### **User Guide**

## **Red Hat OpenShift Service on AWS**



Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

## Red Hat OpenShift Service on AWS: User Guide

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

## **Table of Contents**

What is Red Hat OpenShift Service on AWS?	1
Features	1
ROSA cluster deployment models	1
Accessing ROSA	2
How to get started with ROSA	3
Pricing	3
ROSA service fees	4
AWS infrastructure fees	4
Responsibilities	4
Overview	5
Tasks for shared responsibilities by area	7
Customer responsibilities for data and applications	30
Deployment options	33
Differences between ROSA with HCP and ROSA classic	33
Getting started with ROSA	38
ROSA cluster deployment models	
Getting started guides	38
Getting started with ROSA with HCP	39
Getting started with ROSA classic	
Using ROSA with HCP and the ROSA CLI in auto mode	
Prerequisites	
Step 1: Enable ROSA and configure prerequisites	41
Step 2: Create Amazon VPC architecture for ROSA with HCP clusters	
Step 3: Create the required IAM roles and OpenID Connect configuration	46
Step 4: Create a ROSA with HCP cluster with AWS STS and the ROSA CLI auto mode	47
Step 5: Configure an identity provider and grant cluster access	
Step 6: Grant user access to a cluster	
Step 7: Grant administrator permissions to a user	50
Step 8: Access a cluster through the Red Hat Hybrid Cloud Console	51
Step 9: Deploy an application from the Developer Catalog	
Step 10: Delete a cluster and AWS STS resources	
Using ROSA classic with the ROSA CLI in auto mode	54
Prerequisites	
Step 1: Enable ROSA and configure prerequisites	56

Step 2: Create a ROSA classic cluster with AWS STS and the ROSA CLI auto mode	56
Step 3: Configure an identity provider and grant cluster access	57
Step 4: Grant user access to a cluster	60
Step 5: Grant administrator permissions to a user	60
Step 6: Access a cluster through the web console	61
Step 7: Deploy an application from the Developer Catalog	61
Step 8: Revoke administrator permissions and user access	62
Step 9: Delete a cluster and AWS STS resources	63
Using ROSA classic with the ROSA CLI in manual mode	65
Prerequisites	65
Step 1: Enable ROSA and configure prerequisites	66
Step 2: Create a ROSA classic cluster with AWS STS and the ROSA CLI manual mode	67
Step 3: Configure an identity provider and grant cluster access	69
Step 4: Grant user access to a cluster	71
Step 5: Grant administrator permissions to a user	71
Step 6: Access a cluster through the web console	72
Step 7: Deploy an application from the Developer Catalog	72
Step 8: Revoke administrator permissions and user access	73
Step 9: Delete a cluster and AWS STS resources	74
Using ROSA classic with AWS PrivateLink	76
Prerequisites	77
Step 1: Enable ROSA and configure prerequisites	77
Step 2: Create Amazon VPC architecture for the cluster	78
Step 3: Create a cluster with AWS PrivateLink	82
Step 4: Configure AWS PrivateLink DNS forwarding	84
Step 5: Configure an identity provider and grant cluster access	85
Step 6: Grant user access to a cluster	87
Step 7: Grant administrator permissions to a user	87
Step 8: Access a cluster through the web console	88
Step 9: Deploy an application from the Developer Catalog	88
Step 10: Revoke administrator permissions and user access	90
Step 11: Delete a cluster and AWS STS resources	91
Security	93
Data protection	
Data encryption	94
Internetwork privacy	98

AWS managed IAM policies	98
AWS managed policy: ROSAManageSubscription	98
AWS managed policies for ROSA with HCP account roles	99
AWS managed policies for ROSA with HCP operator roles	99
ROSA updates to AWS managed policies	100
Account policies for ROSA with HCP	106
Operator policies for ROSA with HCP	109
Resilience	114
AWS global infrastructure resilience	114
ROSA cluster resilience	115
Customer-deployed application resilience	116
Infrastructure security	116
Cluster network isolation	116
Pod network isolation	117
Service quotas	118
Required minimum quotas for ROSA	118
Default quotas for ROSA	122
Working with other services	124
ROSA and AWS Marketplace	124
Terminology	124
ROSA payments and billing	125
Subscribing to ROSA Marketplace listings through the console	126
ROSA contracts	126
Private Marketplace	
Troubleshooting	133
Support for ROSA	133
AWS Support	133
Red Hat Support	
ROSA enablement errors in the console	
AWS Organizations service control policy (SCP) is denying required AWS Marketpl	ace
permissions	
User or role does not have the required AWS Marketplace permissions	
Required AWS Marketplace permissions blocked by an administrator	
ROSA cluster creation issues	
Access ROSA cluster debug logs	
Elastic Load Balancing (ELB) role does not exist	136

ROSA cluster fails AWS service quota check during cluster creation	137
Troubleshoot ROSA CLI expired offline access tokens	138
Non-STS cluster issues	138
Failed to create a cluster with an osdCcsAdmin error	138
Document history	140

## What is Red Hat OpenShift Service on AWS?

Red Hat OpenShift Service on AWS (ROSA) is a managed service that you can use to build, scale, and deploy containerized applications with the Red Hat OpenShift enterprise Kubernetes platform on AWS. ROSA streamlines moving on-premises Red Hat OpenShift workloads to AWS, and offers tight integration with other AWS services.

#### **Features**

ROSA is jointly supported and operated by AWS and Red Hat. Each ROSA cluster comes with 24-hour Red Hat site reliability engineer (SRE) support for cluster management, backed by Red Hat's 99.95% uptime service-level agreement (SLA). For more information about the service's support model, see Support for ROSA.

ROSA also provides the following features:

- Red Hat SRE-supported cluster installation, cluster maintenance, and cluster upgrades.
- AWS service integrations include AWS compute, database, analytics, machine learning, networking, and mobile.
- Run and scale the Kubernetes control plane across multiple AWS Availability Zones to ensure high availability.
- Operate clusters using OpenShift APIs and developer productivity tools, including Service Mesh,
   CodeReady Workspaces, and Serverless.

## **ROSA cluster deployment models**

ROSA provides two cluster deployment models: ROSA with hosted control planes (ROSA with HCP) and ROSA classic. With ROSA with HCP, each cluster has a dedicated control plane that is isolated within Red Hat's AWS account and managed by Red Hat. With ROSA classic, cluster control plane infrastructure is hosted in the customer's AWS account.

ROSA with HCP offers a more a more efficient control plane architecture that helps reduce the AWS infrastructure fees incurred when running ROSA and allows for faster cluster creation times. For more information about ROSA with HCP and ROSA classic, see Deployment options.

Features 1



#### Note

ROSA with hosted control planes does not offer compliance certifications or Federal Information Processing Standards (FIPS) at this time. For more information, see Compliance in the Red Hat documentation.

## **Accessing ROSA**

You can define and configure your ROSA service deployments using the following interfaces.

#### **AWS**

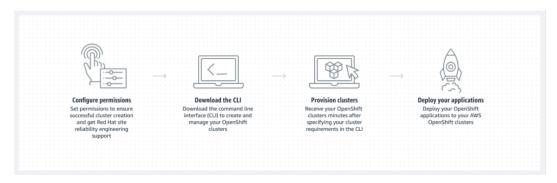
- ROSA console Provides a web interface to enable the ROSA subscription and purchase a ROSA software contract.
- AWS Command Line Interface (AWS CLI) Provides commands for a broad set of AWS services and is supported on Windows, macOS, and Linux. For more information, see AWS Command Line Interface.

#### **Red Hat OpenShift**

- Red Hat Hybrid Cloud Console Provides a web interface to create, update, and manage ROSA clusters, install cluster add-ons, and create and deploy applications to a ROSA cluster.
- ROSA CLI (rosa) Provides commands to create, update, and manage ROSA clusters.
- OpenShift CLI (oc) Provides commands to create applications and manage OpenShift Container Platform projects.
- Knative CLI (kn) Provides commands that can be used to interact with OpenShift Serverless components, such as Knative Serving and Eventing.
- Pipelines CLI (tkn) Provides commands to interact with OpenShift Pipelines using the terminal.
- opm CLI Provides commands that help Operator developers and cluster administrators create and maintain OpenShift Operator catalogs from the terminal.
- Operator SDK CLI Provides commands that an Operator developer can use to build, test, and deploy an OpenShift operator.

Accessing ROSA

## How to get started with ROSA



The following summarizes the getting started process for ROSA. For detailed getting started instructions, see Getting started with ROSA.

#### AWS Management Console/AWS CLI

- 1. Configure permissions for AWS services that ROSA relies on to deliver service functionality. For more information, see Prerequisites.
- 2. Install and configure the latest AWS CLI tool. For more information, see <u>Installing our updating</u> the latest version of the AWS CLI in the AWS CLI User Guide.
- 3. Enable ROSA in the ROSA console.

#### Red Hat Hybrid Cloud Console/ROSA CLI

- 1. Download the latest version of the ROSA CLI and OpenShift CLI from the Red Hat Hybrid Cloud Console. For more information, see Getting started with the ROSA CLI in the Red Hat documentation.
- 2. Create ROSA clusters in the Red Hat Hybrid Cloud Console or with the ROSA CLI.
- 3. When your cluster is ready, configure an identity provider to grant user access to the cluster.
- 4. Deploy and manage workloads on your ROSA cluster the same way that you would with any other OpenShift environment.

## **Pricing**

The total cost of ROSA consists of two components: ROSA service fees and AWS infrastructure fees. For more information about pricing, see Red Hat OpenShift Service on AWS Pricing.

#### **ROSA** service fees

By default, ROSA service fees accrue on demand at an hourly rate per 4 vCPU used by worker nodes. Service fees are uniform across all supported AWS standard Regions. In addition to the worker node service fee, ROSA with hosted control planes (HCP) clusters incur an hourly cluster fee.

ROSA offers 1-year and 3-year service fee contracts that you can purchase for savings on the ondemand service fees for worker nodes. For more information, see ROSA contracts.

#### **AWS** infrastructure fees

AWS infrastructure fees apply to the underlying worker nodes, infrastructure nodes, control plane nodes, storage, and network resources hosted on AWS global infrastructure. AWS infrastructure fees vary by AWS Region.

# Overview of responsibilities for Red Hat OpenShift Service on AWS

This documentation outlines the responsibilities of Amazon Web Services (AWS), Red Hat, and customers for the Red Hat OpenShift Service on AWS (ROSA) managed service. For more information about ROSA and its components, see <u>Policies and service definition</u> in the Red Hat documentation.

The <u>AWS shared responsibility model</u> defines AWS responsibility for protecting the infrastructure that runs all of the services offered in the AWS Cloud, including ROSA. AWS infrastructure includes the hardware, software, networking, and facilities that run AWS Cloud services. This AWS responsibility is commonly referred to as the "security of the cloud". To operate ROSA as a fully managed service, Red Hat and the customer are responsible for the elements of the service that the AWS responsibility model defines as "security in the cloud".

Red Hat is responsible for the ongoing management and security of the ROSA cluster infrastructure, the underlying application platform, and the operating system. While ROSA clusters are hosted on AWS resources in the customer AWS accounts, they are accessed remotely by ROSA service components and Red Hat site reliability engineers (SREs) through IAM roles that the customer creates. Red Hat uses this access to manage the deployment and capacity of all control plane and infrastructure nodes on the cluster, and maintain versions for the control plane nodes, infrastructure nodes, and worker nodes.

ROSA service fees 4

Red Hat and the customer share responsibility for ROSA network management, cluster logging, cluster versioning, and capacity management. While Red Hat manages the ROSA service, the customer is fully responsible for managing and securing any applications, workloads, and data deployed to ROSA.

#### **Overview**

The following table provides an overview of AWS, Red Hat, and customer responsibilities for Red Hat OpenShift Service on AWS.



#### Note

If the cluster-admin role is added to a user, see the responsibilities and exclusion notes in the Red Hat Enterprise Agreement Appendix 4 (Online Subscription Services).

Resource	Incident and operations managemen t	Change managemen t	Access and identity authoriza tion	Security and regulation compliance	Disaster recovery
Customer data	Customer	Customer	Customer	Customer	Customer
Customer applications	Customer	Customer	Customer	Customer	Customer
Developer services	Customer	Customer	Customer	Customer	Customer
Platform monitoring	Red Hat	Red Hat	Red Hat	Red Hat	Red Hat
Logging	Red Hat	Red Hat and customer	Red Hat and customer	Red Hat and customer	Red Hat
Application networking	Red Hat and customer	Red Hat and customer	Red Hat and customer	Red Hat	Red Hat

Overview

Resource	Incident and operations managemen t	Change managemen t	Access and identity authoriza tion	Security and regulation compliance	Disaster recovery
Cluster networking	Red Hat	Red Hat and customer	Red Hat and customer	Red Hat	Red Hat
Virtual networking managemen t	Red Hat and customer	Red Hat and customer	Red Hat and customer	Red Hat and customer	Red Hat and customer
Virtual compute managemen t (control plane, infrastru cture, and worker nodes)	Red Hat	Red Hat	Red Hat	Red Hat	Red Hat
Cluster version	Red Hat	Red Hat and customer	Red Hat	Red Hat	Red Hat
Capacity managemen t	Red Hat	Red Hat and customer	Red Hat	Red Hat	Red Hat
Virtual storage managemen t	Red Hat	Red Hat	Red Hat	Red Hat	Red Hat

Overview

Resource	Incident and operations managemen t	Change managemen t	Access and identity authoriza tion	Security and regulation compliance	Disaster recovery
AWS software (public AWS services)	AWS	AWS	AWS	AWS	AWS
Hardware/ AWS global infrastru cture	AWS	AWS	AWS	AWS	AWS

## Tasks for shared responsibilities by area

AWS, Red Hat, and customers share responsibility for the monitoring and maintenance of ROSA components. This documentation defines ROSA service responsibilities by area and task.

## Incident and operations management

AWS is responsible for protecting the hardware infrastructure that runs all of the services offered in the AWS Cloud. Red Hat is responsible for managing the service components necessary for default platform networking. The customer is responsible for incident and operations management of customer application data and any custom networking the customer may have configured.

Resource	Service responsibilities	Customer responsibilities
Application networking	Red Hat	Customer
	<ul> <li>Monitor native OpenShift router service, and respond to alerts.</li> </ul>	<ul> <li>Monitor health of applicati on routes, and the endpoints behind them.</li> <li>Report outages to AWS and Red Hat.</li> </ul>

Resource	Service responsibilities	Customer responsibilities
Virtual networking management	<ul> <li>Monitor AWS load         balancers, Amazon VPC         subnets, and AWS service         components necessary for         default platform networkin         g. Respond to alerts.</li> </ul>	<ul> <li>Customer</li> <li>Monitor health of AWS load balancer endpoints.</li> <li>Monitor network traffic that is optionally configure d through Amazon VPC-to-VPC connection, AWS VPN connection, or AWS Direct Connect for potential issues or security threats.</li> </ul>
Virtual storage management	<ul> <li>Monitor Amazon EBS         volumes used for cluster         nodes, and Amazon S3         buckets used for the ROSA         service's built-in container         image registry. Respond to         alerts.</li> </ul>	<ul> <li>Customer</li> <li>Monitor health of applicati on data.</li> <li>If customer managed AWS KMS keys are used, create and control the key lifecycle and key policies for Amazon EBS encryption.</li> </ul>
AWS software (public AWS services)	For information about AWS incident and operation s management, see <a href="How AWS maintains operation al resilience and continuit y of service">How AWS maintains operation al resilience and continuit y of service</a> in the AWS whitepaper.	<ul> <li>Customer</li> <li>Monitor health of AWS resources in the customer account.</li> <li>Use IAM tools to apply the appropriate permissions to AWS resources in the customer account.</li> </ul>

Resource	Service responsibilities	Customer responsibilities
infrastructure	• For information about AWS incident and operation s management, see How AWS maintains operation al resilience and continuit y of service in the AWS whitepaper.	<ul> <li>Configure, manage, and monitor customer applications and data to ensure application and data security controls are properly enforced.</li> </ul>

#### **Change management**

AWS is responsible for protecting the hardware infrastructure that runs all of the services offered in the AWS Cloud. Red Hat is responsible for enabling changes to the cluster infrastructure and services that the customer will control, as well as maintaining versions for the control plane nodes, infrastructure nodes, and worker nodes. The customer is responsible for initiating infrastructure changes. The customer is also responsible for installing and maintaining optional services, networking configurations on the cluster, and changes to customer data and applications.

Resource	Service responsibilities	Customer responsibilities
Logging	<ul> <li>Red Hat</li> <li>Centrally aggregate and monitor platform audit logs.</li> <li>Provide and maintain a logging Operator to enable the customer to deploy a logging stack for default application logging.</li> <li>Provide audit logs upon</li> </ul>	<ul> <li>Customer</li> <li>Install the optional default application logging Operator on the cluster.</li> <li>Install, configure, and maintain any optional applogging solutions, such as logging sidecar container s or third-party logging applications.</li> </ul>
	customer request.	<ul> <li>Tune size and frequency of application logs being</li> </ul>

Resource	Service responsibilities	Customer responsibilities
		produced by customer applications if they are affecting the stability of the logging stack or the cluster.  Request platform audit logs through a support case for researching specific incidents.
Application networking	Red Hat	Customer
	<ul> <li>Set up public load balancers. Provide the ability to set up private load balancers and up to one additional load balancer when required.</li> <li>Set up the native OpenShift router service. Provide the ability to set the router as private and add up to one additional router shard.</li> <li>Install, configure, and maintain OpenShift SDN components for default internal pod traffic.</li> <li>Provide the ability for the customer to manage NetworkPolicy and EgressNetworkPolicy (firewall) objects.</li> </ul>	<ul> <li>Configure non-default pod network permissions for project and pod networks, pod ingress, and pod egress using NetworkPolicy objects.</li> <li>Use OpenShift Cluster Manager to request a private load balancer for default application routes.</li> <li>Use OpenShift Cluster Manager to configure up to one additional public or private router shard and corresponding load balancer.</li> <li>Request and configure any additional service load balancers for specific services.</li> <li>Configure any necessary DNS forwarding rules.</li> </ul>

Resource	Service responsibilities	Customer responsibilities
Cluster networking	<ul> <li>Set up cluster management components, such as public or private service endpoints and necessary integration with Amazon VPCcomponents.</li> <li>Set up internal networking components required for internal cluster communication between worker, infrastructure, and control plane nodes.</li> </ul>	<ul> <li>Customer</li> <li>Provide optional non-defau It IP address ranges for machine CIDR, service CIDR, and pod CIDR if needed through OpenShift Cluster Manager when the cluster is provisioned.</li> <li>Request that the API service endpoint be made public or private on cluster creation or after cluster creation through OpenShift Cluster Manager.</li> </ul>

Resource	Service responsibilities	Customer responsibilities
Virtual networking management	<ul> <li>Set up and configure     Amazon VPC component     s required to provision the     cluster, such as subnets,     load balancers, internet     gateways, and NAT     gateways.</li> <li>Provide the ability for the     customer to manage AWS     VPN connectivity with     on-premises resources     , Amazon VPC-to-VPC     connectivity, and AWS     Direct Connect as required     through OpenShift Cluster     Manager.</li> <li>Enable customers to create     and deploy AWS load     balancers for use with     service load balancers.</li> </ul>	Set up and maintain optional Amazon VPC components, such as Amazon VPC-to-VPC connection, AWS VPN connection, or AWS Direct Connect.      Request and configure any additional load balancers for specific services.
Virtual compute management	<ul> <li>Red Hat</li> <li>Set up and configure the ROSA control plane and data plane to use Amazon EC2 instances for cluster compute.</li> <li>Monitor and manage the deployment of Amazon EC2 control plane and infrastru cture nodes on the cluster.</li> </ul>	<ul> <li>Monitor and manage         Amazon EC2 worker nodes         by creating a machine pool         using the OpenShift Cluster         Manager or ROSA CLI.     </li> <li>Manage changes to</li> <li>customer-deployed</li> <li>applications and applicati</li> <li>on data.</li> </ul>

Resource	Service responsibilities	Customer responsibilities
Cluster version	<ul> <li>Red Hat</li> <li>Enable upgrade scheduling process.</li> <li>Monitor upgrade progress and remedy any issues encountered.</li> <li>Publish change logs and release notes for minor and maintenance upgrades.</li> </ul>	<ul> <li>Customer</li> <li>Schedule maintenance version upgrades either immediately, for the future, or have automatic upgrades.</li> <li>Acknowledge and schedule minor version upgrades.</li> <li>Ensure the cluster version stays on a supported minor version.</li> <li>Test customer applications on minor and maintenan ce versions to ensure compatibility.</li> </ul>

Resource	Service responsibilities	Customer responsibilities
Capacity management	Red Hat	Customer
	<ul> <li>Monitor the use of the control plane. Control planes include control plane nodes and infrastru cture nodes.</li> <li>Scale and resize control plane nodes to maintain quality of service.</li> </ul>	<ul> <li>Monitor worker node utilization and, if appropria te, enable the auto scaling feature.</li> <li>Determine the scaling strategy of the cluster. See the additional resources for more information on machine pools.</li> <li>Use the provided OpenShift Cluster Manager controls to add or remove additional worker nodes as required.</li> <li>Respond to Red Hat notifications regarding cluster resource requirements.</li> </ul>

Resource	Service responsibilities	Customer responsibilities
Resource Virtual storage management	<ul> <li>Service responsibilities</li> <li>Red Hat</li> <li>Set up and configure     Amazon EBS to provision     local node storage and     persistent volume storage     for the cluster.</li> <li>Set up and configure the     built-in image registry to     use Amazon S3 bucket     storage.</li> <li>Regularly prune image     registry resources in     Amazon S3 to optimize</li> </ul>	Customer  Optionally configure the Amazon EBS CSI driver or the Amazon EFS CSI driver to provision persistent volumes on the cluster.
	Amazon S3 to optimize  Amazon S3 usage and  cluster performance.	

Resource	Service responsibilities	Customer responsibilities
AWS software (public AWS	AWS	Customer
services)	<ul> <li>Compute</li> <li>Provide the Amazon EC2 service, used for ROSA control plane, infrastru cture, and worker nodes.</li> <li>Storage</li> <li>Provide Amazon EBS to allow the ROSA service to provision local node storage and persistent volume storage for the cluster.</li> </ul>	<ul> <li>Sign requests using an access key ID and secret access key associated with an IAM principal or AWS STS temporary security credentials.</li> <li>Specify VPC subnets for the cluster to use during cluster creation.</li> <li>Optionally configure a customer-managed VPC for use with ROSA clusters.</li> </ul>
	<ul> <li>Networking</li> <li>Provide the following AWS Cloud services to satisfy ROSA virtual networking infrastructure needs: <ul> <li>Amazon VPC</li> <li>Elastic Load Balancing</li> <li>IAM</li> </ul> </li> <li>Provide the following optional AWS service integrations for ROSA: <ul> <li>AWS VPN</li> <li>AWS Direct Connect</li> <li>AWS PrivateLink</li> </ul> </li> </ul>	

Resource	Service responsibilities	Customer responsibilities
	<ul> <li>AWS Transit Gateway</li> </ul>	
Hardware/AWS global infrastructure	<ul> <li>For information on management controls for AWS data centers, see Our Controls on the AWS Cloud Security page.</li> <li>For information on change management best practices , see Guidance for Change Management on AWS in the AWS Solutions Library.</li> </ul>	<ul> <li>Customer</li> <li>Implement change         management best practices         for customer applications         and data hosted on the         AWS Cloud.</li> </ul>

## Access and identity authorization

Access and identity authorization includes responsibilities for managing authorized access to clusters, applications, and infrastructure resources. This includes tasks such as providing access control mechanisms, authentication, authorization, and managing access to resources.

