Red Hat OpenShift Service on AWS 4

Installing, accessing, and deleting ROSA clusters

Installing, accessing, and deleting Red Hat OpenShift Service on AWS (ROSA) clusters.

Last Updated: 2022-06-28

Red Hat OpenShift Service on AWS 4 Installing, accessing, and deleting ROSA clusters

Installing, accessing, and deleting Red Hat OpenShift Service on AWS (ROSA) clusters.

Legal Notice

Abstract

This document provides information on how to install Red Hat OpenShift Service on AWS (ROSA) clusters. The document also provides details on how to access a cluster, configure identity providers, revoke cluster access, and delete a cluster.

Table of Contents

CHAPTER 1. CREATING A ROSA CLUSTER WITH STS USING THE DEFAULT OPTIONS

Create a Red Hat OpenShift Service on AWS (ROSA) cluster quickly by using the default options and automatic AWS Identity and Access Management (IAM) resource creation. You can deploy your cluster by using Red Hat OpenShift Cluster Manager or the ROSA CLI (**rosa**).

The procedures in this document use **auto** mode to immediately create the required IAM resources using the current AWS account, including the account-wide IAM roles and policies, Operator policies, cluster-specific Operator roles, and the OpenID Connect (OIDC) identity provider.

Alternatively, you can use **manual** mode which outputs the **aws** commands needed to create the IAM resources instead of deploying them automatically. For information about the **auto** and **manual** deployment modes, see Understanding the auto and manual deployment modes. For steps to deploy a ROSA cluster using **manual** mode, see Creating a cluster using customizations.

1.1. CREATING A CLUSTER WITH THE DEFAULT OPTIONS

Use the default options and **auto** mode to create a Red Hat OpenShift Service on AWS (ROSA) cluster quickly. You can deploy your cluster by using Red Hat OpenShift Cluster Manager or the ROSA CLI (**rosa**).

1.1.1. Creating a cluster with the default options using OpenShift Cluster Manager

When using Red Hat OpenShift Cluster Manager to create a Red Hat OpenShift Service on AWS (ROSA) cluster that uses the AWS Security Token Service (STS), you can select the default options to create the cluster quickly.

Prerequisites

- You have completed the AWS prerequisites for ROSA with STS.
- You have available AWS service quotas.
- You have enabled the ROSA service in the AWS Console.
- You have installed and configured the latest ROSA CLI (**rosa**) on your installation host.

  NOTE

  To successfully install ROSA 4.10 clusters, use the latest version of the ROSA CLI.

- You have logged in to your Red Hat account by using the **rosa** CLI.
- You have verified that the AWS Elastic Load Balancing (ELB) service role exists in your AWS account.

Procedure

1. Navigate to OpenShift Cluster Manager and select **Create cluster**.
2. On the **Create an OpenShift cluster** page, select **Create cluster** in the **Red Hat OpenShift Service on AWS (ROSA)** row.

3. Review and complete the **Prerequisites** listed on the **Accounts and roles** page. Select the checkbox to acknowledge that you have read and completed all of the prerequisites.

4. Select an AWS account from the **Associated AWS account** drop-down menu. If no associated AWS accounts are found, click **Associate AWS account** and follow these steps:

   a. On the **Authenticate** page, click the copy button next to the **rosa login** command. The provided command includes your ROSA API login token.

      NOTE

      You can also load your API token on the OpenShift Cluster Manager API Token page on OpenShift Cluster Manager.

   b. Run the copied command in the CLI to log in to your ROSA account:

      ```
      $ rosa login --token=<api_login_token> 1
      ```

      1 Replace **<api_login_token>** with the token that is provided in the copied command.

      Example output

      ```
      I: Logged in as '<username>' on 'https://api.openshift.com'
      ```

   c. On the **Authenticate page** in OpenShift Cluster Manager, click **Next**.

   d. On the **OCM role** page, click the copy button next to the **Admin OCM role** command. The admin role enables automatic deployment of the cluster-specific Operator roles and the OpenID Connect (OIDC) provider by using OpenShift Cluster Manager.

   e. Run the copied command in the CLI and follow the prompts to create the OpenShift Cluster Manager IAM role.
      The following example creates an admin OpenShift Cluster Manager IAM role using the default options and **auto** mode for immediate STS resource creation. The example also links the OpenShift Cluster Manager IAM role to a Red Hat organization:

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

      Example output

      ```
      I: Creating ocm role
      ? Role prefix: ManagedOpenShift 1
      ? Permissions boundary ARN (optional): 2
      ? Role creation mode: auto 3
      I: Creating role using 'arn:aws:iam::<aws_account_id>:user/<aws_username>'
      ? Create the 'ManagedOpenShift-OCM-Role-<red_hat_organization_external_id>' role?
      Yes
      I: Created role 'ManagedOpenShift-OCM-Role-<red_hat_organization_external_id>' with
      ARN 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-OCM-Role-
      <red_hat_organization_external_id>'
      I: Linking OCM role
      ? OCM Role ARN: arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-OCM-Role-
      <red_hat_organization_external_id>
      ```

```
? Link the 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-OCM-Role-
<red_hat_organization_external_id>' role with organization '<red_hat_organization_id>'?
Yes ❹
I: Successfully linked role-arn 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-
OCM-Role-<red_hat_organization_external_id>' with organization account
'<red_hat_organization_id>'
```

❶ Specifies the prefix to include in the OpenShift Cluster Manager IAM role name. The default is **ManagedOpenShift**.

❷ Optional: Specifies a permissions boundary Amazon Resource Name (ARN) for the role. For more information, see Permissions boundaries for IAM entities in the AWS documentation.

❸ Selects the role creation mode. You can use **auto** mode to automatically create the OpenShift Cluster Manager IAM role and link it to your Red Hat organization account.

❹ Links the OpenShift Cluster Manager IAM role to your Red Hat organization account.

f. Select **Next** on the OpenShift Cluster Manager **OCM role** page.

g. On the **User role** page, click the copy button for the **User role** command and run the command in the CLI. Follow the prompts to create the user role:

```
$ rosa create user-role
```

Example output

```
I: Creating User role
? Role prefix: ManagedOpenShift ❶
? Permissions boundary ARN (optional): ❷
? Role creation mode: auto ❸
I: Creating ocm user role using 'arn:aws:iam::<aws_account_id>:user/<aws_username>'
? Create the 'ManagedOpenShift-User-<ocm_username>-Role' role? Yes
I: Created role 'ManagedOpenShift-User-<ocm_username>-Role' with ARN
'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-User-<ocm_username>-Role'
I: Linking User role
? User Role ARN: arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-User-
<ocm_username>-Role
? Link the 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-User-
<ocm_username>-Role' role with account '<ocm_user_account_id>'? Yes ❹
I: Successfully linked role ARN 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-
User-<ocm_username>-Role' with account '<ocm_user_account_id>'
```

❶ Specifies the prefix to include in the user role name. The default is **ManagedOpenShift**.

❷ Optional: Specifies a permissions boundary Amazon Resource Name (ARN) for the role. For more information, see Permissions boundaries for IAM entities in the AWS documentation.

❸ Selects the role creation mode. You can use **auto** mode to automatically create the user role and link it to your OpenShift Cluster Manager user account.

4 Links the user role to your OpenShift Cluster Manager user account.

h. On the OpenShift Cluster Manager **User role** page, select **Ok**.

i. Under the **Accounts and roles** page, verify that your AWS account is listed as an **Associated AWS account**

5. If the required AWS IAM **Account roles** are not automatically detected and listed on the **Accounts and roles** page, create the roles and policies:

a. Click the copy buffer next to the **rosa create account-roles** command. Run the command in the CLI to create the required AWS account-wide roles and policies, including the Operator policies::

```
$ rosa create account-roles
```

Example output

```
I: Logged in as '<ocm_username>' on 'https://api.openshift.com'
I: Validating AWS credentials...
I: AWS credentials are valid!
I: Validating AWS quota...
I: AWS quota ok. If cluster installation fails, validate actual AWS resource usage against
https://docs.openshift.com/rosa/rosa_getting_started/rosa-required-aws-service-
quotas.html
I: Verifying whether OpenShift command-line tool is available...
I: Current OpenShift Client Version: 4.9.12
I: Creating account roles
? Role prefix: ManagedOpenShift 1
? Permissions boundary ARN (optional): 2
? Role creation mode: auto 3
I: Creating roles using 'arn:aws:iam::<aws_account_number>:user/<aws_username>'
? Create the 'ManagedOpenShift-Installer-Role' role? Yes 4
I: Created role 'ManagedOpenShift-Installer-Role' with ARN 'arn:aws:iam::
<aws_account_number>:role/ManagedOpenShift-Installer-Role'
? Create the 'ManagedOpenShift-ControlPlane-Role' role? Yes 5
I: Created role 'ManagedOpenShift-ControlPlane-Role' with ARN 'arn:aws:iam::
<aws_account_number>:role/ManagedOpenShift-ControlPlane-Role'
? Create the 'ManagedOpenShift-Worker-Role' role? Yes 6
I: Created role 'ManagedOpenShift-Worker-Role' with ARN 'arn:aws:iam::
<aws_account_number>:role/ManagedOpenShift-Worker-Role'
? Create the 'ManagedOpenShift-Support-Role' role? Yes 7
I: Created role 'ManagedOpenShift-Support-Role' with ARN 'arn:aws:iam::
<aws_account_number>:role/ManagedOpenShift-Support-Role'
? Create the operator policies? Yes 8
I: Created policy with ARN 'arn:aws:iam::
<aws_account_number>:policy/ManagedOpenShift-openshift-cloud-credential-operator-
cloud-crede'
I: Created policy with ARN 'arn:aws:iam::
<aws_account_number>:policy/ManagedOpenShift-openshift-image-registry-installer-
cloud-creden'
I: Created policy with ARN 'arn:aws:iam::
<aws_account_number>:policy/ManagedOpenShift-openshift-ingress-operator-cloud-
credentials'
```

```
I: Created policy with ARN 'arn:aws:iam::
<aws_account_number>:policy/ManagedOpenShift-openshift-cluster-csi-drivers-ebs-
cloud-credent'
I: Created policy with ARN 'arn:aws:iam::
<aws_account_number>:policy/ManagedOpenShift-openshift-cloud-network-config-
controller-cloud'
I: Created policy with ARN 'arn:aws:iam::
<aws_account_number>:policy/ManagedOpenShift-openshift-machine-api-aws-cloud-
credentials'
I: To create a cluster with these roles, run the following command:
rosa create cluster --sts
```

1 Specifies the prefix to include in the account-wide role and policy names. The default is **ManagedOpenShift**.

2 Optional: Specifies a permissions boundary Amazon Resource Name (ARN) for the roles. For more information, see Permissions boundaries for IAM entities in the AWS documentation.

3 Selects the role creation mode. You can use **auto** mode to automatically create the account wide roles and policies.

4 5 6 7 Creates the account-wide installer, control plane, worker and support roles and corresponding inline IAM policies. For more information, see *Account-wide IAM role and policy reference*.

8 Creates the cluster-specific Operator IAM roles that permit the ROSA cluster Operators to carry out core OpenShift functionality. For more information, see *Account-wide IAM role and policy reference*.

b. On the **Accounts and roles** page, click **Refresh ARNs** and verify that the installer, support, worker, and control plane account roles are detected.

6. Select **Next**.

7. On the **Cluster details** page, provide a name for your cluster and specify the cluster details:

    a. Add a **Cluster name**.

    b. Select a cluster version from the **Version** drop-down menu.

    c. Select a cloud provider region from the **Region** drop-down menu.

    d. Select a **Single zone** or **Multi-zone** configuration.

    e. Leave **Enable user workload monitoring** selected to monitor your own projects in isolation from Red Hat Site Reliability Engineer (SRE) platform metrics. This option is enabled by default.

    f. Optional: Select **Enable additional etcd encryption** if you require etcd key value encryption. With this option, the etcd key values are encrypted, but not the keys. This option is in addition to the control plane storage encryption that encrypts the etcd volumes in Red Hat OpenShift Service on AWS clusters by default.

> NOTE
>
> 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. Consider enabling etcd encryption only if you specifically require it for your use case.

   g. Optional: Select **Encrypt persistent volumes with customer keys**if you want to provide your own AWS Key Management Service (KMS) key Amazon Resource Name (ARN). The key is used for encryption of persistent volumes in your cluster.

   h. Click **Next**.

