Red Hat OpenShift Service on AWS 4

Introduction to ROSA

An overview of Red Hat OpenShift Service on AWS architecture

Last Updated: 2022-08-12

Red Hat OpenShift Service on AWS 4 Introduction to ROSA

An overview of Red Hat OpenShift Service on AWS architecture

Legal Notice

Abstract

This document provides an overview of the platform and application architecture in Red Hat OpenShift Service on AWS (ROSA).

Table of Contents

CHAPTER 1. UNDERSTANDING ROSA

Learn about Red Hat OpenShift Service on AWS (ROSA), interacting with ROSA using Red Hat OpenShift Cluster Manager and command-line interface (CLI) tools, consumption experience, and integration with Amazon Web Services (AWS) services.

1.1. ABOUT ROSA

ROSA is a fully-managed, turnkey application platform that allows you to focus on delivering value to your customers by building and deploying applications. Red Hat and AWS Site reliability engineering (SRE) experts manage the underlying platform so you do not have to worry about the complexity of infrastructure management. ROSA provides seamless integration with a wide range of AWS compute, database, analytics, machine learning, networking, mobile, and other services to further accelerate the building and delivering of differentiating experiences to your customers.

You subscribe to the service directly from your AWS account. After the clusters are created, you can operate your clusters with the OpenShift web console or through Red Hat OpenShift Cluster Manager. The ROSA service also uses OpenShift APIs and command-line interface (CLI) tools. These tools provide a standardized OpenShift experience to use your existing skills and tools knowledge.

You receive OpenShift updates with new feature releases and a shared, common source for alignment with OpenShift Container Platform. ROSA supports the same versions of OpenShift as Red Hat OpenShift Dedicated and OpenShift Container Platform to achieve version consistency.

1.2. CREDENTIAL MODES

TIP

AWS Security Token Service (STS) is the recommended credential mode for installing and interacting with clusters on Red Hat OpenShift Service on AWS (ROSA) because it provides enhanced security.

There are two supported credential modes for ROSA clusters. One uses the AWS Security Token Service (STS), which is recommended, and the other uses Identity Access Management (IAM) roles.

1.2.1. ROSA with STS

AWS STS is a global web service that provides short-term credentials for IAM or federated users. ROSA with STS is the recommended credential mode for ROSA clusters. You can use AWS STS with ROSA to allocate temporary, limited-privilege credentials for component-specific IAM roles. The service enables cluster components to make AWS API calls using secure cloud resource management practices.

You can use the **rosa** CLI to create the IAM role, policy, and identity provider resources that are required for ROSA clusters that use STS.

AWS STS aligns with principles of least privilege and secure practices in cloud service resource management. The **rosa** CLI manages the STS credentials that are assigned for unique tasks and takes action upon AWS resources as part of OpenShift functionality. One limitation of using STS is that roles must be created for each ROSA cluster.

The STS credential mode is more secure because:

- It supports an explicit and limited set of roles and policies that you create ahead of time, and tracks every permission asked for and every role used.

- The service is limited to the set permissions.
- When the service is run, it obtains credentials that expire in one hour, so there is no need to rotate or revoke credentials. The expiration also reduces the risks of credentials leaking and being reused.

A listing of the account-wide and per-cluster roles is provided in About IAM resources for ROSA clusters that use STS.

1.2.2. ROSA without STS

This mode makes use of a pre-created IAM user with **AdministratorAccess** within the account that has proper permissions to create other roles and resources as needed. Using this account the service creates all the necessary resources that are needed for the cluster.

1.3. BILLING AND PRICING

ROSA is billed directly to your AWS account. ROSA pricing can be consumption based, with annual commitments or three-year commitments for greater discounting. The total cost of ROSA consists of two components:

- ROSA service fees
- AWS infrastructure fees

Visit the AWS pricing page for more details.

1.4. GETTING STARTED

To get started with deploying your cluster, ensure your AWS account has met the prerequisites, you have a Red Hat account ready, and follow the procedures outlined in Getting started with Red Hat OpenShift Service on AWS.

Additional resources

- OpenShift Cluster Manager
- About IAM resources for ROSA clusters that use STS
- Getting started with Red Hat OpenShift Service on AWS
- AWS pricing page

CHAPTER 2. ROSA ARCHITECTURE

2.1. ARCHITECTURE CONCEPTS

Learn about OpenShift and container basic concepts used in Red Hat OpenShift Service on AWS architecture.

2.1.1. OpenShift

OpenShift is a Kubernetes container platform that provides a trusted environment to run enterprise workloads. It extends the Kubernetes platform with built-in software to enhance app lifecycle development, operations, and security. With OpenShift, you can consistently deploy your workloads across hybrid cloud providers and environments.

2.1.2. Kubernetes

Red Hat OpenShift Service on AWS (ROSA) uses Red Hat OpenShift, which is an enterprise Kubernetes platform. Kubernetes is an open source platform for managing containerized workloads and services across multiple hosts, and offers management tools for deploying, automating, monitoring, and scaling containerized apps with minimal to no manual intervention. For complete information about Kubernetes, see the Kubernetes documentation.

Cluster, compute pool, and compute node

A Kubernetes cluster consists of a control plane and one or more compute nodes. Compute nodes are organized into compute pools of the type or profile of CPU, memory, operating system, attached disks, and other properties. The compute nodes correspond to the Kubernetes **Node** resource, and are managed by a Kubernetes control plane that centrally controls and monitors all Kubernetes resources in the cluster.

When you deploy the resources for a containerized app, the Kubernetes control plane decides which compute node to deploy those resources on, accounting for the deployment requirements and available capacity in the cluster. Kubernetes resources include services, deployments, and pods.

Namespace

Kubernetes namespaces are a way to divide your cluster resources into separate areas that you can deploy apps and restrict access to, such as if you want to share the cluster with multiple teams. For example, system resources that are configured for you are kept in separate namespaces like **kube-system**. If you do not designate a namespace when you create a Kubernetes resource, the resource is automatically created in the **default** namespace.

Pod

Every containerized app that is deployed into a cluster is deployed, run, and managed by a Kubernetes resource that is called a pod. Pods represent small deployable units in a Kubernetes cluster and are used to group the containers that must be treated as a single unit. In most cases, each container is deployed in its own pod. However, an app can require a container and other helper containers to be deployed into one pod so that those containers can be addressed by using the same private IP address.

App

An app can refer to a complete app or a component of an app. You can deploy components of an app in separate pods or separate compute nodes.

Service

A service is a Kubernetes resource that groups a set of pods and provides network connectivity to these pods without exposing the actual private IP address of each pod. You can use a service to make your app available within your cluster or to the public Internet.

Deployment

A deployment is a Kubernetes resource where you can specify information about other resources or capabilities that are required to run your app, such as services, persistent storage, or annotations. You configure a deployment in a configuration YAML file, and then apply it to the cluster. The Kubernetes main resource configures the resources and deploys containers into pods on the compute nodes with available capacity.
Define update strategies for your app, including the number of pods that you want to add during a rolling update and the number of pods that can be unavailable at a time. When you perform a rolling update, the deployment checks whether the update is working and stops the rollout when failures are detected.

A deployment is just one type of workload controller that you can use to manage pods.

2.1.3. Containers

Containers provide a standard way to package your application code, configurations, and dependencies into a single unit. Containers run as isolated processes on compute hosts and share the host operating system and its hardware resources. A container can be moved between environments and run without changes. Unlike virtual machines, containers do not virtualize a device, its operating system, and the underlying hardware. Only the app code, run time, system tools, libraries, and settings are packaged inside the container. This approach makes a container more lightweight, portable, and efficient than a virtual machine.

Built on existing Linux container technology (LXC), the OCI-compliant container images define templates for how to package software into standardized units that include all of the elements that an app needs to run. Red Hat OpenShift Service on AWS (ROSA) uses CRI-O as the container runtime to deploy containers to your cluster.

To run your app in Kubernetes on ROSA, you must first containerize your app by creating a container image that you store in a container registry.

Image

A container image is the base for every container that you want to run. Container images are built from a Dockerfile, a text file that defines how to build the image and which build artifacts to include in it, such as the app, the app configuration, and its dependencies. Images are always built from other images, making them quick to configure.

Registry

An image registry is a place to store, retrieve, and share container images. Images that are stored in a registry can either be publicly available (public registry) or accessible by a small group of users (private registry). ROSA offers public images that you can use to create your first containerized app. For enterprise applications, use a private registry to protect your images from being used by unauthorized users.

2.2. ARCHITECTURE MODELS

ROSA architecture supports the following network configuration types:

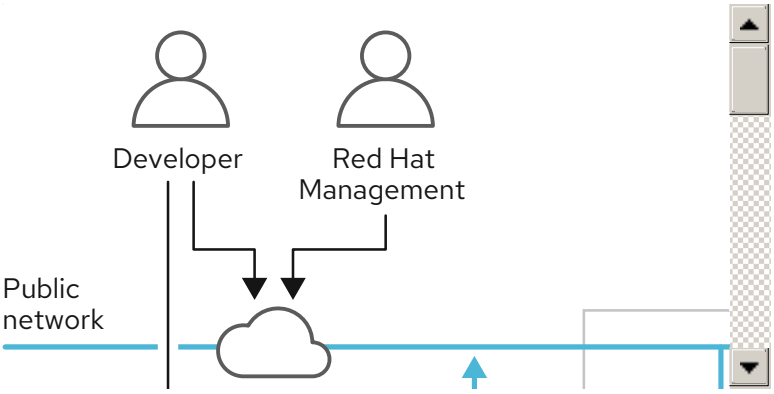
- Public network
- Private network

- AWS PrivateLink

2.2.1. ROSA architecture on public and private networks

You can install ROSA using either a public or private network. Configure a private cluster and private network connection during or after the cluster creation process. Red Hat manages the cluster with limited access through a public network. For more information, see the Service Definition.

Figure 2.1. ROSA deployed on public and private networks



Alternatively, install a cluster using AWS PrivateLink, which is hosted on private subnets only.

2.2.2. AWS PrivateLink architecture

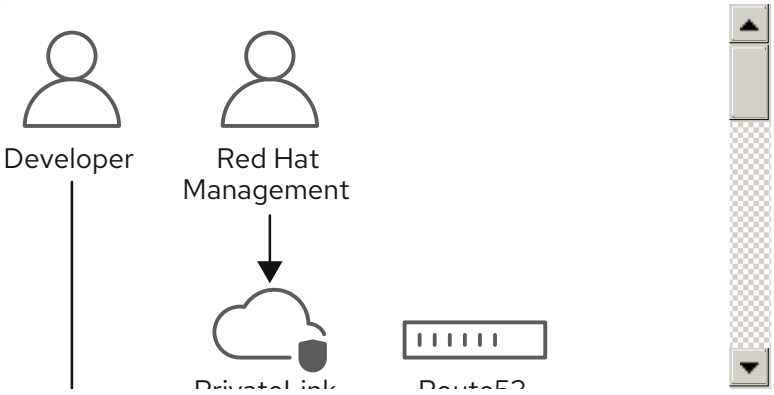
The Red Hat managed infrastructure that creates AWS PrivateLink clusters is hosted on private subnets. The connection between Red Hat and the customer-provided infrastructure is created through AWS PrivateLink VPC endpoints.

NOTE

AWS PrivateLink is supported on existing VPCs only.

The following diagram shows network connectivity of a PrivateLink cluster.

Figure 2.2. Multi-AZ AWS PrivateLink cluster deployed on private subnets



2.2.2.1. AWS reference architectures

AWS provides multiple reference architectures that can be useful to customers when planning how to set up a configuration that uses AWS PrivateLink. Here are three examples:

- VPC with a private subnet and AWS Site-to-Site VPN access.

This configuration enables you to extend your network into the cloud without exposing your network to the internet.

To enable communication with your network over an Internet Protocol Security (IPsec) VPN tunnel, this configuration contains a virtual private cloud (VPC) with a single private subnet and a virtual private gateway. Communication over the internet does not use an internet gateway.

For more information, see VPC with a private subnet only and AWS Site-to-Site VPN access in the AWS documentation.

- VPC with public and private subnets (NAT)
  This configuration enables you to isolate your network so that the public subnet is reachable from the internet but the private subnet is not.

  Only the public subnet can send outbound traffic directly to the internet. The private subnet can access the internet by using a network address translation (NAT) gateway that resides in the public subnet. This allows database servers to connect to the internet for software updates using the NAT gateway, but does not allow connections to be made directly from the internet to the database servers.

  For more information, see VPC with public and private subnets (NAT) in the AWS documentation.

- VPC with public and private subnets and AWS Site-to-Site VPN access
  This configuration enables you to extend your network into the cloud and to directly access the internet from your VPC.

  You can run a multi-tiered application with a scalable web front end in a public subnet, and house your data in a private subnet that is connected to your network by an IPsec AWS Site-to-Site VPN connection.

  For more information, see VPC with public and private subnets and AWS Site-to-Site VPN access in the AWS documentation.

CHAPTER 3. POLICIES AND SERVICE DEFINITION

3.1. ABOUT AVAILABILITY FOR RED HAT OPENSHIFT SERVICE ON AWS

Availability and disaster avoidance are extremely important aspects of any application platform. Although Red Hat OpenShift Service on AWS (ROSA) provides many protections against failures at several levels, customer-deployed applications must be appropriately configured for high availability. To account for outages that might occur with cloud providers, additional options are available such as deploying a cluster across multiple availability zones and maintaining multiple clusters with failover mechanisms.

3.1.1. Potential points of failure

Red Hat OpenShift Service on AWS (ROSA) provides many features and options for protecting your workloads against downtime, but applications must be architected appropriately to take advantage of these features.

ROSA can help further protect you against many common Kubernetes issues by adding Red Hat site reliability engineering (SRE) support and the option to deploy a multiple availability zone cluster, but there are a number of ways in which a container or infrastructure can still fail. By understanding potential points of failure, you can understand risks and appropriately architect both your applications and your clusters to be as resilient as necessary at each specific level.

NOTE

An outage can occur at several different levels of infrastructure and cluster components.

3.1.1.1. Container or pod failure

By design, pods are meant to exist for a short time. Appropriately scaling services so that multiple instances of your application pods are running can protect against issues with any individual pod or container. The OpenShift node scheduler can also make sure these workloads are distributed across different worker nodes to further improve resiliency.

When accounting for possible pod failures, it is also important to understand how storage is attached to your applications. Single persistent volumes attached to single pods cannot leverage the full benefits of pod scaling, whereas replicated databases, database services, or shared storage can.

To avoid disruption to your applications during planned maintenance, such as upgrades, it is important to define a Pod Disruption Budget. These are part of the Kubernetes API and can be managed with **oc** commands such as other object types. They allow for the specification of safety constraints on pods during operations, such as draining a node for maintenance.

3.1.1.2. Worker node failure

Worker nodes are the virtual machines that contain your application pods. By default, a ROSA cluster has a minimum of two worker nodes for a single availability-zone cluster. In the event of a worker node failure, pods are relocated to functioning worker nodes, as long as there is enough capacity, until any issue with an existing node is resolved or the node is replaced. More worker nodes means more protection against single-node outages, and ensures proper cluster capacity for rescheduled pods in the event of a node failure.

NOTE

When accounting for possible node failures, it is also important to understand how storage is affected. EFS volumes are not affected by node failure. However, EBS volumes are not accessible if they are connected to a node that fails.

3.1.1.3. Cluster failure

ROSA clusters have at least three control plane nodes and three infrastructure nodes that are preconfigured for high availability, either in a single zone or across multiple zones, depending on the type of cluster you have selected. Control plane and infrastructure nodes have the same resiliency as worker nodes, with the added benefit of being managed completely by Red Hat.

In the event of a complete control plane outage, the OpenShift APIs will not function, and existing worker node pods are unaffected. However, if there is also a pod or node outage at the same time, the control planes must recover before new pods or nodes can be added or scheduled.