Resource	Service responsibilities	Customer responsibilities
Logging	Red Hat	Customer
	<ul> <li>Adhere to an industry standards-based tiered internal access process for platform audit logs.</li> <li>Provide native OpenShift RBAC capabilities.</li> </ul>	<ul> <li>Configure OpenShift RBAC to control access to projects and by extension a project's application logs.</li> <li>For third-party or custom application logging solutions, the customer is responsible for access management.</li> </ul>

Resource	Service responsibilities	Customer responsibilities
Application networking	Red Hat	Customer
	Provide native OpenShift RBAC and dedicated- admin capabilities.	<ul> <li>Configure OpenShift         dedicated-admin and         RBAC to control access         to route configuration as         required.</li> <li>Manage Red Hat organizat         ion administrators for Red         Hat to grant access to         OpenShift Cluster Manager.         The cluster manager is used         to configure router options         and provide service load         balancer quota.</li> </ul>
Cluster networking	Red Hat	Customer
	<ul> <li>Provide customer access controls through OpenShift Cluster Manager. Provide native OpenShift RBAC and dedicated-admin capabilities.</li> </ul>	<ul> <li>Configure OpenShift         dedicated-admin and         RBAC to control access         to route configuration as         required.</li> <li>Manage Red Hat organizat         ion membership of Red Hat         accounts.</li> <li>Manage organization         administrators for Red         Hat to grant access to         OpenShift Cluster Manager.</li> </ul>

Resource	Service responsibilities	Customer responsibilities
Virtual networking management	<ul> <li>Red Hat</li> <li>Provide customer access controls through OpenShift Cluster Manager.</li> </ul>	<ul> <li>Customer</li> <li>Manage optional user access to AWS components through OpenShift Cluster Manager.</li> </ul>
Virtual compute management	<ul> <li>Provide customer access controls through OpenShift Cluster Manager.</li> </ul>	<ul> <li>Customer</li> <li>Manage optional user access to AWS components through OpenShift Cluster Manager.</li> <li>Create IAM roles and attached policies necessary to enable ROSA service access.</li> </ul>
Virtual storage management	<ul> <li>Provide customer access controls through OpenShift Cluster Manager.</li> </ul>	<ul> <li>Customer</li> <li>Manage optional user access to AWS components through OpenShift Cluster Manager.</li> <li>Create IAM roles and attached policies necessary to enable ROSA service access.</li> </ul>

Resource	Service responsibilities	Customer responsibilities
AWS software (public AWS	AWS	Customer
services)	<ul> <li>Compute</li> <li>Provide the Amazon EC2 service, used for ROSA control plane, infrastru cture, and worker nodes.</li> </ul>	<ul> <li>Create IAM roles and attached policies necessary to enable ROSA service access.</li> <li>Use IAM tools to apply the appropriate permissions to AWS resources in the</li> </ul>
	<ul> <li>Provide Amazon EBS, used to allow ROSA to provision local node storage and persistent volume storage for the cluster.</li> <li>Provide Amazon S3, used for the service's built-in image registry.</li> <li>Networking</li> <li>Provide AWS Identity and Access Management (IAM), used by customers to control access to ROSA resources running on customer accounts.</li> </ul>	<ul> <li>to enable ROSA across your AWS organization, the customer is responsible for managing AWS Organizations administrators.</li> <li>To enable ROSA across your AWS organization, the customer is responsible for distributing the ROSA entitlement grant using AWS License Manager.</li> </ul>

Resource	Service responsibilities	Customer responsibilities
Hardware/AWS global infrastructure	<ul> <li>For information on physical access controls for AWS data centers, see <u>Our Controls</u> on the AWS Cloud Security page.</li> </ul>	<ul> <li>Customer is not responsib         le for AWS global infrastru         cture.</li> </ul>

## Security and regulation compliance

The following are the responsibilities and controls related to compliance:

Resource	Service responsibilities	Customer responsibilities
Logging	<ul> <li>Send cluster audit logs to a Red Hat SIEM to analyze for security events. Retain audit logs for a defined period of time to support forensic analysis.</li> </ul>	<ul> <li>Customer</li> <li>Analyze application logs for security events.</li> <li>Send application logs to an external endpoint through logging sidecar container s or third- party logging applications if longer retention is required than is offered by the default logging stack.</li> </ul>
Virtual networking management	<ul> <li>Red Hat</li> <li>Monitor virtual networkin g components for potential issues and security threats.</li> <li>Use public AWS tools for additional monitoring and protection.</li> </ul>	<ul> <li>Customer</li> <li>Monitor optional configure d virtual networking components for potential issues and security threats.</li> <li>Configure any necessary firewall rules or customer</li> </ul>

Resource	Service responsibilities	Customer responsibilities
		data center protections as required.
Virtual compute management	Red Hat	Customer
management	<ul> <li>Monitor virtual compute components for potential issues and security threats.</li> <li>Use public AWS tools for additional monitoring and protection.</li> </ul>	<ul> <li>Monitor optional configure d virtual networking components for potential issues and security threats.</li> <li>Configure any necessary firewall rules or customer data center protections as required.</li> </ul>

Resource	Service responsibilities	Customer responsibilities
Virtual storage management	<ul> <li>Service responsibilities</li> <li>Red Hat</li> <li>Monitor virtual storage components for potential issues and security threats.</li> <li>Use public AWS tools for additional monitoring and protection.</li> </ul>	<ul> <li>Customer</li> <li>Provision Amazon EBS volumes.</li> <li>Manage Amazon EBS volume storage to ensure enough storage is available to mount as a volume in</li> </ul>
	<ul> <li>Configure the ROSA service to encrypt control plane, infrastructure, and worker node volume data by default using the AWS managed KMS key that Amazon EBS provides.</li> <li>Configure the ROSA service to encrypt customer persistent volumes that use the default storage class with the AWS managed KMS key that Amazon EBS provides.</li> </ul>	ROSA.  • Create the persistent volume claim and generate a persistent volume though OpenShift Cluster Manager.
	<ul> <li>Provide the ability for the customer to use a customer managed KMS key to encrypt persistent volumes.</li> <li>Configure the container image registry to encrypt image registry data at rest using server-side encryption with Amazon S3 managed keys (SSE-3).</li> <li>Provide the ability for the customer to create a public</li> </ul>	

Resource	Service responsibilities	Customer responsibilities
	or private Amazon S3	
	image registry to protect	
	their container images from	
	unauthorized user access.	

Resource	Service responsibilities	Customer responsibilities
Resource  AWS software (public AWS services)	AWS  Compute  Provide Amazon EC2, used for ROSA control plane, infrastructure, and worker nodes. For more informati on, see Infrastructure security in Amazon EC2 in the Amazon EC2 User Guide.  Storage  Provide Amazon EBS, used for ROSA control plane, infrastructure, and worker node volumes, as well as Kubernetes persistent volumes. For more information, see Data protection in Amazon EC2	<ul> <li>Customer</li> <li>Ensure security best practices and the principle of least privilege are followed to protect data on the Amazon EC2 instance. For more information, see Infrastructure security in Amazon EC2 and Data protection in Amazon EC2.</li> <li>Monitor optional configure d virtual networking components for potential issues and security threats.</li> <li>Configure any necessary firewall rules or customer data center protections as required.</li> <li>Create an optional customer managed KMS key and encrypt the</li> </ul>
	as well as Kubernetes persistent volumes. For more information, see Data protection in Amazon EC2 in the Amazon EC2 User Guide.  • Provide AWS KMS, which ROSA uses to encrypt control plane, infrastru cture, and worker node volumes and persisten t volumes. For more	<ul> <li>required.</li> <li>Create an optional customer managed KMS key and encrypt the Amazon EBS persistent volume using the KMS key.</li> <li>Monitor the customer data in virtual storage for potential issues and security threats. For more information, see the AWS Shared Responsibility</li> </ul>
	information, see <u>Amazon</u> <u>EBS encryption</u> in the Amazon EC2 User Guide.	Model.

Resource	Service responsibilities	Customer responsibilities
	<ul> <li>Provide Amazon S3, used for the ROSA service's built- in container image registry.</li> <li>For more information, see <u>Amazon S3 security</u> in the Amazon S3 User Guide.</li> </ul>	
	Networking	
	<ul> <li>Provide security capabilit ies and services to increase privacy and control network access on AWS global infrastructure, including network firewalls built into Amazon VPC, private or dedicated network connections, and automatic encryption of all traffic on the AWS global and regional networks between AWS secured facilities. For more information, see the AWS Shared Responsibility Model and Infrastructure security in the Introduction to AWS Security whitepape r.</li> </ul>	

Resource	Service responsibilities	Customer responsibilities
Hardware/AWS global infrastructure	<ul> <li>Provide the AWS global infrastructure that ROSA uses to deliver service functionality. For more information about AWS security controls, see         <u>Security of the AWS Infrastructure</u> in the AWS whitepaper.</li> <li>Provide documentation for the customer to manage compliance needs and check their security state in AWS using tools such as AWS Artifact and AWS Security Hub.</li> </ul>	<ul> <li>Customer</li> <li>Configure, manage, and monitor customer applications and data to ensure application and data security controls are properly enforced.</li> <li>Use IAM tools to apply the appropriate permissions to AWS resources in the customer account.</li> </ul>

## **Disaster recovery**

Disaster recovery includes data and configuration backup, data replication and configuration of the disaster recovery environment, and failover on disaster events.

Resource	Service responsibilities	Customer responsibilities
Virtual networking management	<ul> <li>Restore or recreate affected virtual network component s that are necessary for the platform to function.</li> </ul>	<ul> <li>Customer</li> <li>Configure virtual networkin g connections with more than one tunnel where possible for protection against outages.</li> </ul>

Resource	Service responsibilities	Customer responsibilities
		<ul> <li>Maintain failover DNS and load balancing if using a global load balancer with multiple clusters.</li> </ul>
Virtual compute management	Red Hat	Customer
	<ul> <li>Monitor the cluster and replace failed Amazon EC2 control plane or infrastru cture nodes.</li> <li>Provide the ability for the customer to manually or automatically replace failed worker nodes.</li> </ul>	<ul> <li>Replace failed Amazon EC2 worker nodes by editing the machine pool configuration through OpenShift Cluster Manager or the ROSA CLI.</li> </ul>
Virtual storage management	Red Hat	Customer
	<ul> <li>For ROSA clusters created with AWS IAM user credentials, back up all Kubernetes objects on the cluster through hourly, daily, and weekly volume snapshots.</li> </ul>	Back up customer applicati ons and application data.

Resource	Service responsibilities	Customer responsibilities
Resource  AWS software (public AWS services)	AWS  Compute  Provide Amazon EC2 features that support data resiliency such as Amazon EBS snapshots and Amazon EC2 Auto Scaling. For more information, see Resilienc e in Amazon EC2 in the Amazon EC2 User Guide.  Storage  Provide the ability for the ROSA service and customers to back up the Amazon EBS volume on the cluster through Amazon EBS volume snapshots.  For information about Amazon S3 features that support data resiliency, see Resilience in Amazon S3.	Customer  Customer  Configure ROSA multi-AZ clusters to improve fault tolerance and cluster availability.  Provision persistent volumes using the Amazon EBS CSI driver to enable volume snapshots.  Create CSI volume snapshots of Amazon EBS persistent volumes.

Resource	Service responsibilities	Customer responsibilities
Hardware/AWS global infrastructure	<ul> <li>Provide AWS global infrastructure that allows ROSA to scale control plane, infrastructure, and worker nodes across Availability Zones. This functionality enables ROSA to orchestrate automatic failover between zones without interruption.</li> <li>For more information about disaster recovery best practices, see Disaster recovery options in the cloud in the AWS Well-Arch itected Framework.</li> </ul>	Customer  Configure ROSA multi-AZ clusters to improve fault tolerance and cluster availability.

## **Customer responsibilities for data and applications**

The customer is responsible for the applications, workloads, and data that they deploy to Red Hat OpenShift Service on AWS. However, AWS and Red Hat provide various tools to help the customer manage data and applications on the platform.

Resource	How AWS and Red Hat helps	Customer responsibilities
Customer data	Red Hat	Customer
	<ul> <li>Maintain platform- level standards for data encryption as defined by industry security and compliance standards.</li> </ul>	<ul> <li>Maintain responsibility for all customer data stored on the platform and how customer applications consume and expose this data.</li> </ul>

Resource	How AWS and Red Hat helps	Customer responsibilities
	<ul> <li>Provide OpenShift         components to help         manage application data,         such as secrets.</li> <li>Enable integration with         data services such as         Amazon RDS to store and         manage data outside of the         cluster and/or AWS.</li> </ul>	
	<ul> <li>AWS</li> <li>Provide Amazon RDS to allow customers to store and manage data outside of the cluster.</li> </ul>	

Resource	How AWS and Red Hat helps	Customer responsibilities
Customer applications	<ul> <li>Provision clusters with OpenShift components installed so that customers can access the OpenShift and Kubernetes APIs to deploy and manage containerized applications</li> <li>Create clusters with image pull secrets so that customer deployments can pull images from the Red Hat Container Catalog registry.</li> <li>Provide access to OpenShift APIs that a customer can use to set up Operators to add community, third-party, AWS, and Red Hat services to the cluster.</li> <li>Provide storage classes and plugins to support persistent volumes for use with customer applications.</li> <li>Provide a container image registry so customers can securely store applicati on container images on the cluster to deploy and manage applications.</li> </ul>	<ul> <li>Maintain responsibility for customer and third-par ty applications, data, and the complete application lifecycle.</li> <li>If a customer adds Red Hat, community, third-party, their own, or other services to the cluster by using Operators or external images, the customer is responsible for these services and for working with the appropriate provider (including Red Hat) to troubleshoot any issues.</li> <li>Use the provided tools and features to configure and deploy; keep up to date; set up resource requests and limits; size the cluster to have enough resources to run apps; set up permissions; integrate with other services; manage any image streams or templates that the customer deploys; externally serve; save, back up, and restore data; and otherwise manage</li> </ul>

Resource	How AWS and Red Hat helps	Customer responsibilities
	<ul> <li>Provide Amazon EBS to support persistent volumes</li> </ul>	their highly available and resilient workloads.
	for use with customer applications.	<ul> <li>Maintain responsibility for monitoring the applications</li> </ul>
	<ul> <li>Provide Amazon S3 to</li> </ul>	run on Red Hat OpenShift
	support Red Hat provision	Service on AWS, including
	ing of the container image registry.	installing and operating software to gather metrics, create alerts, and protect secrets in the application.

# **Deployment options**

ROSA provides two cluster deployment models: ROSA with hosted control planes (ROSA with HCP) and ROSA classic. With ROSA with HCP, each cluster has a dedicated control plane that is isolated within Red Hat's AWS account and managed by Red Hat. With ROSA classic, cluster control plane infrastructure is hosted in the customer's AWS account.

ROSA with HCP offers a more efficient control plane architecture that helps reduces the AWS infrastructure fees incurred when running ROSA and allows for faster cluster creation times. Both cluster deployment models can be enabled in the AWS ROSA console. You have the choice to select which deployment model you want to use when you provision ROSA clusters using the ROSA CLI.



### Note

ROSA with hosted control planes does not offer compliance certifications or Federal Information Processing Standards (FIPS) at this time. For more information, see Compliance in the Red Hat documentation.

# Differences between ROSA with HCP and ROSA classic

There are several technical differences between ROSA with HCP and ROSA classic.

Deployment options

	ROSA with HCP	ROSA classic
Cluster infrastructure hosting	<ul> <li>Control plane component         <ul> <li>s, such as etcd, API server,</li> <li>and oauth, are hosted</li> <li>on Red Hat-owned and</li> <li>managed AWS accounts.</li> </ul> </li> <li>Worker node infrastructure         <ul> <li>is hosted on the customer's</li> <li>AWS account.</li> </ul> </li> </ul>	<ul> <li>Control plane component s are hosted on the customer's AWS account, alongside infrastructure and worker nodes.</li> </ul>
Provisioning time	Approximately 10 minutes.	Approximately 40 minutes.
Architecture	<ul> <li>Control plane infrastru cture is fully managed by Red Hat. Control plane inf rastructure is not directly available to end customers , except through dedicated and explicitly exposed endpoints.</li> <li>Worker nodes are hosted on the customer's AWS account.</li> </ul>	<ul> <li>Control plane infrastructure is hosted in the customer's AWS account.</li> <li>Worker nodes are hosted on the customer's AWS account.</li> </ul>
Minimum Amazon EC2 footprint	<ul> <li>One cluster requires a minimum of two nodes hosted on the customer's AWS account.</li> </ul>	<ul> <li>One cluster requires a minimum of seven nodes hosted on the customer's AWS account.</li> </ul>
Cluster provisioning	<ul> <li>Provision clusters using the ROSA CLI.</li> <li>Customers provision clusters that deploy the control plane component s into Red Hat's AWS account.</li> </ul>	<ul> <li>Provision clusters using the ROSA CLI or web UI.</li> <li>Cluster control plane, worker nodes, and infrastru cture nodes are provision ed into the customer's AWS account.</li> </ul>

	ROSA with HCP	ROSA classic
	<ul> <li>Customers provision     machine pools that deploy     worker nodes into the     customer's AWS account.</li> </ul>	
Upgrades	<ul> <li>Upgrade control plane and machine pools separately.</li> </ul>	<ul> <li>Entire cluster must be upgraded at the same time.</li> </ul>

	ROSA with HCP	ROSA classic
AWS Regions	<ul> <li>US East (N. Virginia) (useast-1)</li> <li>US East (Ohio) (useast-2)</li> <li>US West (Oregon) (usewest-2)</li> <li>Africa (Cape Town) (afsouth-1)</li> <li>Asia Pacific (Hong Kong) (apeast-1)</li> <li>Asia Pacific (Hyderabad) (apesouth-2)</li> <li>Asia Pacific (Jakarta) (apsoutheast-3)</li> <li>Asia Pacific (Melbourne) (apesoutheast-4)</li> <li>Asia Pacific (Mumbai) (apsouth-1)</li> <li>Asia Pacific (Seoul) (apnortheast-2)</li> <li>Asia Pacific (Singapore) (apsoutheast-1)</li> <li>Asia Pacific (Tokyo) (apnortheast-2)</li> <li>Asia Pacific (Tokyo) (apnortheast-1)</li> <li>Canada (Central) (ca-central-1)</li> <li>Europe (Frankfort) (eu-central-1)</li> </ul>	For AWS Region availabil ity, see Red Hat OpenShift Service on AWS endpoints and quotas in the AWS General Reference Guide.  General Reference Guide.
	• Europe (Ireland) (eu-west-1)	

	ROSA with HCP	ROSA classic
	<ul> <li>Europe (London) (eu-west-2)</li> <li>Europe (Milan) (eu-south-1)</li> <li>Europe (Stockholm) (eunorth-1)</li> <li>Middle East (Bahrain) (mesouth-1)</li> <li>South America (São Paulo) (sa-east-1)</li> </ul>	
Compliance	<ul> <li>Compliance certifications and FIPS not yet available.</li> </ul>	<ul> <li>For compliance informati on, see <u>Compliance</u> in the Red Hat documentation.</li> </ul>

# **Getting started with ROSA**

Red Hat OpenShift Service on AWS (ROSA) is a managed service that you can use to build, scale, and deploy containerized applications with the Red Hat OpenShift enterprise Kubernetes platform on AWS.

# **ROSA cluster deployment models**

ROSA supports two cluster deployment models: ROSA with hosted control planes (ROSA with HCP) and ROSA classic. ROSA with HCP offers a more efficient control plane architecture that reduces the AWS infrastructure costs for ROSA and allows for faster cluster creation times. For more information about ROSA with HCP and ROSA classic, see Deployment options.



#### Note

ROSA with hosted control planes does not offer compliance certifications or FIPS at this time. For more information, see Compliance in the Red Hat documentation.

# **Getting started guides**

There are four getting started guides available for deploying an application to a newly created ROSA cluster. Each tutorial covers the following:

- Enabling the ROSA service and configuring AWS prerequisites
- Creating the necessary IAM roles and policies
- Creating the ROSA cluster
- Creating a cluster administrator for quick cluster access
- Configuring an identity provider
- Granting user access to the cluster
- Deploy an application to the cluster
- Deleting the cluster and cluster resources

# **Getting started with ROSA with HCP**

With ROSA with HCP, you can use AWS STS and the ROSA CLI to create a cluster with the necessary IAM roles and policies attached. For more information about IAM policies for ROSA with HCP, see AWS managed IAM policies for ROSA.

Once the cluster is created, you can deploy public application workloads to the cluster using the Red Hat Hybrid Cloud Console or the OpenShift CLI. For steps to deploy an application to a newly created ROSA with HCP cluster, see <a href="Maintenance">Getting started with ROSA with HCP using the ROSA CLI in auto mode.</a>

# **Getting started with ROSA classic**

With ROSA classic, you can use AWS STS and the ROSA CLI to create a cluster with the necessary IAM roles and policies attached. Once the cluster is created, you can then deploy public application workloads to the cluster using the Red Hat Hybrid Cloud Console or the OpenShift CLI. For steps to get started using the ROSA CLI's automatic cluster creation (auto) mode, see <a href="Getting started">Getting started</a> with ROSA classic using the ROSA CLI in auto mode. For steps to get started using the ROSA CLI's manual cluster creation (manual) mode, see <a href="Getting started">Getting started</a> with ROSA classic using the ROSA CLI in manual mode.

If you require the ROSA classic cluster and application workloads to be private, see <u>Getting started</u> with ROSA classic using AWS PrivateLink.

# Getting started with ROSA with HCP using the ROSA CLI in automode

The following sections describe how to get started with ROSA with hosted control planes (ROSA with HCP) using AWS STS and the ROSA CLI. For more information about ROSA with HCP, see <u>Deployment options</u>.

The ROSA CLI uses auto mode or manual mode to create the IAM resources and OpenID Connect (OIDC) configuration required to create a ROSA cluster. auto mode automatically creates the required IAM roles and policies and OIDC provider. manual mode outputs the AWS CLI commands that are needed to create the IAM resources manually. By using manual mode, you can review the generated AWS CLI commands before running them manually. With manual mode, you can also pass the commands to another administrator or group in your organization so they can create the resources.

The procedures in this document use the auto mode of the ROSA CLI to create the required IAM resources and OIDC configuration for ROSA with HCP. For more options to get started, see <u>Getting</u> started with ROSA.