8. On the **Default machine pool** page, select a **Compute node instance type**

> NOTE
>
> After your cluster is created, you can change the number of compute nodes in your cluster, but you cannot change the compute node instance type in the default machine pool. The number and types of nodes available to you depend on whether you use single or multiple availability zones. They also depend on what is enabled and available in your AWS account and the selected region.

9. Optional: Configure autoscaling for the default machine pool:

   a. Select **Enable autoscaling** to automatically scale the number of machines in your default machine pool to meet the deployment needs.

   b. Set the minimum and maximum node count limits for autoscaling. The cluster autoscaler does not reduce or increase the default machine pool node count beyond the limits that you specify.

      - If you deployed your cluster using a single availability zone, set the **Minimum node count** and **Maximum node count**. This defines the minimum and maximum compute node limits in the availability zone.

      - If you deployed your cluster using multiple availability zones, set the **Minimum nodes per zone** and **Maximum nodes per zone**. This defines the minimum and maximum compute node limits per zone.

> NOTE
>
> Alternatively, you can set your autoscaling preferences for the default machine pool after the machine pool is created.

10. If you did not enable autoscaling, select a compute node count for your default machine pool:

    - If you deployed your cluster using a single availability zone, select a **Compute node count** from the drop-down menu. This defines the number of compute nodes to provision to the machine pool for the zone.

    - If you deployed your cluster using multiple availability zones, select a **Compute node count (per zone)** from the drop-down menu. This defines the number of compute nodes to provision to the machine pool per zone.

11. Optional: Expand **Edit node labels** to add labels to your nodes. Click **Add label** to add more node labels and select **Next**.

12. In the **Cluster privacy** section of the **Network configuration** page, select **Public** or **Private** to use either public or private API endpoints and application routes for your cluster.

    IMPORTANT

    If you are using private API endpoints, you cannot access your cluster until you update the network settings in your cloud provider account.

13. Optional: If you opted to use public API endpoints, you can select **Install into an existing VPC** to install your cluster into an existing VPC.

    NOTE

    If you opted to use private API endpoints, you must use an existing VPC and PrivateLink and the **Install into an existing VPC**and **Use a PrivateLink** options are automatically selected. With these options, the Red Hat Site Reliability Engineering (SRE) team can connect to the cluster to assist with support by using only AWS PrivateLink endpoints.

14. Optional: If you are installing your cluster into an existing VPC, select **Configure a cluster-wide proxy** to enable an HTTP or HTTPS proxy to deny direct access to the internet from your cluster.

15. Click **Next**.

16. If you opted to install the cluster in an existing AWS VPC, provide your **Virtual Private Cloud (VPC) subnet settings**.

    NOTE

    You must ensure that your VPC is configured with a public and a private subnet for each availability zone that you want the cluster installed into. If you opted to use PrivateLink, only private subnets are required.

17. In the **CIDR ranges** dialog, configure custom classless inter-domain routing (CIDR) ranges or use the defaults that are provided and click **Next**.

    NOTE

    If you are installing into a VPC, the **Machine CIDR** range must match the VPC subnets.

    IMPORTANT

    CIDR configurations cannot be changed later. Confirm your selections with your network administrator before proceeding.

18. Under the **Cluster roles and policies** page, select **Auto** mode. With this mode, you can automatically create the cluster-specific Operator IAM roles and OIDC provider.

NOTE

To enable **Auto** mode, the OpenShift Cluster Manager IAM role must have administrator capabilities.

If you alternatively want to create the cluster-specific IAM roles and the OIDC provider by using **Manual** mode, see *Creating a cluster using customizations* .

19. Optional: Specify a **Custom operator roles prefix**for your cluster-specific Operator roles.

NOTE

By default, the cluster-specific Operator role names are prefixed with the cluster name and random 4-digit hash. You can optionally specify a custom prefix to replace **<cluster_name>-<hash>** in the role names. The prefix is applied when you create the cluster-specific Operator IAM roles. For information about the prefix, see *About custom Operator IAM role prefixes* .

20. Select **Next**.

21. On the **Cluster update strategy** page, configure your update preferences:

    a. Choose a cluster update method:

    - Select **Individual updates** if you want to schedule each update individually. This is the default option.

    - Select **Recurring updates** to update your cluster on your preferred day and start time, when updates are available.

      IMPORTANT

      Even when you opt for recurring updates, you must update the account-wide and cluster-specific IAM resources before you upgrade your cluster between minor releases.

      NOTE

      You can review the end-of-life dates in the update life cycle documentation for Red Hat OpenShift Service on AWS. For more information, see *Red Hat OpenShift Service on AWS update life cycle* .

    b. If you opted for recurring updates, select a preferred day of the week and upgrade start time in UTC from the drop-down menus.

    c. Optional: You can set a grace period for **Node draining** during cluster upgrades. A **1 hour** grace period is set by default.

    d. Click **Next**.

NOTE

In the event of critical security concerns that significantly impact the security or stability of a cluster, Red Hat Site Reliability Engineering (SRE) might schedule automatic updates to the latest z-stream version that is not impacted. The updates are applied within 48 hours after customer notifications are provided. For a description of the critical impact security rating, see Understanding Red Hat security ratings .

22. Review the summary of your selections and click **Create cluster** to start the cluster installation.

Verification

- You can monitor the progress of the installation in the **Overview** page for your cluster. You can view the installation logs on the same page. Your cluster is ready when the **Status** in the **Details** section of the page is listed as **Ready**.

NOTE

If the installation fails or the cluster **State** does not change to **Ready** after about 40 minutes, check the installation troubleshooting documentation for details. For more information, see *Troubleshooting installations*. For steps to contact Red Hat Support for assistance, see *Getting support for Red Hat OpenShift Service on AWS*.

1.1.2. Creating a cluster with the default options using the CLI

When using the Red Hat OpenShift Service on AWS (ROSA) CLI (**rosa**) to create a cluster that uses the AWS Security Token Service (STS), you can select the default options to create the cluster quickly.

Prerequisites

- You have completed the AWS prerequisites for ROSA with STS.
- You have available AWS service quotas.
- You have enabled the ROSA service in the AWS Console.
- You have installed and configured the latest ROSA CLI (**rosa**) on your installation host.

NOTE

To successfully install ROSA 4.10 clusters, use the latest version of the ROSA CLI.

- You have logged in to your Red Hat account by using the **rosa** CLI.
- You have verified that the AWS Elastic Load Balancing (ELB) service role exists in your AWS account.

Procedure

1. Create the required account-wide roles and policies, including the Operator policies:

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

NOTE

When using **auto** mode, you can optionally specify the **-y** argument to bypass the interactive prompts and automatically confirm operations.

2. Create a cluster with STS using the defaults. When you use the defaults, the latest stable OpenShift version is installed:

```
$ rosa create cluster --cluster-name <cluster_name> --sts --mode auto 1
```

1 Replace **<cluster_name>** with the name of your cluster.

NOTE

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

3. Check the status of your cluster:

```
$ rosa describe cluster --cluster <cluster_name|cluster_id>
```

The following **State** field changes are listed in the output as the cluster installation progresses:

- **waiting (Waiting for OIDC configuration)**
- **pending (Preparing account)**
- **installing (DNS setup in progress)**
- **installing**
- **ready**

NOTE

If the installation fails or the **State** field does not change to **ready** after about 40 minutes, check the installation troubleshooting documentation for details. For more information, see *Troubleshooting installations*. For steps to contact Red Hat Support for assistance, see *Getting support for Red Hat OpenShift Service on AWS*.

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

```
$ rosa logs install --cluster <cluster_name|cluster_id> --watch 1
```

1 Specify the **--watch** flag to watch for new log messages as the installation progresses. This argument is optional.

1.2. NEXT STEPS

- Accessing a ROSA cluster

1.3. ADDITIONAL RESOURCES

- For steps to deploy a ROSA cluster using manual mode, see Creating a cluster using customizations.
- For more information about the AWS Identity Access Management (IAM) resources required to deploy Red Hat OpenShift Service on AWS with STS, see About IAM resources for clusters that use STS.
- For details about optionally setting an Operator role name prefix, see About custom Operator IAM role prefixes.
- For information about the prerequisites to installing ROSA with STS, see AWS prerequisites for ROSA with STS.
- For details about using the **auto** and **manual** modes to create the required STS resources, see Understanding the auto and manual deployment modes .
- For more information about using OpenID Connect (OIDC) identity providers in AWS IAM, see Creating OpenID Connect (OIDC) identity providers in the AWS documentation.
- For more information about troubleshooting ROSA cluster installations, see Troubleshooting installations.
- For steps to contact Red Hat Support for assistance, see Getting support for Red Hat OpenShift Service on AWS.

CHAPTER 2. CREATING A ROSA CLUSTER WITH STS USING CUSTOMIZATIONS

Create a Red Hat OpenShift Service on AWS (ROSA) cluster with the AWS Security Token Service (STS) using customizations. You can deploy your cluster by using Red Hat OpenShift Cluster Manager or the ROSA CLI (**rosa**).

With the procedures in this document, you can also choose between the **auto** and **manual** modes when creating the required AWS Identity and Access Management (IAM) resources.

2.1. UNDERSTANDING THE AUTO AND MANUAL DEPLOYMENT MODES

When installing a Red Hat OpenShift Service on AWS (ROSA) cluster that uses the AWS Security Token Service (STS), you can choose between the **auto** and **manual** ROSA CLI (**rosa**) modes to create the required AWS Identity and Access Management (IAM) resources.

auto mode

With this mode, **rosa** immediately creates the required IAM roles and policies, and an OpenID Connect (OIDC) provider in your AWS account.

manual mode

With this mode, **rosa** outputs the **aws** commands needed to create the IAM resources. The corresponding policy JSON files are also saved to the current directory. By using **manual** mode, you can review the generated **aws** commands before running them manually. **manual** mode also enables you to pass the commands to another administrator or group in your organization so that they can create the resources.

> IMPORTANT
>
> If you opt to use **manual** mode, the cluster installation waits until you create the cluster-specific Operator roles and OIDC provider manually. After you create the resources, the installation proceeds. For more information, see *Creating the Operator roles and OIDC provider using OpenShift Cluster Manager*.

For more information about the AWS IAM resources required to install ROSA with STS, see *About IAM resources for clusters that use STS*.

2.1.1. Creating the Operator roles and OIDC provider using OpenShift Cluster Manager

If you use Red Hat OpenShift Cluster Manager to install your cluster and opt to create the required AWS IAM Operator roles and the OIDC provider using **manual** mode, you are prompted to select one of the following methods to install the resources. The options are provided to enable you to choose a resource creation method that suits the needs of your organization:

AWS CLI (**aws**)

With this method, you can download and extract an archive file that contains the **aws** commands and policy files required to create the IAM resources. Run the provided CLI commands from the directory that contains the policy files to create the Operator roles and the OIDC provider.

ROSA CLI (**rosa**)

You can run the commands provided by this method to create the Operator roles and the OIDC provider for your cluster using **rosa**.

If you use **auto** mode, OpenShift Cluster Manager creates the Operator roles and the OIDC provider automatically, using the permissions provided through the OpenShift Cluster Manager IAM role. To use this feature, you must apply admin privileges to the role.

2.2. SUPPORT CONSIDERATIONS FOR ROSA CLUSTERS WITH STS

The supported way of creating a Red Hat OpenShift Service on AWS cluster that uses the AWS Security Token Service (STS) is by using the steps described in this product documentation.

IMPORTANT

You can use **manual** mode with the Red Hat OpenShift Service on AWS CLI (**rosa**) to generate the AWS Identity and Access Management (IAM) policy files and **aws** commands that are required to install the STS resources.

The files and **aws** commands are generated for review purposes only and must not be modified in any way. Red Hat cannot provide support for ROSA clusters that have been deployed by using modified versions of the policy files or **aws** commands.

2.3. CREATING A CLUSTER USING CUSTOMIZATIONS

Deploy a Red Hat OpenShift Service on AWS (ROSA) with AWS Security Token Service (STS) cluster with a configuration that suits the needs of your environment. You can deploy your cluster with customizations by using Red Hat OpenShift Cluster Manager or the ROSA CLI (**rosa**).

2.3.1. Creating a cluster with customizations by using OpenShift Cluster Manager

When you create a Red Hat OpenShift Service on AWS (ROSA) cluster that uses the AWS Security Token Service (STS), you can customize your installation interactively by using Red Hat OpenShift Cluster Manager.

IMPORTANT

Only public and AWS PrivateLink clusters are supported with STS. Regular private clusters (non-PrivateLink) are not available for use with STS.