All services running on infrastructure nodes are configured by Red Hat to be highly available and distributed across infrastructure nodes. In the event of a complete infrastructure outage, these services are unavailable until these nodes have been recovered.

3.1.1.4. Zone failure

A zone failure from AWS affects all virtual components, such as worker nodes, block or shared storage, and load balancers that are specific to a single availability zone. To protect against a zone failure, ROSA provides the option for clusters that are distributed across three availability zones, known as multiple availability zone clusters. Existing stateless workloads are redistributed to unaffected zones in the event of an outage, as long as there is enough capacity.

3.1.1.5. Storage failure

If you have deployed a stateful application, then storage is a critical component and must be accounted for when thinking about high availability. A single block storage PV is unable to withstand outages even at the pod level. The best ways to maintain availability of storage are to use replicated storage solutions, shared storage that is unaffected by outages, or a database service that is independent of the cluster.

3.2. RESPONSIBILITY ASSIGNMENT MATRIX

This documentation outlines Red Hat, cloud provider, and customer responsibilities for the Red Hat OpenShift Service on AWS (ROSA) managed service.

3.2.1. Overview of responsibilities for Red Hat OpenShift Service on AWS

While Red Hat and Amazon Web Services (AWS) manage the Red Hat OpenShift Service on AWS service, the customer shares certain responsibilities. The Red Hat OpenShift Service on AWS services are accessed remotely, hosted on public cloud resources, created in customer-owned AWS accounts, and have underlying platform and data security that is owned by Red Hat.

IMPORTANT

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 management	Change management	Identity and access management	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	Shared	Shared	Shared	Red Hat
Application networking	Shared	Shared	Shared	Red Hat	Red Hat
Cluster networking	Red Hat	Shared	Shared	Red Hat	Red Hat
Virtual networking	Shared	Shared	Shared	Shared	Shared
Control plane and infrastructure nodes	Red Hat	Red Hat	Red Hat	Red Hat	Red Hat
Worker nodes	Red Hat	Red Hat	Red Hat	Red Hat	Red Hat
Cluster version	Red Hat	Shared	Red Hat	Red Hat	Red Hat
Capacity management	Red Hat	Shared	Red Hat	Red Hat	Red Hat

Resource	Incident and operations management	Change management	Identity and access management	Security and regulation compliance	Disaster recovery
Virtual storage	Red Hat and cloud provider	Red Hat and cloud provider	Red Hat and cloud provider	Red Hat and cloud provider	Red Hat and cloud provider
Physical infrastructure and security	Cloud provider	Cloud provider	Cloud provider	Cloud provider	Cloud provider

3.2.2. Shared responsibility matrix

The customer, Red Hat, and Amazon Web Services (AWS) share responsibility for the monitoring and maintenance of an Red Hat OpenShift Service on AWS cluster. This documentation illustrates the delineation of responsibilities by area and task.

3.2.2.1. Incident and operations management

The customer is responsible for incident and operations management of customer application data and any custom networking the customer may have configured for the cluster network or virtual network.

Resource	Red Hat and AWS responsibilities	Customer responsibilities
Application networking	Monitor cloud load balancers and native OpenShift router service, and respond to alerts.	• Monitor health of service load balancer endpoints. • Monitor health of application routes, and the endpoints behind them. • Report outages to Red Hat.
Virtual networking	Monitor cloud load balancers, subnets, and public cloud components necessary for default platform networking, and respond to alerts.	Monitor network traffic that is optionally configured through VPC to VPC connection, VPN connection, or Direct connection for potential issues or security threats.

3.2.2.2. Change management

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 services, and worker nodes. The customer is responsible for initiating infrastructure change requests and installing and maintaining optional services and networking configurations on the cluster, as well as all changes to customer data and customer applications.

Resource	Red Hat responsibilities	Customer responsibilities
Logging	• Centrally aggregate and monitor platform audit logs. • Provide and maintain a logging Operator to enable the customer to deploy a logging stack for default application logging. • Provide audit logs upon customer request.	• Install the optional default application logging Operator on the cluster. • Install, configure, and maintain any optional app logging solutions, such as logging sidecar containers or third-party logging applications. • Tune size and frequency of application logs being 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	• Set up public cloud load balancers. Provide the ability to set up private load balancers and up to one additional load balancer when required. • Set up native OpenShift router service. Provide the ability to set the router as private and add up to one additional router shard. • Install, configure, and maintain OpenShift SDN components for default internal pod traffic. • Provide the ability for the customer to manage **NetworkPolicy** and **EgressNetworkPolicy** (firewall) objects.	• Configure non-default pod network permissions for project and pod networks, pod ingress, and pod egress using **NetworkPolicy** objects. • Use OpenShift Cluster Manager to request a private load balancer for default application routes. • Use OpenShift Cluster Manager to configure up to one additional public or private router shard and corresponding load balancer. • Request and configure any additional service load balancers for specific services. • Configure any necessary DNS forwarding rules.

Resource	Red Hat responsibilities	Customer responsibilities
Cluster networking	• Set up cluster management components, such as public or private service endpoints and necessary integration with virtual networking components. • Set up internal networking components required for internal cluster communication between worker, infrastructure, and control plane nodes.	• Provide optional non-default IP address ranges for machine CIDR, service CIDR, and pod CIDR if needed through OpenShift Cluster Manager when the cluster is provisioned. • Request that the API service endpoint be made public or private on cluster creation or after cluster creation through OpenShift Cluster Manager.
Virtual networking	• Set up and configure virtual networking components required to provision the cluster, including virtual private cloud, subnets, load balancers, Internet gateways, NAT gateways, etc. • Provide the ability for the customer to manage VPN connectivity with on-premises resources, VPC to VPC connectivity, and Direct connectivity as required through OpenShift Cluster Manager. • Enable customers to create and deploy public cloud load balancers for use with service load balancers.	• Set up and maintain optional public cloud networking components, such as VPC to VPC connection, VPN connection, or Direct connection. • Request and configure any additional service load balancers for specific services.
Cluster version	• Communicate schedule and status of upgrades for minor and maintenance versions. • Publish change logs and release notes for minor and maintenance upgrades.	• Work with Red Hat to establish maintenance start times for upgrades. • Test customer applications on minor and maintenance versions to ensure compatibility.

Resource	Red Hat responsibilities	Customer responsibilities
Capacity management	• Monitor the use of the control plane. Control planes include control plane nodes and infrastructure nodes. • Scale and resize control plane nodes to maintain quality of service. • Monitor the use of customer resources including network, storage and compute capacity. Where autoscaling features are not enabled alert customer for any changes required to cluster resources, such as new compute nodes to scale and additional storage.	• Use the provided OpenShift Cluster Manager controls to add or remove additional worker nodes as required. • Respond to Red Hat notifications regarding cluster resource requirements.

3.2.2.3. Identity and access management

The Identity and Access Management matrix 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	Red Hat responsibilities	Customer responsibilities
Logging	• Adhere to an industry standards-based tiered internal access process for platform audit logs. • Provide native OpenShift RBAC capabilities.	• Configure OpenShift RBAC to control access to projects and by extension a project's application logs. • For third-party or custom application logging solutions, the customer is responsible for access management.
Application networking	Provide native OpenShift RBAC and **dedicated-admin** capabilities.	• Configure OpenShift **dedicated-admin** and RBAC to control access to route configuration as required. • Manage organization administrators for Red Hat to grant access to OpenShift Cluster Manager. The cluster maanger is used to configure router options and provide service load balancer quota.

Resource	Red Hat responsibilities	Customer responsibilities
Cluster networking	• Provide customer access controls through OpenShift Cluster Manager. • Provide native OpenShift RBAC and **dedicated-admin** capabilities.	• Manage Red Hat organization membership of Red Hat accounts. • Manage organization administrators for Red Hat to grant access to OpenShift Cluster Manager. • Configure OpenShift **dedicated-admin** and RBAC to control access to route configuration as required.
Virtual networking	Provide customer access controls through OpenShift Cluster Manager.	Manage optional user access to public cloud components through OpenShift Cluster Manager.

3.2.2.4. Security and regulation compliance

The following are the responsibilities and controls related to compliance:

Resource	Red Hat responsibilities	Customer responsibilities
Logging	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.	Analyze application logs for security events. Send application logs to an external endpoint through logging sidecar containers or third-party logging applications if longer retention is required than is offered by the default logging stack.
Virtual networking	• Monitor virtual networking components for potential issues and security threats. • Leverage additional public cloud provider tools for additional monitoring and protection.	• Monitor optional configured virtual networking components for potential issues and security threats. • Configure any necessary firewall rules or data center protections as required.

3.2.2.5. Disaster recovery

Disaster recovery includes data and configuration backup, replicating data and configuration to the disaster recovery environment, and failover on disaster events.

Resource	Red Hat responsibilities	Customer responsibilities
Virtual networking	Restore or recreate affected virtual network components that are necessary for the platform to function.	• Configure virtual networking connections with more than one tunnel where possible for protection against outages as recommended by the public cloud provider. • Maintain failover DNS and load balancing if using a global load balancer with multiple clusters.

3.2.3. 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, Red Hat provides various tools to help the customer manage data and applications on the platform.

Resource	Red Hat responsibilities	Customer responsibilities
Customer data	• Maintain platform-level standards for data encryption. • Provide OpenShift components to help manage application data, such as secrets. • Enable integration with third-party data services, AWS RDS, to store and manage data outside of the cluster and cloud provider.	Maintain responsibility for all customer data stored on the platform and how customer applications consume and expose this data.

Resource	Red Hat responsibilities	Customer responsibilities
Customer applications	• Provision clusters with OpenShift components installed so that customers can access the OpenShift and Kubernetes APIs to deploy and manage containerized applications. • Create clusters with image pull secrets so that customer deployments can pull images from the Red Hat Container Catalog registry. • Provide access to OpenShift APIs that a customer can use to set up Operators to add community, third-party, and Red Hat services to the cluster. • Provide storage classes and plug-ins to support persistent volumes for use with customer applications.	• Maintain responsibility for customer and third-party applications, data, and their complete lifecycle. • 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. • 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 their highly available and resilient workloads. • Maintain responsibility for monitoring the applications run on Red Hat OpenShift Service on AWS, including installing and operating software to gather metrics and create alerts.

3.3. RED HAT OPENSHIFT SERVICE ON AWS SERVICE DEFINITION

This documentation outlines the service definition for the Red Hat OpenShift Service on AWS (ROSA) managed service.

3.3.1. Account management

This section provides information about the service definition for Red Hat OpenShift Service on AWS account management.

3.3.1.1. Billing

Red Hat OpenShift Service on AWS is billed through Amazon Web Services (AWS) based on the usage of AWS components used by the service, such as load balancers, storage, EC2 instances, other components, and Red Hat subscriptions for the OpenShift service.

Any additional Red Hat software must be purchased separately.

3.3.1.2. Cluster self-service

Customers can self-service their clusters, including, but not limited to:

- Create a cluster
- Delete a cluster
- Add or remove an identity provider
- Add or remove a user from an elevated group
- Configure cluster privacy
- Add or remove machine pools and configure autoscaling
- Define upgrade policies

These tasks can be self-serviced using the **rosa** CLI utility.

3.3.1.3. Compute

Single availability zone clusters require a minimum of 3 control planes, 2 infrastructure nodes, and 2 worker nodes deployed to a single availability zone.

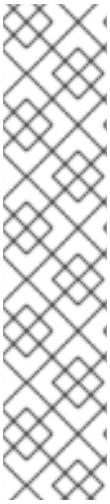
Multiple availability zone clusters require a minimum of 3 control planes. 3 infrastructure nodes, and 3 worker nodes. Additional nodes must be purchased in multiples of three to maintain proper node distribution.

All Red Hat OpenShift Service on AWS clusters support a maximum of 180 worker nodes.

NOTE

The **Default** machine pool node type and size cannot be changed after the cluster is created.

Control plane and infrastructure nodes are deployed and managed by Red Hat. Shutting down the underlying infrastructure through the cloud provider console is unsupported and can lead to data loss. There are at least 3 control plane nodes that handle etcd- and API-related workloads. There are at least 2 infrastructure nodes that handle metrics, routing, the web console, and other workloads. You must not run any workloads on the control and infrastructure nodes. Any workloads you intend to run must be deployed on worker nodes. See the Red Hat Operator support section below for more information about Red Hat workloads that must be deployed on worker nodes.



NOTE

Approximately one vCPU core and 1 GiB of memory are reserved on each worker node and removed from allocatable resources. This reservation of resources is necessary to run processes required by the underlying platform. These processes include system daemons such as udev, kubelet, and container runtime among others. The reserved resources also account for kernel reservations.

OpenShift Container Platform core systems such as audit log aggregation, metrics collection, DNS, image registry, SDN, and others might consume additional allocatable resources to maintain the stability and maintainability of the cluster. The additional resources consumed might vary based on usage.

For additional information, see the Kubernetes documentation.

IMPORTANT

As of the Red Hat OpenShift Service on AWS versions 4.8.35, 4.9.26, 4.10.6, the Red Hat OpenShift Service on AWS default per-pod pid limit is **4096**. If you want to enable this PID limit, you must upgrade your Red Hat OpenShift Service on AWS clusters to these versions or later. Red Hat OpenShift Service on AWS clusters with prior versions use a default PID limit of **1024**.

You cannot configure the per-pod PID limit on any Red Hat OpenShift Service on AWS cluster.