#### **Topics**

- Prerequisites
- Step 1: Enable ROSA and configure prerequisites
- Step 2: Create Amazon VPC architecture for ROSA with HCP clusters
- Step 3: Create the required IAM roles and OpenID Connect configuration
- Step 4: Create a ROSA with HCP cluster with AWS STS and the ROSA CLI auto mode
- Step 5: Configure an identity provider and grant cluster access
- Step 6: Grant user access to a cluster
- Step 7: Grant administrator permissions to a user
- Step 8: Access a cluster through the Red Hat Hybrid Cloud Console
- Step 9: Deploy an application from the Developer Catalog
- Step 10: Delete a cluster and AWS STS resources

# **Prerequisites**

Before getting started, make sure you completed these actions:

- Install and configure the latest AWS CLI. For more information, see <u>Installing or updating the</u> latest version of the AWS CLI.
- Install and configure the latest ROSA CLI and OpenShift Container Platform CLI. For more information, see Getting started with the ROSA CLI.
- Service Quotas must have the required service quotas set for Amazon EC2, Amazon VPC, Amazon EBS, and Elastic Load Balancing that are needed to create and run a ROSA cluster. AWS or Red Hat may request service quota increases on your behalf as required for issue resolution. To view the required quotas, see Red Hat OpenShift Service on AWS endpoints and quotas in the AWS General Reference.
- To receive AWS support for ROSA, you must enable AWS Business, Enterprise On-Ramp, or Enterprise support plans. Red Hat may request AWS support on your behalf as required for issue resolution. For more information, see <u>Support for ROSA</u>. To enable AWS Support, see the <u>AWS</u> <u>Support page</u>.

Prerequisites 40

- If you're using AWS Organizations to manage the AWS accounts that host the ROSA service, the
  organization's service control policy (SCP) must be configured to allow Red Hat to perform policy
  actions that's listed in the SCP without restriction. For more information, see the ROSA SCP
  troubleshooting documentation. For more information about SCPs, see Service control policies
  (SCPs).
- If deploying a ROSA cluster with AWS STS into an enabled AWS Region that's disabled by default, you must update the security token to version 2 for all the Regions in the AWS account with the following command.

```
aws iam set-security-token-service-preferences --global-endpoint-token-version v2Token
```

For more information about enabling Regions, see <u>Managing AWS Regions</u> in the *AWS General Reference*.

# **Step 1: Enable ROSA and configure prerequisites**

To create a ROSA cluster, you must first enable the ROSA service in the AWS ROSA console. The AWS ROSA console verifies if your AWS account has the necessary AWS Marketplace permissions, service quotas, and the Elastic Load Balancing (ELB) service-linked role named AWSServiceRoleForElasticLoadBalancing. If any of these prerequisites are missing, the console provides guidance on how to configure your account to meet the prerequisites.

- 1. Navigate to the ROSA console.
- Choose Get started.
- 3. On the **Verify ROSA prerequisites** page, select **I agree to share my contact information with Red Hat**.
- 4. Choose Enable ROSA.
- 5. Once the page has verified your service quotas meet ROSA prerequisites and the ELB service-linked role is created, open a new terminal session to create your first ROSA cluster using the ROSA CLI.

# Step 2: Create Amazon VPC architecture for ROSA with HCP clusters

To create a ROSA with HCP cluster, you must first configure your own Amazon VPC architecture to deploy your solution into. ROSA with HCP requires that customers configure at least one public

and private subnet per Availability Zone used to create clusters. For single-AZ clusters, only use Availability Zone is used. For multi-AZ clusters, three Availability Zones are needed.



### Important

If Amazon VPC requirements are not met, cluster creation fails.

The following procedure uses the AWS CLI to create both a public and private subnet into a single Availability Zone for a Single-AZ cluster. All cluster resources are in the private subnet. The public subnet routes outbound traffic by using a NAT gateway to the internet.

This example uses the CIDR block 10.0.0.0/16 for the Amazon VPC. However, you can choose a different CIDR block. For more information, see VPC sizing.

1. Set an environment variable for the cluster name by running the following command.

```
ROSA_CLUSTER_NAME=rosa-hcp
```

2. Create a VPC with a 10.0.0.0/16 CIDR block.

```
aws ec2 create-vpc --cidr-block 10.0.0/16 --query Vpc.VpcId --output text
```

The preceding command returns the ID of the new VPC. The following is an example output.

```
vpc-0410832ee325aafea
```

3. Using the VPC ID from the previous step, tag the VPC using the ROSA\_CLUSTER\_NAME variable.

```
aws ec2 create-tags --resources <VPC_ID_VALUE> --tags Key=Name, Value=
$ROSA_CLUSTER_NAME
```

4. Enable DNS hostname support on the VPC.

```
aws ec2 modify-vpc-attribute --vpc-id <VPC_ID_VALUE> --enable-dns-hostnames
```

5. Create a public subnet in the VPC with a 10.0.1.0/24 CIDR block, specifying the Availability Zone where the resource should be created.

#### 

When creating subnets, make sure that subnets are created to an Availability Zone that has ROSA instance types available. If you don't choose a specific Availability Zone, the subnet is created in any one of the Availability Zones in the AWS Region that you specify. To specify a specific Availability Zone, use the --availability zone argument in the create-subnet command. You can use the rosa list instance-types command to list all ROSA instance types available. To check if an instance type is available for a given Availability Zone, use the following command.

aws ec2 describe-instance-type-offerings --location-type availability-zone -filters Name=location, Values=<availability\_zone> --region <region> --output text | egrep "<instance\_type>"

#### 

ROSA with HCP requires that customers configure at least one public and private subnet per Availability Zone used to create clusters. For single-AZ clusters, only one Availability Zone is needed. For multi-AZ clusters, three Availability Zones are needed. If these requirements are not met, cluster creation fails.

```
aws ec2 create-subnet --vpc-id <VPC_ID_VALUE> --cidr-block 10.0.1.0/24 --
availability-zone <AZ_NAME> --query Subnet.SubnetId --output text
```

The preceding command returns the ID of the new subnet. The following is an example output.

```
subnet-0b6a7e8cbc8b75920
```

6. Using the subnet ID from the previous step, tag the subnet using the ROSA\_CLUSTER\_NAME public variable.

```
aws ec2 create-tags --resources <PUBLIC_SUBNET_ID> --tags Key=Name, Value=
$ROSA_CLUSTER_NAME-public
```

7. Create a private subnet in the VPC with a 10.0.0.0/24 CIDR block, specifying the same Availability Zone that the public subnet deployed into.

```
aws ec2 create-subnet --vpc-id <VPC_ID_VALUE> --cidr-block 10.0.0.0/24 --availability-zone <AZ_NAME> --query Subnet.SubnetId --output text
```

The preceding command returns the ID of the new subnet. The following is an example output.

```
subnet-0b6a7e8cbc8b75920
```

8. Using the subnet ID from the previous step, tag the subnet using the ROSA\_CLUSTER\_NAME-private variable.

```
aws ec2 create-tags --resources <PRIVATE_SUBNET_ID> --tags Key=Name, Value=
$ROSA_CLUSTER_NAME-private
```

9. Create an internet gateway for outbound traffic and attach it to the VPC.

10. Tag the internet gateway with the ROSA\_CLUSTER\_NAME variable.

```
aws ec2 create-tags --resources <IG_ID_VALUE> --tags Key=Name, Value=
$ROSA_CLUSTER_NAME
```

11Create a route table for outbound traffic, associate it to the public subnet, and configure traffic to route to the internet gateway.

12.Tag the public route table with the ROSA\_CLUSTER\_NAME variable and verify that the route table was properly configured.

```
aws ec2 create-tags --resources <PUBLIC_RT_ID> --tags Key=Name, Value=
$ROSA_CLUSTER_NAME

aws ec2 describe-route-tables --route-table-id <PUBLIC_RT_ID>
```

13Create a NAT gateway in the public subnet with an elastic IP address to enable traffic to the private subnet.

```
aws ec2 allocate-address --domain vpc --query AllocationId --output text
aws ec2 create-nat-gateway --subnet-id <PUBLIC_SUBNET_ID> --allocation-id
<EIP_ADDRESS> --query NatGateway.NatGatewayId --output text
```

14Tag the NAT gateway and elastic IP address with the \$ROSA\_CLUSTER\_NAME variable.

```
aws ec2 create-tags --resources <EIP_ADDRESS> --resources <NAT_GATEWAY_ID> --tags
Key=Name, Value=$ROSA_CLUSTER_NAME
```

15Create a route table for private subnet traffic, associate it to the private subnet, and configure traffic to route to the NAT gateway.

```
aws ec2 create-route-table --vpc-id <VPC_ID_VALUE> --query RouteTable.RouteTableId --
output text

aws ec2 associate-route-table --subnet-id <PRIVATE_SUBNET_ID> --route-table-id
<PRIVATE_RT_ID>

aws ec2 create-route --route-table-id <PRIVATE_RT_ID> --destination-cidr-block
0.0.0.0/0 --gateway-id <NAT_GATEWAY_ID>
```

16.Tag the private route table and elastic IP address with the \$ROSA\_CLUSTER\_NAME-private variable.

```
aws ec2 create-tags --resources <PRIVATE_RT_ID> <EIP_ADDRESS> --tags Key=Name, Value=
$ROSA_CLUSTER_NAME-private
```

# Step 3: Create the required IAM roles and OpenID Connect configuration

Before creating a ROSA with HCP cluster, you must create the necessary IAM roles and policies and the OpenID Connect (OIDC) configuration. For more information about IAM roles and policies for ROSA with HCP, see AWS managed IAM policies for ROSA.

This procedure uses the auto mode of the ROSA CLI to automatically create the OIDC configuration necessary to create a ROSA with HCP cluster.

1. Create the required IAM account roles and policies.

```
rosa create account-roles --force-policy-creation
```

The --force-policy-creation parameter updates any existing roles and policies that are present. If no roles and policies are present, the command creates these resources instead.

## Note

If your offline access token has expired, the ROSA CLI outputs an error message stating that your authorization token needs updated. For steps to troubleshoot, see <a href="Troubleshoot ROSA CLI expired offline access tokens">Troubleshoot ROSA CLI expired offline access tokens</a>.

2. Create the OpenID Connect (OIDC) configuration that enables user authentication to the cluster. This configuration is registered to be used with OpenShift Cluster Manager (OCM).

```
rosa create oidc-config --mode=auto
```

- 3. Copy the OIDC config ID provided in the ROSA CLI output. The OIDC config ID needs to be provided later to create the ROSA with HCP cluster.
- 4. To verify the OIDC configurations available for clusters associated with your user organization, run the following command.

```
rosa list oidc-config
```

5. Create the required IAM operator roles, replacing <OIDC\_CONFIG\_ID> with the OIDC config ID copied previously.

#### Example



#### Important

You must supply a prefix in <PREFIX NAME> when creating the Operator roles. Failing to do so produces an error.

```
rosa create operator-roles --prefix <PREFIX_NAME> --oidc-config-id <OIDC_CONFIG_ID>
 --hosted-cp
```

6. To verify the IAM operator roles were created, run the following command:

rosa list operator-roles

# Step 4: Create a ROSA with HCP cluster with AWS STS and the ROSA CLI auto mode

You can create a ROSA with HCP cluster using AWS Security Token Service (AWS STS) and the auto mode that's provided in the ROSA CLI. You have the option to create a cluster with a public API and Ingress or a private API and Ingress.

You can create a cluster with a single Availability Zone (Single-AZ) or multiple Availability Zones (Multi-AZ). In either case, your machine's CIDR value must match your VPC's CIDR value.

The following procedure uses the rosa create cluster --hosted-cp command to create a Single-AZ ROSA with HCP cluster. To create a Multi-AZ cluster, specify multi-az in the command and the private subnet IDs for each private subnet you want you to deploy to.

- 1. Create a ROSA with HCP cluster with one of the following commands.
  - Create a ROSA with HCP cluster with a public API and Ingress, specifying the cluster name, operator role prefix, OIDC config ID, and public and private subnet IDs.

rosa create cluster --cluster-name=<CLUSTER\_NAME> --sts --mode=auto --hosted-cp -operator-roles-prefix <OPERATOR\_ROLE\_PREFIX> --oidc-config-id <OIDC\_CONFIG\_ID> -subnet-ids=<PUBLIC\_SUBNET\_ID>,<PRIVATE\_SUBNET\_ID>

 Create a ROSA with HCP cluster with a private API and Ingress, specifying the cluster name, operator role prefix, OIDC config ID, and private subnet IDs.

```
rosa create cluster --private --cluster-name=<CLUSTER_NAME> --sts --mode=auto --
hosted-cp --subnet-ids=<PRIVATE_SUBNET_ID>
```

2. Check the status of your cluster.

```
rosa describe cluster -c <CLUSTER_NAME>
```



### Note

If the creation process fails or the State field doesn't change to a ready status after 10 minutes, see Troubleshoot ROSA cluster creation issues.

To contact AWS Support or Red Hat support for assistance, see Support for ROSA.

3. Track the progress of the cluster creation by watching the OpenShift installer logs.

```
rosa logs install -c <CLUSTER_NAME> --watch
```

# Step 5: Configure an identity provider and grant cluster access

ROSA includes a built-in OAuth server. After your cluster is created, you must configure OAuth to use an identity provider. You can then add users to your configured identity provider to grant them access to your cluster. You can grant these users cluster-admin or dedicated-admin permissions as required.

You can configure different identity provider types for your ROSA cluster. Supported types include GitHub, GitHub Enterprise, GitLab, Google, LDAP, OpenID Connect, and HTPasswd identity providers.

#### Important

The HTPasswd identity provider is included only to enable a single, static administrator user to be created. HTPasswd isn't supported as a general-use identity provider for ROSA.

The following procedure configures a GitHub identity provider as an example. For instructions on how to configure each of the supported identity provider types, see <u>Configuring identity providers</u> for AWS STS.

- 1. Navigate to github.com and log in to your GitHub account.
- 2. If you don't have a GitHub organization to use for identity provisioning for your cluster, create one. For more information, see the steps in the GitHub documentation.
- 3. Using the ROSA CLI's interactive mode, configure an identity provider for your cluster.

```
rosa create idp --cluster=<CLUSTER_NAME> --interactive
```

4. Follow the configuration prompts in the output to restrict cluster access to members of your GitHub organization.

```
I: Interactive mode enabled.
Any optional fields can be left empty and a default will be selected.
? Type of identity provider: github
? Identity provider name: github-1
? Restrict to members of: organizations
? GitHub organizations: <GITHUB_ORG_NAME>
? To use GitHub as an identity provider, you must first register the application:
  - Open the following URL:
    https://github.com/organizations/<GITHUB_ORG_NAME>/settings/
applications/new?oauth_application%5Bcallback_url%5D=https%3A%2F%2Foauth-
openshift.apps.<CLUSTER_NAME>/<RANDOM_STRING>.p1.openshiftapps.com%2Foauth2callback
%2Fgithub-1&oauth_application%5Bname%5D=<CLUSTER_NAME>&oauth_application
%5Burl%5D=https%3A%2F%2Fconsole-openshift-console.apps.<CLUSTER_NAME>/
<RANDOM_STRING>.p1.openshiftapps.com
  - Click on 'Register application'
. . .
```

- 5. Open the URL in the output, replacing <GITHUB\_ORG\_NAME> with the name of your GitHub organization.
- 6. On the GitHub web page, choose **Register application** to register a new OAuth application in your GitHub organization.
- 7. Use the information from the GitHub OAuth page to populate the remaining rosa create idp interactive prompts by running the following command. Replace <GITHUB\_CLIENT\_ID> and <GITHUB\_CLIENT\_SECRET> with the credentials from your GitHub OAuth application.

```
•••
```

```
? Client ID: <GITHUB_CLIENT_ID>
? Client Secret: [? for help] <GITHUB_CLIENT_SECRET>
? GitHub Enterprise Hostname (optional):
? Mapping method: claim
I: Configuring IDP for cluster '<CLUSTER_NAME>'
I: Identity Provider 'github-1' has been created.
    It will take up to 1 minute for this configuration to be enabled.
    To add cluster administrators, see 'rosa grant user --help'.
    To login into the console, open https://console-openshift-
console.apps.<CLUSTER_NAME>.<RANDOM_STRING>.p1.openshiftapps.com and click on github-1.
```

## Note

It might take approximately two minutes for the identity provider configuration to become active. If you configured a cluster-admin user, you can run oc get pods - n openshift-authentication --watch to watch the OAuth pods redeploy with the updated configuration.

8. Verify that the identity provider is configured correctly.

```
rosa list idps --cluster=<CLUSTER_NAME>
```

# Step 6: Grant user access to a cluster

You can grant a user access to your cluster by adding them to the configured identity provider.

The following procedure adds a user to a GitHub organization that's configured for identity provisioning to the cluster.

- 1. Navigate to github.com and log in to your GitHub account.
- 2. Invite users that require cluster access to your GitHub organization. For more information, see Inviting users to join your organization in the GitHub documentation.

# Step 7: Grant administrator permissions to a user

After you add a user to your configured identity provider, you can grant the user cluster-admin or dedicated-admin permissions for your cluster.

## Configure cluster-admin permissions

1. Grant the cluster-admin permissions by running the following command. Replace <IDP\_USER\_NAME> and <CLUSTER\_NAME> with your user and cluster name.

```
rosa grant user cluster-admin --user=<IDP_USER_NAME> --cluster=<CLUSTER_NAME>
```

2. Verify that the user is listed as a member of the cluster-admins group.

```
rosa list users --cluster=<CLUSTER_NAME>
```

## Configure dedicated-admin permissions

Grant the dedicated-admin permissions by using the following command. Replace
 IDP\_USER\_NAME> and <CLUSTER\_NAME> with your user and cluster name by running the following command.

```
rosa grant user dedicated-admin --user=<IDP_USER_NAME> --cluster=<CLUSTER_NAME>
```

2. Verify that the user is listed as a member of the cluster-admins group.

```
rosa list users --cluster=<CLUSTER_NAME>
```

# Step 8: Access a cluster through the Red Hat Hybrid Cloud Console

Log in to your cluster through the Red Hat Hybrid Cloud Console.

1. Obtain the console URL for your cluster using the following command. Replace <CLUSTER\_NAME> with the name of your cluster.

```
rosa describe cluster -c <CLUSTER_NAME> | grep Console
```

2. Navigate to the console URL in the output and log in.

In the **Log in with...** dialog, choose the identity provider name and complete any authorization requests presented by your provider.

# Step 9: Deploy an application from the Developer Catalog

From the Red Hat Hybrid Cloud Console, you can deploy a Developer Catalog test application and expose it with a route.

- 1. Navigate to Red Hat Hybrid Cloud Console and choose the cluster you want to deploy the app into.
- 2. On the cluster's page, choose **Open console**.
- 3. In the Administrator perspective, choose Home > Projects > Create Project.
- 4. Enter a name for your project and optionally add a **Display Name** and **Description**.
- 5. Choose **Create** to create the project.
- 6. Switch to the **Developer** perspective and choose **+Add**. Make sure that the selected project is the one that was just created.
- 7. In the **Developer Catalog** dialog, choose **All services**.
- 8. In the **Developer Catalog** page, choose **Languages** > **JavaScript** from the menu.
- 9. Choose Node.is, and then choose Create Application to open the Create Source-to-Image **Application** page.



#### Note

You might need to choose **Clear All Filters** to display the **Node.js** option.

10In the **Git** section, choose **Try Sample**.

11In the **Name** field, add a unique name.

12Choose Create.



#### Note

The new application takes several minutes to deploy.

13When the deployment is complete, choose the route URL for the application.

A new tab in the browser opens with a message that's similar to the following.

Welcome to your Node.js application on OpenShift

14(Optional) Delete the application and clean up resources:

- a. In the **Administrator** perspective, choose **Home** > **Projects**.
- b. Open the action menu for your project and choose **Delete Project**.

# **Step 10: Delete a cluster and AWS STS resources**

You can use the ROSA CLI to delete a cluster that uses AWS Security Token Service (AWS STS). You can also use the ROSA CLI to delete the IAM roles and OIDC provider created by ROSA. To delete the IAM policies created by ROSA, you can use the IAM console.



#### Important

IAM roles and policies created by ROSA might be used by other ROSA clusters in the same account.

 Delete the cluster and watch the logs. Replace < CLUSTER\_NAME > with the name or ID of your cluster.

```
rosa delete cluster --cluster=<CLUSTER_NAME> --watch
```

#### 

You must wait for the cluster to delete completely before you remove the IAM roles, policies, and OIDC provider. The account IAM roles are required to delete the resources created by the installer. The operator IAM roles are required to clean up the resources created by the OpenShift operators. The operators use the OIDC provider to authenticate.

2. Delete the OIDC provider that the cluster operators use to authenticate by running the following command.

```
rosa delete oidc-provider -c <CLUSTER_ID> --mode auto
```

3. Delete the cluster-specific operator IAM roles.

```
rosa delete operator-roles -c <CLUSTER_ID> --mode auto
```

4. Delete the account IAM roles using the following command. Replace <PREFIX> with the prefix of the account IAM roles to delete. If you specified a custom prefix when creating the account IAM roles, specify the default ManagedOpenShift prefix.

```
rosa delete account-roles --prefix <PREFIX> --mode auto
```

- 5. Delete the IAM policies created by ROSA.
  - a. Log in to the IAM console.
  - b. On the left menu under Access management, choose Policies.
  - c. Select the policy that you want to delete and choose **Actions** > **Delete**.
  - d. Enter the policy name and choose **Delete**.
  - e. Repeat this step to delete each of the IAM policies for the cluster.

# Getting started with ROSA classic using the ROSA CLI in auto mode

The following sections describe how to get started with ROSA classic using AWS STS and the ROSA CLI. For more information about ROSA classic, see Deployment options.

The ROSA CLI uses auto mode or manual mode to create the IAM resources required to provision a ROSA cluster. auto mode immediately creates the required IAM roles and policies and an OpenID Connect (OIDC) provider. manual mode outputs the AWS CLI commands that are needed to create the IAM resources. By using manual mode, you can review the generated AWS CLI commands before running them manually. With manual mode, you can also pass the commands to another administrator or group in your organization so they can create the resources.

The procedures in this document use the auto mode of the ROSA CLI to create the required IAM resources for ROSA classic. For more options to get started, see <u>Getting started with ROSA</u>.

## **Topics**

- Prerequisites
- Step 1: Enable ROSA and configure prerequisites
- Step 2: Create a ROSA classic cluster with AWS STS and the ROSA CLI auto mode
- Step 3: Configure an identity provider and grant cluster access
- Step 4: Grant user access to a cluster

- Step 5: Grant administrator permissions to a user
- Step 6: Access a cluster through the web console
- Step 7: Deploy an application from the Developer Catalog
- Step 8: Revoke administrator permissions and user access
- Step 9: Delete a cluster and AWS STS resources

# **Prerequisites**

Before getting started, make sure you completed these actions:

- Install and configure the latest AWS CLI. For more information, see <u>Installing or updating the</u> latest version of the AWS CLI.
- Install and configure the latest ROSA CLI and OpenShift Container Platform CLI. For more information, see Getting started with the ROSA CLI.
- Service Quotas must have the required service quotas set for Amazon EC2, Amazon VPC, Amazon EBS, and Elastic Load Balancing that are needed to create and run a ROSA cluster. AWS or Red Hat may request service quota increases on your behalf as required for issue resolution. To view the required quotas, see Red Hat OpenShift Service on AWS endpoints and quotas in the AWS General Reference.
- To receive AWS support for ROSA, you must enable AWS Business, Enterprise On-Ramp, or Enterprise support plans. Red Hat may request AWS support on your behalf as required for issue resolution. For more information, see <u>Support for ROSA</u>. To enable AWS Support, see the <u>AWS</u> Support page.
- If you're using AWS Organizations to manage the AWS accounts that host the ROSA service, the
  organization's service control policy (SCP) must be configured to allow Red Hat to perform policy
  actions that's listed in the SCP without restriction. For more information, see the ROSA SCP
  troubleshooting documentation. For more information about SCPs, see Service control policies
  (SCPs).
- If deploying a ROSA cluster with AWS STS into an enabled AWS Region that's disabled by default, you must update the security token to version 2 for all the Regions in the AWS account with the following command.

aws iam set-security-token-service-preferences --global-endpoint-token-version v2Token

Prerequisites 55

For more information about enabling Regions, see <u>Managing AWS Regions</u> in the *AWS General Reference*.