NOTE

AWS Shared VPCs are not currently supported for ROSA installations.

Prerequisites

- You have completed the AWS prerequisites for ROSA with STS.
- You have available AWS service quotas.
- You have enabled the ROSA service in the AWS Console.
- You have installed and configured the latest ROSA CLI (**rosa**) on your installation host.

NOTE

To successfully install ROSA 4.10 clusters, use the latest version of the ROSA CLI.

- You have logged in to your Red Hat account by using the **rosa** CLI.
- You have verified that the AWS Elastic Load Balancing (ELB) service role exists in your AWS account.

Procedure

1. Navigate to OpenShift Cluster Manager and select **Create cluster**.
2. On the **Create an OpenShift cluster** page, select **Create cluster** in the **Red Hat OpenShift Service on AWS (ROSA)** row.
3. Review and complete the **Prerequisites** listed on the **Accounts and roles** page. Select the checkbox to acknowledge that you have read and completed all of the prerequisites.
4. Select an AWS account from the **Associated AWS account** drop-down menu. If no associated AWS accounts are found, click **Associate AWS account** and follow these steps:

   a. On the **Authenticate** page, click the copy button next to the **rosa login** command. The provided command includes your ROSA API login token.

   NOTE

   You can also load your API token on the OpenShift Cluster Manager API Token page on OpenShift Cluster Manager.

   b. Run the copied command in the CLI to log in to your ROSA account.

   ```
   $ rosa login --token=<api_login_token> 1
   ```

   1 Replace **<api_login_token>** with the token that is provided in the copied command.

   #### Example output

   ```
   I: Logged in as '<username>' on 'https://api.openshift.com'
   ```

   c. On the **Authenticate page** in OpenShift Cluster Manager, click **Next**.

   d. On the **OCM role** page, click the copy button next to the **Basic OCM role** or the **Admin OCM role** commands.
   The basic role enables OpenShift Cluster Manager to detect the AWS IAM roles and policies required by ROSA. The admin role also enables the detection of the roles and policies. In addition, the admin role enables automatic deployment of the cluster-specific Operator roles and the OpenID Connect (OIDC) provider by using OpenShift Cluster Manager.

   e. Run the copied command in the CLI and follow the prompts to create the OpenShift Cluster Manager IAM role. The following example creates a basic OpenShift Cluster Manager IAM role using the default options:

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

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::<aws_account_id>:user/<aws_username>'
? Create the 'ManagedOpenShift-OCM-Role-<red_hat_organization_external_id>' role? Yes
I: Created role 'ManagedOpenShift-OCM-Role-<red_hat_organization_external_id>' with ARN 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-OCM-Role-<red_hat_organization_external_id>'
I: Linking OCM role
? OCM Role ARN: arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-OCM-Role-<red_hat_organization_external_id>
? Link the 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-OCM-Role-<red_hat_organization_external_id>' role with organization '<red_hat_organization_id>'? Yes 5
I: Successfully linked role-arn 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-OCM-Role-<red_hat_organization_external_id>' with organization account '<red_hat_organization_id>'
```

1 Specifies the prefix to include in the OpenShift Cluster Manager IAM role name. The default is **ManagedOpenShift**.

2 Enables the admin OpenShift Cluster Manager IAM role, which is equivalent to specifying the **--admin** argument. The admin role is required if you want to use **Auto** mode to automatically provision the cluster-specific Operator roles and the OIDC provider by using OpenShift Cluster Manager.

3 Optional: Specifies a permissions boundary Amazon Resource Name (ARN) for the role. For more information, see Permissions boundaries for IAM entities in the AWS documentation.

4 Selects the role creation mode. You can use **auto** mode to automatically create the OpenShift Cluster Manager IAM role and link it to your Red Hat organization account. In **manual** mode, the **rosa** CLI generates the **aws** commands needed to create and link the role. In **manual** mode, the corresponding policy JSON files are also saved to the current directory. **manual** mode enables you to review the details before running the **aws** commands manually.

5 Links the OpenShift Cluster Manager IAM role to your Red Hat organization account.

f. If you opted not to link the OpenShift Cluster Manager IAM role to your Red Hat organization account in the preceding command, copy the **rosa link** command from the OpenShift Cluster Manager **OCM role** page and run it:

```
$ rosa link ocm-role <arn> 1
```

① Replace **<arn>** with the ARN of the OpenShift Cluster Manager IAM role that is included in the output of the preceding command.

g. Select **Next** on the OpenShift Cluster Manager **OCM role** page.

h. On the **User role** page, click the copy button for the **User role** command and run the command in the CLI. Follow the prompts to create the user role:

```
$ rosa create user-role
```

Example output

```
I: Creating User role
? Role prefix: ManagedOpenShift ①
? Permissions boundary ARN (optional): ②
? Role creation mode: auto ③
I: Creating ocm user role using 'arn:aws:iam::<aws_account_id>:user/<aws_username>'
? Create the 'ManagedOpenShift-User-<ocm_username>-Role' role? Yes
I: Created role 'ManagedOpenShift-User-<ocm_username>-Role' with ARN
'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-User-<ocm_username>-Role'
I: Linking User role
? User Role ARN: arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-User-
<ocm_username>-Role
? Link the 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-User-
<ocm_username>-Role' role with account '<ocm_user_account_id>'? Yes ④
I: Successfully linked role ARN 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-
User-<ocm_username>-Role' with account '<ocm_user_account_id>'
```

① Specifies the prefix to include in the user role name. The default is **ManagedOpenShift**.

② Optional: Specifies a permissions boundary Amazon Resource Name (ARN) for the role. For more information, see Permissions boundaries for IAM entities in the AWS documentation.

③ Selects the role creation mode. You can use **auto** mode to automatically create the user role and link it to your OpenShift Cluster Manager user account. In **manual** mode, the **rosa** CLI generates the **aws** commands needed to create and link the role. In **manual** mode, the corresponding policy JSON files are also saved to the current directory. **manual** mode enables you to review the details before running the **aws** commands manually.

④ Links the user role to your OpenShift Cluster Manager user account.

i. If you opted not to link the user role to your OpenShift Cluster Manager user account in the preceding command, copy the **rosa link** command from the OpenShift Cluster Manager **User role** page and run it:

```
$ rosa link user-role <arn> ①
```

① Replace **<arn>** with the ARN of the user role that is included in the output of the preceding command.

j. On the OpenShift Cluster Manager **User role** page, select **Ok**.

k. Under the **Accounts and roles** page, verify that your AWS account is listed as an **Associated AWS account**

5. If the required AWS IAM **Account roles** are not automatically detected and listed on the **Accounts and roles** page, create the roles and policies:

   a. Click the copy buffer next to the **rosa create account-roles** command. Run the command in the CLI to create the required AWS account-wide roles and policies, including the Operator policies::

```
$ rosa create account-roles
```

Example output

```
I: Logged in as '<ocm_username>' on 'https://api.openshift.com'
I: Validating AWS credentials...
I: AWS credentials are valid!
I: Validating AWS quota...
I: AWS quota ok. If cluster installation fails, validate actual AWS resource usage against
https://docs.openshift.com/rosa/rosa_getting_started/rosa-required-aws-service-
quotas.html
I: Verifying whether OpenShift command-line tool is available...
I: Current OpenShift Client Version: 4.9.12
I: Creating account roles
? Role prefix: ManagedOpenShift 1
? Permissions boundary ARN (optional): 2
? Role creation mode: auto 3
I: Creating roles using 'arn:aws:iam::<aws_account_number>:user/<aws_username>'
? Create the 'ManagedOpenShift-Installer-Role' role? Yes 4
I: Created role 'ManagedOpenShift-Installer-Role' with ARN 'arn:aws:iam::
<aws_account_number>:role/ManagedOpenShift-Installer-Role'
? Create the 'ManagedOpenShift-ControlPlane-Role' role? Yes 5
I: Created role 'ManagedOpenShift-ControlPlane-Role' with ARN 'arn:aws:iam::
<aws_account_number>:role/ManagedOpenShift-ControlPlane-Role'
? Create the 'ManagedOpenShift-Worker-Role' role? Yes 6
I: Created role 'ManagedOpenShift-Worker-Role' with ARN 'arn:aws:iam::
<aws_account_number>:role/ManagedOpenShift-Worker-Role'
? Create the 'ManagedOpenShift-Support-Role' role? Yes 7
I: Created role 'ManagedOpenShift-Support-Role' with ARN 'arn:aws:iam::
<aws_account_number>:role/ManagedOpenShift-Support-Role'
? Create the operator policies? Yes 8
I: Created policy with ARN 'arn:aws:iam::
<aws_account_number>:policy/ManagedOpenShift-openshift-cloud-credential-operator-
cloud-crede'
I: Created policy with ARN 'arn:aws:iam::
<aws_account_number>:policy/ManagedOpenShift-openshift-image-registry-installer-
cloud-creden'
I: Created policy with ARN 'arn:aws:iam::
<aws_account_number>:policy/ManagedOpenShift-openshift-ingress-operator-cloud-
credentials'
I: Created policy with ARN 'arn:aws:iam::
<aws_account_number>:policy/ManagedOpenShift-openshift-cluster-csi-drivers-ebs-
cloud-credent'
```