Additional Resources

- Red Hat Operator Support

3.3.1.4. AWS compute types

Red Hat OpenShift Service on AWS offers the following worker node types and sizes:

Example 3.1. General purpose compute types

- m5.xlarge (4 vCPU, 16 GiB)
- m5.2xlarge (8 vCPU, 32 GiB)
- m5.4xlarge (16 vCPU, 64 GiB)
- m5.8xlarge (32 vCPU, 128 GiB)
- m5.12xlarge (48 vCPU, 192 GiB)
- m5.16xlarge (64 vCPU, 256 GiB)
- m5.24xlarge (96 vCPU, 384 GiB)
- m5d.xlarge (4 vCPU, 16 GiB)
- m5d.2xlarge (8 vCPU, 32 GiB)
- m5d.4xlarge (16 vCPU, 64 GiB)

- m5d.8xlarge (32 vCPU, 128 GiB)
- m5d.12xlarge (48 vCPU, 192 GiB)
- m5d.16xlarge (64 vCPU, 256 GiB)
- m5d.24xlarge (96 vCPU, 384 GiB)
- m5n.xlarge (4 vCPU, 16 GiB)
- m5n.2xlarge (8 vCPU, 32 GiB)
- m5n.4xlarge (16 vCPU, 64 GiB)
- m5n.8xlarge (32 vCPU, 128 GiB)
- m5n.12xlarge (48 vCPU, 192 GiB)
- m5n.16xlarge (64 vCPU, 256 GiB)
- m5n.24xlarge (96 vCPU, 384 GiB)
- m5dn.xlarge (4 vCPU, 16 GiB)
- m5dn.2xlarge (8 vCPU, 32 GiB)
- m5dn.4xlarge (16 vCPU, 64 GiB)
- m5dn.8xlarge (32 vCPU, 128 GiB)
- m5dn.12xlarge (48 vCPU, 192 GiB)
- m5dn.16xlarge (64 vCPU, 256 GiB)
- m5dn.24xlarge (96 vCPU, 384 GiB)
- m5zn.xlarge (4 vCPU, 16 GiB)
- m5zn.2xlarge (8 vCPU, 32 GiB)
- m5zn.3xlarge (12 vCPU, 48 GiB)
- m5zn.6xlarge (24 vCPU, 96 GiB)
- m5zn.12xlarge (48 vCPU, 192 GiB)
- m6i.xlarge (4 vCPU, 16 GiB)
- m6i.2xlarge (8 vCPU, 32 GiB)
- m6i.4xlarge (16 vCPU, 64 GiB)
- m6i.8xlarge (32 vCPU, 128 GiB)
- m6i.12xlarge (48 vCPU, 192 GiB)
- m6i.16xlarge (64 vCPU, 256 GiB)

- m6i.24xlarge (96 vCPU, 384 GiB)
- m6i.32xlarge (128 vCPU, 512 GiB)

Example 3.2. Burstable general purpose compute types

- t3.xlarge (4 vCPU, 16 GiB)
- t3.2xlarge (8 vCPU, 32 GiB)
- t3a.xlarge (4 vCPU, 16 GiB)
- t3a.2xlarge (8 vCPU, 32 GiB)

Example 3.3. Memory-optimized compute types

- r4.xlarge (4 vCPU, 30.5 GiB)
- r4.2xlarge (8 vCPU, 61 GiB)
- r4.4xlarge (16 vCPU, 122 GiB)
- r4.8xlarge (32 vCPU, 244 GiB)
- r4.16xlarge (64 vCPU, 488 GiB)
- r5.xlarge (4 vCPU, 32 GiB)
- r5.2xlarge (8 vCPU, 64 GiB)
- r5.4xlarge (16 vCPU, 128 GiB)
- r5.8xlarge (32 vCPU, 256 GiB)
- r5.12xlarge (48 vCPU, 384 GiB)
- r5.16xlarge (64 vCPU, 512 GiB)
- r5.24xlarge (96 vCPU, 768 GiB)
- r5a.xlarge (4 vCPU, 32 GiB)
- r5a.2xlarge (8 vCPU, 64 GiB)
- r5a.4xlarge (16 vCPU, 128 GiB)
- r5a.8xlarge (32 vCPU, 256 GiB)
- r5a.12xlarge (48 vCPU, 384 GiB)
- r5a.16xlarge (64 vCPU, 512 GiB)
- r5a.24xlarge (96 vCPU, 768 GiB)
- r5ad.xlarge (4 vCPU, 32 GiB)

- r5ad.2xlarge (8 vCPU, 64 GiB)
- r5ad.4xlarge (16 vCPU, 128 GiB)
- r5ad.8xlarge (32 vCPU, 256 GiB)
- r5ad.12xlarge (48 vCPU, 384 GiB)
- r5ad.16xlarge (64 vCPU, 512 GiB)
- r5ad.24xlarge (96 vCPU, 768 GiB)
- r5d.xlarge (4 vCPU, 32 GiB)
- r5d.2xlarge (8 vCPU, 64 GiB)
- r5d.4xlarge (16 vCPU, 128 GiB)
- r5d.8xlarge (32 vCPU, 256 GiB)
- r5d.12xlarge (48 vCPU, 384 GiB)
- r5d.16xlarge (64 vCPU, 512 GiB)
- r5d.24xlarge (96 vCPU, 768 GiB)
- r5n.xlarge (4 vCPU, 32 GiB)
- r5n.2xlarge (8 vCPU, 64 GiB)
- r5n.4xlarge (16 vCPU, 128 GiB)
- r5n.8xlarge (32 vCPU, 256 GiB)
- r5n.12xlarge (48 vCPU, 384 GiB)
- r5n.16xlarge (64 vCPU, 512 GiB)
- r5n.24xlarge (96 vCPU, 768 GiB)
- r5dn.xlarge (4 vCPU, 32 GiB)
- r5dn.2xlarge (8 vCPU, 64 GiB)
- r5dn.4xlarge (16 vCPU, 128 GiB)
- r5dn.8xlarge (32 vCPU, 256 GiB)
- r5dn.12xlarge (48 vCPU, 384 GiB)
- r5dn.16xlarge (64 vCPU, 512 GiB)
- r5dn.24xlarge (96 vCPU, 768 GiB)
- r6i.xlarge (4 vCPU, 32 GiB)
- r6i.2xlarge (8 vCPU, 64 GiB)

- r6i.4xlarge (16 vCPU, 128 GiB)
- r6i.8xlarge (32 vCPU, 256 GiB)
- r6i.12xlarge (48 vCPU, 384 GiB)
- r6i.16xlarge (64 vCPU, 512 GiB)
- r6i.24xlarge (96 vCPU, 768 GiB)
- r6i.32xlarge (128 vCPU, 1,024 GiB)
- z1d.xlarge (4 vCPU, 32 GiB)
- z1d.2xlarge (8 vCPU, 64 GiB)
- z1d.3xlarge (12 vCPU, 96 GiB)
- z1d.6xlarge (24 vCPU, 192 GiB)
- z1d.12xlarge (48 vCPU, 384 GiB)

Example 3.4. Compute-optimized compute types

- c5.xlarge (4 vCPU, 8 GiB)
- c5.2xlarge (8 vCPU, 16 GiB)
- c5.4xlarge (16 vCPU, 32 GiB)
- c5.9xlarge (36 vCPU, 72 GiB)
- c5.12xlarge (48 vCPU, 96 GiB)
- c5.18xlarge (72 vCPU, 144 GiB)
- c5.24xlarge (96 vCPU, 192 GiB)
- c5d.xlarge (4 vCPU, 8 GiB)
- c5d.2xlarge (8 vCPU, 16 GiB)
- c5d.4xlarge (16 vCPU, 32 GiB)
- c5d.9xlarge (36 vCPU, 72 GiB)
- c5d.12xlarge (48 vCPU, 96 GiB)
- c5d.18xlarge (72 vCPU, 144 GiB)
- c5d.24xlarge (96 vCPU, 192 GiB)
- c5a.xlarge (4 vCPU, 8 GiB)
- c5a.2xlarge (8 vCPU, 16 GiB)
- c5a.4xlarge (16 vCPU, 32 GiB)

- c5a.8xlarge (32 vCPU, 64 GiB)
- c5a.12xlarge (48 vCPU, 96 GiB)
- c5a.16xlarge (64 vCPU, 128 GiB)
- c5a.24xlarge (96 vCPU, 192 GiB)
- c5ad.xlarge (4 vCPU, 8 GiB)
- c5ad.2xlarge (8 vCPU, 16 GiB)
- c5ad.4xlarge (16 vCPU, 32 GiB)
- c5ad.8xlarge (32 vCPU, 64 GiB)
- c5ad.12xlarge (48 vCPU, 96 GiB)
- c5ad.16xlarge (64 vCPU, 128 GiB)
- c5ad.24xlarge (96 vCPU, 192 GiB)
- c5n.xlarge (4 vCPU, 10.5 GiB)
- c5n.2xlarge (8 vCPU, 21 GiB)
- c5n.4xlarge (16 vCPU, 42 GiB)
- c5n.9xlarge (36 vCPU, 96 GiB)
- c5n.18xlarge (72 vCPU, 192 GiB)
- c6i.xlarge (4 vCPU, 8 GiB)
- c6i.2xlarge (8 vCPU, 16 GiB)
- c6i.4xlarge (16 vCPU, 32 GiB)
- c6i.8xlarge (32 vCPU, 64 GiB)
- c6i.12xlarge (48 vCPU, 96 GiB)
- c6i.16xlarge (64 vCPU, 128 GiB)
- c6i.24xlarge (96 vCPU, 192 GiB)
- c6i.32xlarge (128 vCPU, 256 GiB)

Example 3.5. Storage-optimized compute types

- i3.xlarge (4 vCPU, 30.5 GiB)
- i3.2xlarge (8 vCPU, 61 GiB)
- i3.4xlarge (16 vCPU, 122 GiB)
- i3.8xlarge (32 vCPU, 244 GiB)

- i3.16xlarge (64 vCPU, 488 GiB)
- i3en.xlarge (4 vCPU, 32 GiB)
- i3en.2xlarge (8 vCPU, 64 GiB)
- i3en.3xlarge (12 vCPU, 96 GiB)
- i3en.6xlarge (24 vCPU, 192 GiB)
- i3en.12xlarge (48 vCPU, 384 GiB)
- i3en.24xlarge (96 vCPU, 768 GiB)

3.3.1.5. Regions and availability zones

The following AWS regions are supported by Red Hat OpenShift 4 and are supported for Red Hat OpenShift Service on AWS. Note: China and GovCloud (US) regions are not supported, regardless of their support on OpenShift 4.

- af-south-1 (Cape Town, AWS opt-in required)
- ap-east-1 (Hong Kong, AWS opt-in required)
- ap-northeast-1 (Tokyo)
- ap-northeast-2 (Seoul)
- ap-northeast-3 (Osaka)
- ap-south-1 (Mumbai)
- ap-southeast-1 (Singapore)
- ap-southeast-2 (Sydney)
- ca-central-1 (Central Canada)
- eu-central-1 (Frankfurt)
- eu-north-1 (Stockholm)
- eu-south-1 (Milan, AWS opt-in required)
- eu-west-1 (Ireland)
- eu-west-2 (London)
- eu-west-3 (Paris)
- me-south-1 (Bahrain, AWS opt-in required)
- sa-east-1 (São Paulo)
- us-east-1 (N. Virginia)

- us-east-2 (Ohio)
- us-west-1 (N. California)
- us-west-2 (Oregon)

Multiple availability zone clusters can only be deployed in regions with at least 3 availability zones. For more information, see the Regions and Availability Zones section in the AWS documentation.

Each new Red Hat OpenShift Service on AWS cluster is installed within an installer-created or preexisting Virtual Private Cloud (VPC) in a single region, with the option to deploy into a single availability zone (Single-AZ) or across multiple availability zones (Multi-AZ). This provides cluster-level network and resource isolation, and enables cloud-provider VPC settings, such as VPN connections and VPC Peering. Persistent volumes (PVs) are backed by AWS Elastic Block Storage (EBS), and are specific to the availability zone in which they are provisioned. Persistent volume claims (PVCs) do not bind to a volume until the associated pod resource is assigned into a specific availability zone to prevent unschedulable pods. Availability zone-specific resources are only usable by resources in the same availability zone.

WARNING

The region and the choice of single or multiple availability zone cannot be changed after a cluster has been deployed.

3.3.1.6. Service Level Agreement (SLA)

Any SLAs for the service itself are defined in Appendix 4 of the Red Hat Enterprise Agreement Appendix 4 (Online Subscription Services).

3.3.1.7. Limited support status

When a cluster transitions to a *Limited Support* status, Red Hat no longer proactively monitors the cluster, the SLA is no longer applicable, and credits requested against the SLA are denied. It does not mean that you no longer have product support. In some cases, the cluster can return to a fully-supported status if you remediate the violating factors. However, in other cases, you might have to delete and recreate the cluster.

A cluster might move to a Limited Support status for many reasons, including the following scenarios:

If you do not upgrade a cluster to a supported version before the end-of-life date

Red Hat does not make any runtime or SLA guarantees for versions after their end-of-life date. To receive continued support, upgrade the cluster to a supported version prior to the end-of-life date. If you do not upgrade the cluster prior to the end-of-life date, the cluster transitions to a Limited Support status until it is upgraded to a supported version.

Red Hat provides commercially reasonable support to upgrade from an unsupported version to a supported version. However, if a supported upgrade path is no longer available, you might have to create a new cluster and migrate your workloads.

If you remove or replace any native Red Hat OpenShift Service on AWS components or any other component that is installed and managed by Red Hat

If cluster administrator permissions were used, Red Hat is not responsible for any of your or your authorized users' actions, including those that affect infrastructure services, service availability, or data loss. If Red Hat detects any such actions, the cluster might transition to a Limited Support status. Red Hat notifies you of the status change and you should either revert the action or create a support case to explore remediation steps that might require you to delete and recreate the cluster.

If you have questions about a specific action that might cause a cluster to move to a Limited Support status or need further assistance, open a support ticket.

3.3.1.8. Support

Red Hat OpenShift Service on AWS includes Red Hat Premium Support, which can be accessed by using the Red Hat Customer Portal .

See Red Hat OpenShift Service on AWS SLAs for support response times.

AWS support is subject to a customer's existing support contract with AWS.

3.3.2. Logging

Red Hat OpenShift Service on AWS provides optional integrated log forwarding to Amazon (AWS) CloudWatch.

3.3.2.1. Cluster audit logging

Cluster audit logs are available through AWS CloudWatch, if the integration is enabled. If the integration is not enabled, you can request the audit logs by opening a support case.

3.3.2.2. Application logging

Application logs sent to **STDOUT** are collected by Fluentd and forwarded to AWS CloudWatch through the cluster logging stack, if it is installed.

3.3.3. Monitoring

This section provides information about the service definition for Red Hat OpenShift Service on AWS monitoring.

3.3.3.1. Cluster metrics

Red Hat OpenShift Service on AWS clusters come with an integrated Prometheus stack for cluster monitoring including CPU, memory, and network-based metrics. This is accessible through the web console. These metrics also allow for horizontal pod autoscaling based on CPU or memory metrics provided by an Red Hat OpenShift Service on AWS user.

3.3.3.2. Cluster status notification

Red Hat communicates the health and status of Red Hat OpenShift Service on AWS clusters through a combination of a cluster dashboard available in OpenShift Cluster Manager, and email notifications sent to the email address of the contact that originally deployed the cluster, and any additional contacts specified by the customer.

3.3.4. Networking

This section provides information about the service definition for Red Hat OpenShift Service on AWS networking.

3.3.4.1. Custom domains for applications

To use a custom hostname for a route, you must update your DNS provider by creating a canonical name (CNAME) record. Your CNAME record should map the OpenShift canonical router hostname to your custom domain. The OpenShift canonical router hostname is shown on the *Route Details* page after a route is created. Alternatively, a wildcard CNAME record can be created once to route all subdomains for a given hostname to the cluster's router.

3.3.4.2. Domain validated certificates

Red Hat OpenShift Service on AWS includes TLS security certificates needed for both internal and external services on the cluster. For external routes, there are two separate TLS wildcard certificates that are provided and installed on each cluster: one is for the web console and route default hostnames, and the other is for the API endpoint. Let's Encrypt is the certificate authority used for certificates. Routes within the cluster, such as the internal API endpoint, use TLS certificates signed by the cluster's built-in certificate authority and require the CA bundle available in every pod for trusting the TLS certificate.

3.3.4.3. Custom certificate authorities for builds

Red Hat OpenShift Service on AWS supports the use of custom certificate authorities to be trusted by builds when pulling images from an image registry.

3.3.4.4. Load Balancers

Red Hat OpenShift Service on AWS uses up to five different load balancers:

- An internal control plane load balancer that is internal to the cluster and used to balance traffic for internal cluster communications.
- An external control plane load balancer that is used for accessing the OpenShift and Kubernetes APIs. This load balancer can be disabled in OpenShift Cluster Manager. If this load balancer is disabled, Red Hat reconfigures the API DNS to point to the internal control plane load balancer.
- An external control plane load balancer for Red Hat that is reserved for cluster management by Red Hat. Access is strictly controlled, and communication is only possible from whitelisted bastion hosts.
- A default external router/ingress load balancer that is the default application load balancer, denoted by **apps** in the URL. The default load balancer can be configured in OpenShift Cluster Manager to be either publicly accessible over the Internet or only privately accessible over a pre-existing private connection. All application routes on the cluster are exposed on this default router load balancer, including cluster services such as the logging UI, metrics API, and registry.
- Optional: A secondary router/ingress load balancer that is a secondary application load balancer, denoted by **apps2** in the URL. The secondary load balancer can be configured in OpenShift Cluster Manager to be either publicly accessible over the Internet or only privately accessible over a pre-existing private connection. If a **Label match** is configured for this router load balancer, then only application routes matching this label are exposed on this router load balancer; otherwise, all application routes are also exposed on this router load balancer.
- Optional: Load balancers for services. Enable non-HTTP/SNI traffic and non-standard ports for

services. These load balancers can be mapped to a service running on Red Hat OpenShift Service on AWS to enable advanced ingress features, such as non-HTTP/SNI traffic or the use of non-standard ports. Each AWS account has a quota which limits the number of Classic Load Balancers that can be used within each cluster.