# Step 1: Enable ROSA and configure prerequisites

To create a ROSA cluster, you must first enable the ROSA service in the AWS ROSA console and verify that AWS prerequisites have been met. The AWS ROSA console verifies if your AWS account has the necessary AWS Marketplace permissions, service quotas, and the Elastic Load Balancing (ELB) service-linked role named AWSServiceRoleForElasticLoadBalancing. If any of these prerequisites are missing, the console provides guidance on how to configure your account to meet the prerequisites.

- 1. Navigate to the ROSA console.
- 2. Choose Get started.
- 3. On the **Verify ROSA prerequisites** page, select **I agree to share my contact information with Red Hat**.
- 4. Choose Enable ROSA.
- 5. Once the page has verified your service quotas meet ROSA prerequisites and the ELB service-linked role is created, open a new terminal session to create your first ROSA classic cluster using the ROSA CLI.

# Step 2: Create a ROSA classic cluster with AWS STS and the ROSA CLI auto mode

You can create a ROSA classic cluster using AWS Security Token Service (AWS STS) and the auto mode that's provided in the ROSA CLI.

1. Create the required IAM account roles and policies.

rosa create account-roles --mode auto



#### Note

If your offline access token has expired, the ROSA CLI outputs an error message stating that your authorization token needs updated. For steps to troubleshoot, see Troubleshoot ROSA CLI expired offline access tokens.

2. Create a cluster with AWS STS using the defaults in the ROSA CLI's auto mode. When using the defaults, the latest stable OpenShift version is installed.

rosa create cluster --cluster-name <CLUSTER\_NAME> --sts --mode auto



#### Note

When you specify --mode auto, the rosa create cluster command creates the cluster-specific operator IAM roles and the OIDC provider automatically. The operators use the OIDC provider to authenticate.

3. Check the status of your cluster.

rosa describe cluster -c <CLUSTER\_NAME>



#### Note

If the provisioning process fails or the State field doesn't change to a ready status after 40 minutes, see Troubleshoot ROSA cluster provisioning issues.

To contact AWS Support or Red Hat support for assistance, see Support for ROSA.

4. Track the progress of the cluster creation by watching the OpenShift installer logs.

rosa logs install -c <CLUSTER\_NAME> --watch

# Step 3: Configure an identity provider and grant cluster access

ROSA includes a built-in OAuth server. After your cluster is created, you must configure OAuth to use an identity provider. You can then add users to your configured identity provider to grant them access to your cluster. You can grant these users cluster-admin or dedicated-admin permissions as required.

You can configure different identity provider types for your ROSA cluster. Supported types include GitHub, GitHub Enterprise, GitLab, Google, LDAP, OpenID Connect, and HTPasswd identity providers.

#### Important

The HTPasswd identity provider is included only to enable a single, static administrator user to be created. HTPasswd isn't supported as a general-use identity provider for ROSA.

The following procedure configures a GitHub identity provider as an example. For instructions on how to configure each of the supported identity provider types, see Configuring identity providers for AWS STS.

- 1. Navigate to github.com and log in to your GitHub account.
- 2. If you don't have a GitHub organization to use for identity provisioning for your cluster, create one. For more information, see the steps in the GitHub documentation.
- Using the ROSA CLI's interactive mode, configure an identity provider for your cluster.

```
rosa create idp --cluster=<CLUSTER_NAME> --interactive
```

4. Follow the configuration prompts in the output to restrict cluster access to members of your GitHub organization.

```
I: Interactive mode enabled.
Any optional fields can be left empty and a default will be selected.
? Type of identity provider: github
? Identity provider name: github-1
? Restrict to members of: organizations
? GitHub organizations: <GITHUB_ORG_NAME>
? To use GitHub as an identity provider, you must first register the application:
  - Open the following URL:
    https://github.com/organizations/<GITHUB_ORG_NAME>/settings/
applications/new?oauth_application%5Bcallback_url%5D=https%3A%2F%2Foauth-
openshift.apps.<CLUSTER_NAME>/<RANDOM_STRING>.p1.openshiftapps.com%2Foauth2callback
%2Fgithub-1&oauth_application%5Bname%5D=<CLUSTER_NAME>&oauth_application
```

```
%5Burl%5D=https%3A%2F%2Fconsole-openshift-console.apps.<CLUSTER_NAME>/
<RANDOM_STRING>.p1.openshiftapps.com
  - Click on 'Register application'
...
```

- 5. Open the URL in the output, replacing <GITHUB\_ORG\_NAME> with the name of your GitHub organization.
- 6. On the GitHub web page, choose **Register application** to register a new OAuth application in your GitHub organization.
- 7. Use the information from the GitHub OAuth page to populate the remaining rosa create idp interactive prompts by running the following command. Replace <GITHUB\_CLIENT\_ID> and <GITHUB\_CLIENT\_SECRET> with the credentials from your GitHub OAuth application.

```
...
? Client ID: <GITHUB_CLIENT_ID>
? Client Secret: [? for help] <GITHUB_CLIENT_SECRET>
? GitHub Enterprise Hostname (optional):
? Mapping method: claim
I: Configuring IDP for cluster '<CLUSTER_NAME>'
I: Identity Provider 'github-1' has been created.
    It will take up to 1 minute for this configuration to be enabled.
    To add cluster administrators, see 'rosa grant user --help'.
    To login into the console, open https://console-openshift-
console.apps.<CLUSTER_NAME>.<RANDOM_STRING>.pl.openshiftapps.com and click on github-1.
```

## Note

It might take approximately two minutes for the identity provider configuration to become active. If you configured a cluster-admin user, you can run oc get pods - n openshift-authentication --watch to watch the OAuth pods redeploy with the updated configuration.

8. Verify that the identity provider is configured correctly.

```
rosa list idps --cluster=<CLUSTER_NAME>
```

# Step 4: Grant user access to a cluster

You can grant a user access to your cluster by adding them to the configured identity provider.

The following procedure adds a user to a GitHub organization that's configured for identity provisioning to the cluster.

- 1. Navigate to github.com and log in to your GitHub account.
- 2. Invite users that require cluster access to your GitHub organization. For more information, see Inviting users to join your organization in the GitHub documentation.

# Step 5: Grant administrator permissions to a user

After you add a user to your configured identity provider, you can grant the user cluster-admin or dedicated-admin permissions for your cluster.

# Configure cluster-admin permissions

1. Grant the cluster-admin permissions by running the following command. Replace <IDP\_USER\_NAME> and <CLUSTER\_NAME> with your user and cluster name.

```
rosa grant user cluster-admin --user=<IDP_USER_NAME> --cluster=<CLUSTER_NAME>
```

2. Verify that the user is listed as a member of the cluster-admins group.

```
rosa list users --cluster=<CLUSTER_NAME>
```

## Configure dedicated-admin permissions

 Grant the dedicated-admin permissions by using the following command. Replace <IDP\_USER\_NAME> and <CLUSTER\_NAME> with your user and cluster name by running the following command.

```
rosa grant user dedicated-admin --user=<IDP_USER_NAME> --cluster=<CLUSTER_NAME>
```

2. Verify that the user is listed as a member of the cluster-admins group.

```
rosa list users --cluster=<CLUSTER_NAME>
```

# Step 6: Access a cluster through the web console

After you create a cluster administrator user or added a user to your configured identity provider, you can log in to your cluster through the Red Hat Hybrid Cloud Console.

Obtain the console URL for your cluster using the following command. Replace
 <CLUSTER\_NAME> with the name of your cluster.

```
rosa describe cluster -c <CLUSTER_NAME> | grep Console
```

- 2. Navigate to the console URL in the output and log in.
  - If you created a cluster-admin user, log in using the provided credentials.
  - If you configured an identity provider for your cluster, choose the identity provider name in the **Log in with...** dialog and complete any authorization requests presented by your provider.

# Step 7: Deploy an application from the Developer Catalog

From the Red Hat Hybrid Cloud Console, you can deploy a Developer Catalog test application and expose it with a route.

- Navigate to Red Hat Hybrid Cloud Console and choose the cluster you want to deploy the app into.
- 2. On the cluster's page, choose **Open console**.
- 3. In the **Administrator** perspective, choose **Home** > **Projects** > **Create Project**.
- 4. Enter a name for your project and optionally add a **Display Name** and **Description**.
- 5. Choose **Create** to create the project.
- 6. Switch to the **Developer** perspective and choose **+Add**. Make sure that the selected project is the one that was just created.
- 7. In the **Developer Catalog** dialog, choose **All services**.
- 8. In the **Developer Catalog** page, choose **Languages** > **JavaScript** from the menu.
- 9. Choose **Node.js**, and then choose **Create Application** to open the **Create Source-to-Image Application** page.



#### Note

You might need to choose **Clear All Filters** to display the **Node.js** option.

10In the **Git** section, choose **Try Sample**.

11In the **Name** field, add a unique name.

12Choose Create.



#### Note

The new application takes several minutes to deploy.

13. When the deployment is complete, choose the route URL for the application.

A new tab in the browser opens with a message that's similar to the following.

Welcome to your Node.js application on OpenShift

14(Optional) Delete the application and clean up resources:

- a. In the **Administrator** perspective, choose **Home** > **Projects**.
- b. Open the action menu for your project and choose **Delete Project**.

# Step 8: Revoke administrator permissions and user access

You can revoke cluster-admin or dedicated-admin permissions from a user by using the ROSA CLI.

To revoke access from a user, you must remove the user from your configured identity provider.

# Revoke cluster-admin permissions from a user

 Revoke the cluster-admin permissions using the following command. Replace <IDP\_USER\_NAME> and <CLUSTER\_NAME> with your user and cluster name.

```
rosa revoke user cluster-admin --user=<IDP_USER_NAME> --cluster=<CLUSTER_NAME>
```

2. Verify that the user isn't listed as a member of the cluster-admins group.

rosa list users --cluster=<CLUSTER\_NAME>

# Revoke dedicated-admin permissions from a user

1. Revoke the dedicated-admin permissions by using the following command. Replace <IDP\_USER\_NAME> and <CLUSTER\_NAME> with your user and cluster name.

```
rosa revoke user dedicated-admin --user=<IDP_USER_NAME> --cluster=<CLUSTER_NAME>
```

2. Verify that the user isn't listed as a member of the dedicated-admins group.

```
rosa list users --cluster=<CLUSTER_NAME>
```

#### Revoke user access to a cluster

You can revoke cluster access for an identity provider user by removing them from the configured identity provider.

You can configure different types of identity providers for your cluster. The following procedure revokes cluster access for a member of a GitHub organization.

- 1. Navigate to github.com and log in to your GitHub account.
- 2. Remove the user from your GitHub organization. For more information, see Removing a member from your organization in the GitHub documentation.

# **Step 9: Delete a cluster and AWS STS resources**

You can use the ROSA CLI to delete a cluster that uses AWS Security Token Service (AWS STS). You can also use the ROSA CLI to delete the IAM roles and OIDC provider created by ROSA. To delete the IAM policies created by ROSA, you can use the IAM console.

#### Important

IAM roles and policies created by ROSA might be used by other ROSA clusters in the same account.

 Delete the cluster and watch the logs. Replace < CLUSTER NAME > with the name or ID of your cluster.

```
rosa delete cluster --cluster=<CLUSTER_NAME> --watch
```



#### Important

You must wait for the cluster to delete completely before you remove the IAM roles, policies, and OIDC provider. The account IAM roles are required to delete the resources created by the installer. The operator IAM roles are required to clean up the resources created by the OpenShift operators. The operators use the OIDC provider to authenticate.

2. Delete the OIDC provider that the cluster operators use to authenticate by running the following command.

```
rosa delete oidc-provider -c <CLUSTER_ID> --mode auto
```

3. Delete the cluster-specific operator IAM roles.

```
rosa delete operator-roles -c <CLUSTER_ID> --mode auto
```

4. Delete the account IAM roles using the following command. Replace <PREFIX> with the prefix of the account IAM roles to delete. If you specified a custom prefix when creating the account IAM roles, specify the default ManagedOpenShift prefix.

```
rosa delete account-roles --prefix <PREFIX> --mode auto
```

- 5. Delete the IAM policies created by ROSA.
  - a. Log in to the IAM console.
  - b. On the left menu under Access management, choose Policies.
  - c. Select the policy that you want to delete and choose **Actions** > **Delete**.
  - d. Enter the policy name and choose **Delete**.
  - e. Repeat this step to delete each of the IAM policies for the cluster.

# Getting started with ROSA classic using the ROSA CLI in manual mode

The following sections describe how to get started with ROSA classic using AWS STS and the ROSA CLI. For more information about ROSA classic, see <u>Deployment options</u>.

The ROSA CLI uses auto mode or manual mode to create the IAM resources that are required to provision a ROSA cluster. auto mode immediately creates the required IAM roles and policies and an OpenID Connect (OIDC) provider. manual mode outputs the AWS CLI commands that are needed to create the IAM resources. By using manual mode, you can review the generated AWS CLI commands before running them manually. You can also use manual to pass the commands to another administrator or group in your organization so they can create the resources.

The procedures in this document use the manual mode of the ROSA CLI to create the required IAM resources for ROSA classic. For more options to get started, see Getting started with ROSA.

### **Topics**

- Prerequisites
- Step 1: Enable ROSA and configure prerequisites
- Step 2: Create a ROSA classic cluster with AWS STS and the ROSA CLI manual mode
- Step 3: Configure an identity provider and grant cluster access
- Step 4: Grant user access to a cluster
- Step 5: Grant administrator permissions to a user
- Step 6: Access a cluster through the web console
- Step 7: Deploy an application from the Developer Catalog
- Step 8: Revoke administrator permissions and user access
- Step 9: Delete a cluster and AWS STS resources

# **Prerequisites**

Before getting started, make sure you completed these actions:

• Install and configure the latest AWS CLI. For more information, see <u>Installing or updating the</u> latest version of the AWS CLI.

- Install and configure the latest ROSA CLI and OpenShift Container Platform CLI. For more information, see Getting started with the ROSA CLI.
- Service Quotas must have the required service quotas set for Amazon EC2, Amazon VPC, Amazon EBS, and Elastic Load Balancing that are needed to create and run a ROSA cluster. AWS or Red Hat may request service quota increases on your behalf as required for issue resolution. To view the required quotas, see Red Hat OpenShift Service on AWS endpoints and quotas in the AWS General Reference.
- To receive AWS support for ROSA, you must enable AWS Business, Enterprise On-Ramp, or Enterprise support plans. Red Hat may request AWS support on your behalf as required for issue resolution. For more information, see <u>Support for ROSA</u>. To enable AWS Support, see the <u>AWS</u> <u>Support page</u>.
- If you're using AWS Organizations to manage the AWS accounts that host the ROSA service, the
  organization's service control policy (SCP) must be configured to allow Red Hat to perform policy
  actions that's listed in the SCP without restriction. For more information, see the <u>ROSA SCP</u>
  troubleshooting documentation. For more information about SCPs, see <u>Service control policies</u>
  (SCPs).
- If deploying a ROSA cluster with AWS STS into an enabled AWS Region that's disabled by default, you must update the security token to version 2 for all the Regions in the AWS account with the following command.

```
aws iam set-security-token-service-preferences --global-endpoint-token-version v2Token
```

For more information about enabling Regions, see <u>Managing AWS Regions</u> in the *AWS General Reference*.

# Step 1: Enable ROSA and configure prerequisites

To create a ROSA cluster, you must first enable the ROSA service in the AWS ROSA console. The AWS ROSA console verifies if your AWS account has the necessary AWS Marketplace permissions, service quotas, and the Elastic Load Balancing (ELB) service-linked role named AWSServiceRoleForElasticLoadBalancing. If any of these prerequisites are missing, the console provides guidance on how to configure your account to meet the prerequisites.

- 1. Navigate to the <u>ROSA console</u>.
- 2. Choose **Get started**.

- 3. On the Verify ROSA prerequisites page, select I agree to share my contact information with Red Hat.
- 4. Choose Enable ROSA.
- 5. Once the page has verified your service quotas meet ROSA prerequisites and the ELB servicelinked role is created, open a new terminal session to create your first ROSA cluster using the ROSA CLI.

# Step 2: Create a ROSA classic cluster with AWS STS and the ROSA CLI manual mode

You can create a ROSA classic cluster using AWS Security Token Service (AWS STS) and the manual mode that's provided in the ROSA CLI.

When you create a cluster, you can run rosa create cluster --interactive to customize your deployment with a series of interactive prompts. For more information, see Interactive cluster creation mode reference in the Red Hat documentation.

After the cluster is provisioned, a single command is provided in the output. Run this command to deploy further clusters that use the exact same custom configuration.



#### Note

AWS shared VPCs aren't currently supported for ROSA installations.

1. Create the required IAM account roles and policies.

rosa create account-roles --mode manual



#### Note

If your offline access token has expired, the ROSA CLI outputs an error message stating that your authorization token needs updated. For steps to troubleshoot, see Troubleshoot ROSA CLI expired offline access tokens.

2. Run the AWS CLI commands generated in the output to create the roles and policies.

3. Create a cluster with AWS STS in --interactive mode to specify any custom settings.

rosa create cluster --interactive --sts



## 

After you enable etcd encryption for the key values in etcd, you incur a performance overhead of approximately 20%. The overhead is a result of introducing this second layer of encryption, in addition to the default Amazon EBS encryption that encrypts the etcd volumes.

4. To create the cluster-specific operator IAM roles, generate the operator policy JSON files in the current working directory and output the AWS CLI commands for review.

rosa create operator-roles --mode manual --cluster <CLUSTER\_NAME|CLUSTER\_ID>

- 5. Run the AWS CLI commands from the output.
- 6. Create the OpenID Connect (OIDC) provider the cluster operators use to authenticate.

rosa create oidc-provider --mode auto --cluster <CLUSTER\_NAME|CLUSTER\_ID>

7. Check the status of your cluster.

rosa describe cluster -c <CLUSTER\_NAME>



#### Note

If the creation process fails or the State field doesn't change to a ready status after 40 minutes, see Troubleshoot ROSA cluster creation issues.

To contact AWS Support or Red Hat support for assistance, see Support for ROSA.

8. Track the progress of the cluster creation by watching the OpenShift installer logs.

rosa logs install -c <CLUSTER\_NAME> --watch

# Step 3: Configure an identity provider and grant cluster access

ROSA includes a built-in OAuth server. After your cluster is created, you must configure OAuth to use an identity provider. You can then add users to your configured identity provider to grant them access to your cluster. You can grant these users cluster-admin or dedicated-admin permissions as required.

You can configure different identity provider types for your cluster. Supported types include GitHub, GitHub Enterprise, GitLab, Google, LDAP, OpenID Connect, and HTPasswd identity providers.

## Important

The HTPasswd identity provider is included only to enable a single, static administrator user to be created. HTPasswd isn't supported as a general-use identity provider for ROSA.

The following procedure configures a GitHub identity provider as an example. For instructions on how to configure each of the supported identity provider types, see Configuring identity providers for AWS STS.

- 1. Navigate to github.com and log in to your GitHub account.
- 2. If you don't have a GitHub organization to use for identity provisioning for your ROSA cluster, create one. For more information, see the steps in the GitHub documentation.
- Using the ROSA CLI's interactive mode, configure an identity provider for your cluster.

```
rosa create idp --cluster=<CLUSTER_NAME> --interactive
```

4. Follow the configuration prompts in the output to restrict cluster access to members of your GitHub organization.

```
I: Interactive mode enabled.
Any optional fields can be left empty and a default will be selected.
? Type of identity provider: github
? Identity provider name: github-1
? Restrict to members of: organizations
? GitHub organizations: <GITHUB_ORG_NAME>
? To use GitHub as an identity provider, you must first register the application:
  - Open the following URL:
```

```
https://github.com/organizations/<GITHUB_ORG_NAME>/settings/
applications/new?oauth_application%5Bcallback_url%5D=https%3A%2F%2Foauth-
openshift.apps.<CLUSTER_NAME>/<RANDOM_STRING>.p1.openshiftapps.com%2Foauth2callback
%2Fgithub-1&oauth_application%5Bname%5D=<CLUSTER_NAME>&oauth_application
%5Burl%5D=https%3A%2F%2Fconsole-openshift-console.apps.<CLUSTER_NAME>/
<RANDOM_STRING>.p1.openshiftapps.com
- Click on 'Register application'
...
```

- 5. Open the URL in the output with the following command. Replace <GITHUB\_ORG\_NAME> with the name of your GitHub organization.
- 6. On the GitHub web page, choose **Register application** to register a new OAuth application in your GitHub organization.
- 7. Use the information from the GitHub OAuth page to populate the remaining rosa create idp interactive prompts using the following command. Replace <GITHUB\_CLIENT\_ID> and <GITHUB\_CLIENT\_SECRET> with the credentials from your GitHub OAuth application.

```
...
? Client ID: <GITHUB_CLIENT_ID>
? Client Secret: [? for help] <GITHUB_CLIENT_SECRET>
? GitHub Enterprise Hostname (optional):
? Mapping method: claim
I: Configuring IDP for cluster '<CLUSTER_NAME>'
I: Identity Provider 'github-1' has been created.
    It will take up to 1 minute for this configuration to be enabled.
    To add cluster administrators, see 'rosa grant user --help'.
    To login into the console, open https://console-openshift-
console.apps.<CLUSTER_NAME>.<RANDOM_STRING>.pl.openshiftapps.com and click on github-1.
```

## Note

It might take about two minutes for the identity provider configuration to become active. If you configured a cluster-admin user, you can run the oc get pods -n openshift-authentication --watch command to watch the OAuth pods redeploy with the updated configuration.

8. Verify the identity provider has been configured correctly using the following command.

rosa list idps --cluster=<CLUSTER\_NAME>

# Step 4: Grant user access to a cluster

You can grant a user access to your cluster by adding them to the configured identity provider.

The following procedure adds a user to a GitHub organization that's configured for identity provisioning to the cluster.

- 1. Navigate to github.com and log in to your GitHub account.
- 2. Invite users that require cluster access to your GitHub organization. For more information, see <a href="Inviting users to join your organization">Inviting users to join your organization</a> in the documentation on Github.

# Step 5: Grant administrator permissions to a user

After you add a user to your configured identity provider, you can grant the user cluster-admin or dedicated-admin permissions for your cluster.