```
I: Created policy with ARN 'arn:aws:iam::
<aws_account_number>:policy/ManagedOpenShift-openshift-cloud-network-config-
controller-cloud'
I: Created policy with ARN 'arn:aws:iam::
<aws_account_number>:policy/ManagedOpenShift-openshift-machine-api-aws-cloud-
credentials'
I: To create a cluster with these roles, run the following command:
rosa create cluster --sts
```

1 Specifies the prefix to include in the OpenShift Cluster Manager IAM role name. The default is **ManagedOpenShift**.

2 Optional: Specifies a permissions boundary Amazon Resource Name (ARN) for the role. For more information, see Permissions boundaries for IAM entities in the AWS documentation.

3 Selects the role creation mode. You can use **auto** mode to automatically create the account wide roles and policies. In **manual** mode, the **rosa** CLI generates the **aws** commands needed to create the roles and policies. In **manual** mode, the corresponding policy JSON files are also saved to the current directory. **manual** mode enables you to review the details before running the **aws** commands manually.

4 5 6 7 Creates the account-wide installer, control plane, worker and support roles and corresponding inline IAM policies. For more information, see *Account-wide IAM role and policy reference*.

8 Creates the cluster-specific Operator IAM roles that permit the ROSA cluster Operators to carry out core OpenShift functionality. For more information, see *Account-wide IAM role and policy reference* .

b. On the **Accounts and roles** page, click **Refresh ARNs** and verify that the installer, support, worker, and control plane account roles are detected.

6. Select **Next**.

7. On the **Cluster details** page, provide a name for your cluster and specify the cluster details:

   a. Add a **Cluster name**.

   b. Select a cluster version from the **Version** drop-down menu.

   c. Select a cloud provider region from the **Region** drop-down menu.

   d. Select a **Single zone** or **Multi-zone** configuration.

   e. Leave **Enable user workload monitoring** selected to monitor your own projects in isolation from Red Hat Site Reliability Engineer (SRE) platform metrics. This option is enabled by default.

   f. Optional: Select **Enable additional etcd encryption** if you require etcd key value encryption. With this option, the etcd key values are encrypted, but not the keys. This option is in addition to the control plane storage encryption that encrypts the etcd volumes in Red Hat OpenShift Service on AWS clusters by default.

> **NOTE**
>
> 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. Consider enabling etcd encryption only if you specifically require it for your use case.

g. Optional: Select **Encrypt persistent volumes with customer keys**if you want to provide your own AWS Key Management Service (KMS) key Amazon Resource Name (ARN). The key is used for encryption of persistent volumes in your cluster.

h. Click **Next**.

8. On the **Default machine pool** page, select a **Compute node instance type**

> **NOTE**
>
> After your cluster is created, you can change the number of compute nodes in your cluster, but you cannot change the compute node instance type in the default machine pool. The number and types of nodes available to you depend on whether you use single or multiple availability zones. They also depend on what is enabled and available in your AWS account and the selected region.

9. Optional: Configure autoscaling for the default machine pool:

   a. Select **Enable autoscaling** to automatically scale the number of machines in your default machine pool to meet the deployment needs.

   b. Set the minimum and maximum node count limits for autoscaling. The cluster autoscaler does not reduce or increase the default machine pool node count beyond the limits that you specify.

      - If you deployed your cluster using a single availability zone, set the **Minimum node count** and **Maximum node count**. This defines the minimum and maximum compute node limits in the availability zone.

      - If you deployed your cluster using multiple availability zones, set the **Minimum nodes per zone** and **Maximum nodes per zone**. This defines the minimum and maximum compute node limits per zone.

> **NOTE**
>
> Alternatively, you can set your autoscaling preferences for the default machine pool after the machine pool is created.

10. If you did not enable autoscaling, select a compute node count for your default machine pool:

    - If you deployed your cluster using a single availability zone, select a **Compute node count** from the drop-down menu. This defines the number of compute nodes to provision to the machine pool for the zone.

    - If you deployed your cluster using multiple availability zones, select a **Compute node count (per zone)** from the drop-down menu. This defines the number of compute nodes to provision to the machine pool per zone.

11. Optional: Expand **Edit node labels** to add labels to your nodes. Click **Add label** to add more node labels and select **Next**.

12. In the **Cluster privacy** section of the **Network configuration** page, select **Public** or **Private** to use either public or private API endpoints and application routes for your cluster.

    IMPORTANT

    If you are using private API endpoints, you cannot access your cluster until you update the network settings in your cloud provider account.

13. Optional: If you opted to use public API endpoints, you can select **Install into an existing VPC** to install your cluster into an existing VPC.

    NOTE

    If you opted to use private API endpoints, you must use an existing VPC and PrivateLink and the **Install into an existing VPC**and **Use a PrivateLink** options are automatically selected. With these options, the Red Hat Site Reliability Engineering (SRE) team can connect to the cluster to assist with support by using only AWS PrivateLink endpoints.

14. Optional: If you are installing your cluster into an existing VPC, select **Configure a cluster-wide proxy** to enable an HTTP or HTTPS proxy to deny direct access to the internet from your cluster.

15. Click **Next**.

16. If you opted to install the cluster in an existing AWS VPC, provide your **Virtual Private Cloud (VPC) subnet settings**.

    NOTE

    You must ensure that your VPC is configured with a public and a private subnet for each availability zone that you want the cluster installed into. If you opted to use PrivateLink, only private subnets are required.

17. In the **CIDR ranges** dialog, configure custom classless inter-domain routing (CIDR) ranges or use the defaults that are provided and click **Next**.

    NOTE

    If you are installing into a VPC, the **Machine CIDR** range must match the VPC subnets.

    IMPORTANT

    CIDR configurations cannot be changed later. Confirm your selections with your network administrator before proceeding.

18. Under the **Cluster roles and policies** page, select your preferred cluster-specific Operator IAM role and OIDC provider creation mode.
    With **Manual** mode, you can use either **rosa** CLI commands or **aws** CLI commands to generate

the required Operator roles and OIDC provider for your cluster. **Manual** mode enables you to review the details before using your preferred option to create the IAM resources manually and complete your cluster installation.

Alternatively, you can use **Auto** mode to automatically create the Operator roles and OIDC provider.

> NOTE
>
> To enable **Auto** mode, the OpenShift Cluster Manager IAM role must have administrator capabilities.

19. Optional: Specify a **Custom operator roles prefix**for your cluster-specific Operator IAM roles.

> NOTE
>
> By default, the cluster-specific Operator role names are prefixed with the cluster name and random 4-digit hash. You can optionally specify a custom prefix to replace **<cluster_name>-<hash>** in the role names. The prefix is applied when you create the cluster-specific Operator IAM roles. For information about the prefix, see *About custom Operator IAM role prefixes* .

20. Select **Next**.

21. On the **Cluster update strategy** page, configure your update preferences:

    a. Choose a cluster update method:

    - Select **Individual updates** if you want to schedule each update individually. This is the default option.

    - Select **Recurring updates** to update your cluster on your preferred day and start time, when updates are available.

    > IMPORTANT
    >
    > Even when you opt for recurring updates, you must update the account-wide and cluster-specific IAM resources before you upgrade your cluster between minor releases.

    > NOTE
    >
    > You can review the end-of-life dates in the update life cycle documentation for Red Hat OpenShift Service on AWS. For more information, see *Red Hat OpenShift Service on AWS update life cycle* .

    b. If you opted for recurring updates, select a preferred day of the week and upgrade start time in UTC from the drop-down menus.

    c. Optional: You can set a grace period for **Node draining** during cluster upgrades. A **1 hour** grace period is set by default.

    d. Click **Next**.

> NOTE
>
> In the event of critical security concerns that significantly impact the security or stability of a cluster, Red Hat Site Reliability Engineering (SRE) might schedule automatic updates to the latest z-stream version that is not impacted. The updates are applied within 48 hours after customer notifications are provided. For a description of the critical impact security rating, see Understanding Red Hat security ratings .

22. Review the summary of your selections and click **Create cluster** to start the cluster installation.

23. If you opted to use **Manual** mode, create the cluster-specific Operator roles and OIDC provider manually to continue the installation:

    a. In the **Action required to continue installation** dialog, select either the **AWS CLI** or **ROSA CLI** tab and manually create the resources:

    - If you opted to use the **AWS CLI** method, click **Download .zip**, save the file, and then extract the AWS CLI command and policy files. Then, run the provided **aws** commands in the CLI.

      > NOTE
      >
      > You must run the **aws** commands in the directory that contains the policy files.

    - If you opted to use the **ROSA CLI** method, click the copy button next to the **rosa create** commands and run them in the CLI.

    b. In the **Action required to continue installation** dialog, click **x** to return to the **Overview** page for your cluster.

    c. Verify that the cluster **Status** in the **Details** section of the **Overview** page for your cluster has changed from **Waiting** to **Installing**. There might be a short delay of approximately two minutes before the status changes.

> NOTE
>
> If you opted to use **Auto** mode, OpenShift Cluster Manager creates the Operator roles and the OIDC provider automatically.

Verification

- You can monitor the progress of the installation in the **Overview** page for your cluster. You can view the installation logs on the same page. Your cluster is ready when the **Status** in the **Details** section of the page is listed as **Ready**.

> NOTE
>
> If the installation fails or the cluster **State** does not change to **Ready** after about 40 minutes, check the installation troubleshooting documentation for details. For more information, see *Troubleshooting installations*. For steps to contact Red Hat Support for assistance, see *Getting support for Red Hat OpenShift Service on AWS*.

2.3.2. Creating a cluster with customizations using the CLI

When you create a Red Hat OpenShift Service on AWS (ROSA) cluster that uses the AWS Security Token Service (STS), you can customize your installation interactively.

When you run **rosa create cluster --interactive** at cluster creation time, you are presented with a series of interactive prompts that enable you to customize your deployment. For more information, see *Interactive cluster creation mode reference* .

After a cluster installation using the interactive mode completes, a single command is provided in the output that enables you to deploy further clusters using the same custom configuration.

> **IMPORTANT**
>
> Only public and AWS PrivateLink clusters are supported with STS. Regular private clusters (non-PrivateLink) are not available for use with STS.

> **NOTE**
>
> AWS Shared VPCs are not currently supported for ROSA installations.

Prerequisites

- You have completed the AWS prerequisites for ROSA with STS.
- You have available AWS service quotas.
- You have enabled the ROSA service in the AWS Console.
- You have installed and configured the latest ROSA (**rosa**) and AWS (**aws**) CLIs on your installation host.

  > **NOTE**
  >
  > To successfully install ROSA 4.10 clusters, use latest version of the ROSA CLI.

- If you are using a customer-managed AWS Key Management Service (KMS) key for encryption, you have created a symmetric KMS key and you have the key ID and Amazon Resource Name (ARN). For more information about creating AWS KMS keys, see the AWS documentation.

Procedure

1. Create the required account-wide roles and policies, including the Operator policies:

   a. Generate the IAM policy JSON files in the current working directory and output the **aws** CLI commands for review:

   ```
   $ rosa create account-roles --mode manual 1
   ```

   1 **manual** mode generates the **aws** CLI commands and JSON files needed to create the account-wide roles and policies. After review, you must run the commands manually to create the resources.

b. After review, run the **aws** commands manually to create the roles and policies. Alternatively, you can run the preceding command using **--mode auto** to run the **aws** commands immediately.

2. Optional: If you are using your own AWS KMS key to encrypt the control plane, infrastructure, and worker node root volumes, add the ARN for the account-wide installer role to your KMS key policy.

a. Save the key policy for your KMS key to a file on your local machine. The following example saves the output to **kms-key-policy.json** in the current working directory:

```
$ aws kms get-key-policy --key-id <key_id_or_arn> --policy-name default --output text >
kms-key-policy.json 1
```

1 Replace **<key_id_or_arn>** with the ID or ARN of your KMS key.

b. Add the ARN for the account-wide installer role that you created in the preceding step to the **Statement.Principal.AWS** section in the file. In the following example, the ARN for the default **ManagedOpenShift-Installer-Role** role is added:

```
{
   "Version": "2012-10-17",
   "Id": "key-rosa-policy-1",
   "Statement": [
      {
         "Sid": "Enable IAM User Permissions",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::<aws-account-id>:root"
         },
         "Action": "kms:*",
         "Resource": "*"
      },
      {
         "Sid": "Allow ROSA use of the key",
         "Effect": "Allow",
         "Principal": {
            "AWS": [
               "arn:aws:iam::<aws-account-id>:role/ManagedOpenShift-Support-Role", 1
               "arn:aws:iam::<aws-account-id>:role/ManagedOpenShift-Installer-Role",
               "arn:aws:iam::<aws-account-id>:role/ManagedOpenShift-Worker-Role",
               "arn:aws:iam::<aws-account-id>:role/ManagedOpenShift-ControlPlane-Role"
            ]
         },
         "Action": [
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:DescribeKey"
         ],
         "Resource": "*"
      },
      {
         "Sid": "Allow attachment of persistent resources",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::<aws-account-id>:role/ManagedOpenShift-Support-Role", 2
          "arn:aws:iam::<aws-account-id>:role/ManagedOpenShift-Installer-Role",
          "arn:aws:iam::<aws-account-id>:role/ManagedOpenShift-Worker-Role",
          "arn:aws:iam::<aws-account-id>:role/ManagedOpenShift-ControlPlane-Role"
        ]
      },
      "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": "*",
      "Condition": {
        "Bool": {
          "kms:GrantIsForAWSResource": "true"
        }
      }
    }
  ]
}
```

1 2 You must specify the ARN for the account-wide role that will be used when you create the ROSA cluster. The ARNs listed in the section must be comma-separated.

c. Apply the changes to your KMS key policy:

```
$ aws kms put-key-policy --key-id <key_id_or_arn> \ 1
  --policy file://kms-key-policy.json \ 2
  --policy-name default
```

1 Replace **<key_id_or_arn>** with the ID or ARN of your KMS key.

2 You must include the **file://** prefix when referencing a key policy in a local file.

You can reference the ARN of your KMS key when you create the cluster in the next step.

3. Create a cluster with STS using custom installation options. You can use the **--interactive** mode to interactively specify custom settings:

```
$ rosa create cluster --interactive --sts
```

Example output

```
I: Interactive mode enabled.
Any optional fields can be left empty and a default will be selected.
? Cluster name: <cluster_name>
? OpenShift version: 4.8.9 1
I: Using arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-Installer-Role for the
Installer role 2
I: Using arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-ControlPlane-Role for the
```