3.3.4.5. Cluster ingress

Project administrators can add route annotations for many different purposes, including ingress control through IP allow-listing.

Ingress policies can also be changed by using **NetworkPolicy** objects, which leverage the **ovs-networkpolicy** plug-in. This allows for full control over the ingress network policy down to the pod level, including between pods on the same cluster and even in the same namespace.

All cluster ingress traffic will go through the defined load balancers. Direct access to all nodes is blocked by cloud configuration.

3.3.4.6. Cluster egress

Pod egress traffic control through **EgressNetworkPolicy** objects can be used to prevent or limit outbound traffic in Red Hat OpenShift Service on AWS.

Public outbound traffic from the control plane and infrastructure nodes is required and necessary to maintain cluster image security and cluster monitoring. This requires that the **0.0.0.0/0** route belongs only to the Internet gateway; it is not possible to route this range over private connections.

OpenShift 4 clusters use NAT gateways to present a public, static IP for any public outbound traffic leaving the cluster. Each availability zone a cluster is deployed into receives a distinct NAT gateway, therefore up to 3 unique static IP addresses can exist for cluster egress traffic. Any traffic that remains inside the cluster, or that does not go out to the public Internet, will not pass through the NAT gateway and will have a source IP address belonging to the node that the traffic originated from. Node IP addresses are dynamic; therefore, a customer must not rely on whitelisting individual IP addresses when accessing private resources.

Customers can determine their public static IP addresses by running a pod on the cluster and then querying an external service. For example:

```
$ oc run ip-lookup --image=busybox -i -t --restart=Never --rm -- /bin/sh -c "/bin/nslookup -type=a
myip.opendns.com resolver1.opendns.com | grep -E 'Address: [0-9.]+'"
```

3.3.4.7. Cloud network configuration

Red Hat OpenShift Service on AWS allows for the configuration of a private network connection through AWS-managed technologies:

- VPN connections
- VPC peering
- Transit Gateway
- Direct Connect

IMPORTANT

Red Hat site reliability engineers (SREs) do not monitor private network connections. Monitoring these connections is the responsibility of the customer.

3.3.4.8. DNS forwarding

For Red Hat OpenShift Service on AWS clusters that have a private cloud network configuration, a customer can specify internal DNS servers available on that private connection, that should be queried for explicitly provided domains.

3.3.5. Storage

This section provides information about the service definition for Red Hat OpenShift Service on AWS storage.

3.3.5.1. Encrypted-at-rest OS and node storage

Control plane nodes use encrypted-at-rest AWS Elastic Block Store (EBS) storage.

3.3.5.2. Encrypted-at-rest PV

EBS volumes that are used for PVs are encrypted-at-rest by default.

3.3.5.3. Block storage (RWO)

Persistent volumes (PVs) are backed by AWS EBS, which is Read-Write-Once.

PVs can be attached only to a single node at a time and are specific to the availability zone in which they were provisioned. However, PVs can be attached to any node in the availability zone.

Each cloud provider has its own limits for how many PVs can be attached to a single node. See AWS instance type limits for details.

3.3.5.4. Shared Storage (RWX)

The AWS CSI Driver can be used to provide RWX support for Red Hat OpenShift Service on AWS. A community Operator is provided to simplify setup. See AWS EFS Setup for OpenShift Dedicated and Red Hat OpenShift Service on AWS for details.

3.3.6. Platform

This section provides information about the service definition for the Red Hat OpenShift Service on AWS (ROSA) platform.

3.3.6.1. Cluster backup policy

IMPORTANT

It is critical that customers have a backup plan for their applications and application data.

Application and application data backups are not a part of the Red Hat OpenShift Service on AWS service. The following table outlines the cluster backup policy.

Component	Snapshot frequency	Retention	Notes
Full object store backup, all cluster persistent volumes (PVs)	Daily	7 days	This is a full backup of all Kubernetes objects like etcd, as well as all PVs in the cluster.
	Weekly	30 days	
Full object store backup	Hourly	24 hour	This is a full backup of all Kubernetes objects like etcd. No PVs are backed up in this backup schedule.
Node root volume	Never	N/A	Nodes are considered to be short-term. Nothing critical should be stored on a node's root volume.

3.3.6.2. Autoscaling

Node autoscaling is available on Red Hat OpenShift Service on AWS. You can configure the autoscaler option to automatically scale the number of machines in a cluster.

Additional resources

- About autoscaling nodes on a cluster

3.3.6.3. Daemonsets

Customers can create and run daemonsets on Red Hat OpenShift Service on AWS. To restrict daemonsets to only running on worker nodes, use the following **nodeSelector**:

```
...
spec:
  nodeSelector:
    role: worker
...
```

3.3.6.4. Multiple availability zone

In a multiple availability zone cluster, control plane nodes are distributed across availability zones and at least one worker node is required in each availability zone.

3.3.6.5. Node labels

Custom node labels are created by Red Hat during node creation and cannot be changed on Red Hat OpenShift Service on AWS clusters at this time. However, custom labels are supported when creating new machine pools.

3.3.6.6. OpenShift version

Red Hat OpenShift Service on AWS is run as a service and is kept up to date with the latest OpenShift Container Platform version. Upgrade scheduling to the latest version is available.

3.3.6.7. Upgrades

Upgrades can be scheduled using the **rosa** CLI utility or through OpenShift Cluster Manager.

See the Red Hat OpenShift Service on AWS Life Cycle for more information on the upgrade policy and procedures.

3.3.6.8. Windows Containers

Red Hat OpenShift support for Windows Containers is not available on Red Hat OpenShift Service on AWS at this time.

3.3.6.9. Container engine

Red Hat OpenShift Service on AWS runs on OpenShift 4 and uses CRI-O as the only available container engine.

3.3.6.10. Operating system

Red Hat OpenShift Service on AWS runs on OpenShift 4 and uses Red Hat CoreOS as the operating system for all control plane and worker nodes.

3.3.6.11. Red Hat Operator support

Red Hat workloads typically refer to Red Hat-provided Operators made available through Operator Hub. Red Hat workloads are not managed by the Red Hat SRE team, and must be deployed on worker nodes. These Operators may require additional Red Hat subscriptions, and may incur additional cloud infrastructure costs. Examples of these Red Hat-provided Operators are:

- Red Hat Quay
- Red Hat Advanced Cluster Management
- Red Hat Advanced Cluster Security
- Red Hat OpenShift Service Mesh
- OpenShift Serverless
- Red Hat OpenShift Logging
- Red Hat OpenShift Pipelines

3.3.6.12. Kubernetes Operator support

All Operators listed in the Operator Hub marketplace should be available for installation. These operators are considered customer workloads, and are not monitored by Red Hat SRE.

3.3.7. Security

This section provides information about the service definition for Red Hat OpenShift Service on AWS security.

3.3.7.1. Authentication provider

Authentication for the cluster can be configured using either OpenShift Cluster Manager or cluster creation process or using the **rosa** CLI. Red Hat OpenShift Service on AWS is not an identity provider, and all access to the cluster must be managed by the customer as part of their integrated solution. The use of multiple identity providers provisioned at the same time is supported. The following identity providers are supported:

- GitHub or GitHub Enterprise
- GitLab
- Google
- LDAP
- OpenID Connect

3.3.7.2. Privileged containers

Privileged containers are available for users with the **cluster-admin** role. Usage of privileged containers as **cluster-admin** is subject to the responsibilities and exclusion notes in the Red Hat Enterprise Agreement Appendix 4 (Online Subscription Services).

3.3.7.3. Customer administrator user

In addition to normal users, Red Hat OpenShift Service on AWS provides access to an Red Hat OpenShift Service on AWS-specific group called **dedicated-admin**. Any users on the cluster that are members of the **dedicated-admin** group:

- Have administrator access to all customer-created projects on the cluster.
- Can manage resource quotas and limits on the cluster.
- Can add and manage **NetworkPolicy** objects.
- Are able to view information about specific nodes and PVs in the cluster, including scheduler information.
- Can access the reserved **dedicated-admin** project on the cluster, which allows for the creation of service accounts with elevated privileges and also gives the ability to update default limits and quotas for projects on the cluster.

3.3.7.4. Cluster administration role

The administrator of Red Hat OpenShift Service on AWS has default access to the **cluster-admin** role for your organization's cluster. While logged into an account with the **cluster-admin** role, users have increased permissions to run privileged security contexts.

3.3.7.5. Project self-service

By default, all users have the ability to create, update, and delete their projects. This can be restricted if a member of the **dedicated-admin** group removes the **self-provisioner** role from authenticated users:

```
$ oc adm policy remove-cluster-role-from-group self-provisioner system:authenticated:oauth
```

Restrictions can be reverted by applying:

```
$ oc adm policy add-cluster-role-to-group self-provisioner system:authenticated:oauth
```

3.3.7.6. Regulatory compliance

See Understanding process and security for ROSA for the latest compliance information.

3.3.7.7. Network security

With Red Hat OpenShift Service on AWS, AWS provides a standard DDoS protection on all load balancers, called AWS Shield. This provides 95% protection against most commonly used level 3 and 4 attacks on all the public facing load balancers used for Red Hat OpenShift Service on AWS. A 10-second timeout is added for HTTP requests coming to the **haproxy** router to receive a response or the connection is closed to provide additional protection.

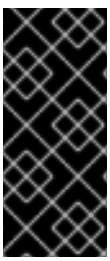
3.3.7.8. etcd encryption

In Red Hat OpenShift Service on AWS, the control plane storage is encrypted at rest by default and this includes encryption of the etcd volumes. This storage-level encryption is provided through the storage layer of the cloud provider.

You can also enable etcd encryption, which encrypts the key values in etcd, but not the keys. If you enable etcd encryption, the following Kubernetes API server and OpenShift API server resources are encrypted:

- Secrets
- Config maps
- Routes
- OAuth access tokens
- OAuth authorize tokens

The etcd encryption feature is not enabled by default and it can be enabled only at cluster installation time. Even with etcd encryption enabled, the etcd key values are accessible to anyone with access to the control plane nodes or **cluster-admin** privileges.



IMPORTANT

By enabling etcd encryption for the key values in etcd, you will incur a performance overhead of approximately 20%. The overhead is a result of introducing this second layer of encryption, in addition to the default control plane storage encryption that encrypts the etcd volumes. Red Hat recommends that you enable etcd encryption only if you specifically require it for your use case.

3.3.8. Additional resources

- See Understanding process and security for ROSA for the latest compliance information.
- See ROSA life cycle

3.4. RED HAT OPENSHIFT SERVICE ON AWS UPDATE LIFE CYCLE

3.4.1. Overview

Red Hat provides a published product life cycle for Red Hat OpenShift Service on AWS in order for customers and partners to effectively plan, deploy, and support their applications running on the platform. Red Hat publishes this life cycle in order to provide as much transparency as possible and might make exceptions from these policies as conflicts arise.

Red Hat OpenShift Service on AWS is a managed instance of Red Hat OpenShift and maintains an independent release schedule. More details about the managed offering can be found in the Red Hat OpenShift Service on AWS service definition. The availability of Security Advisories and Bug Fix Advisories for a specific version are dependent upon the Red Hat OpenShift Container Platform life cycle policy and subject to the Red Hat OpenShift Service on AWS maintenance schedule.

Additional resources

- Red Hat OpenShift Service on AWS service definition

3.4.2. Definitions

Table 3.1. Version reference

Version format	Major	Minor	Patch	Major.minor.patch
	x	y	z	x.y.z
Example	4	5	21	4.5.21

Major releases or X-releases

Referred to only as *major releases* or *X-releases* (X.y.z).

Examples

- "Major release 5" → 5.y.z
- "Major release 4" → 4.y.z
- "Major release 3" → 3.y.z

Minor releases or Y-releases

Referred to only as *minor releases* or *Y-releases* (x.Y.z).

Examples

- "Minor release 4" → 4.4.z
- "Minor release 5" → 4.5.z
- "Minor release 6" → 4.6.z

Patch releases or Z-releases

Referred to only as *patch releases* or *Z-releases* (x.y.Z).

Examples

- "Patch release 14 of minor release 5" → 4.5.14
- "Patch release 25 of minor release 5" → 4.5.25
- "Patch release 26 of minor release 6" → 4.6.26

3.4.3. Major versions (X.y.z)

Major versions of Red Hat OpenShift Service on AWS, for example version 4, are supported for one year following the release of a subsequent major version or the retirement of the product.

Example

- If version 5 were made available on Red Hat OpenShift Service on AWS on January 1, version 4 would be allowed to continue running on managed clusters for 12 months, until December 31. After this time, clusters would need to be upgraded or migrated to version 5.

3.4.4. Minor versions (x.Y.z)

Starting with the 4.8 OpenShift Container Platform minor version, Red Hat supports all minor versions within a 9 month period following general availability of the given minor version. Patch versions do not affect the 9 month supportability period.

Customers are notified 60, 30, and 15 days prior to the end of the 9 month period. Clusters must be upgraded to a supported minor version prior to the end of the 9 month period, or the cluster will enter a "Limited Support" status.

Example

1. A customer's cluster is currently running on 4.8.14. The 4.8 minor version became generally available on July 27, 2021.
2. On Feb 26, March 28, and April 12, 2022, the customer is notified that their cluster will enter "Limited Support" status on April 27, 2022 if the cluster has not already been upgraded to a supported minor version.
3. The cluster must be upgraded to 4.9 or later by April 27, 2022.
4. If the upgrade has not been performed, the cluster will be flagged as being in a "Limited Support" status.

Additional resources

- Red Hat OpenShift Service on AWS limited support status

3.4.5. Patch versions (x.y.Z)

During the period in which a minor version is supported, Red Hat supports all OpenShift Container Platform patch versions unless otherwise specified.

For reasons of platform security and stability, a patch release may be deprecated, which would prevent installations of that release and trigger mandatory upgrades off that release.

Example

1. 4.7.6 is found to contain a critical CVE.
2. Any releases impacted by the CVE will be removed from the supported patch release list. In addition, any clusters running 4.7.6 will be scheduled for automatic upgrades within 48 hours.

3.4.6. Limited support status

When a cluster transitions to a *Limited Support* status, Red Hat no longer proactively monitors the cluster, the SLA is no longer applicable, and credits requested against the SLA are denied. It does not mean that you no longer have product support. In some cases, the cluster can return to a fully-supported status if you remediate the violating factors. However, in other cases, you might have to delete and recreate the cluster.

A cluster might transition to a Limited Support status for many reasons, including the following scenarios:

If you do not upgrade a cluster to a supported version before the end-of-life date

Red Hat does not make any runtime or SLA guarantees for versions after their end-of-life date. To receive continued support, upgrade the cluster to a supported version prior to the end-of-life date. If you do not upgrade the cluster prior to the end-of-life date, the cluster transitions to a Limited Support status until it is upgraded to a supported version.

Red Hat provides commercially reasonable support to upgrade from an unsupported version to a supported version. However, if a supported upgrade path is no longer available, you might have to create a new cluster and migrate your workloads.

If you remove or replace any native Red Hat OpenShift Service on AWS components or any other component that is installed and managed by Red Hat

If cluster administrator permissions were used, Red Hat is not responsible for any of your or your authorized users' actions, including those that affect infrastructure services, service availability, or data loss. If Red Hat detects any such actions, the cluster might transition to a Limited Support status. Red Hat notifies you of the status change and you should either revert the action or create a support case to explore remediation steps that might require you to delete and recreate the cluster.

If you have questions about a specific action that might cause a cluster to transition to a Limited Support status or need further assistance, open a support ticket.