## Configure cluster-admin permissions

Grant the cluster-admin permissions using the following command. Replace
 IDP\_USER\_NAME> and <CLUSTER\_NAME> with your user and cluster name.

```
rosa grant user cluster-admin --user=<IDP_USER_NAME> --cluster=<CLUSTER_NAME>
```

2. Verify that the user is listed as a member of the cluster-admins group.

```
rosa list users --cluster=<CLUSTER_NAME>
```

## Configure dedicated-admin permissions

1. Grant the dedicated-admin permissions using the following command. Replace <IDP\_USER\_NAME> and <CLUSTER\_NAME> with your user and cluster name.

```
rosa grant user dedicated-admin --user=<IDP_USER_NAME> --cluster=<CLUSTER_NAME>
```

2. Verify that the user is listed as a member of the cluster-admins group.

```
rosa list users --cluster=<CLUSTER_NAME>
```

# Step 6: Access a cluster through the web console

After you create a cluster administrator user or add a user to your configured identity provider, you can log in to your cluster through the Red Hat Hybrid Cloud Console.

 Obtain the console URL for your cluster by using the following command. Replace <CLUSTER\_NAME> with the name of your cluster.

```
rosa describe cluster -c <CLUSTER_NAME> | grep Console
```

- 2. Navigate to the console URL in the output and log in.
  - If you create a cluster-admin user, log in using the provided credentials.
  - If you configure an identity provider for your cluster, choose the identity provider name in the Log in with... dialog and complete any authorization requests that are presented by your provider.

# Step 7: Deploy an application from the Developer Catalog

From the Red Hat Hybrid Cloud Console, you can deploy a Developer Catalog test application and expose it with a route.

- Navigate to Red Hat Hybrid Cloud Console and choose the cluster you want to deploy the app into.
- 2. On the cluster's page, choose **Open console**.
- 3. In the **Administrator** perspective, choose **Home** > **Projects** > **Create Project**.
- 4. Enter a name for your project and optionally add a **Display Name** and **Description**.
- 5. Choose **Create** to create the project.
- 6. Switch to the **Developer** perspective and choose **+Add**. Make sure that the selected project is the one that was just created.
- 7. In the **Developer Catalog** dialog, choose **All services**.
- 8. In the **Developer Catalog** page, choose **Languages** > **JavaScript** from the menu.

9. Choose Node.is, and then choose Create Application to open the Create Source-to-Image **Application** page.



## Note

You might need to choose **Clear All Filters** to display the **Node.js** option.

10In the **Git** section, choose **Try Sample**.

11In the Name field, add a unique name.

12Choose Create.



### Note

The new application takes several minutes to deploy.

13. When the deployment is complete, choose the route URL for the application.

A new tab in the browser opens with a message similar to the following.

Welcome to your Node.js application on OpenShift

14(Optional) Delete the application and clean up resources.

- a. In the **Administrator** perspective, choose **Home** > **Projects**.
- b. Open the action menu for your project and choose **Delete Project**.

# Step 8: Revoke administrator permissions and user access

You can revoke cluster-admin or dedicated-admin permissions from a user by using the ROSA CLI.

To revoke access from a user, you must remove the user from your configured identity provider.

# Revoke cluster-admin permissions from a user

1. Revoke the cluster-admin permission by using the following command. Replace <IDP\_USER\_NAME> and <CLUSTER\_NAME> with your user and cluster name.

rosa revoke user cluster-admin --user=<IDP\_USER\_NAME> --cluster=<CLUSTER\_NAME>

2. Verify that the user isn't listed as a member of the cluster-admins group.

```
rosa list users --cluster=<CLUSTER_NAME>
```

## Revoke dedicated-admin permissions from a user

Revoke the dedicated-admin permission using the following command. Replace
 IDP\_USER\_NAME> and <CLUSTER\_NAME> with your user and cluster name.

```
rosa revoke user dedicated-admin --user=<IDP_USER_NAME> --cluster=<CLUSTER_NAME>
```

2. Verify that the user isn't listed as a member of the dedicated-admins group.

```
rosa list users --cluster=<CLUSTER_NAME>
```

### Revoke user access to a cluster

You can revoke cluster access for an identity provider user by removing them from the configured identity provider.

You can configure different types of identity providers for your cluster. The following procedure revokes cluster access for a member of a GitHub organization.

- 1. Navigate to github.com and log in to your GitHub account.
- 2. Remove the user from your GitHub organization. For more information, see Removing a member from your organization in the GitHub documentation.

## **Step 9: Delete a cluster and AWS STS resources**

You can use the ROSA CLI to delete a cluster that uses AWS Security Token Service (AWS STS). You can also use the ROSA CLI to delete the IAM roles and OIDC provider created by ROSA. To delete the IAM policies created by ROSA, you can use the IAM console.



## Important

IAM roles and policies created by ROSA might be used by other ROSA clusters in the same account.

 Delete the cluster and watch the logs. Replace < CLUSTER\_NAME > with the name or ID of your cluster.

```
rosa delete cluster --cluster=<CLUSTER_NAME> --watch
```



### Important

You must wait for the cluster to delete completely before you remove the IAM roles, policies, and OIDC provider. The account IAM roles are required to delete the resources created by the installer. The operator IAM roles are required to clean up the resources created by the OpenShift operators. The operators use the OIDC provider to authenticate.

2. Delete the OIDC provider that the cluster operators use to authenticate by running the following command.

```
rosa delete oidc-provider -c <CLUSTER_ID> --mode auto
```

3. Delete the cluster-specific operator IAM roles.

```
rosa delete operator-roles -c <CLUSTER_ID> --mode auto
```

4. Delete the account IAM roles using the following command. Replace <PREFIX> with the prefix of the account IAM roles to delete. If you specified a custom prefix when creating the account IAM roles, specify the default ManagedOpenShift prefix.

```
rosa delete account-roles --prefix <PREFIX> --mode auto
```

- 5. Delete the IAM policies created by ROSA.
  - a. Log in to the IAM console.
  - b. On the left menu under **Access management**, choose **Policies**.
  - c. Select the policy that you want to delete and choose **Actions** > **Delete**.

- d. Enter the policy name and choose **Delete**.
- e. Repeat this step to delete each of the IAM policies for the cluster.

# Getting started with ROSA classic using AWS PrivateLink

ROSA classic clusters can be deployed in a few different ways: public, private, or private with AWS PrivateLink. For more information about ROSA classic, see <u>Deployment options</u>. For both public and private cluster configurations, the OpenShift cluster has access to the internet, and privacy is set on the application workloads at the application layer.

If you require both the cluster and the application workloads to be private, you can configure AWS PrivateLink with ROSA classic. AWS PrivateLink is a highly available, scalable technology that ROSA uses to create a private connection between the ROSA service and cluster resources in the AWS customer account. With AWS PrivateLink, the Red Hat site reliability engineering (SRE) team can access the cluster for support and remediation purposes by using a private subnet connected to the cluster's AWS PrivateLink endpoint.

For more information about AWS PrivateLink, see What is AWS PrivateLink?

## **Topics**

- Prerequisites
- Step 1: Enable ROSA and configure prerequisites
- Step 2: Create Amazon VPC architecture for the cluster
- Step 3: Create a cluster with AWS PrivateLink
- Step 4: Configure AWS PrivateLink DNS forwarding
- Step 5: Configure an identity provider and grant cluster access
- Step 6: Grant user access to a cluster
- Step 7: Grant administrator permissions to a user
- Step 8: Access a cluster through the web console
- Step 9: Deploy an application from the Developer Catalog
- Step 10: Revoke administrator permissions and user access
- Step 11: Delete a cluster and AWS STS resources

## **Prerequisites**

Before getting started, make sure you completed these actions:

- Install and configure the latest AWS CLI. For more information, see <u>Installing or updating the</u> latest version of the AWS CLI.
- Install and configure the latest ROSA CLI and OpenShift Container Platform CLI. For more information, see Getting started with the ROSA CLI.
- Service Quotas must have the required service quotas set for Amazon EC2, Amazon VPC, Amazon EBS, and Elastic Load Balancing that are needed to create and run a ROSA cluster. AWS or Red Hat may request service quota increases on your behalf as required for issue resolution. To view the required quotas, see Red Hat OpenShift Service on AWS endpoints and quotas in the AWS General Reference.
- To receive AWS support for ROSA, you must enable AWS Business, Enterprise On-Ramp, or Enterprise support plans. Red Hat may request AWS support on your behalf as required for issue resolution. For more information, see <u>Support for ROSA</u>. To enable AWS Support, see the <u>AWS</u> <u>Support page</u>.
- If you're using AWS Organizations to manage the AWS accounts that host the ROSA service, the
  organization's service control policy (SCP) must be configured to allow Red Hat to perform policy
  actions that's listed in the SCP without restriction. For more information, see the ROSA SCP
  troubleshooting documentation. For more information about SCPs, see Service control policies
  (SCPs).
- If deploying a ROSA cluster with AWS STS into an enabled AWS Region that's disabled by default, you must update the security token to version 2 for all the Regions in the AWS account with the following command.

```
aws iam set-security-token-service-preferences --global-endpoint-token-version v2Token
```

For more information about enabling Regions, see <u>Managing AWS Regions</u> in the *AWS General Reference*.

# **Step 1: Enable ROSA and configure prerequisites**

To create a ROSA cluster, you must first enable the ROSA service in the AWS ROSA console. The AWS ROSA console verifies if your AWS account has the necessary AWS Marketplace

Prerequisites 77

permissions, service quotas, and the Elastic Load Balancing (ELB) service-linked role named AWSServiceRoleForElasticLoadBalancing. If any of these prerequisites are missing, the console provides guidance on how to configure your account to meet the prerequisites.

- Navigate to the ROSA console.
- 2. Choose **Get started**.
- 3. On the Verify ROSA prerequisites page, select I agree to share my contact information with Red Hat.
- 4. Choose Enable ROSA.
- 5. Once the page has verified your service quotas meet ROSA prerequisites and the ELB servicelinked role is created, open a new terminal session to create your first ROSA cluster using the ROSA CLI.

# **Step 2: Create Amazon VPC architecture for the cluster**

To create a ROSA cluster that uses AWS PrivateLink, you must first configure your own Amazon VPC architecture to deploy your solution into. ROSA requires that customers configure at least one public and private subnet per Availability Zone used to create clusters. For single-AZ clusters, only use Availability Zone is used. For multi-AZ clusters, three Availability Zones are needed.

### Important

If Amazon VPC requirements are not met, cluster creation fails.

The following procedure uses the AWS CLI to create both a public and private subnet into a single Availability Zone for a Single-AZ cluster. All cluster resources are in the private subnet. The public subnet routes outbound traffic by using a NAT gateway to the internet.

This example uses the CIDR block 10.0.0.0/16 for the Amazon VPC. However, you can choose a different CIDR block. For more information, see VPC sizing.

1. Set an environment variable for the cluster name by running the following command.

ROSA\_CLUSTER\_NAME=rosa-privatelink

2. Create a VPC with a 10.0.0.0/16 CIDR block.

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --query Vpc.VpcId --output text
```

The preceding command returns the ID of the new VPC. The following is an example output.

```
vpc-0410832ee325aafea
```

3. Using the VPC ID from the previous step, tag the VPC using the ROSA\_CLUSTER\_NAME variable.

```
aws ec2 create-tags --resources <VPC_ID_VALUE> --tags Key=Name, Value=
$ROSA_CLUSTER_NAME
```

4. Enable DNS hostname support on the VPC.

```
aws ec2 modify-vpc-attribute --vpc-id <VPC_ID_VALUE> --enable-dns-hostnames
```

5. Create a public subnet in the VPC with a 10.0.1.0/24 CIDR block, specifying the Availability Zone where the resource should be created.

## Important

When creating subnets, make sure that subnets are created to an Availability Zone that has ROSA instance types available. If you don't choose a specific Availability Zone, the subnet is created in any one of the Availability Zones in the AWS Region that you specify. To specify a specific Availability Zone, use the --availability zone argument in the create-subnet command. You can use the rosa list instance-types command to list all ROSA instance types available. To check if an instance type is available for a given Availability Zone, use the following command.

```
aws ec2 describe-instance-type-offerings --location-type availability-zone --
filters Name=location, Values=<availability_zone> --region <region> --output
  text | egrep "<instance_type>"
```

## 

ROSA requires that customers configure at least one public and private subnet per Availability Zone used to create clusters. For single-AZ clusters, only one Availability Zone is needed. For multi-AZ clusters, three Availability Zones are needed. If these requirements are not met, cluster creation fails.

```
aws ec2 create-subnet --vpc-id <VPC_ID_VALUE> --cidr-block 10.0.1.0/24 -- availability-zone <AZ_NAME> --query Subnet.SubnetId --output text
```

The preceding command returns the ID of the new subnet. The following is an example output.

```
subnet-0b6a7e8cbc8b75920
```

6. Using the subnet ID from the previous step, tag the subnet using the ROSA\_CLUSTER\_NAME-public variable.

```
aws ec2 create-tags --resources <PUBLIC_SUBNET_ID> --tags Key=Name, Value=
$ROSA_CLUSTER_NAME-public
```

7. Create a private subnet in the VPC with a 10.0.0.0/24 CIDR block, specifying the same Availability Zone that the public subnet deployed into.

```
aws ec2 create-subnet --vpc-id <VPC_ID_VALUE> --cidr-block 10.0.0.0/24 -- availability-zone <AZ_NAME> --query Subnet.SubnetId --output text
```

The preceding command returns the ID of the new subnet. The following is an example output.

```
subnet-0b6a7e8cbc8b75920
```

8. Using the subnet ID from the previous step, tag the subnet using the ROSA\_CLUSTER\_NAME-private variable.

```
aws ec2 create-tags --resources <PRIVATE_SUBNET_ID> --tags Key=Name, Value=
$ROSA_CLUSTER_NAME-private
```

9. Create an internet gateway for outbound traffic and attach it to the VPC.

```
aws ec2 create-internet-gateway --query InternetGateway.InternetGatewayId --output
text
```

```
aws ec2 attach-internet-gateway --vpc-id <VPC_ID_VALUE> --internet-gateway-id
  <IG_ID_VALUE>
```

10.Tag the internet gateway with the ROSA\_CLUSTER\_NAME variable.

```
aws ec2 create-tags --resources <IG_ID_VALUE> --tags Key=Name, Value=
$ROSA_CLUSTER_NAME
```

11Create a route table for outbound traffic, associate it to the public subnet, and configure traffic to route to the internet gateway.

```
aws ec2 create-route-table --vpc-id <VPC_ID_VALUE> --query RouteTable.RouteTableId --
output text

aws ec2 associate-route-table --subnet-id <PUBLIC_SUBNET_ID> --route-table-id
<PUBLIC_RT_ID>

aws ec2 create-route --route-table-id <PUBLIC_RT_ID> --destination-cidr-block
0.0.0.0/0 --gateway-id <IG_ID_VALUE>
```

12. Tag the public route table with the ROSA\_CLUSTER\_NAME variable and verify that the route table was properly configured.

```
aws ec2 create-tags --resources <PUBLIC_RT_ID> --tags Key=Name, Value=
$ROSA_CLUSTER_NAME

aws ec2 describe-route-tables --route-table-id <PUBLIC_RT_ID>
```

13Create a NAT gateway in the public subnet with an elastic IP address to enable traffic to the private subnet.

14Tag the NAT gateway and elastic IP address with the \$ROSA\_CLUSTER\_NAME variable.

```
aws ec2 create-tags --resources <EIP_ADDRESS> --resources <NAT_GATEWAY_ID> --tags
Key=Name, Value=$ROSA_CLUSTER_NAME
```

15Create a route table for private subnet traffic, associate it to the private subnet, and configure traffic to route to the NAT gateway.

```
aws ec2 create-route-table --vpc-id <VPC_ID_VALUE> --query RouteTable.RouteTableId --
output text

aws ec2 associate-route-table --subnet-id <PRIVATE_SUBNET_ID> --route-table-id
<PRIVATE_RT_ID>

aws ec2 create-route --route-table-id <PRIVATE_RT_ID> --destination-cidr-block
0.0.0.0/0 --gateway-id <NAT_GATEWAY_ID>
```

16.Tag the private route table and elastic IP address with the \$ROSA\_CLUSTER\_NAME-private variable.

```
aws ec2 create-tags --resources <PRIVATE_RT_ID> <EIP_ADDRESS> --tags Key=Name, Value=
$ROSA_CLUSTER_NAME-private
```

# Step 3: Create a cluster with AWS PrivateLink

You can use AWS PrivateLink and the ROSA CLI to create a cluster with a single Availability Zone (Single-AZ) or multiple Availability Zones (Multi-AZ). In either case, your machine's CIDR value must match your VPC's CIDR value.

The following procedure uses the rosa create cluster command to create a Single-AZ ROSA cluster. To create a Multi-AZ cluster, specify multi-az in the command and the private subnet IDs for each private subnet you want you to deploy to.

## Note

If you use a firewall, you must configure it so that ROSA can access the sites that it requires to function.

For more information, see <u>AWS firewall prerequisites</u> in the Red Hat OpenShift documentation.

1. Create a Single-AZ cluster by running the following command.

rosa create cluster --private-link --cluster-name=<CLUSTER\_NAME> --machinecidr=10.0.0.0/16 --subnet-ids=<PRIVATE\_SUBNET\_ID>



### Note

To create a cluster that uses AWS PrivateLink with AWS Security Token Service (AWS STS) short-lived credentials, append --sts --mode auto or --sts --mode manual to the end of the rosa create cluster command.

2. Create the cluster operator IAM roles by following the interactive prompts.

```
rosa create operator-roles --interactive -c <CLUSTER_NAME>
```

3. Create the OpenID Connect (OIDC) provider the cluster operators use to authenticate.

```
rosa create oidc-provider --interactive -c <CLUSTER_NAME>
```

4. Check the status of your cluster.

```
rosa describe cluster -c <CLUSTER_NAME>
```

## Example



#### Note

It may take up to 40 minutes for the cluster State field to show the ready status. If provisioning fails or doesn't show as ready after 40 minutes, see Troubleshoot ROSA cluster provisioning issues.

To contact AWS Support or Red Hat support for assistance, see Support for ROSA.

5. Track the progress of the cluster creation by watching the OpenShift installer logs.

```
rosa logs install -c <CLUSTER_NAME> --watch
```

# Step 4: Configure AWS PrivateLink DNS forwarding

Clusters that use AWS PrivateLink create a public hosted zone and a private hosted zone in Route 53. Records within the Route 53 private hosted zone are resolvable only from within the VPC that it's assigned to.

The Let's Encrypt DNS-01 validation requires a public zone so that valid and publicly trusted certificates can be issued for the domain. The validation records are deleted after Let's Encrypt validation is complete. The zone is still required for issuing and renewing these certificates, which are typically required every 60 days. Although these zones usually appear empty, a public zone serves a critical role in the validation process.

For more information about AWS private hosted zones, see <u>Working with private zones</u>. For more information about public hosted zones, see <u>Working with public hosted zones</u>.

## **Configure a Route 53 Resolver inbound endpoint**

To allow for records such as api.<cluster\_domain> and \*.apps.<cluster\_domain> to resolve outside of the VPC, configure a Route 53 Resolver inbound endpoint.

- 1. Open the Route 53 console.
- 2. In the navigation pane under **Resolver**, choose **Inbound endpoints**.
- 3. Choose Configure endpoints.
- 4. In the upper right, use the AWS Region selector to choose the AWS Region that contains the VPC used for the cluster.
- 5. Under **Basic configuration**, choose **Inbound only** and then choose **Next**.
- 6. On the **Configure inbound endpoint** page, complete the **General settings for inbound endpoint** section. Under **Security group for this endpoint**, choose a security group that allows inbound UDP and TCP traffic from the remote network on destination port 53.
- 7. In the **IP address** section, choose the Availability Zones and private subnets that were used when creating the cluster and choose **Next**.
- 8. (Optional) Complete the **Tags** section.
- 9. Choose Submit.

## Configure DNS forwarding for the cluster

After the Route 53 Resolver internal endpoint is associated and operational, configure DNS forwarding so DNS queries can be handled by the designated servers on your network.

- Configure your corporate network to forward DNS queries to those IP addresses for the top-level domain, such as drow-pl-01.htno.pl.openshiftapps.com.
- 2. If you're forwarding DNS queries from one VPC to another VPC, follow the instructions in Managing forwarding rules.
- 3. If you're configuring your remote network DNS server, see your specific DNS server documentation to configure selective DNS forwarding for the installed cluster domain.

# Step 5: Configure an identity provider and grant cluster access

ROSA includes a built-in OAuth server. After your ROSA cluster is created, you must configure OAuth to use an identity provider. You can then add users to your configured identity provider to grant them access to your cluster. You can grant these users cluster-admin or dedicatedadmin permissions as required.

You can configure different identity provider types for your cluster. The supported types include GitHub, GitHub Enterprise, GitLab, Google, LDAP, OpenID Connect, and HTPasswd identity providers.



### Important

The HTPasswd identity provider is included only to enable a single, static administrator user to be created. HTPasswd isn't supported as a general-use identity provider for ROSA.

The following procedure configures a GitHub identity provider as an example. For instructions on how to configure each of the supported identity provider types, see Configuring identity providers for AWS STS.

- 1. Navigate to github.com and log in to your GitHub account.
- 2. If you don't have a GitHub organization to use for identity provisioning for your ROSA cluster, create one. For more information, see the steps in the GitHub documentation.

3. Using the ROSA CLI's interactive mode, configure an identity provider for your cluster by running the following command.

```
rosa create idp --cluster=<CLUSTER_NAME> --interactive
```

4. Follow the configuration prompts in the output to restrict cluster access to members of your GitHub organization.

```
I: Interactive mode enabled.

Any optional fields can be left empty and a default will be selected.

? Type of identity provider: github

? Identity provider name: github-1

? Restrict to members of: organizations

? GitHub organizations: <GITHUB_ORG_NAME>

? To use GitHub as an identity provider, you must first register the application:

- Open the following URL:

https://github.com/organizations/<GITHUB_ORG_NAME>/settings/
applications/new?oauth_application%5Bcallback_url%5D=https%3A%2F%2Foauth-openshift.apps.<CLUSTER_NAME>/<RANDOM_STRING>.p1.openshiftapps.com%2Foauth2callback
%2Fgithub-1&oauth_application%5Bname%5D=<CLUSTER_NAME>&oauth_application
%5Burl%5D=https%3A%2F%2Fconsole-openshift-console.apps.<CLUSTER_NAME>/
<RANDOM_STRING>.p1.openshiftapps.com

- Click on 'Register application'
...
```

- 5. Open the URL in the output, replacing <GITHUB\_ORG\_NAME> with the name of your GitHub organization.
- 6. On the GitHub web page, choose **Register application** to register a new OAuth application in your GitHub organization.
- 7. Use the information from the GitHub OAuth page to populate the remaining rosa create idp interactive prompts, replacing <GITHUB\_CLIENT\_ID> and <GITHUB\_CLIENT\_SECRET> with the credentials from your GitHub OAuth application.

```
...
? Client ID: <GITHUB_CLIENT_ID>
? Client Secret: [? for help] <GITHUB_CLIENT_SECRET>
? GitHub Enterprise Hostname (optional):
? Mapping method: claim
I: Configuring IDP for cluster '<CLUSTER_NAME>'
I: Identity Provider 'github-1' has been created.
It will take up to 1 minute for this configuration to be enabled.
```

To add cluster administrators, see 'rosa grant user --help'. To login into the console, open https://console-openshiftconsole.apps.<CLUSTER\_NAME>.<RANDOM\_STRING>.p1.openshiftapps.com and click on github-1.



#### Note

It might take around two minutes for the identity provider configuration to become active. If you configured a cluster-admin user, you can run the oc get pods -n openshift-authentication --watch command to watch the OAuth pods redeploy with the updated configuration.

8. Verify the identity provider has been configured correctly.