```
ControlPlane role
I: Using arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-Worker-Role for the Worker role
I: Using arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-Support-Role for the Support role
? External ID (optional):
? Operator roles prefix: <cluster_name>-<random_string> 3
? Multiple availability zones (optional): No 4
? AWS region: us-east-1
? PrivateLink cluster (optional): No
? Install into an existing VPC (optional): No
? Enable Customer Managed key (optional): No 5
? Compute nodes instance type (optional):
? Enable autoscaling (optional): No
? Compute nodes: 2
? Machine CIDR: 10.0.0.0/16
? Service CIDR: 172.30.0.0/16
? Pod CIDR: 10.128.0.0/14
? Host prefix: 23
? Encrypt etcd data (optional): No 6
? Disable Workload monitoring (optional): No
I: Creating cluster '<cluster_name>'
I: To create this cluster again in the future, you can run:
   rosa create cluster --cluster-name <cluster_name> --role-arn arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-Installer-Role --support-role-arn arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-Support-Role --master-iam-role arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-ControlPlane-Role --worker-iam-role arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-Worker-Role --operator-roles-prefix <cluster_name>-<random_string> --region us-east-1 --version 4.8.9 --compute-nodes 2 --machine-cidr 10.0.0.0/16 --service-cidr 172.30.0.0/16 --pod-cidr 10.128.0.0/14 --host-prefix 23 7
I: To view a list of clusters and their status, run 'rosa list clusters'
I: Cluster '<cluster_name>' has been created.
I: Once the cluster is installed you will need to add an Identity Provider before you can login into the cluster. See 'rosa create idp --help' for more information.
I: To determine when your cluster is Ready, run 'rosa describe cluster -c <cluster_name>'.
I: To watch your cluster installation logs, run 'rosa logs install -c <cluster_name> --watch'.
```

1. When creating the cluster, the listed **OpenShift version** options include the major, minor, and patch versions, for example **4.8.9**.

2. If more than one matching set of account-wide roles are available in your account for a cluster version, an interactive list of options is provided.

3. Optional: By default, the cluster-specific Operator role names are prefixed with the cluster name and random 4-digit hash. You can optionally specify a custom prefix to replace **<cluster_name>-<hash>** in the role names. The prefix is applied when you create the cluster-specific Operator IAM roles. For information about the prefix, see *Defining an Operator IAM role prefix*.

4. Multiple availability zones are recommended for production workloads. The default is a single availability zone.

5. Enable this option if you are using your own AWS KMS key to encrypt the control plane, infrastructure, and worker node root volumes. Specify the ARN for the KMS key that you added to the account-wide role ARN to in the preceding step.

6 Enable this option only if your use case requires etcd key value encryption in addition to the control plane storage encryption that encrypts the etcd volumes by default. With this

> 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.

7 The output includes a custom command that you can run to create a cluster with the same configuration in the future.

As an alternative to using the **--interactive** mode, you can specify the customization options directly when you run **rosa create cluster**. Run **rosa create cluster --help** to view a list of available CLI options.

> IMPORTANT
>
> You must complete the following steps to create the Operator IAM roles and the OpenID Connect (OIDC) provider to move the state of the cluster to **ready**.

4. Create the cluster-specific Operator IAM roles:

   a. Generate the Operator IAM policy JSON files in the current working directory and output the **aws** CLI commands for review:

   ```
   $ rosa create operator-roles --mode manual --cluster <cluster_name|cluster_id> 1
   ```

   1 **manual** mode generates the **aws** CLI commands and JSON files needed to create the Operator roles. After review, you must run the commands manually to create the resources.

   b. After review, run the **aws** commands manually to create the Operator IAM roles and attach the managed Operator policies to them. Alternatively, you can run the preceding command again using **--mode auto** to run the **aws** commands immediately.

   > NOTE
   >
   > A custom prefix is applied to the Operator role names if you specified the prefix in the preceding step.

5. Create the OpenID Connect (OIDC) provider that the cluster Operators use to authenticate:

   ```
   $ rosa create oidc-provider --mode auto --cluster <cluster_name|cluster_id> 1
   ```

   1 **auto** mode immediately runs the **aws** CLI command that creates the OIDC provider.

6. Check the status of your cluster:

```
$ rosa describe cluster --cluster <cluster_name|cluster_id>
```

Example output

```
Name:                      <cluster_name>
ID:                        <cluster_id>
External ID:               <external_id>
OpenShift Version:         <version>
Channel Group:             stable
DNS:                       <cluster_name>.xxxx.p1.openshiftapps.com
AWS Account:               <aws_account_id>
API URL:                   https://api.<cluster_name>.xxxx.p1.openshiftapps.com:6443
Console URL:               https://console-openshift-console.apps.
<cluster_name>.xxxx.p1.openshiftapps.com
Region:                    <aws_region>
Multi-AZ:                  false
Nodes:
 - Master:                 3
 - Infra:                  2
 - Compute:                2
Network:
 - Service CIDR:           172.30.0.0/16
 - Machine CIDR:           10.0.0.0/16
 - Pod CIDR:               10.128.0.0/14
 - Host Prefix:            /23
STS Role ARN:              arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-Installer-
Role
Support Role ARN:          arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-Support-
Role
Instance IAM Roles:
 - Master:                 arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-ControlPlane-
Role
 - Worker:                 arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-Worker-Role
Operator IAM Roles:
 - arn:aws:iam::<aws_account_id>:role/<cluster_name>-xxxx-openshift-ingress-operator-
cloud-credentials
 - arn:aws:iam::<aws_account_id>:role/<cluster_name-xxxx-openshift-cluster-csi-drivers-ebs-
cloud-credent
 - arn:aws:iam::<aws_account_id>:role/<cluster_name-xxxx-openshift-machine-api-aws-
cloud-credentials
 - arn:aws:iam::<aws_account_id>:role/<cluster_name-xxxx-openshift-cloud-credential-
operator-cloud-crede
 - arn:aws:iam::<aws_account_id>:role/<cluster_name-xxxx-openshift-image-registry-
installer-cloud-creden
State:                     ready
Private:                   No
Created:                   Oct  1 2021 08:12:25 UTC
Details Page:              https://console.redhat.com/openshift/details/s/<subscription_id>
OIDC Endpoint URL:         https://rh-oidc.s3.<aws_region>.amazonaws.com/<cluster_id>
```

The following **State** field changes are listed in the output as the cluster installation progresses:

- **waiting (Waiting for OIDC configuration)**

- **pending (Preparing account)**

- **installing (DNS setup in progress)**
- **installing**
- **ready**

NOTE

If the installation fails or the **State** field does not change to **ready** after about 40 minutes, check the installation troubleshooting documentation for details. For more information, see *Troubleshooting installations*. For steps to contact Red Hat Support for assistance, see *Getting support for Red Hat OpenShift Service on AWS*.

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

```
$ rosa logs install --cluster <cluster_name|cluster_id> --watch 1
```

1 Specify the **--watch** flag to watch for new log messages as the installation progresses. This argument is optional.

2.4. NEXT STEPS

- Accessing a ROSA cluster

2.5. ADDITIONAL RESOURCES

- For more information about the AWS Identity Access Management (IAM) resources required to deploy Red Hat OpenShift Service on AWS with STS, see About IAM resources for clusters that use STS.
- For details about optionally setting an Operator role name prefix, see About custom Operator IAM role prefixes.
- For an overview of the options that are presented when you create a cluster using interactive mode, see Interactive cluster creation mode reference .
- For information about the prerequisites to installing ROSA with STS, see AWS prerequisites for ROSA with STS.
- For more information about using OpenID Connect (OIDC) identity providers in AWS IAM, see Creating OpenID Connect (OIDC) identity providers in the AWS documentation.
- For more information about etcd encryption, see the etcd encryption service definition .
- For more information about troubleshooting ROSA cluster installations, see Troubleshooting cluster deployments.
- For steps to contact Red Hat Support for assistance, see Getting support for Red Hat OpenShift Service on AWS.

CHAPTER 3. INTERACTIVE CLUSTER CREATION MODE REFERENCE

This section provides an overview of the options that are presented when you use the interactive mode to create a cluster by using the Red Hat OpenShift Service on AWS (ROSA) CLI (**rosa**).

3.1. UNDERSTANDING THE INTERACTIVE CLUSTER CREATION MODE OPTIONS

You can create a Red Hat OpenShift Service on AWS cluster with the AWS Security Token Service (STS) by using the interactive mode. You can enable the mode by specifying the **--interactive** option when you run **rosa create cluster**. The following table describes the interactive mode options.

Table 3.1. **--interactive** mode options

Field	Description
Cluster name	Enter a name for your cluster, for example **my-rosa-cluster**.
Deploy cluster using AWS STS	Create an OpenShift cluster that uses the AWS Security Token Service (STS) to allocate temporary, limited-privilege credentials for component-specific AWS Identity and Access Management (IAM) roles. The service enables cluster components to make AWS API calls using secure cloud resource management practices. The default is **Yes**.
OpenShift version	Select the version of OpenShift to install, for example **4.3.12**. The default is the latest version.
External ID (optional)	Specify an unique identifier that is passed by OpenShift Cluster Manager and the OpenShift installer when an account role is assumed. This option is only required for custom account roles that expect an external ID.
Operator roles prefix	Enter a prefix to assign to the cluster-specific Operator IAM roles. The default is the name of the cluster and a 4-digit random string, for example **my-rosa-cluster-a0b1**.
Multiple availability zones	Deploy the cluster to multiple availability zones in the AWS region. The default is **No**, which results in a cluster being deployed to a single availability zone. If you deploy a cluster into multiple availability zones, the AWS region must have at least 3 availability zones. Multiple availability zones are recommended for production workloads.
AWS region	Specify the AWS region to deploy the cluster in. This overrides the **AWS_REGION** environment variable.

Field	Description
PrivateLink cluster	Create a cluster using AWS PrivateLink. This option provides private connectivity between Virtual Private Clouds (VPCs), AWS services, and your on-premise networks, without exposing your traffic to the public internet. To provide support, Red Hat Site Reliability Engineering (SRE) can connect to the cluster by using AWS PrivateLink Virtual Private Cloud (VPC) endpoints. This option cannot be changed after a cluster is created. The default is **No**.
Install into an existing VPC	Install a cluster into an existing AWS VPC. To use this option, your VPC must have 2 subnets for each availability zone that you are installing the cluster into. The default is **No**.
Enable customer managed key	Enable this option to use a specific AWS Key Management Service (KMS) key as the encryption key for persistent data. This key is used as the encryption key for control plane, infrastructure, and worker node root volumes. When disabled, the account KMS key for the specified region is used by default to ensure persistent data is always encrypted. The default is **No**.
Compute nodes instance type	Select a compute node instance type. The default is **m5.xlarge**.
Enable autoscaling	Enable compute node autoscaling. The autoscaler adjusts the size of the cluster to meet your deployment demands. The default is **No**.
Compute nodes	Specify the number of compute nodes to provision into each availability zone. Clusters deployed in a single availability zone require at least 2 nodes. Clusters deployed in multiple zones must have at least 3 nodes. The maximum number of worker nodes is 180 nodes. The default value is **2**.
Machine CIDR	Specify the IP address range for machines (cluster nodes), which must encompass all CIDR address ranges for your VPC subnets. Subnets must be contiguous. A minimum IP address range of 128 addresses, using the subnet prefix **/25**, is supported for single availability zone deployments. A minimum address range of 256 addresses, using the subnet prefix **/24**, is supported for deployments that use multiple availability zones. The default is **10.0.0.0/16**. This range must not conflict with any connected networks.
Service CIDR	Specify the IP address range for services. The range must be large enough to accommodate your workload. The address block must not overlap with any external service accessed from within the cluster. The default is **172.30.0.0/16**. It is recommended that they are the same between clusters.

Field	Description
Pod CIDR	Specify the IP address range for pods. The range must be large enough to accommodate your workload. The address block must not overlap with any external service accessed from within the cluster. The default is **10.128.0.0/14**. It is recommended that they are the same between clusters.
Host prefix	Specify the subnet prefix length assigned to pods scheduled to individual machines. The host prefix determines the pod IP address pool for each machine. For example, if the host prefix is set to /**23**, each machine is assigned a/**23** subnet from the pod CIDR address range. The default is /**23**, allowing 512 cluster nodes and 512 pods per node, both of which are beyond our supported maximums. For information on the supported maximums, see the Additional resources section below.
Encrypt etcd data (optional)	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. You can additionally enable the **Encrypt etcd data** option to encrypt the key values for some resources in etcd, but not the keys. 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.
Disable workload monitoring	Disable monitoring for user-defined projects. Monitoring for user-defined projects is enabled by default.

3.2. ADDITIONAL RESOURCES

- For a list of the supported maximums, see ROSA tested cluster maximums.
- For detailed 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 detailed steps to create a ROSA cluster with STS using customizations, including the AWS IAM resources, see Creating a ROSA cluster with STS using customizations .
- For more information about etcd encryption, see the etcd encryption service definition .
- For an example VPC architecture, see this sample VPC architecture.

CHAPTER 4. CREATING AN AWS PRIVATELINK CLUSTER ON ROSA

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

4.1. UNDERSTANDING AWS PRIVATELINK

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

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

4.2. REQUIREMENTS FOR USING AWS PRIVATELINK CLUSTERS

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

Table 4.1. Required AWS resources

<table>
<tr><th>Component</th><th>AWS Type</th><th>Description</th></tr>
<tr><td>VPC</td><td>

- AWS::EC2::VPC
- AWS::EC2::VPCEndpoint

</td><td>You must provide a VPC for the cluster to use.</td></tr>
<tr><td>Network access control</td><td>

- AWS::EC2::NetworkAcl
- AWS::EC2::NetworkAclEntry

</td><td>You must allow access to the following ports:
<table>
<tr><th>Port</th><th>Reason</th></tr>
<tr><td>80</td><td>Inbound HTTP traffic</td></tr>
<tr><td>443</td><td>Inbound HTTPS traffic</td></tr>
<tr><td>22</td><td>Inbound SSH traffic</td></tr>
<tr><td>1024-65535</td><td>Inbound ephemeral traffic</td></tr>
<tr><td>0-65535</td><td>Outbound ephemeral traffic</td></tr>
</table>
</td></tr>
</table>

Component	AWS Type	Description
Private subnets	- AWS::EC2::Subnet - AWS::EC2::RouteTable - AWS::EC2::SubnetRouteTableAssociation	Your VPC must have private subnets in 1 availability zone for Single-AZ deployments or 3 availability zones for Multi-AZ deployments. You must provide appropriate routes and route tables.

4.3. CREATING AN AWS PRIVATELINK CLUSTER

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

NOTE

AWS PrivateLink is supported on existing VPCs only.

Prerequisites

You have installed Red Hat OpenShift Service on AWS.

Procedure

Creating a cluster can take up to 40 minutes.

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

   IMPORTANT

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

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

   - To create a Single-AZ cluster:

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

   - To create a Multi-AZ cluster:

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

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

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

NOTE

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

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

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

4.4. CONFIGURING AWS PRIVATELINK DNS FORWARDING

With AWS PrivateLink clusters, a public hosted zone and a private hosted zone are created in Route 53. With the private hosted zone, records within the zone are resolvable only from within the VPC to which it is assigned.

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

For more information about private hosted zones, see AWS private hosted zones documentation . For more information about private hosted zones, see AWS public hosted zones documentation .

Prerequisites

- Your corporate network or other VPC has connectivity
- UDP port 53 and TCP port 53 ARE enabled across your networks to allow for DNS queries
- You have created an AWS PrivateLink cluster using Red Hat OpenShift Service on AWS

Procedure

1. To allow for records such as **api.<cluster_domain>** and ***.apps.<cluster_domain>** to resolve outside of the VPC, configure a Route 53 Resolver Inbound Endpoint .
2. When you configure the inbound endpoint, select the VPC and private subnets that were used when you created the cluster.
3. After the endpoints are operational and associated, configure your corporate network to forward DNS queries to those IP addresses for the top-level cluster domain, such as **drow-pl-01.htno.p1.openshiftapps.com**.
4. If you are forwarding DNS queries from one VPC to another VPC, configure forwarding rules.
5. If you are configuring your remote network DNS server, see your specific DNS server documentation to configure selective DNS forwarding for the installed cluster domain.

4.5. NEXT STEPS

Configure identity providers

4.6. ADDITIONAL RESOURCES

- AWS PrivateLink firewall prerequisites
- Overview of the ROSA with STS deployment workflow
- Deleting a ROSA cluster
- ROSA architecture

CHAPTER 5. ACCESSING A ROSA CLUSTER

It is recommended that you access your Red Hat OpenShift Service on AWS (ROSA) cluster using an identity provider (IDP) account. However, the cluster administrator who created the cluster can access it using the quick access procedure.

This document describes how to access a cluster and set up an IDP using the **rosa** CLI. Alternatively, you can create an IDP account using OpenShift Cluster Manager console.

5.1. ACCESSING YOUR CLUSTER QUICKLY

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

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

Procedure

1. Enter the following command:

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

Example output

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

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

Example output

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

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