3.4.7. Supported versions exception policy

Red Hat reserves the right to add or remove new or existing versions, or delay upcoming minor release versions, that have been identified to have one or more critical production impacting bugs or security issues without advance notice.

3.4.8. Installation policy

While Red Hat recommends installation of the latest support release, Red Hat OpenShift Service on AWS supports installation of any supported release as covered by the preceding policy.

3.4.9. Mandatory upgrades

In the event that a Critical or Important CVE, or other bug identified by Red Hat, significantly impacts the security or stability of the cluster, the customer must upgrade to the next supported patch release within two business days.

In extreme circumstances and based on Red Hat's assessment of the CVE criticality to the environment, if the upgrade to the next supported patch release has not been performed within two business days of notification, the cluster will be automatically updated to the latest patch release to mitigate potential security breach or instability.

3.4.10. Life cycle dates

Version	General availability	End of life
4.10	Mar 10, 2022	Jan 10, 2023
4.9	Oct 18, 2021	Sep 28, 2022
4.8	Jul 27, 2021	Aug 31, 2022
4.7	Mar 24, 2021	Dec 17, 2021 [a]

[a] 4.7 minor version follows previous Y-1 life cycle

3.5. UNDERSTANDING PROCESS AND SECURITY FOR RED HAT OPENSHIFT SERVICE ON AWS

This document details the Red Hat responsibilities for the managed Red Hat OpenShift Service on AWS (ROSA).

Acronyms and terms

- **AWS** - Amazon Web Services
- **CEE** - Customer Experience and Engagement (Red Hat Support)
- **CI/CD** - Continuous Integration / Continuous Delivery
- **CVE** - Common Vulnerabilities and Exposures
- **PVs** - Persistent Volumes
- **ROSA** - Red Hat OpenShift Service on AWS
- **SRE** - Red Hat Site Reliability Engineering
- **VPC** - Virtual Private Cloud

3.5.1. Incident and operations management

This documentation details the Red Hat responsibilities for the Red Hat OpenShift Service on AWS (ROSA) managed service.

3.5.1.1. Platform monitoring

Red Hat site reliability engineers (SREs) maintain a centralized monitoring and alerting system for all ROSA cluster components, the SRE services, and underlying AWS accounts. Platform audit logs are securely forwarded to a centralized security information and event monitoring (SIEM) system, where they may trigger configured alerts to the SRE team and are also subject to manual review. Audit logs are retained in the SIEM system for one year. Audit logs for a given cluster are not deleted at the time the cluster is deleted.

3.5.1.2. Incident management

An incident is an event that results in a degradation or outage of one or more Red Hat services. An incident can be raised by a customer or a Customer Experience and Engagement (CEE) member through a support case, directly by the centralized monitoring and alerting system, or directly by a member of the SRE team.

Depending on the impact on the service and customer, the incident is categorized in terms of severity.

When managing a new incident, Red Hat uses the following general workflow:

1. An SRE first responder is alerted to a new incident and begins an initial investigation.
2. After the initial investigation, the incident is assigned an incident lead, who coordinates the recovery efforts.
3. An incident lead manages all communication and coordination around recovery, including any relevant notifications and support case updates.
4. The incident is recovered.
5. The incident is documented and a root cause analysis (RCA) is performed within 3 business days of the incident.
6. An RCA draft document will be shared with the customer within 7 business days of the incident.

3.5.1.3. Notifications

Platform notifications are configured using email. Some customer notifications are also sent to an account's corresponding Red Hat account team, including a Technical Account Manager, if applicable.

The following activities can trigger notifications:

- Platform incident
- Performance degradation
- Cluster capacity warnings
- Critical vulnerabilities and resolution
- Upgrade scheduling

3.5.1.4. Backup and recovery for ROSA clusters with STS

There is no backup method available for ROSA clusters with STS.

3.5.1.5. Backup and recovery

All Red Hat OpenShift Service on AWS cluster metadata from OpenShift Cluster Manager is securely backed up by Red Hat. The following table outlines backup and recovery strategies:

Component	Snapshot frequency	Retention	Notes
Full object store backup, all cluster persistent volumes (PVs)	Daily	7 days	This is a full backup of all Kubernetes objects like etcd, as well as all PVs in the cluster.
	Weekly	30 days	
Full object store backup	Hourly	24 hour	This is a full backup of all Kubernetes objects like etcd. No PVs are backed up in this backup schedule.
Node root volume	Never	N/A	Nodes are considered to be short-term. Nothing critical should be stored on a node's root volume.

- Red Hat does not commit to any Recovery Point Objective (RPO) or Recovery Time Objective (RTO).
- Customers are responsible for taking regular backups of their data
- Customers should deploy multi-AZ clusters with workloads that follow Kubernetes best practices to ensure high availability within a region.
- If an entire cloud region is unavailable, customers must install a new cluster in a different region and restore their apps using their backup data.

3.5.1.6. Cluster capacity

Evaluating and managing cluster capacity is a responsibility that is shared between Red Hat and the customer. Red Hat SRE is responsible for the capacity of all control plane and infrastructure nodes on the cluster.

Red Hat SRE also evaluates cluster capacity during upgrades and in response to cluster alerts. The impact of a cluster upgrade on capacity is evaluated as part of the upgrade testing process to ensure that capacity is not negatively impacted by new additions to the cluster. During a cluster upgrade, additional worker nodes are added to make sure that total cluster capacity is maintained during the upgrade process.

Capacity evaluations by the Red Hat SRE staff also happen in response to alerts from the cluster, after usage thresholds are exceeded for a certain period of time. Such alerts can also result in a notification to the customer.

3.5.2. Change management

This section describes the policies about how cluster and configuration changes, patches, and releases are managed.

3.5.2.1. Customer-initiated changes

You can initiate changes using self-service capabilities such as cluster deployment, worker node scaling, or cluster deletion.

Change history is captured in the **Cluster History** section in the OpenShift Cluster Manager **Overview tab**, and is available for you to view. The change history includes, but is not limited to, logs from the following changes:

- Adding or removing identity providers
- Adding or removing users to or from the **dedicated-admins** group
- Scaling the cluster compute nodes
- Scaling the cluster load balancer
- Scaling the cluster persistent storage
- Upgrading the cluster

3.5.2.2. Red Hat-initiated changes

Red Hat site reliability engineering (SRE) manages the infrastructure, code, and configuration of Red Hat OpenShift Service on AWS using a GitOps workflow and fully automated CI/CD pipelines. This process ensures that Red Hat can safely introduce service improvements on a continuous basis without negatively impacting customers.

Every proposed change undergoes a series of automated verifications immediately upon check-in. Changes are then deployed to a staging environment where they undergo automated integration testing. Finally, changes are deployed to the production environment. Each step is fully automated.

An authorized SRE reviewer must approve advancement to each step. The reviewer cannot be the same individual who proposed the change. All changes and approvals are fully auditable as part of the GitOps workflow.

Some changes are released to production incrementally, using feature flags to control availability of new features to specified clusters or customers.

3.5.2.3. Patch management

OpenShift Container Platform software and the underlying immutable Red Hat CoreOS (RHCOS) operating system image are patched for bugs and vulnerabilities in regular z-stream upgrades. Read more about RHCOS architecture in the OpenShift Container Platform documentation.

3.5.2.4. Release management

Red Hat does not automatically upgrade your clusters. You can schedule to upgrade the clusters at regular intervals (recurring upgrade) or just once (individual upgrade) using the OpenShift Cluster Manager web console. Red Hat might forcefully upgrade a cluster to a new z-stream version only if the cluster is affected by a critical impact CVE.

NOTE

Because the required permissions can change between y-stream releases, the policies might have to be updated before an upgrade can be performed. Therefore, you cannot schedule a recurring upgrade on ROSA clusters with STS.

You can review the history of all cluster upgrade events in the OpenShift Cluster Manager web console. For more information about releases, see the Life Cycle policy.

3.5.3. Identity and access management

Most access by Red Hat site reliability engineering (SRE) teams is done by using cluster Operators through automated configuration management.

3.5.3.1. Subprocessors

For a list of the available subprocessors, see the Red Hat Subprocessor List on the Red Hat Customer Portal.

3.5.3.2. SRE access to all Red Hat OpenShift Service on AWS clusters

SREs access Red Hat OpenShift Service on AWS clusters through the web console or command-line tools. Authentication requires multi-factor authentication (MFA) with industry-standard requirements for password complexity and account lockouts. SREs must authenticate as individuals to ensure auditability. All authentication attempts are logged to a Security Information and Event Management (SIEM) system.

SREs access private clusters using an encrypted HTTP connection. Connections are permitted only from a secured Red Hat network using either an IP allowlist or a private cloud provider link.

3.5.3.3. Privileged access controls in Red Hat OpenShift Service on AWS

SRE adheres to the principle of least privilege when accessing Red Hat OpenShift Service on AWS and AWS components. There are four basic categories of manual SRE access:

- SRE admin access through the Red Hat Portal with normal two-factor authentication and no privileged elevation.
- SRE admin access through the Red Hat corporate SSO with normal two-factor authentication and no privileged elevation.
- OpenShift elevation, which is a manual elevation using Red Hat SSO. Access is limited to 2 hours, is fully audited, and requires management approval.
- AWS access or elevation, which is a manual elevation for AWS console or CLI access. Access is limited to 60 minutes and is fully audited.

Each of these access types have different levels of access to components:

Component	Typical SRE admin access (Red Hat Portal)	Typical SRE admin access (Red Hat SSO)	OpenShift elevation	Cloud provider access or elevation
OpenShift Cluster Manager	R/W	No access	No access	No access
OpenShift console	No access	R/W	R/W	No access
Node operating system	No access	A specific list of elevated OS and network permissions.	A specific list of elevated OS and network permissions.	No access
AWS Console	No access	No access, but this is the account used to request cloud provider access.	No access	All cloud provider permissions using the SRE identity.

3.5.3.4. SRE access to AWS accounts

Red Hat personnel do not access AWS accounts in the course of routine Red Hat OpenShift Service on AWS operations. For emergency troubleshooting purposes, the SREs have well-defined and auditable procedures to access cloud infrastructure accounts.

SREs generate a short-lived AWS access token for a reserved role using the AWS Security Token Service (STS). Access to the STS token is audit-logged and traceable back to individual users. Both STS and non-STS clusters use the AWS STS service for SRE access. For non-STS clusters, the **BYOCAdminAccess** role has the **AdministratorAccess** IAM policy attached, and this role is used for administration. For STS clusters, the **ManagedOpenShift-Support-Role** has the **ManagedOpenShift-Support-Access** policy attached, and this role is used for administration.

3.5.3.5. Red Hat support access

Members of the Red Hat Customer Experience and Engagement (CEE) team typically have read-only access to parts of the cluster. Specifically, CEE has limited access to the core and product namespaces and does not have access to the customer namespaces.

Role	Core namespace	Layered product namespace	Customer namespace	AWS account*
OpenShift SRE	Read: All Write: Very limited [1]	Read: All Write: None	Read: None[2] Write: None	Read: All [3] Write: All [3]

Role	Core namespace	Layered product namespace	Customer namespace	AWS account*
CEE	Read: All Write: None	Read: All Write: None	Read: None[2] Write: None	Read: None Write: None
Customer administrator	Read: None Write: None	Read: None Write: None	Read: All Write: All	Read: All Write: All
Customer user	Read: None Write: None	Read: None Write: None	Read: Limited[4] Write: Limited[4]	Read: None Write: None
Everybody else	Read: None Write: None	Read: None Write: None	Read: None Write: None	Read: None Write: None

1. Limited to addressing common use cases such as failing deployments, upgrading a cluster, and replacing bad worker nodes.
2. Red Hat associates have no access to customer data by default.
3. SRE access to the AWS account is an emergency procedure for exceptional troubleshooting during a documented incident.
4. Limited to what is granted through RBAC by the Customer Administrator, as well as namespaces created by the user.

3.5.3.6. Customer access

Customer access is limited to namespaces created by the customer and permissions that are granted using RBAC by the Customer Administrator role. Access to the underlying infrastructure or product namespaces is generally not permitted without **cluster-admin** access. More information on customer access and authentication can be found in the "Understanding Authentication" section of the documentation.

3.5.3.7. Access approval and review

New SRE user access requires management approval. Separated or transferred SRE accounts are removed as authorized users through an automated process. Additionally, the SRE performs periodic access review, including management sign-off of authorized user lists.

3.5.4. Security and regulation compliance

Security and regulation compliance includes tasks such as the implementation of security controls and compliance certification.

3.5.4.1. Data classification

Red Hat defines and follows a data classification standard to determine the sensitivity of data and highlight inherent risk to the confidentiality and integrity of that data while it is collected, used, transmitted, stored, and processed. Customer-owned data is classified at the highest level of sensitivity and handling requirements.

3.5.4.2. Data management

Red Hat OpenShift Service on AWS (ROSA) uses AWS Key Management Service (KMS) to help securely manage keys for encrypted data. These keys are used for control plane data volumes that are encrypted by default.

When a customer deletes their ROSA cluster, all cluster data is permanently deleted, including control plane data volumes and customer application data volumes, such as persistent volumes (PV).

3.5.4.3. Vulnerability management

Red Hat performs periodic vulnerability scanning of ROSA using industry standard tools. Identified vulnerabilities are tracked to their remediation according to timelines based on severity. Vulnerability scanning and remediation activities are documented for verification by third-party assessors in the course of compliance certification audits.

3.5.4.4. Network security

3.5.4.4.1. Firewall and DDoS protection

Each ROSA cluster is protected by a secure network configuration using firewall rules for AWS Security Groups. ROSA customers are also protected against DDoS attacks with AWS Shield Standard.

3.5.4.4.2. Private clusters and network connectivity

Customers can optionally configure their ROSA cluster endpoints, such as web console, API, and application router, to be made private so that the cluster control plane and applications are not accessible from the Internet. Red Hat SRE still requires Internet-accessible endpoints that are protected with IP allow-lists.

AWS customers can configure a private network connection to their ROSA cluster through technologies such as AWS VPC peering, AWS VPN, or AWS Direct Connect.

3.5.4.4.3. Cluster network access controls

Fine-grained network access control rules can be configured by customers, on a per-project basis, using **NetworkPolicy** objects and the OpenShift SDN.

3.5.4.5. Penetration testing

Red Hat performs periodic penetration tests against ROSA. Tests are performed by an independent internal team by using industry standard tools and best practices.

Any issues that may be discovered are prioritized based on severity. Any issues found belonging to open source projects are shared with the community for resolution.

3.5.4.6. Compliance

Red Hat OpenShift Service on AWS follows common industry best practices for security and controls. The certifications are outlined in the following table.

Table 3.2. Security and control certifications for Red Hat OpenShift Service on AWS

Certification	Red Hat OpenShift Service on AWS
ISO 27001	Yes
PCI DSS	Yes
SOC 2 Type 2	Yes

Additional resources

- See Red Hat Subprocessor List for information on SRE residency.

3.5.5. Disaster recovery

Red Hat OpenShift Service on AWS (ROSA) provides disaster recovery for failures that occur at the pod, worker node, infrastructure node, control plane node, and availability zone levels.

All disaster recovery requires that the customer use best practices for deploying highly available applications, storage, and cluster architecture, such as single-zone deployment or multi-zone deployment, to account for the level of desired availability.

One single-zone cluster will not provide disaster avoidance or recovery in the event of an availability zone or region outage. Multiple single-zone clusters with customer-maintained failover can account for outages at the zone or at the regional level.

One multi-zone cluster will not provide disaster avoidance or recovery in the event of a full region outage. Multiple multi-zone clusters with customer-maintained failover can account for outages at the regional level.

3.5.6. Additional resources

- For more information about customer or shared responsibilities, see the ROSA Responsibilities document.
- For more information about ROSA and its components, see the ROSA Service Definition.

