.

   ```bash
   $ rosa describe cluster --cluster=<cluster_name>
   ```

   **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.org
   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:

   ```bash
   $ rosa logs install --cluster=<cluster_name> --watch
   ```

### 6.2. NEXT STEPS

Configure identity providers

### 6.3. ADDITIONAL RESOURCES

- Getting started workflow
- Deleting a ROSA cluster
- ROSA architecture
CHAPTER 7. CREATING AN AWS PRIVATELINK CLUSTER ON ROSA

This document describes how to create a ROSA cluster using AWS PrivateLink. Alternatively, you can create a ROSA cluster without AWS PrivateLink.

7.1. UNDERSTANDING PRIVATELINK

A Red Hat OpenShift Service on AWS cluster can be created without any requirements on public subnets, internet gateways, or network address translation (NAT) gateways. In this configuration, Red Hat uses AWS PrivateLink (PrivateLink) to manage and monitor a cluster in order to avoid all public ingress network traffic.

For more information, see AWS PrivateLink on the AWS website.

7.2. REQUIREMENTS FOR USING PRIVATELINK CLUSTERS

For PrivateLink clusters, internet gateways, NAT gateways and public subnets are not required, but the private subnets must have internet connectivity provided in order to install required components. At least one single private subnet is required for Single-AZ clusters and at least 3 private subnets are required for Multi-AZ clusters. The following table shows the AWS resources that are required for a successful installation:

<table>
<thead>
<tr>
<th>Component</th>
<th>AWS Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPC</td>
<td>AWS::EC2::VPC</td>
<td>You must provide a VPC for the cluster to use.</td>
</tr>
<tr>
<td></td>
<td>AWS::EC2::VPCEndpoint</td>
<td></td>
</tr>
<tr>
<td>Network access control</td>
<td>AWS::EC2::NetworkAcl</td>
<td>You must allow access to the following ports:</td>
</tr>
<tr>
<td></td>
<td>AWS::EC2::NetworkAclEntry</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port</td>
<td>Reason</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>Inbound HTTP traffic</td>
</tr>
<tr>
<td></td>
<td>443</td>
<td>Inbound HTTPS traffic</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>Inbound SSH traffic</td>
</tr>
<tr>
<td></td>
<td>1024-65535</td>
<td>Inbound ephemeral traffic</td>
</tr>
<tr>
<td></td>
<td>0-65535</td>
<td>Outbound ephemeral traffic</td>
</tr>
</tbody>
</table>
### 7.3. CREATING AN AWS PRIVATELINK CLUSTER

You can create an AWS PrivateLink cluster using the `rosa` CLI.

#### Prerequisites

You have installed Red Hat OpenShift Service on AWS.

#### Procedure

Creating a cluster can take up to 40 minutes.

1. With AWS PrivateLink, you can create a cluster with a single availability zone (Single-AZ) or multiple availability zones (Multi-AZ). In either case, your machine’s classless inter-domain routing (CIDR) must match your virtual private cloud’s CIDR. See [Requirements for using your own VPC](#) and [VPC Validation](#) for more information.

   **IMPORTANT**

   If you use a firewall, you must configure it so that Red Hat OpenShift Service on AWS can access the sites that it requires to function.

   For more information, see the AWS PrivateLink firewall prerequisites section.

   - To create a Single-AZ cluster:
     ```
     $ rosa create cluster --private-link --cluster-name=<cluster-name> [--machine-cidr=<VPC CIDR>/16] --subnet-ids=<private-subnet-id>
     ```

   - To create a Multi-AZ cluster:
     ```
     $ rosa create cluster --private-link --multi-az --cluster-name=<cluster-name> [--machine-cidr=<VPC CIDR>/16] --subnet-ids=<private-subnet-id1>,<private-subnet-id2>,<private-subnet-id3>
     ```