```
$ oc whoami
```

Example output

```
cluster-admin
```

5.2. ACCESSING YOUR CLUSTER WITH AN IDP ACCOUNT

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

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

Procedure

To access your cluster using an IDP account:

1. Add an IDP.

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

    ```
    $ rosa create idp --cluster=<cluster_name> --interactive
    ```

    b. Enter the following values:

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

    #### Example output

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

    c. Follow the URL in the output and select **Register application** to register a new OAuth application in your GitHub organization. By registering the application, you enable the OAuth server that is built into ROSA to authenticate members of your GitHub organization into your cluster.

NOTE

The fields in the **Register a new OAuth application** GitHub form are automatically filled with the required values through the URL that is defined by the **rosa** CLI tool.

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

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

Continued example output

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

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

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

```
$ rosa list idps --cluster=<cluster_name>
```

Example output

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

2. Log in to your cluster.

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

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

Example output

```
Name:        rh-rosa-test-cluster1
ID:        1de87g7c30g75qechgh7l5b2bha6r04e
```

```
External ID: 34322be7-b2a7-45c2-af39-2c684ce624e1
API URL:     https://api.rh-rosa-test-cluster1.j9n4.s1.devshift.org:6443
Console URL: https://console-openshift-console.apps.rh-rosa-test-
cluster1.j9n4.s1.devshift.org
Nodes:       Master: 3, Infra: 3, Compute: 4
Region:      us-east-2
State:       ready
Created:     May 27, 2020
```

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

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

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

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

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

Example output

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

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

Using project "default".
```

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

```
$ oc version
```

Example output

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

5.3. GRANTING CLUSTER-ADMIN ACCESS

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

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

Prerequisites

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

Procedure

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

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

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

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

Example output

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

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

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

Example output

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

5.4. GRANTING DEDICATED-ADMIN ACCESS

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

Prerequisites

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

Procedure

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

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

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

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

   Example output

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

   NOTE

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

5.5. ADDITIONAL RESOURCES

- Configuring identity providers using Red Hat OpenShift Cluster Manager console
- Understanding the ROSA with STS deployment workflow

CHAPTER 6. CONFIGURING IDENTITY PROVIDERS FOR STS

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

The following topics describe how to configure an identity provider using OpenShift Cluster Manager console. Alternatively, you can use the **rosa** CLI to configure an identity provider and access the cluster.

6.1. UNDERSTANDING IDENTITY PROVIDERS

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

6.1.1. Supported identity providers

You can configure the following types of identity providers:

Identity provider	Description
GitHub or GitHub Enterprise	Configure a GitHub identity provider to validate usernames and passwords against GitHub or GitHub Enterprise’s OAuth authentication server.
GitLab	Configure a GitLab identity provider to use GitLab.com or any other GitLab instance as an identity provider.
Google	Configure a Google identity provider using Google’s OpenID Connect integration.
LDAP	Configure an LDAP identity provider to validate usernames and passwords against an LDAPv3 server, using simple bind authentication.
OpenID Connect	Configure an OpenID Connect (OIDC) identity provider to integrate with an OIDC identity provider using an Authorization Code Flow.
HTPasswd	Configure an HTPasswd identity provider for a single, static administration user. You can log in to the cluster as the user to troubleshoot issues. **IMPORTANT** The HTPasswd identity provider option is included only to enable the creation of a single, static administration user. HTPasswd is not supported as a general-use identity provider for Red Hat OpenShift Service on AWS. For the steps to configure the single user, see *Configuring an HTPasswd identity provider*.

6.1.2. Identity provider parameters

The following parameters are common to all identity providers:

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

NOTE

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

6.2. CONFIGURING A GITHUB IDENTITY PROVIDER

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

WARNING

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

Prerequisites

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

Procedure

1. From OpenShift Cluster Manager, navigate to the **Clusters** page and select the cluster that you need to configure identity providers for.
2. Click the **Access control** tab.
3. Click **Add identity provider**.

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

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

     ```
     https://oauth-openshift.apps.<cluster_name>.
     <cluster_domain>/oauth2callback/<idp_provider_name>
     ```

     For example:

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

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

Verification

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

6.3. CONFIGURING A GITLAB IDENTITY PROVIDER

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

Prerequisites

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

Procedure

1. From OpenShift Cluster Manager, navigate to the **Clusters** page and select the cluster that you need to configure identity providers for.

2. Click the **Access control** tab.

3. Click **Add identity provider**.

   NOTE

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

4. Select **GitLab** from the drop-down menu.

5. Enter a unique name for the identity provider. This name cannot be changed later.

   - An **OAuth callback URL** is automatically generated in the provided field. You will provide this URL to GitLab.

     ```
     https://oauth-openshift.apps.<cluster_name>.
     <cluster_domain>/oauth2callback/<idp_provider_name>
     ```

     For example:

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

6. Add a new application in GitLab .

7. Return to Red Hat OpenShift Service on AWS and select a mapping method from the drop-down menu. **Claim** is recommended in most cases.

8. Enter the **Client ID** and **Client secret** provided by GitLab.

9. Enter the **URL** of your GitLab provider.

10. Optional: You can use a certificate authority (CA) file to validate server certificates for the configured GitLab URL. Click **Browse** to locate and attach a **CA file** to the identity provider.

11. Click **Confirm**.

Verification

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

6.4. CONFIGURING A GOOGLE IDENTITY PROVIDER

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

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

Procedure

1. From OpenShift Cluster Manager, navigate to the **Clusters** page and select the cluster that you need to configure identity providers for.

2. Click the **Access control** tab.

3. Click **Add identity provider**.

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

4. Select **Google** from the drop-down menu.

5. Enter a unique name for the identity provider. This name cannot be changed later.

   - An **OAuth callback URL** is automatically generated in the provided field. You will provide this URL to Google.

     ```
     https://oauth-openshift.apps.<cluster_name>.
     <cluster_domain>/oauth2callback/<idp_provider_name>
     ```

     For example:

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

6. Configure a Google identity provider using Google’s OpenID Connect integration.

7. Return to Red Hat OpenShift Service on AWS and select a mapping method from the drop-down menu. **Claim** is recommended in most cases.

8. Enter the **Client ID** of a registered Google project and the **Client secret** issued by Google.

9. Enter a hosted domain to restrict users to a Google Apps domain.

10. Click **Confirm**.

Verification

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

6.5. CONFIGURING A LDAP IDENTITY PROVIDER

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

Prerequisites

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

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

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

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

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

IMPORTANT

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

Procedure

1. From OpenShift Cluster Manager, navigate to the **Clusters** page and select the cluster that you need to configure identity providers for.
2. Click the **Access control** tab.
3. Click **Add identity provider**.

   NOTE

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

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

    IMPORTANT

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

12. Click **Confirm**.

Verification

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

6.6. CONFIGURING AN OPENID IDENTITY PROVIDER

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

IMPORTANT

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

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

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

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

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

See the OpenID claims documentation for more information.

Prerequisites

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

Procedure

1. From OpenShift Cluster Manager, navigate to the **Clusters** page and select the cluster that you need to configure identity providers for.
2. Click the **Access control** tab.
3. Click **Add identity provider**.

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

4. Select **OpenID** from the drop-down menu.

5. Enter a unique name for the identity provider. This name cannot be changed later.

   - An **OAuth callback URL** is automatically generated in the provided field.

     ```
     https://oauth-openshift.apps.<cluster_name>.
     <cluster_domain>/oauth2callback/<idp_provider_name>
     ```

     For example:

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

6. Create an authorization request using an Authorization Code Flow .

7. Return to Red Hat OpenShift Service on AWS and select a mapping method from the drop-down menu. **Claim** is recommended in most cases.

8. Enter a **Client ID** and **Client secret** provided from OpenID.

9. Enter an **Issuer URL**. This is the URL that the OpenID provider asserts as the Issuer Identifier. It must use the https scheme with no URL query parameters or fragments.

10. Enter an **Email** attribute whose value should be used as the email address. Click **Add more** to add multiple email attributes.

11. Enter a **Name** attribute whose value should be used as the preferred username. Click **Add more** to add multiple preferred usernames.

12. Enter a **Preferred username** attribute whose value should be used as the display name. Click **Add more** to add multiple display names.

13. Optional: Click **Show advanced Options** to add a certificate authority (CA) file to your OpenID identity provider.

14. Optional: Under the advanced options, you can add **Additional scopes**. By default, the **OpenID** scope is requested.

15. Click **Confirm**.

Verification

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

6.7. CONFIGURING AN HTPASSWD IDENTITY PROVIDER

Configure an HTPasswd identity provider to create a single, static user with cluster administration privileges. You can log in to your cluster as the user to troubleshoot issues.

IMPORTANT

The HTPasswd identity provider option is included only to enable the creation of a single, static administration user. HTPasswd is not supported as a general-use identity provider for Red Hat OpenShift Service on AWS.

Procedure

1. From OpenShift Cluster Manager, navigate to the **Clusters** page and select your cluster.
2. Select **Access control → Identity providers**.
3. Click **Add identity provider**.
4. Select **HTPasswd** from the **Identity Provider** drop-down menu.
5. Add a unique name in the **Name** field for the identity provider.
6. Use the suggested username and password for the static user, or create your own.