CHAPTER 4. ABOUT IAM RESOURCES FOR ROSA CLUSTERS THAT USE STS

To deploy a Red Hat OpenShift Service on AWS (ROSA) cluster that uses the AWS Security Token Service (STS), you must create the following AWS Identity Access Management (IAM) resources:

- Specific account-wide IAM roles and policies that provide the STS permissions required for ROSA support, installation, control plane, and compute functionality. This includes account-wide Operator policies.
- Cluster-specific Operator IAM roles that permit the ROSA cluster Operators to carry out core OpenShift functionality.
- An OpenID Connect (OIDC) provider that the cluster Operators use to authenticate.
- If you deploy ROSA by using OpenShift Cluster Manager, you must create the additional resources:
  - An OpenShift Cluster Manager IAM role to complete the installation on your cluster.
  - A user role without any permissions to verify your AWS account identity.

This document provides reference information about the IAM resources that you must deploy when you create a ROSA cluster that uses STS. It also includes the **aws** CLI commands that are generated when you use **manual** mode with the **rosa create** command.

Additional resources

- For steps to quickly create a ROSA cluster with STS, including the AWS IAM resources, see Creating a ROSA cluster with STS using the default options .
- For steps to create a ROSA cluster with STS using customizations, including the AWS IAM resources, see Creating a ROSA cluster with STS using customizations .

4.1. OPENSHIFT CLUSTER MANAGER ROLES AND PERMISSIONS

If you create ROSA clusters by using OpenShift Cluster Manager, you must have the following AWS IAM roles linked to your AWS account to create and manage the clusters. For more information about linking your IAM roles to your AWS account, see Associating your AWS account.

TIP

If you only use the **rosa** CLI tool, then you do not need to create these IAM roles.

These AWS IAM roles are as follows:

- The ROSA user role is an AWS role used by Red Hat to verify the customer's AWS identity. This role has no additional permissions, and the role has a trust relationship with the Red Hat installer account.
- An **ocm-role** resource grants the required permissions for installation of ROSA clusters in OpenShift Cluster Manager. You can apply basic or administrative permissions to the **ocm-role** resource. If you create an administrative **ocm-role** resource, OpenShift Cluster Manager can

create the needed AWS Operator roles and OpenID Connect (OIDC) provider. This IAM role also creates a trust relationship with the Red Hat installer account as well.

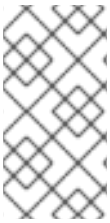
NOTE

The **ocm-role** IAM resource refers to the combination of the IAM role and the necessary policies created with it.

You must create this user role as well as an administrative **ocm-role** resource, if you want to use the auto mode in OpenShift Cluster Manager to create your Operator role policies and OIDC provider.

4.1.1. Understanding the OpenShift Cluster Manager role

Creating ROSA clusters in OpenShift Cluster Manager require an **ocm-role** IAM role. The basic **ocm-role** IAM role permissions let you to perform cluster maintenance within OpenShift Cluster Manager. To automatically create the operator roles and OpenID Connect (OIDC) provider, you must add the **--admin** option to the **rosa create** command. This command creates an **ocm-role** resource with additional permissions needed for administrative tasks.



NOTE

This elevated IAM role allows OpenShift Cluster Manager to automatically create the cluster-specific Operator roles and OIDC provider during cluster creation. For more information about this automatic role and policy creation, see the "Methods of account-wide role creation" link in Additional resources.

4.1.1.1. Understanding the user role

In addition to an **ocm-role** IAM role, you must create a user role so that Red Hat OpenShift Service on AWS can verify your AWS identity. This role has no permissions, and it is only used to create a trust relationship between the installer account and your **ocm-role** resources.

The following tables show the associated basic and administrative permissions for the **ocm-role** resource.

Table 4.1. Associated permissions for the basic ocm-role resource

Resource	Description
iam:GetOpenIDConnectProvider	This permission allows the basic role to retrieve information about the specified OpenID Connect (OIDC) provider.
iam:GetRole	This permission allows the basic role to retrieve any information for a specified role. Some of the data returned include the role's path, GUID, ARN, and the role's trust policy that grants permission to assume the role.
iam:ListRoles	This permission allows the basic role to list the roles within a path prefix.
iam:ListRoleTags	This permission allows the basic role to list the tags on a specified role.

Resource	Description
ec2:DescribeRegions	This permission allows the basic role to return information about all of the enabled regions on your account.
ec2:DescribeRouteTables	This permission allows the basic role to return information about all of your route tables.
ec2:DescribeSubnets	This permission allows the basic role to return information about all of your subnets.
ec2:DescribeVpcs	This permission allows the basic role to return information about all of your virtual private clouds (VPCs).
sts:AssumeRole	This permission allows the basic role to retrieve temporary security credentials to access AWS resources that are beyond its normal permissions.
sts:AssumeRoleWithWebIdentity	This permission allows the basic role to retrieve temporary security credentials for users authenticated their account with a web identity provider.

Table 4.2. Additional permissions for the admin**ocm-role** resource

Resource	Description
iam:AttachRolePolicy	This permission allows the admin role to attach a specified policy to the desired IAM role.
iam:CreateOpenIDConnectProvider	This permission creates a resource that describes an identity provider, which supports OpenID Connect (OIDC). When you create an OIDC provider with this permission, this provider establishes a trust relationship between the provider and AWS.
iam:CreateRole	This permission allows the admin role to create a role for your AWS account.
iam:ListPolicies	This permission allows the admin role to list any policies associated with your AWS account.
iam:ListPolicyTags	This permission allows the admin role to list any tags on a designated policy.
iam:PutRolePermissionsBoundary	This permission allows the admin role to change the permissions boundary for a user based on a specified policy.
iam:TagRole	This permission allows the admin role to add tags to an IAM role.

Additional resources

- Methods of account-wide role creation

Creating an OpenShift Cluster Manager IAM role

You create your OpenShift Cluster Manager IAM roles by using the command-line interface (CLI).

Prerequisites

- You have an AWS account.
- You have Red Hat Organization Administrator privileges in the OpenShift Cluster Manager organization.
- You have the permissions required to install AWS account-wide roles.
- You have installed and configured the latest AWS (**aws**) and ROSA (**rosa**) CLIs on your installation host.

Procedure

- To create an ocm-role IAM role with basic privileges, run the following command:

```
$ rosa create ocm-role
```

- To create an ocm-role IAM role with admin privileges, run the following command:

```
$ rosa create ocm-role --admin
```

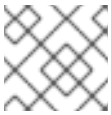
  This command allows you create the role by specifying specific attributes. The following example output shows the "auto mode" selected, which lets the **rosa** CLI to create your Operator roles and policies. See "Methods of account-wide role creation" in the Additional resources for more information.

Example output

```
I: Creating ocm role
? Role prefix: ManagedOpenShift 1
? Enable admin capabilities for the OCM role (optional): No 2
? Permissions boundary ARN (optional):  3
? Role creation mode: auto 4
I: Creating role using 'arn:aws:iam::<ARN>:user/<UserName>'
? Create the 'ManagedOpenShift-OCM-Role-182' role? Yes 5
I: Created role 'ManagedOpenShift-OCM-Role-182' with ARN  'arn:aws:iam::
<ARN>:role/ManagedOpenShift-OCM-Role-182'
I: Linking OCM role
? OCM Role ARN: arn:aws:iam::<ARN>:role/ManagedOpenShift-OCM-Role-182 6
? Link the 'arn:aws:iam::<ARN>:role/ManagedOpenShift-OCM-Role-182' role with organization
'<AWS ARN'? Yes 7
I: Successfully linked role-arn 'arn:aws:iam::<ARN>:role/ManagedOpenShift-OCM-Role-182' with
organization account '<AWS ARN>'
```

1. A prefix value for all of the created AWS resources. In this example, **ManagedOpenShift** prepends all of the AWS resources.

2 Choose if you want this role to have the additional admin permissions.



NOTE

You do not see this prompt if you used the **--admin** option.

3 The Amazon Resource Name (ARN) of the policy to set permission boundaries.

4 Choose the method of how to create your AWS roles. Using **auto**, the **rosa** CLI tool generates and links the roles and policies. In the **auto** mode, you receive some different prompts to create the AWS roles.

5 The auto method asks if you want to create a specific **ocm-role** using your prefix.

6 Confirm that you want to associate your IAM role with your OpenShift Cluster Manager.

7 Links the created role with your AWS organization.

AWS IAM roles link to your AWS account to create and manage the clusters. For more information about linking your IAM roles to your AWS account, see Associating your AWS account.

Additional resources

- AWS Identity and Access Management Data Types
- Amazon Elastic Computer Cloud Data Types
- AWS Token Security Service Data Types
- Methods of account-wide role creation

4.2. ACCOUNT-WIDE IAM ROLE AND POLICY REFERENCE

This section provides details about the account-wide IAM roles and policies that are required for ROSA deployments that use STS, including the Operator policies. It also includes the JSON files that define the policies.

The account-wide roles and policies are specific to an OpenShift minor release version, for example OpenShift 4.8, and are backward compatible. You can minimize the required STS resources by reusing the account-wide roles and policies for multiple clusters of the same minor version, regardless of their patch version.

4.2.1. Methods of account-wide role creation

You can create account-wide roles by using the **rosa** CLI tool or the OpenShift Cluster Manager guided installation. You can create the roles manually or by using an automatic process that uses pre-defined names for these roles and polices.

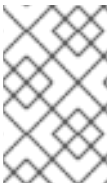
You can create account-wide roles by using the **rosa** CLI tool. You can create the roles manually or by using an automatic process that uses pre-defined names for these roles and polices.

Manual ocm-role resource creation
You can use the manual creation method if you have the necessary CLI access to create these roles on your system. You can run this option in your desired CLI tool or from OpenShift Cluster Manager. After

you start the manual creation process, the CLI presents a series of commands for you to run that create the roles and link them to the needed policies.

Automatic ocm-role resource creation
If you created an **ocm-role** resource with administrative permissions, you can use the automatic creation method from OpenShift Cluster Manager. The **rosa** CLI does not require that you have this admin **ocm-role** IAM resource to automatically create these roles and polices. Selecting this method creates the roles and policies that uses the default names.

If you use the ROSA guided installation on OpenShift Cluster Manager, you must have created an **ocm-role** resource with administrative permissions in the first step of the guided cluster installation. Without this role, you cannot use the automatic Operator role and policy creation option, but you can still create the cluster and its roles and policies with the manual process.



NOTE

The account number present in the **sts_installer_trust_policy.json** and **sts_support_trust_policy.json** samples represents the Red Hat account that is allowed to assume the required roles.

Table 4.3. ROSA installer role, policy, and policy files

Resource	Description
ManagedOpenShift-Installer-Role	An IAM role used by the ROSA installer.
ManagedOpenShift-Installer-Role-Policy	An IAM policy that provides the ROSA installer with the permissions required to complete cluster installation tasks.

Example 4.1. sts_installer_trust_policy.json

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::710019948333:role/RH-Managed-OpenShift-Installer"
                ]
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
    ]
}
```

Example 4.2. sts_installer_permission_policy.json

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:DescribeAutoScalingGroups",
                "ec2:AllocateAddress",
                "ec2:AssociateAddress",
                "ec2:AssociateDhcpOptions",
                "ec2:AssociateRouteTable",
                "ec2:AttachInternetGateway",
                "ec2:AttachNetworkInterface",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CopyImage",
                "ec2:CreateDhcpOptions",
                "ec2:CreateInternetGateway",
                "ec2:CreateNatGateway",
                "ec2:CreateNetworkInterface",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSubnet",
                "ec2:CreateTags",
                "ec2:CreateVolume",
                "ec2:CreateVpc",
                "ec2:CreateVpcEndpoint",
                "ec2:DeleteDhcpOptions",
                "ec2:DeleteInternetGateway",
                "ec2:DeleteNatGateway",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteRoute",
                "ec2:DeleteRouteTable",
                "ec2:DeleteSecurityGroup",
                "ec2:DeleteSnapshot",
                "ec2:DeleteSubnet",
                "ec2:DeleteTags",
                "ec2:DeleteVolume",
                "ec2:DeleteVpc",
                "ec2:DeleteVpcEndpoints",
                "ec2:DeregisterImage",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAddresses",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeImages",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeInstanceCreditSpecifications",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeNatGateways",
                "ec2:DescribeNetworkAcls",
```

```
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribePrefixLists",
                "ec2:DescribeRegions",
                "ec2:DescribeReservedInstancesOfferings",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeTags",
                "ec2:DescribeVolumes",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcClassicLink",
                "ec2:DescribeVpcClassicLinkDnsSupport",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcs",
                "ec2:DetachInternetGateway",
                "ec2:DisassociateRouteTable",
                "ec2:GetConsoleOutput",
                "ec2:GetEbsDefaultKmsKeyId",
                "ec2:ModifyInstanceAttribute",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:ReleaseAddress",
                "ec2:ReplaceRouteTableAssociation",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:RunInstances",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:TerminateInstances",
                "elasticloadbalancing:AddTags",
                "elasticloadbalancing:ApplySecurityGroupsToLoadBalancer",
                "elasticloadbalancing:AttachLoadBalancerToSubnets",
                "elasticloadbalancing:ConfigureHealthCheck",
                "elasticloadbalancing:CreateListener",
                "elasticloadbalancing:CreateLoadBalancer",
                "elasticloadbalancing:CreateLoadBalancerListeners",
                "elasticloadbalancing:CreateTargetGroup",
                "elasticloadbalancing:DeleteLoadBalancer",
                "elasticloadbalancing:DeleteTargetGroup",
                "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
                "elasticloadbalancing:DeregisterTargets",
                "elasticloadbalancing:DescribeInstanceHealth",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeLoadBalancerAttributes",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeTags",
                "elasticloadbalancing:DescribeTargetGroupAttributes",
                "elasticloadbalancing:DescribeTargetGroups",
                "elasticloadbalancing:DescribeTargetHealth",
                "elasticloadbalancing:ModifyLoadBalancerAttributes",
                "elasticloadbalancing:ModifyTargetGroup",
                "elasticloadbalancing:ModifyTargetGroupAttributes",
                "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
                "elasticloadbalancing:RegisterTargets",
                "elasticloadbalancing:SetLoadBalancerPoliciesOfListener",
```

```
                "iam:AddRoleToInstanceProfile",
                "iam:CreateInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:GetInstanceProfile",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:GetUser",
                "iam:ListAttachedRolePolicies",
                "iam:ListInstanceProfiles",
                "iam:ListInstanceProfilesForRole",
                "iam:ListRolePolicies",
                "iam:ListRoles",
                "iam:ListUserPolicies",
                "iam:ListUsers",
                "iam:PassRole",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:SimulatePrincipalPolicy",
                "iam:TagRole",
                "iam:UntagRole",
                "route53:ChangeResourceRecordSets",
                "route53:ChangeTagsForResource",
                "route53:CreateHostedZone",
                "route53:DeleteHostedZone",
                "route53:GetChange",
                "route53:GetHostedZone",
                "route53:ListHostedZones",
                "route53:ListHostedZonesByName",
                "route53:ListResourceRecordSets",
                "route53:ListTagsForResource",
                "route53:UpdateHostedZoneComment",
                "s3:CreateBucket",
                "s3:DeleteBucket",
                "s3:DeleteObject",
                "s3:GetAccelerateConfiguration",
                "s3:GetBucketAcl",
                "s3:GetBucketCORS",
                "s3:GetBucketLocation",
                "s3:GetBucketLogging",
                "s3:GetBucketObjectLockConfiguration",
                "s3:GetBucketReplication",
                "s3:GetBucketRequestPayment",
                "s3:GetBucketTagging",
                "s3:GetBucketVersioning",
                "s3:GetBucketWebsite",
                "s3:GetEncryptionConfiguration",
                "s3:GetLifecycleConfiguration",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectTagging",
                "s3:GetObjectVersion",
                "s3:GetReplicationConfiguration",
                "s3:ListBucket",
                "s3:ListBucketVersions",
                "s3:PutBucketAcl",
                "s3:PutBucketTagging",
                "s3:PutEncryptionConfiguration",
```

```
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutObjectTagging",
                "servicequotas:GetServiceQuota",
                "servicequotas:ListAWSDefaultServiceQuotas",
                "sts:AssumeRole",
                "sts:AssumeRoleWithWebIdentity",
                "sts:GetCallerIdentity",
                "tag:GetResources",
                "tag:UntagResources",
                "ec2:CreateVpcEndpointServiceConfiguration",
                "ec2:DeleteVpcEndpointServiceConfigurations",
                "ec2:DescribeVpcEndpointServiceConfigurations",
                "ec2:DescribeVpcEndpointServicePermissions",
                "ec2:DescribeVpcEndpointServices",
                "ec2:ModifyVpcEndpointServicePermissions"
                "kms:DescribeKey",
                "cloudwatch:GetMetricData"
            ],
            "Resource": "*"
        }
    ]
}
```

Table 4.4. ROSA control plane role, policy, and policy files

Resource	Description
ManagedOpenShift-ControlPlane-Role	An IAM role used by the ROSA control plane.
ManagedOpenShift-ControlPlane-Role-Policy	An IAM policy that provides the ROSA control plane with the permissions required to manage its components.