```
rosa list idps --cluster=<CLUSTER_NAME>
```

# Step 6: Grant user access to a cluster

You can grant a user access to your cluster by adding them to the configured identity provider.

The following procedure adds a user to a GitHub organization that's configured for identity provisioning to the cluster.

- 1. Navigate to github.com and log in to your GitHub account.
- 2. Invite users that require cluster access to your GitHub organization. For more information, see Inviting users to join your organization in the GitHub documentation.

# Step 7: Grant administrator permissions to a user

After you added a user to your configured identity provider, you can grant the user clusteradmin or dedicated-admin permissions for your cluster.

# Configure cluster-admin permissions

 Grant the cluster-admin permissions using the following command. Replace <IDP USER NAME> and <CLUSTER NAME> with your user and cluster name.

```
rosa grant user cluster-admin --user=<IDP_USER_NAME> --cluster=<CLUSTER_NAME>
```

2. Verify the user is listed as a member of the cluster-admins group.

```
rosa list users --cluster=<CLUSTER_NAME>
```

## Configure dedicated-admin permissions

1. Grant the dedicated-admin permissions with the following command. Replace <IDP\_USER\_NAME> and <CLUSTER\_NAME> with your user and cluster name.

```
rosa grant user dedicated-admin --user=<IDP_USER_NAME> --cluster=<CLUSTER_NAME>
```

2. Verify the user is listed as a member of the cluster-admins group.

```
rosa list users --cluster=<CLUSTER_NAME>
```

# Step 8: Access a cluster through the web console

After you created a cluster administrator user or added a user to your configured identity provider, you can log in to your cluster through the Red Hat Hybrid Cloud Console.

 Obtain the console URL for your cluster using the following command. Replace <CLUSTER\_NAME> with the name of your cluster.

```
rosa describe cluster -c <CLUSTER_NAME> | grep Console
```

- 2. Navigate to the console URL in the output and log in.
  - If you created a cluster-admin user, log in using the provided credentials.
  - If you configured an identity provider for your cluster, choose the identity provider name in the **Log in with...** dialog and complete any authorization requests presented by your provider.

# Step 9: Deploy an application from the Developer Catalog

From the Red Hat Hybrid Cloud Console, you can deploy a Developer Catalog test application and expose it with a route.

- 1. Navigate to Red Hat Hybrid Cloud Console and choose the cluster that you want to deploy the app into.
- 2. On the cluster's page, choose **Open console**.
- 3. In the Administrator perspective, choose Home > Projects > Create Project.
- 4. Enter a name for your project and optionally add a **Display Name** and **Description**.
- 5. Choose **Create** to create the project.
- 6. Switch to the **Developer** perspective and choose **+Add**. Make sure that the selected project is the one that was just created.
- 7. In the **Developer Catalog** dialog, choose **All services**.
- 8. In the **Developer Catalog** page, choose **Languages** > **JavaScript** from the menu.
- 9. Choose **Node.js**, and then choose **Create Application** to open the **Create Source-to-Image Application** page.



### Note

You might need to choose **Clear All Filters** to display the **Node.js** option.

10In the **Git** section, choose **Try Sample**.

11In the **Name** field, add a unique name.

12Choose Create.



#### Note

The new application takes several minutes to deploy.

13When the deployment is complete, choose the route URL for the application.

A new tab in the browser opens with a message that's similar to the following.

Welcome to your Node.js application on OpenShift

14(Optional) Delete the application and clean up resources.

- a. In the **Administrator** perspective, choose **Home** > **Projects**.
- b. Open the action menu for your project and choose **Delete Project**.

# Step 10: Revoke administrator permissions and user access

You can revoke cluster-admin or dedicated-admin permissions from a user by using the ROSA CLI.

To revoke access from a user, you must remove the user from your configured identity provider.

## Revoke cluster-admin permissions from a user

1. Revoke the cluster-admin permissions using the following command. Replace <IDP\_USER\_NAME> and <CLUSTER\_NAME> with your user and cluster name.

```
rosa revoke user cluster-admin --user=<IDP_USER_NAME> --cluster=<CLUSTER_NAME>
```

2. Verify that the user isn't listed as a member of the cluster-admins group.

```
rosa list users --cluster=<CLUSTER_NAME>
```

# Revoke dedicated-admin permissions from a user

1. Revoke the dedicated-admin permissions using the following command. Replace <IDP\_USER\_NAME> and <CLUSTER\_NAME> with your user and cluster name.

```
rosa revoke user dedicated-admin --user=<IDP_USER_NAME> --cluster=<CLUSTER_NAME>
```

2. Verify that the user isn't listed as a member of the dedicated-admins group.

```
rosa list users --cluster=<CLUSTER_NAME>
```

### Revoke user access to a cluster

You can revoke cluster access for an identity provider user by removing them from the configured identity provider.

You can configure different types of identity providers for your cluster. The following procedure revokes cluster access for a member of a GitHub organization.

1. Navigate to github.com and log in to your GitHub account.

2. Remove the user from your GitHub organization. For more information, see Removing a member from your organization in the GitHub documentation.

# **Step 11: Delete a cluster and AWS STS resources**

You can use the ROSA CLI to delete a cluster that uses AWS Security Token Service (AWS STS). You can also use the ROSA CLI to delete the IAM roles and OIDC provider created by ROSA. To delete the IAM policies created by ROSA, you can use the IAM console.



### Important

IAM roles and policies created by ROSA might be used by other ROSA clusters in the same account.

 Delete the cluster and watch the logs. Replace < CLUSTER\_NAME > with the name or ID of your cluster.

rosa delete cluster --cluster=<CLUSTER\_NAME> --watch

## 

You must wait for the cluster to delete completely before you remove the IAM roles, policies, and OIDC provider. The account IAM roles are required to delete the resources created by the installer. The operator IAM roles are required to clean up the resources created by the OpenShift operators. The operators use the OIDC provider to authenticate.

2. Delete the OIDC provider that the cluster operators use to authenticate by running the following command.

rosa delete oidc-provider -c <CLUSTER\_ID> --mode auto

3. Delete the cluster-specific operator IAM roles.

rosa delete operator-roles -c <CLUSTER\_ID> --mode auto

4. Delete the account IAM roles using the following command. Replace <PREFIX> with the prefix of the account IAM roles to delete. If you specified a custom prefix when creating the account IAM roles, specify the default ManagedOpenShift prefix.