    NOTE

    The credentials defined in this step are not visible after you select **Add** in the following step. If you lose the credentials, you must recreate the identity provider and define the credentials again.

7. Select **Add** to create the HTPasswd identity provider and the single, static user.
8. Grant the static user permission to manage the cluster:
    a. Under **Access control → Cluster Roles and Access**, select **Add user**.
    b. Enter the **User ID** of the static user that you created in the preceding step.
    c. Select a **Group**. Users in the **dedicated-admins** group have standard administrative privileges for Red Hat OpenShift Service on AWS. Users in the **cluster-admins** group have full administrative access to the cluster.
    d. Select **Add user** to grant the administration privileges to the user.

Verification

- The configured HTPasswd identity provider is visible on the **Access control → Identity providers** page.

    NOTE

    After creating the identity provider, synchronization usually completes within two minutes. You can log in to the cluster as the user after the HTPasswd identity provider becomes available.

- The single, administrative user is visible on the **Access control → Cluster Roles and Access** page. The administration group membership of the user is also displayed.

6.8. ADDITIONAL RESOURCES

- Accessing a cluster
- Understanding the ROSA with STS deployment workflow

CHAPTER 7. REVOKING ACCESS TO A ROSA CLUSTER

An identity provider (IDP) controls access to a Red Hat OpenShift Service on AWS (ROSA) cluster. To revoke access of a user to a cluster, you must configure that within the IDP that was set up for authentication.

7.1. REVOKING ADMINISTRATOR ACCESS USING THEROSA CLI

You can revoke the administrator access of users so that they can access the cluster without administrator privileges. To remove the administrator access for a user, you must revoke the **dedicated-admin** or **cluster-admin** privileges. You can revoke the administrator privileges using the **rosa** command-line utility or using OpenShift Cluster Manager console.

7.1.1. Revoking dedicated-admin access using the rosa CLI

You can revoke access for a **dedicated-admin** user if you are the user who created the cluster, the organization administrator user, or the super administrator user.

Prerequisites

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

Procedure

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

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

2. Enter the following command to verify that your user no longer has **dedicated-admin** access. The output does not list the revoked user.

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

7.1.2. Revoking cluster-admin access using the rosa CLI

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

Prerequisites

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

Procedure

1. Enter the following command to revoke the **cluster-admin** access of a user:

```
$ rosa revoke user cluster-admins --user=myusername --cluster=mycluster
```

2. Enter the following command to verify that the user no longer has **cluster-admin** access. The output does not list the revoked user.

```
$ oc get groups cluster-admins
```

7.2. REVOKING ADMINISTRATOR ACCESS USING OPENSHIFT CLUSTER MANAGER CONSOLE

You can revoke the **dedicated-admin** or **cluster-admin** access of users through OpenShift Cluster Manager console. Users will be able to access the cluster without administrator privileges.

Prerequisites

- You have added an Identity Provider (IDP) to your cluster.
- You have the IDP user name for the user whose privileges you are revoking.
- You are logged in to OpenShift Cluster Manager console using an OpenShift Cluster Manager account that you used to create the cluster, the organization administrator user, or the super administrator user.

Procedure

1. On the **Clusters** tab of OpenShift Cluster Manager, select the name of your cluster to view the cluster details.
2. Select **Access control** > **Cluster Roles and Access**.
3. For the user that you want to remove, click the **Options** menu ⋮ to the right of the user and group combination and click **Delete**.

CHAPTER 8. DELETING A ROSA CLUSTER

This document provides steps to delete a Red Hat OpenShift Service on AWS (ROSA) cluster that uses the AWS Security Token Service (STS). After deleting your cluster, you can also delete the AWS Identity and Access Management (IAM) resources that are used by the cluster.

8.1. PREREQUISITES

- If Red Hat OpenShift Service on AWS created a VPC, you must remove the following items from your cluster before you can successfully delete your cluster:
  - Network configurations, such as VPN configurations and VPC peering connections
  - Any additional services that were added to the VPC

If these configurations and services remain, the cluster does not delete properly.

8.2. DELETING A ROSA CLUSTER AND THE CLUSTER-SPECIFIC IAM RESOURCES

You can delete a Red Hat OpenShift Service on AWS (ROSA) with AWS Security Token Service (STS) cluster by using the ROSA CLI (**rosa**) or Red Hat OpenShift Cluster Manager.

After deleting the cluster, you can clean up the cluster-specific Identity and Access Management (IAM) resources in your AWS account by using the ROSA CLI (**rosa**). The cluster-specific resources include the Operator roles and the OpenID Connect (OIDC) provider.

IMPORTANT

The cluster deletion must complete before you remove the IAM resources, because the resources are used in the cluster deletion and clean-up processes.

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

Prerequisites

- You have installed a ROSA cluster.
- You have installed and configured the latest ROSA CLI (**rosa**) on your installation host.

Procedure

1. Obtain the cluster ID, the Amazon Resource Names (ARNs) for the cluster-specific Operator roles and the endpoint URL for the OIDC provider:

```
$ rosa describe cluster --cluster=<cluster_name> 1
```

1 Replace **<cluster_name>** with the name of your cluster.

Example output

```
Name:                       mycluster
ID:                     1s3v4x39lhs8sm49m90mi0822o34544a ①
...
Operator IAM Roles: ②
 - arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-machine-api-aws-cloud-
credentials
 - arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-cloud-credential-operator-
cloud-crede
 - arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-image-registry-installer-
cloud-creden
 - arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-ingress-operator-cloud-
credentials
 - arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-cluster-csi-drivers-ebs-
cloud-credent
 - arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-cloud-network-config-
controller-cloud
State:                      ready
Private:                     No
Created:                     May 13 2022 11:26:15 UTC
Details Page:
https://console.redhat.com/openshift/details/s/296kyEFwzoy1CREQicFRdZybrc0
OIDC Endpoint URL:          https://rh-oidc.s3.us-east-
1.amazonaws.com/1s5v4k39lhm8sm59m90mi0822o31844a ③
```

① Lists the cluster ID.

② Specifies the ARNs for the cluster-specific Operator roles. For example, in the sample output the ARN for the role required by the Machine Config Operator is **arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-machine-api-aws-cloud-credentials**.

③ Specifies the endpoint URL for the cluster-specific OIDC provider.

IMPORTANT

You require the cluster ID to delete the cluster-specific STS resources using the ROSA CLI (**rosa**) after the cluster is deleted.

2. Delete the cluster:
   - To delete the cluster by using Red Hat OpenShift Cluster Manager:
     a. Navigate to OpenShift Cluster Manager.
     b. Click the Options menu ⋮ next to your cluster and select **Delete cluster**.
     c. Type the name of your cluster at the prompt and click **Delete**.
   - To delete the cluster using the ROSA CLI (**rosa**):
     a. Enter the following command to delete the cluster and watch the logs, replacing **<cluster_name>** with the name or ID of your cluster:

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

IMPORTANT

You must wait for the cluster deletion to complete before you remove the Operator roles and the OIDC provider. The cluster-specific Operator roles are required to clean-up the resources created by the OpenShift Operators. The Operators use the OIDC provider to authenticate.

3. Delete the OIDC provider that the cluster Operators use to authenticate:

```
$ rosa delete oidc-provider -c <cluster_id> --mode auto 1
```

1 Replace **<cluster_id>** with the ID of the cluster.

NOTE

You can use the **-y** option to automatically answer yes to the prompts.

4. Delete the cluster-specific Operator IAM roles:

```
$ rosa delete operator-roles -c <cluster_id> --mode auto 1
```

1 Replace **<cluster_id>** with the ID of the cluster.

Additional resources

- For steps to delete the account-wide IAM roles and policies, see Deleting the account-wide IAM roles and policies.
- For steps to delete the OpenShift Cluster Manager and user IAM roles, see Unlinking and deleting the OpenShift Cluster Manager and user IAM roles.

8.3. DELETING THE ACCOUNT-WIDE IAM RESOURCES

After you have deleted all Red Hat OpenShift Service on AWS (ROSA) with AWS Security Token Services (STS) clusters that depend on the account-wide AWS Identity and Access Management (IAM) resources, you can delete the account-wide resources.

If you no longer need to install a ROSA with STS cluster by using Red Hat OpenShift Cluster Manager, you can also delete the OpenShift Cluster Manager and user IAM roles.

IMPORTANT

The account-wide IAM roles and policies might be used by other ROSA clusters in the same AWS account. You must only remove the resources if they are not required by other clusters.

The OpenShift Cluster Manager and user IAM roles are required if you want to install and manage other ROSA clusters in the same AWS account by using OpenShift Cluster Manager. You must only remove the roles if you no longer need to install ROSA clusters in your account by using OpenShift Cluster Manager.

8.3.1. Deleting the account-wide IAM roles and policies

This section provides steps to delete the account-wide IAM roles and inline policies that you created for ROSA with STS deployments, along with the account-wide Operator policies. You can delete the account-wide AWS Identity and Access Management (IAM) roles and policies only after deleting all of the Red Hat OpenShift Service on AWS (ROSA) with AWS Security Token Services (STS) clusters that depend on them.

IMPORTANT

The account-wide IAM roles and policies might be used by other ROSA clusters in the same AWS account. You must only remove the roles if they are not required by other clusters.

Prerequisites

- You have installed a ROSA cluster.
- You have installed and configured the latest ROSA CLI (**rosa**) on your installation host.

Procedure

1. Delete the account-wide roles:

   a. List the account-wide roles in your AWS account by using the ROSA CLI (**rosa**):

   ```
   $ rosa list account-roles
   ```

   Example output

   ```
   I: Fetching account roles
   ROLE NAME                          ROLE TYPE      ROLE ARN
   OPENSHIFT VERSION
   ManagedOpenShift-ControlPlane-Role  Control plane  arn:aws:iam::
   <aws_account_id>:role/ManagedOpenShift-ControlPlane-Role  4.10
   ManagedOpenShift-Installer-Role     Installer      arn:aws:iam::
   <aws_account_id>:role/ManagedOpenShift-Installer-Role     4.10
   ManagedOpenShift-Support-Role       Support        arn:aws:iam::
   <aws_account_id>:role/ManagedOpenShift-Support-Role       4.10
   ManagedOpenShift-Worker-Role        Worker         arn:aws:iam::
   <aws_account_id>:role/ManagedOpenShift-Worker-Role        4.10
   ```

   b. Delete the account-wide roles:

```
$ rosa delete account-roles --prefix <prefix> --mode auto 1
```

1 You must include the **--<prefix>** argument. Replace **<prefix>** with the prefix of the account-wide roles to delete. If you did not specify a custom prefix when you created the account-wide roles, specify the default prefix, **ManagedOpenShift**.

IMPORTANT

The account-wide IAM roles might be used by other ROSA clusters in the same AWS account. You must only remove the roles if they are not required by other clusters.

2. Delete the account-wide in-line and Operator policies:

   a. Under the **Policies** page in the AWS IAM Console, filter the list of policies by the prefix that you specified when you created the account-wide roles and policies.

   NOTE

   If you did not specify a custom prefix when you created the account-wide roles, search for the default prefix, **ManagedOpenShift**.

   b. Delete the account-wide in-line policies and Operator policies by using the AWS IAM Console. For more information about deleting IAM policies by using the AWS IAM Console, see Deleting IAM policies in the AWS documentation.

   IMPORTANT

   The account-wide in-line and Operator IAM policies might be used by other ROSA clusters in the same AWS account. You must only remove the roles if they are not required by other clusters.

8.3.2. Unlinking and deleting the OpenShift Cluster Manager and user IAM roles

If you installed a Red Hat OpenShift Service on AWS (ROSA) cluster by using Red Hat OpenShift Cluster Manager, you created OpenShift Cluster Manager and user Identity and Access Management (IAM) roles and linked them to your Red Hat organization. After deleting your cluster, you can unlink and delete the roles by using the ROSA CLI (**rosa**).

IMPORTANT

The OpenShift Cluster Manager and user IAM roles are required if you want to use OpenShift Cluster Manager to install and manage other ROSA clusters in the same AWS account. You must only remove the roles if you no longer need to use OpenShift Cluster Manager to install ROSA clusters.

Prerequisites

- You created OpenShift Cluster Manager and user IAM roles and linked them to your Red Hat organization.
- You have installed and configured the latest ROSA CLI (**rosa**) on your installation host.