Example 4.3. **sts_instance_controlplane_trust_policy.json**

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
    ]
}
```

Example 4.4. **sts_instance_controlplane_permission_policy.json**

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateSecurityGroup",
                "ec2:CreateTags",
                "ec2:CreateVolume",
                "ec2:DeleteSecurityGroup",
                "ec2:DeleteVolume",
                "ec2:Describe*",
                "ec2:DetachVolume",
                "ec2:ModifyInstanceAttribute",
                "ec2:ModifyVolume",
                "ec2:RevokeSecurityGroupIngress",
                "elasticloadbalancing:AddTags",
                "elasticloadbalancing:AttachLoadBalancerToSubnets",
                "elasticloadbalancing:ApplySecurityGroupsToLoadBalancer",
                "elasticloadbalancing:CreateListener",
                "elasticloadbalancing:CreateLoadBalancer",
                "elasticloadbalancing:CreateLoadBalancerPolicy",
                "elasticloadbalancing:CreateLoadBalancerListeners",
                "elasticloadbalancing:CreateTargetGroup",
                "elasticloadbalancing:ConfigureHealthCheck",
                "elasticloadbalancing:DeleteListener",
                "elasticloadbalancing:DeleteLoadBalancer",
                "elasticloadbalancing:DeleteLoadBalancerListeners",
                "elasticloadbalancing:DeleteTargetGroup",
                "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
                "elasticloadbalancing:DeregisterTargets",
                "elasticloadbalancing:Describe*",
                "elasticloadbalancing:DetachLoadBalancerFromSubnets",
                "elasticloadbalancing:ModifyListener",
                "elasticloadbalancing:ModifyLoadBalancerAttributes",
                "elasticloadbalancing:ModifyTargetGroup",
                "elasticloadbalancing:ModifyTargetGroupAttributes",
                "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
                "elasticloadbalancing:RegisterTargets",
                "elasticloadbalancing:SetLoadBalancerPoliciesForBackendServer",
                "elasticloadbalancing:SetLoadBalancerPoliciesOfListener",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

Table 4.5. ROSA compute node role, policy, and policy files

Resource	Description
ManagedOpenShift-Worker-Role	An IAM role used by the ROSA compute instances.
ManagedOpenShift-Worker-Role-Policy	An IAM policy that provides the ROSA compute instances with the permissions required to manage their components.

Example 4.5. **sts_instance_worker_trust_policy.json**

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
    ]
}
```

Example 4.6. **sts_instance_worker_permission_policy.json**

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

Table 4.6. ROSA support role, policy, and policy files

Resource	Description
ManagedOpenShift-Support-Role	An IAM role used by the Red Hat Site Reliability Engineering (SRE) support team.
ManagedOpenShift-Support-Role-Policy	An IAM policy that provides the Red Hat SRE support team with the permissions required to support ROSA clusters.

Example 4.7. **sts_support_trust_policy.json**

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::710019948333:role/RH-Technical-Support-Access"
                ]
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
    ]
}
```

Example 4.8. **sts_support_permission_policy.json**

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudtrail:DescribeTrails",
                "cloudtrail:LookupEvents",
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey",
                "ec2:CopySnapshot",
                "ec2:CreateNetworkInsightsPath",
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots",
                "ec2:CreateTags",
                "ec2:DeleteNetworkInsightsAnalysis",
                "ec2:DeleteNetworkInsightsPath",
                "ec2:DeleteTags",
                "ec2:DescribeAccountAttributes",
```

```
                "ec2:DescribeAddresses",
                "ec2:DescribeAddressesAttribute",
                "ec2:DescribeAggregateIdFormat",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeByoipCidrs",
                "ec2:DescribeCapacityReservations",
                "ec2:DescribeCarrierGateways",
                "ec2:DescribeClassicLinkInstances",
                "ec2:DescribeClientVpnAuthorizationRules",
                "ec2:DescribeClientVpnConnections",
                "ec2:DescribeClientVpnEndpoints",
                "ec2:DescribeClientVpnRoutes",
                "ec2:DescribeClientVpnTargetNetworks",
                "ec2:DescribeCoipPools",
                "ec2:DescribeCustomerGateways",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeEgressOnlyInternetGateways",
                "ec2:DescribeIamInstanceProfileAssociations",
                "ec2:DescribeIdentityIdFormat",
                "ec2:DescribeIdFormat",
                "ec2:DescribeImageAttribute",
                "ec2:DescribeImages",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceTypeOfferings",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeIpv6Pools",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLocalGatewayRouteTables",
                "ec2:DescribeLocalGatewayRouteTableVirtualInterfaceGroupAssociations",
                "ec2:DescribeLocalGatewayRouteTableVpcAssociations",
                "ec2:DescribeLocalGateways",
                "ec2:DescribeLocalGatewayVirtualInterfaceGroups",
                "ec2:DescribeLocalGatewayVirtualInterfaces",
                "ec2:DescribeManagedPrefixLists",
                "ec2:DescribeNatGateways",
                "ec2:DescribeNetworkAcls",
                "ec2:DescribeNetworkInsightsAnalyses",
                "ec2:DescribeNetworkInsightsPaths",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribePlacementGroups",
                "ec2:DescribePrefixLists",
                "ec2:DescribePrincipalIdFormat",
                "ec2:DescribePublicIpv4Pools",
                "ec2:DescribeRegions",
                "ec2:DescribeReservedInstances",
                "ec2:DescribeRouteTables",
                "ec2:DescribeScheduledInstances",
                "ec2:DescribeSecurityGroupReferences",
                "ec2:DescribeSecurityGroupRules",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSnapshotAttribute",
                "ec2:DescribeSnapshots",
```

```
                "ec2:DescribeSpotFleetInstances",
                "ec2:DescribeStaleSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeTags",
                "ec2:DescribeTransitGatewayAttachments",
                "ec2:DescribeTransitGatewayConnectPeers",
                "ec2:DescribeTransitGatewayConnects",
                "ec2:DescribeTransitGatewayMulticastDomains",
                "ec2:DescribeTransitGatewayPeeringAttachments",
                "ec2:DescribeTransitGatewayRouteTables",
                "ec2:DescribeTransitGateways",
                "ec2:DescribeTransitGatewayVpcAttachments",
                "ec2:DescribeVolumeAttribute",
                "ec2:DescribeVolumeStatus",
                "ec2:DescribeVolumes",
                "ec2:DescribeVolumesModifications",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcClassicLink",
                "ec2:DescribeVpcClassicLinkDnsSupport",
                "ec2:DescribeVpcEndpointConnectionNotifications",
                "ec2:DescribeVpcEndpointConnections",
                "ec2:DescribeVpcEndpointServiceConfigurations",
                "ec2:DescribeVpcEndpointServicePermissions",
                "ec2:DescribeVpcEndpointServices",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcPeeringConnections",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpnConnections",
                "ec2:DescribeVpnGateways",
                "ec2:GetAssociatedIpv6PoolCidrs",
                "ec2:GetConsoleOutput",
                "ec2:GetManagedPrefixListEntries",
                "ec2:GetSerialConsoleAccessStatus",
                "ec2:GetTransitGatewayAttachmentPropagations",
                "ec2:GetTransitGatewayMulticastDomainAssociations",
                "ec2:GetTransitGatewayPrefixListReferences",
                "ec2:GetTransitGatewayRouteTableAssociations",
                "ec2:GetTransitGatewayRouteTablePropagations",
                "ec2:ModifyInstanceAttribute",
                "ec2:RebootInstances",
                "ec2:RunInstances",
                "ec2:SearchLocalGatewayRoutes",
                "ec2:SearchTransitGatewayMulticastGroups",
                "ec2:SearchTransitGatewayRoutes",
                "ec2:StartInstances",
                "ec2:StartNetworkInsightsAnalysis",
                "ec2:StopInstances",
                "ec2:TerminateInstances",
                "elasticloadbalancing:ConfigureHealthCheck",
                "elasticloadbalancing:DescribeAccountLimits",
                "elasticloadbalancing:DescribeInstanceHealth",
                "elasticloadbalancing:DescribeListenerCertificates",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeLoadBalancerAttributes",
                "elasticloadbalancing:DescribeLoadBalancerPolicies",
                "elasticloadbalancing:DescribeLoadBalancerPolicyTypes",
```

```
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeRules",
                "elasticloadbalancing:DescribeSSLPolicies",
                "elasticloadbalancing:DescribeTags",
                "elasticloadbalancing:DescribeTargetGroupAttributes",
                "elasticloadbalancing:DescribeTargetGroups",
                "elasticloadbalancing:DescribeTargetHealth",
                "iam:GetRole",
                "iam:ListRoles",
                "kms:CreateGrant",
                "route53:GetHostedZone",
                "route53:GetHostedZoneCount",
                "route53:ListHostedZones",
                "route53:ListHostedZonesByName",
                "route53:ListResourceRecordSets",
                "s3:GetBucketTagging",
                "s3:GetObjectAcl",
                "s3:GetObjectTagging",
                "s3:ListAllMyBuckets"
                "sts:DecodeAuthorizationMessage",
                "tiros:CreateQuery",
                "tiros:GetQueryAnswer",
                "tiros:GetQueryExplanation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": [
                "arn:aws:s3:::managed-velero*",
                "arn:aws:s3:::*image-registry*"
            ]
        }
    ]
}
```

Table 4.7. ROSA Ingress Operator IAM policy and policy file

Resource	Description
ManagedOpenShift-openshift-ingress-operator-cloud-credentials	An IAM policy that provides the ROSA Ingress Operator with the permissions required to manage external access to a cluster.

Example 4.9. openshift_ingress_operator_cloud_credentials_policy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "elasticloadbalancing:DescribeLoadBalancers",
        "route53:ListHostedZones",
        "route53:ChangeResourceRecordSets",
        "tag:GetResources"
      ],
      "Resource": "*"
    }
  ]
}
```

Table 4.8. ROSA back-end storage IAM policy and policy file

Resource	Description
ManagedOpenShift-openshift-cluster-csi-drivers-ebs-cloud-credentials	An IAM policy required by ROSA to manage back-end storage through the Container Storage Interface (CSI).

Example 4.10. **openshift_cluster_csi_drivers_ebs_cloud_credentials_policy.json**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:CreateSnapshot",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2:DeleteSnapshot",
        "ec2:DeleteTags",
        "ec2:DeleteVolume",
        "ec2:DescribeInstances",
        "ec2:DescribeSnapshots",
        "ec2:DescribeTags",
        "ec2:DescribeVolumes",
        "ec2:DescribeVolumesModifications",
        "ec2:DetachVolume",
        "ec2:ModifyVolume"
      ],
      "Resource": "*"
    }
  ]
}
```

Table 4.9. ROSA Machine Config Operator policy and policy file

Resource	Description
ManagedOpenShift-openshift-machine-api-aws-cloud-credentials	An IAM policy that provides the ROSA Machine Config Operator with the permissions required to perform core cluster functionality.

Example 4.11. **openshift_machine_api_aws_cloud_credentials_policy.json**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeImages",
        "ec2:DescribeInstances",
        "ec2:DescribeInternetGateways",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth",
        "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
        "elasticloadbalancing:RegisterTargets",
        "elasticloadbalancing:DeregisterTargets",
        "iam:PassRole",
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlainText",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:RevokeGrant",
        "kms:CreateGrant",
        "kms:ListGrants"
      ],
```

```
      "Resource": "*",
      "Condition": {
       "Bool": {
        "kms:GrantIsForAWSResource": true
       }
      }
     }
    ]
}
```

Table 4.10. ROSA Cloud Credential Operator policy and policy file

Resource	Description
ManagedOpenShift-openshift-cloud-credential-operator-cloud-credentials	An IAM policy that provides the ROSA Cloud Credential Operator with the permissions required to manage cloud provider credentials.

Example 4.12. **openshift_cloud_credential_operator_cloud_credential_operator_iam_ro_creds_policy.json**

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

Table 4.11. ROSA Image Registry Operator policy and policy file

Resource	Description
ManagedOpenShift-openshift-image-registry-installer-cloud-credentials	An IAM policy that provides the ROSA Image Registry Operator with the permissions required to manage the internal registry storage in AWS S3 for a cluster.

Example 4.13. **openshift_image_registry_installer_cloud_credentials_policy.json**

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:PutBucketTagging",
        "s3:GetBucketTagging",
        "s3:PutBucketPublicAccessBlock",
        "s3:GetBucketPublicAccessBlock",
        "s3:PutEncryptionConfiguration",
        "s3:GetEncryptionConfiguration",
        "s3:PutLifecycleConfiguration",
        "s3:GetLifecycleConfiguration",
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:ListBucketMultipartUploads",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts"
      ],
      "Resource": "*"
    }
  ]
}
```

Additional resources

- For a definition of OpenShift major, minor, and patch versions, see the Red Hat OpenShift Service on AWS update life cycle.

4.2.2. Account-wide IAM role and policy AWS CLI reference

This section lists the **aws** CLI commands that the **rosa** command generates in the terminal. You can run the command in either manual or automatic mode.

Using manual mode for account role creation

The manual role creation mode generates the **aws** commands for you to review and run. The following command starts that process:

```
$ rosa create account-roles --mode manual
```

NOTE

The provided command examples include the **ManagedOpenShift** prefix. The **ManagedOpenShift** prefix is the default value, if you do not specify a custom prefix by using the **--prefix** option.