```
rosa delete account-roles --prefix <PREFIX> --mode auto
```

- 5. Delete the IAM policies created by ROSA.
  - a. Log in to the IAM console.
  - b. On the left menu under Access management, choose Policies.
  - c. Select the policy that you want to delete and choose **Actions** > **Delete**.
  - d. Enter the policy name and choose **Delete**.
  - e. Repeat this step to delete each of the IAM policies for the cluster.

# **Security in ROSA**

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The <u>shared responsibility model</u> describes this as security of the cloud and security in the cloud:

- Security of the cloud AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the <u>AWS</u>
   <u>Compliance Programs</u>. To learn about the compliance programs that apply to ROSA, see <u>AWS</u> services in Scope by Compliance Program.
- **Security in the cloud** Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using ROSA. It shows you how to configure ROSA to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your ROSA resources.

## **Topics**

- Data protection in ROSA
- AWS managed IAM policies for ROSA
- Resilience in ROSA
- Infrastructure security in ROSA

# **Data protection in ROSA**

The <u>Overview of responsibilities for ROSA</u> documentation and <u>AWS shared responsibility model</u> define data protection in ROSA. AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. Red Hat is responsible for protecting the cluster infrastructure and underlying service platform. The customer is responsible for maintaining control over content that

Data protection 93

is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the <a href="Data">Data</a><a href="Privacy FAQ">Privacy FAQ</a>. For information about data protection in Europe, see the <a href="AWS Shared Responsibility">AWS Shared Responsibility</a><a href="Model and GDPR">Model and GDPR</a> blog post on the <a href="AWS Security Blog">AWS Security Blog</a>.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form fields such as a **Name** field. This includes when you work with ROSA or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into ROSA or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.

### **Topics**

- Protecting data using encryption
- Internetwork traffic privacy

## Protecting data using encryption

Data protection refers to protecting data while in transit (as it travels to and from ROSA) and at rest (while it is stored on disks in AWS data centers).

Red Hat OpenShift Service on AWS provides secure access to Amazon Elastic Block Store (Amazon EBS) storage volumes attached to Amazon EC2 instances for ROSA control plane, infrastructure, and worker nodes, as well as Kubernetes persistent volumes for persistent storage. ROSA encrypts volume data at rest and in transit, and uses AWS Key Management Service (AWS KMS) to help protect your encrypted data. The service uses Amazon S3 for container image registry storage, which is encrypted at rest by default.

## Important

Because ROSA is a managed service, AWS and Red Hat manage the infrastructure that ROSA uses. Customers should not attempt to manually shut down the Amazon EC2 instances that ROSA uses from the AWS console or CLI. This action can lead to customer data loss.

## Data encryption for Amazon EBS-backed storage volumes

Red Hat OpenShift Service on AWS uses the Kubernetes persistent volume (PV) framework to allow cluster administrators to provision a cluster with persistent storage. Persistent volumes, as well as the control plane, infrastructure, and worker nodes, are backed by Amazon Elastic Block Store (Amazon EBS) storage volumes attached to Amazon EC2 instances.

For ROSA persistent volumes and nodes backed by Amazon EBS, encryption operations occur on the servers that host EC2 instances, ensuring the security of both data at rest and data in transit between an instance and its attached storage. For more information, see Amazon EBS encryption in the Amazon EC2 User Guide.

## Data encryption for the Amazon EBS CSI driver and Amazon EFS CSI driver

ROSA defaults to using the Amazon EBS CSI driver to provision Amazon EBS storage. The Amazon EBS CSI driver and Amazon EBS CSI Driver Operator are installed on the cluster by default in the openshift-cluster-csi-drivers namespace. The Amazon EBS CSI driver and operator allow you to dynamically provision persistent volumes and create volume snapshots.

ROSA is also capable of provisioning persistent volumes using the Amazon EFS CSI driver and Amazon EFS CSI Driver Operator. The Amazon EFS driver and operator also allow you to share file system data between pods or with other applications within or outside of Kubernetes.

Volume data is secured in transit for both the Amazon EBS CSI driver and Amazon EFS CSI driver. For more information, see Using Container Storage Interface (CSI) in the Red Hat documentation.

## Important

When dynamically provisioning ROSA persistent volumes using the Amazon EFS CSI driver, Amazon EFS considers the user ID, group ID (GID), and secondary group IDs of the access point when evaluating file system permissions. Amazon EFS replaces the user and group IDs on files with the user and group IDs on the access point and ignores NFS client IDs. As a result, Amazon EFS silently ignores fsGroup settings. ROSA is not able to replace the GIDs of files by using fsGroup. Any pod that can access a mounted Amazon EFS access point can access any file on the volume. For more information, see Working with Amazon EFS access points in the Amazon EFS User Guide.

## etcd encryption

ROSA provides the option to enable encryption of etcd key values within the etcd volume during cluster creation, adding an additional layer of encryption. Once etcd is encrypted, you will incur approximately 20% additional performance overhead. We recommend that you enable etcd encryption only if you specifically require it for your use case. For more information, see etcd encryption in the ROSA service definition.

## **Key management**

ROSA uses KMS keys to securely manage control plane, infrastructure, and worker data volumes and persistent volumes for customer applications. During cluster creation, you have the choice of using the default AWS managed KMS key provided by Amazon EBS, or specifying your own customer managed key. For more information, see Data encryption using KMS.

## Data encryption for the built-in image registry

ROSA provides a built-in container image registry to store, retrieve, and share container images via Amazon S3 bucket storage. The registry is configured and managed by the OpenShift Image Registry Operator. It provides an out-of-the-box solution for users to manage the images that run their workloads, and runs on top of the existing cluster infrastructure. For more information, see Registry in the Red Hat documentation.

ROSA offers public and private image registries. For enterprise applications, we recommend using a private registry to protect your images from being used by unauthorized users. To protect your registry's data at rest, ROSA uses server-side encryption by default with Amazon S3 managed keys (SSE-S3). This does not require any action on your part, and is offered at no additional charge. For

more information, see <u>Protecting data using server-side encryption with Amazon S3 managed</u> encryption keys (SSE-S3) in the *Amazon S3 User Guide*.

ROSA uses Transport Layer Security (TLS) protocol to secure data in transit to and from the image registry. For more information, see Registry in the Red Hat documentation.

## **Data encryption using KMS**

ROSA uses AWS KMS to securely manage keys for encrypted data. Control plane, infrastructure, and worker node volumes are encrypted by default using the AWS managed KMS key provided by Amazon EBS. This KMS key has the alias aws/ebs. Persistent volumes that use the default gp3 storage class are also encrypted by default using this KMS key.

Newly created ROSA clusters are configured to use the default gp3 storage class to encrypt persistent volumes. Persistent volumes created by using any other storage class are only encrypted if the storage class is configured to be encrypted. For more information about ROSA pre-built storage classes, see <a href="Configuring persistent storage">Configuring persistent storage</a> in the Red Hat documentation. Newly created ROSA clusters are configured to use the default gp3 storage class to encrypt persistent volumes. Persistent volumes created by using any other storage class are only encrypted if the storage class is configured to be encrypted. For more information about ROSA pre-built storage classes, see <a href="Configuring persistent storage">Configuring persistent storage</a> in the Red Hat documentation.

During cluster creation, you can choose to encrypt the persistent volumes in your cluster using the default Amazon EBS-provided key, or specify your own customer managed symmetric KMS key. For more information about creating keys, see <u>Creating symmetric encryption KMS keys</u> in the *AWS KMS Developer Guide*.

You can also encrypt persistent volumes for individual containers within a cluster by defining a KMS key. This is useful when you have explicit compliance and security guidelines when deploying to AWS. For more information, see <a href="Encrypting container persistent volumes on AWS with a KMS">Encrypting container persistent volumes on AWS with a KMS</a> key in the Red Hat documentation.

The following points should be considered when encrypting persistent volumes using your own KMS keys:

- When you use KMS encryption with your own KMS key, the key must exist in the same AWS Region as your cluster.
- There is a cost associated with creating and using your own KMS keys. For more information, see AWS Key Management Service pricing.

# Internetwork traffic privacy

Red Hat OpenShift Service on AWS uses Amazon Virtual Private Cloud (Amazon VPC) to create boundaries between resources in your ROSA cluster and control traffic between them, your on-premises network, and the internet. For more information about Amazon VPC security, see Internetwork traffic privacy in Amazon VPC in the *Amazon VPC User Guide*.

Within the VPC, you can configure your ROSA clusters to use an HTTP or HTTPS proxy server to deny direct internet access. If you are a cluster administrator, you can also define network policies at the pod level that restrict internetwork traffic to pods in your ROSA cluster. For more information, see Infrastructure security in ROSA.

# **AWS managed IAM policies for ROSA**

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining <u>customer managed policies</u> that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the IAM User Guide.

# AWS managed policy: ROSAManageSubscription

You can attach the ROSAManageSubscription policy to your IAM entities. Before you enable ROSA in the AWS ROSA console, you must first attach this policy to a console role.

This policy grants the AWS Marketplace permissions required for you to manage the ROSA subscription.

Internetwork privacy 98

#### **Permissions details**

This policy includes the following permissions.

- aws-marketplace: Subscribe Grants permission to subscribe to the AWS Marketplace product for ROSA.
- aws-marketplace: Unsubscribe Allows principals to remove subscriptions to AWS Marketplace products.
- aws-marketplace: ViewSubscriptions Allows principals to view subscriptions from AWS Marketplace. This is required so that the IAM principal can view the available AWS Marketplace subscriptions.

To view the full JSON policy document, see <u>ROSAManageSubscription</u> in the *AWS Managed Policy Reference Guide*.

# AWS managed policies for ROSA with HCP account roles

You can attach these AWS managed policies to the account roles needed to use ROSA with hosted control planes (HCP). The permissions are required for Red Hat site reliability engineering (SRE) support on the cluster, cluster creation, and compute functionality.

The following managed policies are required:

- ROSAWorkerInstancePolicy Allows the ROSA service to manage Amazon EC2 instance lifecycles in a ROSA cluster.
- <u>ROSASRESupportPolicy</u> Grants required permissions to Red Hat site reliability engineers
  (SREs) to directly observe, diagnose, and support AWS resources associated with ROSA clusters,
  including the ability to change ROSA cluster node state.
- <u>ROSAInstallerPolicy</u> Grants required permissions to the installer to manage AWS resources that support cluster installation.

# AWS managed policies for ROSA with HCP operator roles

You can attach these AWS managed policies to the operator roles needed to use ROSA with hosted control planes (HCP). The permissions are required to allow OpenShift operators to manage ROSA with HCP cluster nodes.

## The following managed policies are required:

- ROSAAmazonEBSCSIDriverOperatorPolicy Grants required permissions to the Amazon EBS CSI
   Driver Operator to install and maintain the Amazon EBS CSI driver on a ROSA cluster.
- <u>ROSAIngressOperatorPolicy</u> Grants required permissions to the Ingress Operator to provision
  and manage load balancers and DNS configurations for ROSA clusters. The policy allows read
  access to tag values. The operator then filters the tag values for Route 53 resources to discover
  hosted zones.
- ROSAImageRegistryOperatorPolicy Grants required permissions to the Image Registry
  Operator to provision and manage resources for the ROSA in-cluster image registry and
  dependent services, including S3.
- <u>ROSACloudNetworkConfigOperatorPolicy</u> Grants required permissions to the Cloud Network Config Controller Operator to provision and manage networking resources for the ROSA cluster networking overlay.
- <u>ROSAKubeControllerPolicy</u> Grants required permissions to kube controller to manage Amazon EC2, Elastic Load Balancing, and AWS KMS resources for a ROSA with hosted control planes cluster.
- <u>ROSANodePoolManagementPolicy</u> Grants required permissions to the NodePool controller to describe, run, and terminate Amazon EC2 instances managed as worker nodes. This policy also enables disk encryption of the worker node root volume using AWS KMS keys.
- ROSAKMSProviderPolicy Grants required permissions to the built-in AWS Encryption Provider
  to manage AWS KMS keys that support etcd data encryption. This policy allows Amazon EC2 to
  encrypt and decrypt etcd data using the KMS keys provided by the AWS Encryption Provider.
- <u>ROSAControlPlaneOperatorPolicy</u> Grants required permissions to the Control Plane Operator to manage Amazon EC2 and Route 53 resources for ROSA with hosted control planes clusters.

To view managed policy permissions, see <u>AWS managed policies</u> in the *AWS Managed Policy Reference Guide*.

# **ROSA updates to AWS managed policies**

View details about updates to AWS managed policies for ROSA since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the ROSA document history page.

Change	Description	Date
ROSAInstallerPolicy — Policy updated	ROSA updated the policy to allow the service to provide installer alert messages when cluster installation fails due to a missing customer-specified cluster OIDC provider. This policy was also updated with permissions that allow the service to retrieve existing DNS name servers so that cluster provisioning operation s are idempotent. To learn more, see ROSAInstallerPolic Y.	January 26, 2024
ROSASRESupportPolicy — Policy updated	ROSA updated the policy to allow the service to perform read operations on security groups using the DescribeS ecurityGroups API. To learn more, see ROSASRESU pportPolicy.	January 22, 2024
ROSAImageRegistryO peratorPolicy — Policy updated	ROSA updated the policy to allow the Image Registry Operator to take actions on Amazon S3 buckets in Regions with 14-charac ter names. To learn more, see ROSAImageRegistryOperatorPolicy.	December 12, 2023
ROSAKubeControllerPolicy — Policy updated	ROSA updated the policy to allow the kube-cont roller-manager to describe	October 16, 2023

Change	Description	Date
	Availability Zones, Amazon EC2 instances, route tables, security groups, VPCs, and subnets. To learn more, see ROSAKubeControllerPolicy.	
ROSAManageSubscription — Policy updated	ROSA updated the policy to add the ROSA with hosted control planes ProductId. To learn more, see ROSAManag eSubscription.	August 1, 2023
ROSAKubeControllerPolicy — Policy updated	ROSA updated the policy to allow the kube-controller-manager to create Network Load Balancers as Kubernete s service load balancers.  Network Load Balancers provide greater ability to handle volatile workloads and support static IP addresses for the load balancer. To learn more, see ROSAKubeC ontrollerPolicy.	July 13, 2023

Change	Description	Date
ROSANodePoolManage mentPolicy — New policy added	ROSA added a new policy to allow the NodePool controller to describe, run, and terminate Amazon EC2 instances managed as worker nodes. This policy also enables disk encryption of the worker node root volume using AWS KMS keys. To learn more, see ROSANodePoolManagementPolicy.	June 8, 2023
ROSAInstallerPolicy — New policy added	ROSA added a new policy to allow the installer to manage AWS resources that support cluster installation. To learn more, see ROSAInstallerPolic Y.	June 6, 2023
ROSASRESupportPolicy — New policy added	ROSA added a new policy to allow Red Hat SREs to directly observe, diagnose and support AWS resources associated with ROSA clusters, including the ability to change ROSA cluster node state. To learn more, see ROSASRESupportPolicy.	June 1, 2023

Change	Description	Date
ROSAKMSProviderPolicy — New policy added	ROSA added a new policy to allow the built-in AWS Encryption Provider to manage AWS KMS keys to support etcd data encryptio n. To learn more, see ROSAKMSProviderPolicy.	April 27, 2023
ROSAKubeControllerPolicy — New policy added	ROSA added a new policy to allow the kube controlle r to manage Amazon EC2, Elastic Load Balancing, and AWS KMS resources for ROSA with hosted control planes clusters. To learn more, see ROSAKubeControllerPolicy.	April 27, 2023
ROSAImageRegistryO peratorPolicy — New policy added	ROSA added a new policy to allow the Image Registry Operator to provision and manage resources for the ROSA in-cluster image registry and dependent services, including S3. To learn more, see ROSAImage RegistryOperatorPolicy.	April 27, 2023
ROSAControlPlaneOp eratorPolicy — New policy added	ROSA added a new policy to allow the Control Plane Operator to manage Amazon EC2 and Route 53 resources for ROSA with hosted control planes clusters. To learn more, see ROSAControl PlaneOperatorPolicy.	April 24, 2023

Change	Description	Date
ROSACloudNetworkCo nfigOperatorPolicy — New policy added	ROSA added a new policy to allow the Cloud Network Config Controller Operator to provision and manage networking resources for the ROSA cluster networking overlay. To learn more, see ROSACloudNetworkConfigOperatorPolicy.	April 20, 2023
ROSAIngressOperatorPolicy — New policy added	ROSA added a new policy to allow the Ingress Operator to provision and manage load balancers and DNS configurations for ROSA clusters. To learn more, see ROSAIngressOperatorPolicy.	April 20, 2023
ROSAAmazonEBSCSIDr iverOperatorPolicy — New policy added	ROSA added a new policy to allow the Amazon EBS CSI Driver Operator to install and maintain the Amazon EBS CSI driver on a ROSA cluster. To learn more, see ROSAAmazo nEBSCSIDriverOperatorPolicy.	April 20, 2023
ROSAWorkerInstancePolicy — New policy added	ROSA added a new policy to allow the service to manage cluster resources. To learn more, see ROSAWorke rInstancePolicy.	April 20, 2023

Change	Description	Date
ROSAManageSubscription – New policy added	ROSA added a new policy to grant the AWS Marketpla ce permissions required to manage the ROSA subscript ion. To learn more, see ROSAManageSubscription.	April 11, 2022
Red Hat OpenShift Service on AWS started tracking changes	Red Hat OpenShift Service on AWS started tracking changes for its AWS managed policies.	March 2, 2022

## AWS managed policies for ROSA with HCP account roles



#### Note

These AWS managed policies are intended for use by ROSA with hosted control planes, an upcoming service update. ROSA clusters created with the existing architecture will continue to use customer managed IAM policies. For more information about customer managed IAM policies used by ROSA, see About IAM resources for ROSA clusters that use AWS STS.

These AWS managed policies add permissions used by ROSA IAM roles. The permissions are required for Red Hat site reliability engineering (SRE) support on the cluster, cluster creation, and compute functionality.

## **Topics**

- AWS managed policy: ROSAWorkerInstancePolicy
- AWS managed policy: ROSASRESupportPolicy
- AWS managed policy: ROSAInstallerPolicy

## AWS managed policy: ROSAWorkerInstancePolicy

You can attach ROSAWorkerInstancePolicy to your IAM entities. Before you create a ROSA with hosted control planes cluster, you must first attach this policy to a worker IAM role.

#### **Permissions details**

This policy includes the following permissions that allow the ROSA service to complete the following tasks:

 ec2 — Review AWS Region and Amazon EC2 instance details as part of the lifecycle management of worker nodes in a ROSA cluster.

To view the full JSON policy document, see <u>ROSAWorkerInstancePolicy</u> in the *AWS Managed Policy Reference Guide*.

#### AWS managed policy: ROSASRESupportPolicy

You can attach ROSASRESupportPolicy to your IAM entities.

Before you create a ROSA with hosted control planes cluster, you must first attach this policy to a support IAM role. This policy grants required permissions to Red Hat site reliability engineers (SREs) to directly observe, diagnose, and support AWS resources associated with ROSA clusters, including the ability to change ROSA cluster node state.

#### **Permissions details**

This policy includes the following permissions that allow Red Hat SREs to complete the following tasks:

- cloudtrail Read AWS CloudTrail events and trails relevant to the cluster.
- cloudwatch Read Amazon CloudWatch metrics relevant to the cluster.
- ec2 Read, describe, and review Amazon EC2 components related to the cluster's health such as security groups, VPC endpoint connections, and volume status. Launch, stop, reboot, and terminate Amazon EC2 instances.
- elasticloadbalancing Read, describe, and review Elastic Load Balancing parameters related to the cluster's health.
- iam Evaluate IAM roles that relate to the cluster's health.
- route53 Review DNS settings related to the cluster's health.
- sts DecodeAuthorizationMessage Read IAM messages for debugging purposes.

To view the full JSON policy document, see <u>ROSASRESupportPolicy</u> in the *AWS Managed Policy Reference Guide*.

## AWS managed policy: ROSAInstallerPolicy

You can attach ROSAInstallerPolicy to your IAM entities.

Before you create a ROSA with hosted control planes cluster, you must first attach this policy to an IAM role named [Prefix]-ROSA-Worker-Role. This policy allows entities to add any role that follows the [Prefix]-ROSA-Worker-Role pattern to an instance profile. This policy grants necessary permissions to the installer to manage AWS resources that support ROSA cluster installation.

#### **Permissions details**

This policy includes the following permissions that allow the installer to complete the following tasks:

- ec2 Run Amazon EC2 instances using AMIs hosted in AWS accounts owned and managed by Red Hat. Describe Amazon EC2 instances, volumes, and network resources associated with Amazon EC2 nodes. This is required so that the Kubernetes control plane can join instances to a cluster. This is also required so that the cluster can evaluate its presence within Amazon VPC.
- elasticloadbalancing Add load balancers to target nodes on a cluster. Remove load balancers from target nodes on a cluster. This permission is required so that the Kubernetes control plane can dynamically provision load balancers requested by Kubernetes services and OpenShift application services.
- kms Read an AWS KMS key, create and manage grants to Amazon EC2, and return a unique symmetric data key for use outside of AWS KMS. This is required for the use of encrypted etcd data when etcd encryption is enabled at cluster creation.
- iam Validate IAM roles and policies. Dynamically provision and manage Amazon EC2 instance profiles relevant to the cluster. Add tags to an IAM instance profile by using the iam: TagInstanceProfile permission. Provide installer error messages when cluster installation fails due to a missing customer-specified cluster OIDC provider.
- route53 Manage Route 53 resources needed to create clusters.
- servicequotas Evaluate service quotas required to create a cluster.
- sts Create temporary AWS STS credentials for ROSA components. Assume the credentials for cluster creation.
- secretsmanager Read a secret value to securely allow customer-managed OIDC configuration as part of cluster provisioning.

To view the full JSON policy document, see ROSAInstallerPolicy in the AWS Managed Policy Reference Guide.

## AWS managed policies for ROSA with HCP operator roles



#### Note

These AWS managed policies are intended for use by ROSA with hosted control planes (HCP), an upcoming service update. ROSA clusters created with the existing architecture will continue to use customer managed IAM policies. For more information about customer managed IAM policies used by ROSA, see About IAM resources for ROSA clusters that use AWS STS.

These AWS managed policies add permissions used by ROSA IAM roles. The permissions are required for OpenShift operators on the ROSA with HCP cluster to manage cluster nodes.

## **Topics**

- AWS managed policy: ROSAAmazonEBSCSIDriverOperatorPolicy
- AWS managed policy: ROSAIngressOperatorPolicy
- AWS managed policy: ROSAImageRegistryOperatorPolicy
- AWS managed policy: ROSACloudNetworkConfigOperatorPolicy
- AWS managed policy: ROSAKubeControllerPolicy
- AWS managed policy: ROSANodePoolManagementPolicy
- AWS managed policy: ROSAKMSProviderPolicy
- AWS managed policy: ROSAControlPlaneOperatorPolicy

## AWS managed policy: ROSAAmazonEBSCSIDriverOperatorPolicy

You can attach ROSAAmazonEBSCSIDriverOperatorPolicy to your IAM entities. You must attach this policy to an operator IAM role to allow a ROSA with hosted control planes cluster to make calls to other AWS services. A unique set of operator roles is required for each cluster.

This policy grants necessary permissions to the Amazon EBS CSI Driver Operator to install and maintain the Amazon EBS CSI driver on a ROSA cluster. For more information about the operator, see aws-ebs-csi-driver operator in the OpenShift GitHub documentation.

#### **Permissions details**

This policy includes the following permissions that allow the Amazon EBS Driver Operator to complete the following tasks:

• ec2 — Create, modify, attach, detach, and delete Amazon EBS volumes that are attached to Amazon EC2 instances. Create and delete Amazon EBS volume snapshots and list Amazon EC2 instances, volumes, and snapshots.

To view the full JSON policy document, see <u>ROSAAmazonEBSCSIDriverOperatorPolicy</u> in the *AWS Managed Policy Reference Guide*.

## AWS managed policy: ROSAIngressOperatorPolicy

You can attach ROSAIngressOperatorPolicy to your IAM entities. You must attach this policy to an operator IAM role to allow a ROSA with hosted control planes cluster to make calls to other AWS services. A unique set of operator roles is required for each cluster.

This policy grants required permissions to the Ingress Operator to provision and manage load balancers and DNS configurations for ROSA clusters. The policy allows read access to tag values. The operator then filters the tag values for Route 53 resources to discover hosted zones. For more information about the operator, see <a href="OpenShift Ingress Operator">OpenShift Ingress Operator</a> in the OpenShift GitHub documentation.

#### **Permissions details**

This policy includes the following permissions that allow the Ingress Operator to complete the following tasks:

- elasticloadbalancing Describe the state of provisioned load balancers.
- route53 List Route 53 hosted zones and edit records that manage the DNS controlled by the ROSA cluster.
- tag Manage tagged resources by using the tag: GetResources permission.

To view the full JSON policy document, see <u>ROSAIngressOperatorPolicy</u> in the *AWS Managed Policy Reference Guide*.

## AWS managed policy: ROSAImageRegistryOperatorPolicy

You can attach ROSAImageRegistryOperatorPolicy to your IAM entities. You must attach this policy to an operator IAM role to allow a ROSA with hosted control planes cluster to make calls to other AWS services. A unique set of operator roles is required for each cluster.

This policy grants required permissions to the Image Registry Operator to provision and manage resources for the ROSA in-cluster image registry and dependent services, including S3. This is required so that the operator can install and maintain the internal registry of a ROSA cluster. For more information about the operator, see <a href="Image Registry Operator">Image Registry Operator</a> in the OpenShift GitHub documentation.

#### **Permissions details**

This policy includes the following permissions that allow the Image Registry Operator to complete the following actions:

• s3 — Manage and evaluate Amazon S3 buckets as persistent storage for container image content and cluster metadata.

To view the full JSON policy document, see <u>ROSAImageRegistryOperatorPolicy</u> in the *AWS Managed Policy Reference Guide*.

## AWS managed policy: ROSACloudNetworkConfigOperatorPolicy

You can attach ROSACloudNetworkConfigOperatorPolicy to your IAM entities. You must attach this policy to an operator IAM role to allow a ROSA with hosted control planes cluster to make calls to other AWS services. A unique set of operator roles is required for each cluster.

This policy grants required permissions to the Cloud Network Config Controller Operator to provision and manage networking resources for the ROSA cluster networking overlay. The operator uses these permissions to manage private IP addresses for Amazon EC2 instances as part of the ROSA cluster. For more information about the operator, see <a href="Cloud-network-config-controller">Cloud-network-config-controller</a> in the OpenShift GitHub documentation.

#### **Permissions details**

This policy includes the following permissions that allow the Cloud Network Config Controller Operator to complete the following tasks:

• ec2 — Read, assign, and describe configurations for connecting Amazon EC2 instances, Amazon VPC subnets, and elastic network interfaces in a ROSA cluster.

To view the full JSON policy document, see <u>ROSACloudNetworkConfigOperatorPolicy</u> in the *AWS Managed Policy Reference Guide*.

## AWS managed policy: ROSAKubeControllerPolicy

You can attach ROSAKubeControllerPolicy to your IAM entities. You must attach this policy to an operator IAM role to allow a ROSA with hosted control planes cluster to make calls to other AWS services. A unique set of operator roles is required for each cluster.

This policy grants required permissions to the kube controller to manage Amazon EC2, Elastic Load Balancing, and AWS KMS resources for a ROSA with hosted control planes cluster. For more information about this controller, see Controller architecture in the OpenShift documentation.

#### **Permissions details**

This policy includes the following permissions that allow the kube controller to complete the following tasks:

- ec2 Create, delete, and add tags to Amazon EC2 instance security groups. Add inbound rules
  to security groups. Describe Availability Zones, Amazon EC2 instances, route tables, security
  groups, VPCs, and subnets.
- elasticloadbalancing Create and manage load balancers and their policies, create
  and manage load balancer listeners, register targets with target groups and manage target
  groups, register and de-register Amazon EC2 instances with a load balancer, and add tags to load
  balancers.
- kms Retrieve detailed information about an AWS KMS key. This is required for the use of encrypted etcd data when etcd encryption is enabled at cluster creation.

To view the full JSON policy document, see <u>ROSAKubeControllerPolicy</u> in the *AWS Managed Policy Reference Guide*.

## AWS managed policy: ROSANodePoolManagementPolicy

You can attach ROSANodePoolManagementPolicy to your IAM entities.

Before you create a ROSA with hosted control planes cluster, you must first attach this policy to an IAM role named [Prefix]-ROSA-Worker-Role. A unique set of operator roles is required for each cluster.

This policy grants required permissions to the NodePool controller to describe, run, and terminate Amazon EC2 instances managed as worker nodes. This policy also grants permissions to allow for disk encryption of the worker node root volume using AWS KMS keys. For more information about this controller, see Controller architecture in the OpenShift documentation.

#### **Permissions details**

This policy includes the following permissions that allow the NodePool controller to complete the following tasks:

- ec2 Run Amazon EC2 instances using AMIs hosted in AWS accounts owned and managed by Red Hat. Manage EC2 lifecycles in the ROSA cluster. Dynamically create and integrate worker nodes with Elastic Load Balancing, Amazon VPC, Route 53, Amazon EBS, and Amazon EC2.
- iam Use Elastic Load Balancing via the service-linked role named
   AWSServiceRoleForElasticLoadBalancing. Assign roles to Amazon EC2 instance profiles.
- kms Read an AWS KMS key, create and manage grants to Amazon EC2, and return a unique symmetric data key for use outside of AWS KMS. This is required to allow for disk encryption of the worker node root volume.

To view the full JSON policy document, see <u>ROSANodePoolManagementPolicy</u> in the *AWS Managed Policy Reference Guide*.

## AWS managed policy: ROSAKMSProviderPolicy

You can attach ROSAKMSProviderPolicy to your IAM entities. You must attach this policy to an operator IAM role to allow a ROSA with hosted control planes cluster to make calls to other AWS services. A unique set of operator roles is required for each cluster.

This policy grants required permissions to the built-in AWS Encryption Provider to manage AWS KMS keys that support etcd data encryption. This policy allows Amazon EC2 to use KMS keys that the AWS Encryption Provider provides to encrypt and decrypt etcd data. For more information about this provider, see AWS Encryption Provider in the Kubernetes GitHub documentation.

#### **Permissions details**

This policy includes the following permissions that allow the AWS Encryption Provider to complete the following tasks:

• kms — Encrypt, decrypt, and retrieve an AWS KMS key. This is required for the use of encrypted etcd data when etcd encryption is enabled at cluster creation.

To view the full JSON policy document, see <u>ROSAKMSProviderPolicy</u> in the *AWS Managed Policy Reference Guide*.

## AWS managed policy: ROSAControlPlaneOperatorPolicy

You can attach ROSAControlPlaneOperatorPolicy to your IAM entities. You must attach this policy to an operator IAM role to allow a ROSA with hosted control planes cluster to make calls to other AWS services. A unique set of operator roles is required for each cluster.

This policy grants required permissions to the Control Plane Operator to manage Amazon EC2 and Route 53 resources for ROSA with hosted control planes clusters. For more information about this operator, see Controller architecture in the OpenShift documentation.

#### **Permissions details**

This policy includes the following permissions that allow the Control Plane Operator to complete the following tasks:

- ec2 Create and manage Amazon VPC endpoints.
- route53 List and change Route 53 record sets and list hosted zones.

To view the full JSON policy document, see <u>ROSAControlPlaneOperatorPolicy</u> in the *AWS Managed Policy Reference Guide*.

## **Resilience in ROSA**

## AWS global infrastructure resilience

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected through low-latency, high-throughput, and highly redundant networking. With Availability Zones, you

Resilience 114

can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

ROSA provides customers with the option to run the Kubernetes control plane and data plane in a single AWS Availability Zone, or across multiple Availability Zones. While Single-AZ clusters can be useful for experimentation, customers are encouraged to run their workloads in more than one Availability Zone. This ensures that applications can withstand even a complete Availability Zone failure - a very rare event in itself.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

## **ROSA** cluster resilience

The ROSA control plane consists of at least three OpenShift control plane nodes. Each control plane node is made up of an API server instance, an etcd instance, and controllers. In the event of a control plane node failure, all API requests are automatically routed to the other available nodes to ensure cluster availability.

The ROSA data plane consists of at least two OpenShift infrastructure nodes and two OpenShift worker nodes. Infrastructure nodes run pods that support OpenShift cluster infrastructure components such as the default router, the built-in OpenShift registry, and the components for cluster metrics and monitoring. OpenShift worker nodes run end-user application pods.

Red Hat site reliability engineers (SREs) fully manage the control plane and infrastructure nodes. Red Hat SREs proactively monitor the ROSA cluster, and are responsible for replacing any failed control plane nodes and infrastructure nodes. For more information, see Overview of responsibilities for ROSA.



#### Important

Because ROSA is a managed service, Red Hat is responsible for managing the underlying AWS infrastructure that ROSA uses. Customers should not attempt to manually shut down the Amazon EC2 instances that ROSA uses from the AWS console or AWS CLI. This action can lead to customer data loss.

If a worker node fails on the data plane, the control plane relocates unscheduled pods to the functioning worker node(s) until the failed node is recovered or replaced. Failed worker nodes can

ROSA cluster resilience 115 be replaced manually or automatically by enabling automatic scaling of machines in a cluster. For more information, see Cluster autoscaling in the Red Hat documentation.

## **Customer-deployed application resilience**

Although ROSA provides many protections to ensure high availability of the service, customers are responsible for building their deployed applications for high availability to protect workloads against downtime. For more information, see <a href="About availability for ROSA">About availability for ROSA</a> in the Red Hat documentation.

## Infrastructure security in ROSA

As a managed service, Red Hat OpenShift Service on AWS is protected by the AWS global network security. For information about AWS security services and how AWS protects infrastructure, see <a href="AWS Cloud Security">AWS Cloud Security</a>. To design your AWS environment using the best practices for infrastructure security, see <a href="Infrastructure Protection">Infrastructure Protection</a> in Security Pillar — AWS Well-Architected Framework.

You use AWS published API calls to access ROSA through the AWS network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the <u>AWS Security Token Service</u> (AWS STS) to generate temporary security credentials to sign requests.

## Cluster network isolation

Red Hat site reliability engineers (SREs) are responsible for the ongoing management and network security of the cluster and underlying application platform. For more information on Red Hat responsibilities for ROSA, see Overview of responsibilities for ROSA.

When you create a new cluster, ROSA provides the option of creating a public Kubernetes API server endpoint and application routes or a private Kubernetes API endpoint and application routes. This connection is used to communicate with your cluster (using OpenShift management

tools such as the ROSA CLI and OpenShift CLI). A private connection allows all communication between your nodes and the API server to stay within your VPC. If you enable private access to the API server and application routes, you must use an existing VPC and AWS PrivateLink to connect the VPC to the OpenShift backend service.

Kubernetes API server access is secured using a combination of AWS Identity and Access Management (IAM) and native Kubernetes role-based access control (RBAC). For more information on Kubernetes RBAC, see Using RBAC Authorization in the Kubernetes documentation.

ROSA allows you to create secured application routes using several types of TLS termination to serve certificates to the client. For more information, see <u>Secured routes</u> in the Red Hat documentation.

If you create a ROSA cluster in an existing VPC, you specify the VPC subnets and Availability Zones for your cluster to use. You also define the CIDR ranges for the cluster network to use, and match these CIDR ranges to the VPC subnets. For more information, see <u>CIDR range definitions</u> in the Red Hat documentation.

For clusters that use the public API endpoint, ROSA requires that your VPC is configured with a public and private subnet for each Availability Zone that you want the cluster deployed into. For clusters that use the private API endpoint, only private subnets are required.

If you are using an existing VPC, you can configure your ROSA clusters to use an HTTP or HTTPS proxy server during or after cluster creation to encrypt cluster web traffic, adding another layer of security for your data. When you enable a proxy, the core cluster components are denied direct access to the internet. The proxy does not deny internet access for user workloads. For more information, see Configuring a cluster-wide proxy in the Red Hat documentation.

## Pod network isolation

If you are a cluster administrator, you can define network policies at the pod level that restrict traffic to pods in your ROSA cluster. For more information, see <a href="Network policy">Network policy</a> in the Red Hat documentation.

Pod network isolation 117

## **ROSA** service quotas

Red Hat OpenShift Service on AWS (ROSA) uses service quotas for Amazon EC2, Amazon Virtual Private Cloud (Amazon VPC), Amazon Elastic Block Store (Amazon EBS), and Elastic Load Balancing (ELB) to provision clusters.

## **Required minimum quotas for ROSA**

For the following Amazon EC2 and Amazon EBS quotas, ROSA requires a higher quota than the default service provides. To use ROSA, you may need to request an increase for these quotas. For more information, see Requesting a quota increase in the Service Quotas User Guide.

#### Important

For On-Demand Standard (A, C, D, H, I, M, R, T, Z) Amazon EC2 instances, the default value of 5 vCPUs is not sufficient to create ROSA clusters. ROSA requires 100 vCPUs or greater for cluster creation. To increase this quota, open the Service Quotas console and request a quota increase.

#### Note

You can check your quotas using the AWS SDKs, but the SDK calculation does not include existing ROSA resources. The quota check in the SDK may pass, and ROSA cluster creation may fail. To fix this issue, open the Service Quotas console and request a quota increase.

Name	Service code	Default	Minimum required	Adjustable	Description
Running On- Demand Standard (A, C, D, H, I, M, R, T, Z) instances	ec2	5	100	<u>Yes</u>	Maximum number of vCPUs assigned to the Running On-Demand

Name	Service code	Default	Minimum required	Adjustable	Description
					Standard (A, C, D, H, I, M, R, T, Z) instances.  The default value of 5 vCPUs is not sufficient to create ROSA clusters. ROSA requires 100 vCPUs for cluster creation.

Name	Service code	Default	Minimum required	Adjustable	Description
Storage for General Purpose SSD (gp3) volumes, in TiB	ebs	50	300	Yes	The maximum aggregated amount of storage, in TiB, that can be provision ed across General Purpose SSD (gp3) volumes in this Region.  300 TiB of storage is required for optimal performance.

Name	Service code	Default	Minimum required	Adjustable	Description
Storage for General Purpose SSD (gp2) volumes, in TiB	ebs	50	300	Yes	The maximum aggregated amount of storage, in TiB, that can be provision ed across General Purpose SSD (gp2) volumes in this Region.  300 TiB of storage is required for optimal performance.

Name	Service code	Default	Minimum required	Adjustable	Description
Storage for Provision ed IOPS SSD (io1) volumes, in TiB	ebs	50	300	Yes	The maximum aggregated amount of storage, in TiB, that can be provision ed across Provision ed IOPS SSD (io1) volumes in this Region.  300 TiB of storage is required for optimal performance.

## Note

The default values are the initial quotas set by AWS, which are separate from the actual applied quota value and maximum possible service quota. For more information, see Terminology in Service Quotas in the Service Quotas User Guide.

# **Default quotas for ROSA**

ROSA uses the following default quotas for Amazon EC2, Amazon VPC, Amazon EBS, and Elastic Load Balancing. For information on increasing quotas, see Requesting a quota increase in the Service Quotas User Guide.

Default quotas for ROSA 122

#### Amazon EC2

• EC2-VPC Elastic IPs

#### **Amazon VPC**

- VPCs per Region
- Network interfaces per Region
- Internet gateways per Region

#### **Amazon EBS**

- Snapshots per Region
- IOPS for Provisioned IOPS SSD (io1) volumes

## **Elastic Load Balancing**

- Application Load Balancers per Region
- Classic Load Balancers per Region

Default quotas for ROSA 123

## **AWS services integrated with ROSA**

ROSA works with other AWS services to provide additional solutions for your business challenges. This topic identifies services that either use ROSA to add functionality, or services that ROSA uses to perform tasks.

#### **Topics**

How ROSA works with AWS Marketplace

## How ROSA works with AWS Marketplace

AWS Marketplace is a curated digital catalog that you can use to find, buy, deploy, and manage third-party software, data, and services that you need to build solutions and run your business. AWS Marketplace simplifies software licensing and procurement with flexible pricing options and multiple deployment methods.

ROSA uses AWS Marketplace for service metering and billing. ROSA classic is metered and billed through an AWS Marketplace Amazon Machine Image (AMI)-based product, whereas ROSA with hosted control planes (HCP) is metered and billed through an AWS Marketplace software as a service (SaaS)-based product.

This page explains how ROSA works with AWS Marketplace for payments, billing, subscriptions, and contract purchases.

## **Terminology**

This page uses the following terms when discussing ROSA's integration with AWS Marketplace.

## Amazon Machine Image (AMI)

An image of a server, including an operating system and additional software, that runs on AWS.

## **AMI** subscription

In AWS Marketplace, AMI-based software products such as ROSA classic use an hourly with annual subscription pricing model. Hourly pricing is the default pricing model, but you have the option to purchase a year's worth of usage upfront for one Amazon EC2 instance type.

ROSA and AWS Marketplace 124

#### SaaS subscription

In AWS Marketplace, software-as-a-service (SaaS) products such as ROSA with HCP adopt a usage-based subscription model. The software seller tracks your usage and you pay only for what you use.

#### **Public offer**

Public offers allow you to purchase AWS Marketplace software and services directly from the AWS Management Console.

#### **Private offer**

Private offers are a purchasing program that allow sellers and buyers to negotiate custom prices and end user licensing agreement (EULA) terms for purchases in AWS Marketplace.

#### **ROSA** service fees

Fees that ROSA charges for the OpenShift software and cluster management by Red Hat site reliability engineers (SREs). ROSA service fees are metered through AWS Marketplace and appear on your AWS bill.

#### AWS infrastructure fees

Standard fees that AWS charges for the AWS services underlying ROSA clusters, including Amazon EC2, Amazon EBS, Amazon S3, and Elastic Load Balancing. Fees are metered through the AWS service being used and appear on your AWS bill.

## **ROSA** payments and billing

ROSA integrates with AWS Marketplace to enable metering and billing of ROSA service fees. ROSA service fees cover access to OpenShift software and cluster management by Red Hat site reliability engineers (SREs). ROSA service fees are uniform across all supported AWS standard Regions. ROSA with HCP service fees accrue on demand by default at a flat hourly rate based on the number of running clusters and worker node vCPUs running in those clusters. ROSA classic service fees accrue on demand based on the number of worker node vCPUs. ROSA classic does not charge service fees for the control plane or required infrastructure nodes.

ROSA customers also pay standard AWS infrastructure fees for the AWS services underlying ROSA clusters, including Amazon EC2, Amazon EBS, Amazon S3, and Elastic Load Balancing. AWS infrastructure fees are a separate billing item from the ROSA service fees that are metered through

ROSA payments and billing 125

AWS Marketplace. AWS infrastructure fees vary by AWS Region and are based on hourly usage by default. For additional AWS infrastructure cost savings, you can purchase Amazon EC2 savings plans or reserved instances. For more information, see Compute Savings Plans and Reserved Instances in the Amazon EC2 User Guide.

ROSA does not charge fees until you create a ROSA cluster, or purchase a ROSA contract. For more information, see Red Hat OpenShift Service on AWS pricing.

You can view ROSA service fees and AWS infrastructure fees and manage payments in the AWS Billing console. You can also view your costs and monitor usage using the AWS Cost Explorer Service interface free of charge. For more information, see Viewing your bill in the AWS Billing and Cost Management User Guide and Analyzing your costs with AWS Cost Explorer Service in the AWS Cost Management User Guide.

## Subscribing to ROSA Marketplace listings through the console

When you enable ROSA in the ROSA console, your AWS account is subscribed to the ROSA classic and ROSA with HCP listings on AWS Marketplace. There is no fee for enabling ROSA subscriptions.

For AWS Organizations users, ROSA allows you to share ROSA classic subscriptions with other accounts in your organization. For more information, see Sharing subscriptions in an organization in the AWS Marketplace Buyer Guide.

#### **ROSA** contracts

ROSA uses AWS Marketplace to provide optional contracts for ROSA with HCP and ROSA classic. Contracts provide savings on ROSA worker node service fees. ROSA contracts do not impact the fees charged for AWS infrastructure.

## 12-month contracts

You can purchase 12-month public offer contracts for ROSA classic and ROSA with HCP from the ROSA console.



#### Note

ROSA classic must be enabled on your account before you can purchase 12-month contracts from the console.



#### Note

12-month contracts cannot be transferred to a private offer.

#### Purchasing a ROSA classic 12-month contract

When you purchase a ROSA classic 12-month contract, you make a upfront payment for an annual term and pay no hourly service fee for the next 12 months for the covered instances. Contract cost is based on the Amazon EC2 instance type and number of instances that you select. The contract does not cover the AWS infrastructure fees that ROSA charges for the underlying AWS services that are used. For more information, see Red Hat OpenShift Service on AWS Pricing.

The contract covers only the instance types that you specify during contract creation (m5.xlarge for example). You can purchase additional 12-month contracts for cost savings on more than one Amazon EC2 instance type. Usage outside of your 12-month contract incurs ROSA service fees at the on-demand rate.



#### Note

ROSA classic 12-month contracts do not auto renew.

#### To purchase a 12-month contract for ROSA classic



#### Note

If you are using the ROSA console in a Region that does not yet support ROSA with HCP, this workflow is not yet available. For a list of Regions that support ROSA with HCP, see Differences between ROSA with HCP and ROSA classic.

To purchase ROSA classic contracts in Regions without ROSA with HCP support, go to the ROSA console and choose Purchase a software contract and view existing contracts.

- 1. Go to the ROSA console.
- 2. On the left navigation pane, choose **Contracts**.
- 3. Choose Contracts for ROSA classic.

- 4. Choose Purchase contract.
- 5. Select the EC2 instance type and number of instances that you need.
- 6. Choose **Review contract**.
- 7. Review the contract details and choose **Purchase contract**.



#### Note

ROSA 12-month contracts cannot be downgraded or cancelled after creation using the console. If you need to downgrade or cancel the contract during the active contract duration, go to AWS Support Center and open a support case.

#### Purchasing a ROSA with HCP 12-month contract

When you enable ROSA with HCP in the console, a no-cost 12-month ROSA with HCP contract is initially created on your account to facilitate on-demand billing. If you choose to purchase a ROSA with HCP contract to save on worker node service fees, the initial contract is modified to cover usage costs for the worker node vCPUs and control planes that you specify.

When you purchase a ROSA with HCP 12-month contract, you make an upfront payment for an annual term and pay no hourly usage fee for the next 12 months for the covered worker node vCPUs and control planes. Contract cost is based on the number of worker node vCPUs and control planes that you select. The contract covers only the worker node vCPUs and control planes that you specify during contract creation. The contract does not cover the AWS infrastructure fees that ROSA charges for the underlying AWS services that are used. For more information, see Red Hat OpenShift Service on AWS Pricing.

#### Monthly usage quota

Upon purchase, your prepaid vCPUs and control planes are converted to a monthly usage quota. Hourly on-demand usage rates apply for vCPU and control plane usage that exceeds the monthly quota. ROSA with HCP uses the following formulas to calculate the monthly quota associated with the contract:

- Worker node vCPUs: number of vCPUs x 24 hours x 365 days / 12 months
- Control planes: number of control planes x 24 hours x 365 days / 12 months

For example, a purchase of 4,000 worker node vCPUs and 8 control planes would convert to a monthly guota of 2,920,000 worker node vCPU hours and 5,840 control plane hours consumable per month.

#### To purchase a ROSA with HCP 12-month contract



#### Note

If you are using the Red Hat OpenShift Service on AWS console in a Region that doesn't yet support ROSA with hosted control planes, this workflow is not yet available. For a list of Regions that support ROSA with HCP, see Differences between ROSA with HCP and ROSA classic.

- 1. Go to the ROSA console.
- 2. On the left navigation pane, choose **Contracts**.
- 3. Choose Contracts for ROSA with HCP.
- 4. Choose Purchase contract.
- 5. Enter the number of vCPUs to purchase. Specify in multiples of 4.
- 6. Enter the number of control planes to purchase.
- 7. Choose **Review contract**.
- 8. Review the contract details and choose **Purchase contract**.



#### Note

ROSA 12-month contracts cannot be downgraded or cancelled after creation using the console. If you need to downgrade or cancel the contract during the active contract duration, go to AWS Support Center and open a support case.

#### **Upgrading a ROSA with HCP 12-month contract**

You can upgrade your active ROSA with HCP 12-month contract at any time with additional worker node vCPUs and control planes. When you upgrade your ROSA with HCP 12-month contract, you make an upfront prorated payment for the added resources. Prorated amounts are calculated based on the number of days remaining on the contract. The contract covers only the worker node

vCPUs and control planes that you specify during contract creation. Contract upgrades do not impact the fees charged for AWS infrastructure.

Upon upgrade, the added vCPUs and control planes are converted to a monthly usage quota using the same formulas as the original contract purchase. Hourly on-demand usage rates apply for vCPU and control plane usage that exceeds the monthly quota. For more information, see Monthly usage quota.

#### To upgrade a ROSA with HCP 12-month contract

- 1. Go to the ROSA console.
- 2. On the left navigation pane, choose **Contracts**.
- 3. Choose Contracts for ROSA with HCP.
- 4. Choose **Upgrade**.
- 5. Enter the number of vCPUs to add. Specify in multiples of 4.
- 6. Enter the number of control planes to add to the contract.
- 7. Choose **Review upgrade**.
- 8. Review the contract details and choose **Purchase upgrade**.



#### Note

ROSA classic 12-month contracts cannot be upgraded. Additional 12-month ROSA classic contracts can be purchased at any time using the ROSA console.

## Obtaining a private offer

You can request an AWS Marketplace private offer for ROSA with HCP or ROSA classic to receive product pricing and end user licensing agreement (EULA) terms negotiated with Red Hat. For more information, see Private offers in the AWS Marketplace Buyer Guide.

#### To obtain a ROSA private offer



#### Note

If you are an AWS Organizations user and received a private offer that was issued to your payer and member accounts, follow the procedure below to subscribe to ROSA directly on each account in your organization.

If you receive a ROSA classic private offer that was only issued to the AWS Organizations payer account, you will need to share the subscription with members accounts in your organization. For more information, see Sharing subscriptions in an organization in the AWS Marketplace Buyer Guide.

- 1. Once a private offer has been issued, sign in to the AWS Marketplace console.
- 2. Open the email with a ROSA private offer link.
- 3. Follow the link to directly access the private offer.



#### Note

Following this link before logging in to the correct account will result in a **Page note** found (404) error.

- 4. Review the terms and conditions.
- 5. Choose **Accept terms**.



#### Note

If an AWS Marketplace private offer is not accepted, the ROSA service fees from AWS Marketplace will continue to be billed at the public hourly rate.

- 6. To verify the offer details, select **Show details** in the product listing.
- 7. To get started using ROSA, choose **Continue to configuration**. You will be redirected to the ROSA console.

## **Private Marketplace**

Private Marketplace enables administrators to build customized digital catalogs of approved products from AWS Marketplace. Administrators can create unique sets of vetted software

Private Marketplace 131 available in AWS Marketplace for AWS organizational units or different AWS accounts within their organization to purchase.

If your organization uses a private marketplace, an administrator must add the AWS Marketplace listings for ROSA to the private marketplace before users can enable the service. For more information, see Getting started with private marketplace in the AWS Marketplace Buyer Guide.

Private Marketplace 132

## **Troubleshooting**

The following documentation covers how to troubleshoot issues that might occur when enabling ROSA and provisioning ROSA clusters.

#### **Topics**

- Support for ROSA
- Troubleshoot ROSA enablement errors in the ROSA console
- Troubleshoot ROSA cluster creation issues
- Troubleshoot non-STS ROSA cluster issues

## **Support for ROSA**

With ROSA, you can receive troubleshooting support from AWS Support and the Red Hat support teams. Support cases can be opened with either organization, and are routed to the correct team to resolve your issue.

## **AWS Support**

An AWS Developer Support plan is required to open ROSA technical cases, but an AWS Business or Enterprise On-Ramp Support plan is recommended for continuous access to ROSA technical support and architectural guidance. Red Hat uses the AWS Support API to open cases for customers when necessary. AWS Business Support and AWS Enterprise On-Ramp enable continuous phone, web, and chat access to support engineers. For more information about AWS Support plans, see AWS Support.

For steps to enable an AWS Support plan, see How do I sign up for an AWS Support plan?

For information about creating an AWS Support case, see <u>Creating support cases and case</u> management.

## **Red Hat Support**

ROSA includes Red Hat Premium Support. To receive Red Hat Premium Support, navigate to the <u>Red Hat Customer Portal</u> and use the support case tool to create a support ticket. For more information, see How to engage with Red Hat support.

Support for ROSA 133

## Troubleshoot ROSA enablement errors in the ROSA console

ROSA uses AWS Marketplace to facilitate subscription management, billing, and metering. When you enable ROSA, the AWS ROSA console page subscribes to a ROSA listing on AWS Marketplace. If your IAM principal is missing required aws-marketplace subscription permissions when you enable ROSA in the ROSA console, the console generates an error message.

This section covers how to troubleshoot AWS Marketplace subscription permission issues that you might encounter when you choose **Enable ROSA** in the ROSA console.

With ROSA, you can also receive troubleshooting support from AWS Support and the Red Hat support teams. For more information, see Support for ROSA.

#### **Topics**

- AWS Organizations service control policy (SCP) is denying required AWS Marketplace permissions
- User or role does not have the required AWS Marketplace permissions
- Required AWS Marketplace permissions blocked by an administrator

# AWS Organizations service control policy (SCP) is denying required AWS Marketplace permissions

## **Description**

If your AWS Organizations service control policy (SCP) isn't configured to allow the required aws-marketplace:Subscribe permission when you choose **Enable ROSA**, the ROSA console generates the following error message: An error occurred while enabling Red Hat OpenShift Service on AWS (ROSA), because an AWS Organizations Service Control Policy (SCP) is denying required permissions. Contact your AWS Organizations management account administrator, and consult the documentation for troubleshooting.

#### Solution

Your organization's management account administrator can enable ROSA in the organization's management account. Once ROSA is enabled, they can then use AWS License Manager to distribute the AWS Marketplace license entitlement for ROSA to member accounts within your organization.

Contact your account administrator and request that they distribute and activate the ROSA entitlement for your account.

## User or role does not have the required AWS Marketplace permissions

## Description

If your IAM principal doesn't have the required aws-marketplace: Subscribe permission when you choose **Enable ROSA**, the ROSA console generates the following error message: An error occurred while enabling Red Hat OpenShift Service on AWS (ROSA), because your user or role does not have the required permissions.

#### Solution

- Navigate to the <u>ROSA console</u> and attach the AWS managed policy ROSAManageSubscription to your IAM identity. For more information about ROSAManageSubscription, see <u>AWS</u> <u>managed policy: ROSAManageSubscription</u>.
- 2. Choose **Get started**.
- 3. On the **Verify ROSA prerequisites** page, select **I agree to share my contact information with Red Hat**.
- 4. Choose Enable ROSA.

If you don't have permission to view or to update your permission set in IAM, contact your AWS account administrator and ask them to enable ROSA for your account.

## Required AWS Marketplace permissions blocked by an administrator

## **Description**

If your account administrator blocked the required aws-marketplace: Subscribe permission, the ROSA console generates the following error message when you choose **Enable ROSA**: An error occurred while enabling Red Hat OpenShift Service on AWS (ROSA), because required permissions have been blocked by an administrator. ROSAManageSubscription, an AWS managed policy, includes the permissions required to enable ROSA. Consult the documentation and try again.

#### Solution

Contact your AWS account administrator and ask them to take the following action:

- Navigate to the ROSA console.
- 2. Choose Get started.
- 3. On the **Verify ROSA prerequisites** page, select **I agree to share my contact information with Red Hat**.
- 4. Choose Enable ROSA. This action enables ROSA for all IAM identities under the AWS account.

## **Troubleshoot ROSA cluster creation issues**

This section contains solutions to issues you may have when creating ROSA clusters.

With ROSA, you can also receive troubleshooting support from AWS Support and the Red Hat support teams. For more information, see Support for ROSA.

#### **Topics**

- Access ROSA cluster debug logs
- Elastic Load Balancing (ELB) role does not exist
- ROSA cluster fails AWS service quota check during cluster creation
- Troubleshoot ROSA CLI expired offline access tokens

## Access ROSA cluster debug logs

To begin to troubleshoot issues with your application, first review the debug logs. The ROSA CLI debug logs provide details on the error messages that are produced when a cluster fails to create.

To display cluster debug information, run the following ROSA CLI command. In the command, replace <cluster\_name> with the name of your cluster.