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 --cluster=<cluster_name>
   ```
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 --cluster=<cluster_name> --watch

7.4. NEXT STEPS

Configure identity providers

7.5. ADDITIONAL RESOURCES

- AWS PrivateLink firewall prerequisites
- Getting started workflow
- Deleting a ROSA cluster
- ROSA architecture
CHAPTER 8. 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.

8.1. 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.

8.1.1. Supported identity providers

You can configure the following types of identity providers:

<table>
<thead>
<tr>
<th>Identity provider</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub or GitHub Enterprise</td>
<td>Configure a <a href="#">github</a> identity provider to validate user names and passwords against GitHub or GitHub Enterprise’s OAuth authentication server.</td>
</tr>
<tr>
<td>GitLab</td>
<td>Configure a <a href="#">gitlab</a> identity provider to use GitLab.com or any other GitLab instance as an identity provider.</td>
</tr>
<tr>
<td>Google</td>
<td>Configure a <a href="#">google</a> identity provider using Google’s OpenID Connect integration.</td>
</tr>
<tr>
<td>LDAP</td>
<td>Configure the <a href="#">ldap</a> identity provider to validate user names and passwords against an LDAPv3 server, using simple bind authentication.</td>
</tr>
<tr>
<td>OpenID Connect</td>
<td>Configure an <a href="#">oidc</a> identity provider to integrate with an OpenID Connect identity provider using an Authorization Code Flow.</td>
</tr>
</tbody>
</table>

8.1.2. Identity provider parameters

The following parameters are common to all identity providers:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The provider name is prefixed to provider user names to form an identity name.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>mappingMethod</strong></td>
<td>Defines how new identities are mapped to users when they log in. Enter one of the following values:</td>
</tr>
<tr>
<td><strong>claim</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>lookup</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>generate</strong></td>
<td>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, <em>myuser2</em>. 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.</td>
</tr>
<tr>
<td><strong>add</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>

**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**.

### 8.2. 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 that 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.

8.3. CONFIGURING A GITLAB IDENTITY PROVIDER

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

Prerequisites

- 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 that 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.

8.4. 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 that 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/
     ```


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.

### 8.5. CONFIGURING A LDAP IDENTITY PROVIDER

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

**Prerequisites**

- 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\`

<table>
<thead>
<tr>
<th>URL component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ldap</strong></td>
<td>For regular LDAP, use the string <strong>ldap</strong>. For secure LDAP (LDAPS), use <strong>ldaps</strong> instead.</td>
</tr>
<tr>
<td><strong>host:port</strong></td>
<td>The name and port of the LDAP server. Defaults to <strong>localhost:389</strong> for ldap and <strong>localhost:636</strong> for LDAPS.</td>
</tr>
<tr>
<td><strong>basedn</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>attribute</strong></td>
<td>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 <strong>uid</strong>. It is recommended to choose an attribute that will be unique across all entries in the subtree you will be using.</td>
</tr>
<tr>
<td><strong>scope</strong></td>
<td>The scope of the search. Can be either <strong>one</strong> or <strong>sub</strong>. If the scope is not provided, the default is to use <strong>sub</strong>.</td>
</tr>
<tr>
<td><strong>filter</strong></td>
<td>A valid LDAP search filter. If not provided, defaults to <strong>(objectClass=*)</strong></td>
</tr>
</tbody>
</table>

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

\`(&(<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 that 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.

8.6. 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:

<table>
<thead>
<tr>
<th>Claim</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>preferred_username</td>
<td>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.</td>
</tr>
<tr>
<td>email</td>
<td>Email address.</td>
</tr>
<tr>
<td>name</td>
<td>Display name.</td>
</tr>
</tbody>
</table>

See the OpenID claims documentation for more information.

**Prerequisites**

- 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 that 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/
     ```


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.

8.7. NEXT STEPS
Access a cluster.

8.8. ADDITIONAL RESOURCES
- Getting started workflow
CHAPTER 9. 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.

9.1. ACCESSING YOUR CLUSTER QUICKLY

You can use this quick access procedure to log in to your cluster.

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

Procedure

1. Enter the following command:

   $ rosa create admin --cluster=<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.ornagnization.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
9.2. 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=<cluster_name> --interactive
      ```

   b. Enter the following values:

      - Type of identity provider: gh

      - 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 organization)

   **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&oath_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>`  
? 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=<cluster_name>
```

**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 --cluster=<cluster_name>
```

**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.

```bash
$ 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.

```bash
$ oc version
```

**Example output**

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

### 9.3. 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

<table>
<thead>
<tr>
<th>GROUP</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cluster-admins</td>
</tr>
<tr>
<td></td>
<td>rh-rosa-test-user</td>
</tr>
<tr>
<td></td>
<td>dedicated-admins</td>
</tr>
<tr>
<td></td>
<td>rh-rosa-test-user</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>pod/apiserver-6ndg2</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>17h</td>
</tr>
<tr>
<td>pod/apiserver-lrmxs</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>17h</td>
</tr>
<tr>
<td>pod/apiserver-tsqhz</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>17h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLUSTER-IP</th>
<th>EXTERNAL-IP</th>
<th>PORT(S)</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>service/api</td>
<td>ClusterIP</td>
<td>172.30.23.241</td>
<td>&lt;none&gt;</td>
<td>443/TCP</td>
<td>18h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESIRED</th>
<th>CURRENT</th>
<th>READY</th>
<th>UP-TO-DATE</th>
<th>AVAILABLE</th>
<th>NODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>daemonset.apps/apiserver</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>node-role.kubernetes.io/master=</td>
</tr>
</tbody>
</table>

9.4. 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:
9.5. ADDITIONAL RESOURCES

- Getting started workflow
CHAPTER 10. DELETING ACCESS TO A ROSA CLUSTER

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

10.1. 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**:

   ```shell
   $ 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.

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

   **NOTE**

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

10.2. 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:

   ```shell
   $ 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 11. DELETING A ROSA CLUSTER

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

### 11.1. DELETING A CLUSTER

You can delete a 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 12. COMMAND QUICK REFERENCE FOR CREATING CLUSTERS AND USERS

12.1. COMMAND QUICK REFERENCE LIST

If you have already created your first cluster and users, this list can serve as a command quick reference list when creating additional clusters and users.

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

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

12.2. ADDITIONAL RESOURCES

- Getting started workflow