Command output

```
aws iam create-role \
```

```
 --role-name ManagedOpenShift-Installer-Role \
 --assume-role-policy-document file://sts_installer_trust_policy.json \
 --tags Key=rosa_openshift_version,Value=4.8 Key=rosa_role_prefix,Value=ManagedOpenShift
Key=rosa_role_type,Value=installer

aws iam put-role-policy \
 --role-name ManagedOpenShift-Installer-Role \
 --policy-name ManagedOpenShift-Installer-Role-Policy \
 --policy-document file://sts_installer_permission_policy.json

aws iam create-role \
 --role-name ManagedOpenShift-ControlPlane-Role \
 --assume-role-policy-document file://sts_instance_controlplane_trust_policy.json \
 --tags Key=rosa_openshift_version,Value=4.8 Key=rosa_role_prefix,Value=ManagedOpenShift
Key=rosa_role_type,Value=instance_controlplane

aws iam put-role-policy \
 --role-name ManagedOpenShift-ControlPlane-Role \
 --policy-name ManagedOpenShift-ControlPlane-Role-Policy \
 --policy-document file://sts_instance_controlplane_permission_policy.json

aws iam create-role \
 --role-name ManagedOpenShift-Worker-Role \
 --assume-role-policy-document file://sts_instance_worker_trust_policy.json \
 --tags Key=rosa_openshift_version,Value=4.8 Key=rosa_role_prefix,Value=ManagedOpenShift
Key=rosa_role_type,Value=instance_worker

aws iam put-role-policy \
 --role-name ManagedOpenShift-Worker-Role \
 --policy-name ManagedOpenShift-Worker-Role-Policy \
 --policy-document file://sts_instance_worker_permission_policy.json

aws iam create-role \
 --role-name ManagedOpenShift-Support-Role \
 --assume-role-policy-document file://sts_support_trust_policy.json \
 --tags Key=rosa_openshift_version,Value=4.8 Key=rosa_role_prefix,Value=ManagedOpenShift
Key=rosa_role_type,Value=support

aws iam put-role-policy \
 --role-name ManagedOpenShift-Support-Role \
 --policy-name ManagedOpenShift-Support-Role-Policy \
 --policy-document file://sts_support_permission_policy.json

aws iam create-policy \
 --policy-name ManagedOpenShift-openshift-ingress-operator-cloud-credentials \
 --policy-document file://openshift_ingress_operator_cloud_credentials_policy.json \
 --tags Key=rosa_openshift_version,Value=4.8 Key=rosa_role_prefix,Value=ManagedOpenShift
Key=operator_namespace,Value=openshift-ingress-operator Key=operator_name,Value=cloud-
credentials

aws iam create-policy \
 --policy-name ManagedOpenShift-openshift-cluster-csi-drivers-ebs-cloud-credent \
 --policy-document file://openshift_cluster_csi_drivers_ebs_cloud_credentials_policy.json \
 --tags Key=rosa_openshift_version,Value=4.8 Key=rosa_role_prefix,Value=ManagedOpenShift
Key=operator_namespace,Value=openshift-cluster-csi-drivers Key=operator_name,Value=ebs-cloud-
credentials
```

```
aws iam create-policy \
 --policy-name ManagedOpenShift-openshift-machine-api-aws-cloud-credentials \
 --policy-document file://openshift_machine_api_aws_cloud_credentials_policy.json \
 --tags Key=rosa_openshift_version,Value=4.8 Key=rosa_role_prefix,Value=ManagedOpenShift
Key=operator_namespace,Value=openshift-machine-api Key=operator_name,Value=aws-cloud-
credentials

aws iam create-policy \
 --policy-name ManagedOpenShift-openshift-cloud-credential-operator-cloud-crede \
 --policy-document
file://openshift_cloud_credential_operator_cloud_credential_operator_iam_ro_creds_policy.json \
 --tags Key=rosa_openshift_version,Value=4.8 Key=rosa_role_prefix,Value=ManagedOpenShift
Key=operator_namespace,Value=openshift-cloud-credential-operator
Key=operator_name,Value=cloud-credential-operator-iam-ro-creds

aws iam create-policy \
 --policy-name ManagedOpenShift-openshift-image-registry-installer-cloud-creden \
 --policy-document file://openshift_image_registry_installer_cloud_credentials_policy.json \
 --tags Key=rosa_openshift_version,Value=4.8 Key=rosa_role_prefix,Value=ManagedOpenShift
Key=operator_namespace,Value=openshift-image-registry Key=operator_name,Value=installer-
cloud-credentials
```

Using auto mode for role creation

When you add the **--mode auto** argument, the **rosa** CLI tool creates your roles and policies. The following command starts that process:

```
$ rosa create account-roles --mode auto
```

NOTE

The provided command examples include the **ManagedOpenShift** prefix. The **ManagedOpenShift** prefix is the default value, if you do not specify a custom prefix by using the **--prefix** option.

Command output

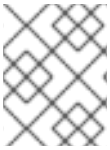
```
I: Creating roles using 'arn:aws:iam::<ARN>:user/<UserID>'
? Create the 'ManagedOpenShift-Installer-Role' role? Yes
I: Created role 'ManagedOpenShift-Installer-Role' with ARN 'arn:aws:iam::
<ARN>:role/ManagedOpenShift-Installer-Role'
? Create the 'ManagedOpenShift-ControlPlane-Role' role? Yes
I: Created role 'ManagedOpenShift-ControlPlane-Role' with ARN 'arn:aws:iam::
<ARN>:role/ManagedOpenShift-ControlPlane-Role'
? Create the 'ManagedOpenShift-Worker-Role' role? Yes
I: Created role 'ManagedOpenShift-Worker-Role' with ARN 'arn:aws:iam::
<ARN>:role/ManagedOpenShift-Worker-Role'
? Create the 'ManagedOpenShift-Support-Role' role? Yes
I: Created role 'ManagedOpenShift-Support-Role' with ARN 'arn:aws:iam::
<ARN>:role/ManagedOpenShift-Support-Role'
? Create the operator policies? Yes
I: Created policy with ARN 'arn:aws:iam::<ARN>:policy/ManagedOpenShift-openshift-machine-api-
aws-cloud-credentials'
I: Created policy with ARN 'arn:aws:iam::<ARN>:policy/ManagedOpenShift-openshift-cloud-
```

```
credential-operator-cloud-crede'
I: Created policy with ARN 'arn:aws:iam::<ARN>:policy/ManagedOpenShift-openshift-image-registry-installer-cloud-creden'
I: Created policy with ARN 'arn:aws:iam::<ARN>:policy/ManagedOpenShift-openshift-ingress-operator-cloud-credentials'
I: Created policy with ARN 'arn:aws:iam::<ARN>:policy/ManagedOpenShift-openshift-cluster-csi-drivers-ebs-cloud-credent'
I: Created policy with ARN 'arn:aws:iam::<ARN>:policy/ManagedOpenShift-openshift-cloud-network-config-controller-cloud'
I: To create a cluster with these roles, run the following command:
rosa create cluster --sts
```

4.3. CLUSTER-SPECIFIC OPERATOR IAM ROLE REFERENCE

This section provides details about the Operator IAM roles that are required for Red Hat OpenShift Service on AWS (ROSA) deployments that use STS. The cluster Operators use the Operator roles to obtain the temporary permissions required to carry out cluster operations, such as managing back-end storage, cloud provider credentials, and external access to a cluster.

When you create the Operator roles, the account-wide Operator policies for the matching cluster version are attached to the roles. The Operator policies are tagged with the Operator and version they are compatible with. The correct policy for an Operator role is determined by using the tags.



NOTE

If more than one matching policy is available in your account for an Operator role, an interactive list of options is provided when you create the Operator.

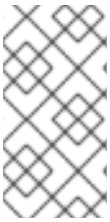
Table 4.12. ROSA cluster-specific Operator roles

Resource	Description
<cluster_name>-<hash>-openshift-cluster-csi-drivers-ebs-cloud-credentials	An IAM role required by ROSA to manage back-end storage through the Container Storage Interface (CSI).
<cluster_name>-<hash>-openshift-machine-api-aws-cloud-credentials	An IAM role required by the ROSA Machine Config Operator to perform core cluster functionality.
<cluster_name>-<hash>-openshift-cloud-credential-operator-cloud-credentials	An IAM role required by the ROSA Cloud Credential Operator to manage cloud provider credentials.
<cluster_name>-<hash>-openshift-image-registry-installer-cloud-credentials	An IAM role required by the ROSA Image Registry Operator to manage the internal registry storage in AWS S3 for a cluster.
<cluster_name>-<hash>-openshift-ingress-operator-cloud-credentials	An IAM role required by the ROSA Ingress Operator to manage external access to a cluster.

4.3.1. Operator IAM role AWS CLI reference

This section lists the **aws** CLI commands that are shown in the terminal when you run the following **rosa** command using **manual** mode:

```
$ rosa create operator-roles --mode manual --cluster <cluster_name>
```



NOTE

When using **manual** mode, the **aws** commands are printed to the terminal for your review. After reviewing the **aws** commands, you must run them manually. Alternatively, you can specify **--mode auto** with the **rosa create** command to run the **aws** commands immediately.

Command output

```
aws iam create-role \
 --role-name <cluster_name>-<hash>-openshift-cluster-csi-drivers-ebs-cloud-credent \
 --assume-role-policy-document file://operator_cluster_csi_drivers_ebs_cloud_credentials_policy.json
\
 --tags Key=rosa_cluster_id,Value=<id> Key=rosa_openshift_version,Value=4.8
Key=rosa_role_prefix,Value= Key=operator_namespace,Value=openshift-cluster-csi-drivers
Key=operator_name,Value=ebs-cloud-credentials

aws iam attach-role-policy \
 --role-name <cluster_name>-<hash>-openshift-cluster-csi-drivers-ebs-cloud-credent \
 --policy-arn arn:aws:iam::<aws_account_id>:policy/ManagedOpenShift-openshift-cluster-csi-drivers-
ebs-cloud-credent

aws iam create-role \
 --role-name <cluster_name>-<hash>-openshift-machine-api-aws-cloud-credentials \
 --assume-role-policy-document file://operator_machine_api_aws_cloud_credentials_policy.json \
 --tags Key=rosa_cluster_id,Value=<id> Key=rosa_openshift_version,Value=4.8
Key=rosa_role_prefix,Value= Key=operator_namespace,Value=openshift-machine-api
Key=operator_name,Value=aws-cloud-credentials

aws iam attach-role-policy \
 --role-name <cluster_name>-<hash>-openshift-machine-api-aws-cloud-credentials \
 --policy-arn arn:aws:iam::<aws_account_id>:policy/ManagedOpenShift-openshift-machine-api-aws-
cloud-credentials

aws iam create-role \
 --role-name <cluster_name>-<hash>-openshift-cloud-credential-operator-cloud-crede \
 --assume-role-policy-document
file://operator_cloud_credential_operator_cloud_credential_operator_iam_ro_creds_policy.json \
 --tags Key=rosa_cluster_id,Value=<id> Key=rosa_openshift_version,Value=4.8
Key=rosa_role_prefix,Value= Key=operator_namespace,Value=openshift-cloud-credential-operator
Key=operator_name,Value=cloud-credential-operator-iam-ro-creds

aws iam attach-role-policy \
 --role-name <cluster_name>-<hash>-openshift-cloud-credential-operator-cloud-crede \
 --policy-arn arn:aws:iam::<aws_account_id>:policy/ManagedOpenShift-openshift-cloud-credential-
operator-cloud-crede
```

```
aws iam create-role \
 --role-name <cluster_name>-<hash>-openshift-image-registry-installer-cloud-creden \
 --assume-role-policy-document file://operator_image_registry_installer_cloud_credentials_policy.json
\
 --tags Key=rosa_cluster_id,Value=<id> Key=rosa_openshift_version,Value=4.8
Key=rosa_role_prefix,Value= Key=operator_namespace,Value=openshift-image-registry
Key=operator_name,Value=installer-cloud-credentials

aws iam attach-role-policy \
 --role-name <cluster_name>-<hash>-openshift-image-registry-installer-cloud-creden \
 --policy-arn arn:aws:iam::<aws_account_id>:policy/ManagedOpenShift-openshift-image-registry-
installer-cloud-creden

aws iam create-role \
 --role-name <cluster_name>-<hash>-openshift-ingress-operator-cloud-credentials \
 --assume-role-policy-document file://operator_ingress_operator_cloud_credentials_policy.json \
 --tags Key=rosa_cluster_id,Value=<id> Key=rosa_openshift_version,Value=4.8
Key=rosa_role_prefix,Value= Key=operator_namespace,Value=openshift-ingress-operator
Key=operator_name,Value=cloud-credentials

aws iam attach-role-policy \
 --role-name <cluster_name>-<hash>-openshift-ingress-operator-cloud-credentials \
 --policy-arn arn:aws:iam::<aws_account_id>:policy/ManagedOpenShift-openshift-ingress-operator-
cloud-credentials
```

NOTE

The command examples provided in the table include Operator roles that use the **ManagedOpenShift** prefix. If you defined a custom prefix when you created the account-wide roles and policies, including the Operator policies, you must reference it by using the **--prefix <prefix_name>** option when you create the Operator roles.

4.3.2. About custom Operator IAM role prefixes

Each Red Hat OpenShift Service on AWS (ROSA) cluster that uses the AWS Security Token Service (STS) requires cluster-specific Operator IAM roles.

By default, the Operator role names are prefixed with the cluster name and a random 4-digit hash. For example, the Cloud Credential Operator IAM role for a cluster named **mycluster** has the default name **mycluster-<hash>-openshift-cloud-credential-operator-cloud-credentials**, where **<hash>** is a random 4-digit string.

This default naming convention enables you to easily identify the Operator IAM roles for a cluster in your AWS account.

When you create the Operator roles for a cluster, you can optionally specify a custom prefix to use instead of **<cluster_name>-<hash>**. By using a custom prefix, you can prepend logical identifiers to your Operator role names to meet the requirements of your environment. For example, you might prefix the cluster name and the environment type, such as **mycluster-dev**. In that example, the Cloud Credential Operator role name with the custom prefix is **mycluster-dev-openshift-cloud-credential-operator-cloud-credenti**.

NOTE

The role names are truncated to 64 characters.

Additional resources

- For steps to create the cluster-specific Operator IAM roles using a custom prefix, see Creating a cluster with customizations using the CLI or Creating a cluster with customizations by using OpenShift Cluster Manager.

4.4. OIDC PROVIDER REQUIREMENTS FOR OPERATOR AUTHENTICATION

For ROSA installations that use STS, you must create a cluster-specific OIDC provider that is used by the cluster Operators to authenticate.

4.4.1. OIDC provider AWS CLI reference

This section lists the **aws** CLI command that is shown in the terminal when you run the following **rosa** command using **manual** mode:

```
$ rosa create oidc-provider --mode manual --cluster <cluster_name>
```

NOTE

When using **manual** mode, the **aws** command is printed to the terminal for your review. After reviewing the **aws** command, you must run it manually. Alternatively, you can specify **--mode auto** with the **rosa create** command to run the **aws** command immediately.

Command output

```
aws iam create-open-id-connect-provider \
 --url https://rh-oidc.s3.<aws_region>.amazonaws.com/<cluster_id> \
 --client-id-list openshift sts.amazonaws.com \
 --thumbprint-list <thumbprint> (1)
```

(1) The thumbprint is generated automatically when you run the **rosa create oidc-provider** command. For more information about using thumbprints with AWS Identity and Access Management (IAM) OpenID Connect (OIDC) identity providers, see the AWS documentation.

CHAPTER 5. GETTING SUPPORT FOR RED HAT OPENSHIFT SERVICE ON AWS

Get support for Red Hat OpenShift Service on AWS (ROSA).

5.1. GETTING SUPPORT

If you experience difficulty with a procedure described in this documentation, visit the Red Hat Customer Portal. Through the Customer Portal, you can:

- Search or browse through the Red Hat Knowledgebase of technical support articles about Red Hat products.
- Access other product documentation.
- Submit a support case to Red Hat Support:
    a. Click **Open a New Case**.
    b. Select the reason for the support ticket, such as **Defect/Bug** or **Account/Customer Service Request**.
    c. In the **Product** field, enter **OpenShift** to filter the list. Select **Red Hat OpenShift Service on AWS** and the version from the drop-down menus.
    d. Complete the remaining fields.
    e. On the *Review* page, select the correct cluster ID that you are contacting support about, and click **Submit**.

You can also get support from AWS Support as long as you have a valid AWS support contract.

If you have a suggestion for improving this documentation or have found an error, submit a Jira issue for the most relevant documentation component. Be sure to provide specific details, such as the section name and Red Hat OpenShift Service on AWS version.