```
rosa describe cluster -c <cluster_name> --debug
```

## Elastic Load Balancing (ELB) role does not exist

## Description

If you didn't create a load balancer in your AWS account, the AWSServiceRoleForElasticLoadBalacing role might have not been created. If you

ROSA cluster creation issues 136

don't configure the Elastic Load Balancing role correctly and attempt to create a ROSA cluster, the following error message is returned: Error creating network Load Balancer: AccessDenied.

#### Solution

1. Check if your account has the AWSServiceRoleForElasticLoadBalancing role.

```
aws iam get-role --role-name "AWSServiceRoleForElasticLoadBalancing"
```

2. If you don't have this role, create the role by running the following command.

```
aws iam get-role --role-name "AWSServiceRoleForElasticLoadBalancing" || aws iam
create-service-linked-role --aws-service-name "elasticloadbalancing.amazonaws.com"
```

## ROSA cluster fails AWS service quota check during cluster creation

## **Description**

To use ROSA, the service quotas for your account may need increased. For more information, see ROSA service quotas.

#### **Solution**

1. Run the following command to identify your account's quotas.

rosa verify quota



#### Note

Quotas are different in different AWS Regions. Make sure to verify each of the quotas for your Regions.

- 2. If you need to increase your quota, navigate to the Service Quotas console.
- 3. On the navigation pane, choose **AWS services**.
- 4. Choose the service that needs a quota increase.
- 5. Select the quota that needs to be increased and choose **Request quota increase**.

6. For **Request quota increase**, enter the total amount that you want the quota to be and choose Request.

## **Troubleshoot ROSA CLI expired offline access tokens**

## Description

If you use the ROSA CLI and your api.openshift.com offline access token expires, an error message appears. This happens when sso.redhat.com invalidates the token.

#### Solution

- 1. Navigate to the OpenShift Cluster Manager API Token page and choose **Load Token**.
- 2. Copy and paste the following authentication command in the terminal.

```
rosa login --token="<api_token>"
```

## Troubleshoot non-STS ROSA cluster issues

This section covers how to troubleshoot issues that you might encounter when provisioning non-STS ROSA clusters.

We recommend that you provision ROSA clusters using AWS Security Token Service (STS) shortlived credentials for better security protection. For more information about provisioning ROSA STS clusters, see Getting started with ROSA using AWS STS in auto mode.

With ROSA, you can also receive troubleshooting support from AWS Support or the Red Hat support teams. For more information, see Support for ROSA.

## Failed to create a cluster with an osdCcsAdmin error



#### Note

This error occurs only when you use the non-STS method of provisioning ROSA clusters. To avoid this issue, provision your ROSA clusters using AWS STS. For more information, see Getting started with ROSA using AWS STS in auto mode.

## **Description**

If your cluster fails to create, you might receive the following error message:

Failed to create cluster: Unable to create cluster spec: Failed to get access keys for user 'osdCcsAdmin': NoSuchEntity: The user with name osdCcsAdmin cannot be found.

## **Solution**

1. Delete the stack.

rosa init --delete-stack

2. Reinitialize your account.

rosa init

# **Document history for the ROSA User Guide**

The following table covers all of the documentation updates for ROSA.

Change	Description	Date
ROSA with HCP AWS Region expansion	ROSA with hosted control planes (HCP) is now available in the Asia Pacific (Hong Kong) AWS Region.	April 8, 2024
ROSA with HCP AWS Region expansion	ROSA with hosted control planes (HCP) is now available in the South America (São Paulo) AWS Region.	April 1, 2024
ROSA with HCP AWS Region expansion	ROSA with hosted control planes (HCP) is now available in the Middle East (Bahrain) AWS Region.	March 25, 2024
ROSA with HCP AWS Region expansion	ROSA with hosted control planes (HCP) is now available in the Asia Pacific (Seoul) AWS Region.	March 14, 2024
ROSA with HCP AWS Region expansion	ROSA with hosted control planes (HCP) is now available in the Africa (Cape Town) AWS Region.	March 5, 2024
Updated ROSAInstallerPolicy	Updated the AWS managed policy ROSAInstallerPolicy.	January 26, 2024
Updated ROSASRESu pportPolicy	Updated the AWS managed policy ROSASRESupportPolicy.	January 22, 2024

<u>Updated ROSAImage</u> <u>RegistryOperatorPolicy</u>	Updated the AWS managed policy ROSAImageRegistryO peratorPolicy.	December 12, 2023
<u>Updated ROSAKubeController</u> <u>Policy</u>	Updated the AWS managed policy ROSAKubeController Policy.	October 16, 2023
<u>Updated ROSAManag</u> <u>eSubscription</u>	Updated the AWS managed policy ROSAManageSubscrip tion.	August 1, 2023
Updated ROSAKubeController Policy	Updated the AWS managed policy ROSAKubeController Policy.	July 13, 2023
Added ROSA security pages	Resilience in ROSA, Infrastru cture security in ROSA, and Data protection in ROSA pages were added.	June 30, 2023
Added the deployment options page	Deployment options page was added.	June 9, 2023
Added new AWS managed policy ROSANodePoolManage mentPolicy	New AWS managed policy ROSANodePoolManage mentPolicy was added.	June 8, 2023
Added new AWS managed policy ROSAInstallerPolicy	New AWS managed policy ROSAInstallerPolicy was added.	June 6, 2023
Added new AWS managed policy ROSASRESupportPolicy	New AWS managed policy ROSASRESupportPolicy was added.	June 1, 2023
Added Overview of responsib ilities for ROSA	Added Overview of responsib ilities for ROSA page.	May 26, 2023

Updated What is Red Hat OpenShift Service on AWS?	Updated the What is Red Hat OpenShift Service on AWS page.	May 24, 2023
Added new AWS managed policies for ROSA operator roles	New AWS managed policies ROSAImageRegistryO peratorPolicy, ROSAKubeC ontrollerPolicy, and ROSAKMSProviderPolicy were added.	April 27, 2023
Added new AWS managed policy ROSAControlPlaneOp eratorPolicy	New AWS managed policy ROSAControlPlaneOp eratorPolicy was added.	April 24, 2023
Added new AWS managed policies for ROSA account roles	New AWS managed policy pages for ROSA account and operator roles page were added.	April 20, 2023
Added the ROSA service quotas page	The ROSA service quotas page was added	December 22, 2022
Added troubleshooting pages	Troubleshooting pages were added	November 1, 2022
Added getting started pages	Getting started pages were added	August 12, 2022
Added new AWS managed policy ROSAManageSubscription	New AWS managed policy ROSAManageSubscription was added.	April 11, 2022
<u>Initial release</u>	The initial release of the Red Hat OpenShift Service on AWS User Guide	March 24, 2021