- You have organization administrator privileges in your Red Hat organization.

Procedure

1. Unlink the OpenShift Cluster Manager IAM role from your Red Hat organization and delete the role:

   a. List the OpenShift Cluster Manager IAM roles in your AWS account:

   ```
   $ rosa list ocm-roles
   ```

   #### Example output

   ```
   I: Fetching ocm roles
   ROLE NAME                          ROLE ARN
   LINKED  ADMIN
   ManagedOpenShift-OCM-Role-<red_hat_organization_external_id>  arn:aws:iam::
   <aws_account_id>:role/ManagedOpenShift-OCM-Role-
   <red_hat_organization_external_id>  Yes     Yes
   ```

   b. If your OpenShift Cluster Manager IAM role is listed as linked in the output of the preceding command, unlink the role from your Red Hat organization:

   ```
   $ rosa unlink ocm-role --role-arn <arn> 1
   ```

   1 Replace **<arn>** with the Amazon Resource Name (ARN) for your OpenShift Cluster Manager IAM role. The ARN is specified in the output of the preceding command. In the preceding example, the ARN is in the format **arn:aws:iam::<aws_account_external_id>:role/ManagedOpenShift-OCM-Role-<red_hat_organization_external_id>**.

   #### Example output

   ```
   I: Unlinking OCM role
   ? Unlink the 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-OCM-Role-
   <red_hat_organization_external_id>' role from organization '<red_hat_organization_id>'?
   Yes
   I: Successfully unlinked role-arn 'arn:aws:iam::
   <aws_account_id>:role/ManagedOpenShift-OCM-Role-
   <red_hat_organization_external_id>' from organization account
   '<red_hat_organization_id>'
   ```

   c. Delete the OpenShift Cluster Manager IAM role and policies:

   ```
   $ rosa delete ocm-role --role-arn <arn>
   ```

   #### Example output

   ```
   I: Deleting OCM role
   ? OCM Role ARN: arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-OCM-Role-
   <red_hat_organization_external_id>
   ? Delete 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-OCM-Role-
   ```

```
<red_hat_organization_external_id>' ocm role? Yes
? OCM role deletion mode: auto 1
I: Successfully deleted the OCM role
```

1 Specifies the deletion mode. You can use **auto** mode to automatically delete the OpenShift Cluster Manager IAM role and policies. In **manual** mode, the **rosa** CLI generates the **aws** commands needed to delete the role and policies. **manual** mode enables you to review the details before running the **aws** commands manually.

2. Unlink the user IAM role from your Red Hat organization and delete the role:

   a. List the user IAM roles in your AWS account:

   ```
   $ rosa list user-roles
   ```

   Example output

   ```
   I: Fetching user roles
   ROLE NAME                              ROLE ARN
   LINKED
   ManagedOpenShift-User-<ocm_user_name>-Role  arn:aws:iam::
   <aws_account_id>:role/ManagedOpenShift-User-<ocm_user_name>-Role  Yes
   ```

   b. If your user IAM role is listed as linked in the output of the preceding command, unlink the role from your Red Hat organization:

   ```
   $ rosa unlink user-role --role-arn <arn> 1
   ```

   1 Replace **<arn>** with the Amazon Resource Name (ARN) for your user IAM role. The ARN is specified in the output of the preceding command. In the preceding example, the ARN is in the format **arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-User-<ocm_user_name>-Role**.

   Example output

   ```
   I: Unlinking user role
   ? Unlink the 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-User-
   <ocm_user_name>-Role' role from the current account '<ocm_user_account_id>'? Yes
   I: Successfully unlinked role ARN 'arn:aws:iam::
   <aws_account_id>:role/ManagedOpenShift-User-<ocm_user_name>-Role' from account
   '<ocm_user_account_id>'
   ```

   c. Delete the user IAM role:

   ```
   $ rosa delete user-role --role-arn <arn>
   ```

   Example output

   ```
   I: Deleting user role
   ? User Role ARN: arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-User-
   <ocm_user_name>-Role
   ```

```
? Delete the 'arn:aws:iam::<aws_account_id>:role/ManagedOpenShift-User-
<ocm_user_name>-Role' role from the AWS account? Yes
? User role deletion mode: auto 1
I: Successfully deleted the user role
```

1. Specifies the deletion mode. You can use **auto** mode to automatically delete the user IAM role. In **manual** mode, the **rosa** CLI generates the **aws** command needed to delete the role. **manual** mode enables you to review the details before running the **aws** command manually.

8.4. ADDITIONAL RESOURCES

- For information about the AWS IAM resources for ROSA clusters that use STS, see About IAM resources for ROSA clusters that use STS.

CHAPTER 9. DEPLOYING ROSA WITHOUT AWS STS

9.1. AWS PREREQUISITES FOR ROSA

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

You must ensure that the prerequisites are met before installing ROSA. This requirements document does not apply to AWS Security Token Service (STS). If you are using STS, see the STS-specific requirements.

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.

9.1.1. Deployment Prerequisites

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

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

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

9.1.2. Customer Requirements

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

NOTE

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

9.1.2.1. Account

- The customer ensures that the AWS limits are sufficient to support Red Hat OpenShift Service on AWS provisioned within the customer's AWS account.

- The customer's AWS account should be in the customer's AWS Organizations with the applicable service control policy (SCP) applied.

NOTE

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

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

NOTE

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

9.1.2.2. Access requirements

- To appropriately manage the Red Hat OpenShift Service on AWS service, Red Hat must have the **AdministratorAccess** policy applied to the administrator role at all times. This requirement does **not** apply if you are using AWS Security Token Service (STS).

NOTE

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

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

9.1.2.3. Support requirements

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

9.1.2.4. Security requirements

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

9.1.3. Required customer procedure

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

Procedure

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

9.1.3.1. Minimum required service control policy (SCP)

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

NOTE

The minimum SCP requirement does not apply when using AWS security token service (STS). For more information about STS, see AWS prerequisites for ROSA with STS.

	Service	Actions	Effect
Required	Amazon EC2	All	Allow
	Amazon EC2 Auto Scaling	All	Allow
	Amazon S3	All	Allow
	Identity And Access Management	All	Allow
	Elastic Load Balancing	All	Allow

	Service	Actions	Effect
	Elastic Load Balancing V2	All	Allow
	Amazon CloudWatch	All	Allow
	Amazon CloudWatch Events	All	Allow
	Amazon CloudWatch Logs	All	Allow
	AWS Support	All	Allow
	AWS Key Management Service	All	Allow
	AWS Security Token Service	All	Allow
	AWS Resource Tagging	All	Allow
	AWS Route53 DNS	All	Allow
	AWS Service Quotas	ListServices GetRequestedServiceQuotaChange GetServiceQuota RequestServiceQuotaIncrease ListServiceQuotas	Allow
Optional	AWS Billing	ViewAccount Viewbilling ViewUsage	Allow
	AWS Cost and Usage Report	All	Allow
	AWS Cost Explorer Services	All	Allow

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "events:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "support:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sts:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "tag:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "route53:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "servicequotas:ListServices",
                "servicequotas:GetRequestedServiceQuotaChange",
                "servicequotas:GetServiceQuota",
                "servicequotas:RequestServiceQuotaIncrease",
                "servicequotas:ListServiceQuotas"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

9.1.4. Red Hat managed IAM references for AWS

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

9.1.4.1. IAM Policies

NOTE

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

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

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

9.1.4.2. IAM users

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

9.1.5. Provisioned AWS Infrastructure

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

9.1.5.1. EC2 instances

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

Instance types can vary for control plane and infrastructure nodes, depending on the worker node count. At a minimum, the following EC2 instances will be deployed:

- Three **m5.2xlarge** control plane nodes
- Two **r5.xlarge** infrastructure nodes
- Two **m5.xlarge** customizable worker nodes

For further guidance on worker node counts, see the link to "Initial Planning Considerations" in the "Additional resources" section of this page.

9.1.5.2. AWS Elastic Block Store (EBS) storage

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

Volume requirements for each EC2 instance:

- Control Plane Volume
  - Size: 350GB
  - Type: io1
  - Input/Output Operations Per Second: 1000
- Infrastructure Volume
  - Size: 300GB
  - Type: gp2
  - Input/Output Operations Per Second: 900
- Worker Volume
  - Size: 300GB
  - Type: gp2
  - Input/Output Operations Per Second: 900

9.1.5.3. Elastic load balancers

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

9.1.5.4. S3 storage

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

NOTE

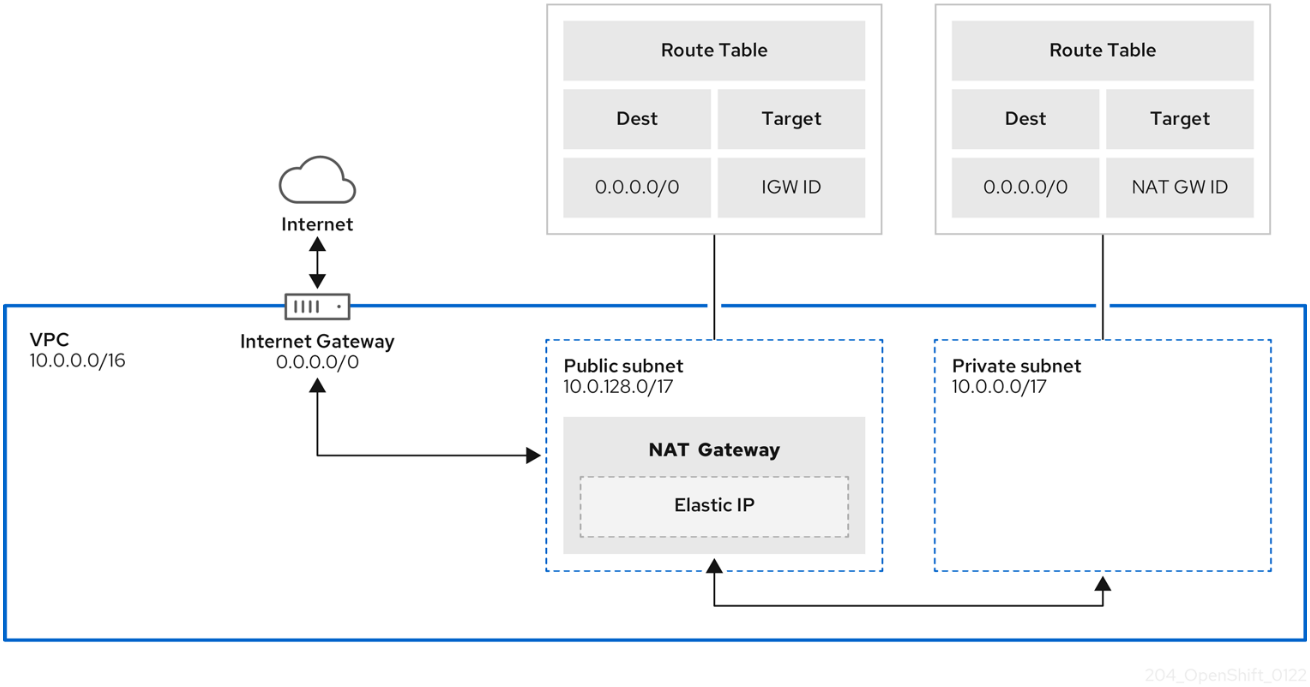
Two buckets are required with a typical size of 2TB each.

9.1.5.5. VPC

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

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

9.1.5.5.1. Sample VPC Architecture



Route Table	
Dest	Target
0.0.0.0/0	IGW ID

Route Table	
Dest	Target
0.0.0.0/0	NAT GW ID

9.1.5.6. Security groups

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

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

9.1.6. AWS firewall prerequisites

IMPORTANT

Only ROSA clusters deployed with PrivateLink can use a firewall to control egress traffic.

This section provides the necessary details that enable you to control egress traffic from your Red Hat OpenShift Service on AWS cluster. If you are using a firewall to control egress traffic, you must configure your firewall to grant access to the domain and port combinations below. Red Hat OpenShift Service on AWS requires this access to provide a fully managed OpenShift service.

Procedure

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

Domain	Port	Function
registry.redhat.io	443	Provides core container images.
quay.io	443	Provides core container images.
***.quay.io**	443	Provides core container images.

Domain	Port	Function
sso.redhat.com	443, 80	Required. The **https://console.redhat.com/openshift** site uses authentication from **sso.redhat.com** to download the pull secret and use Red Hat SaaS solutions to facilitate monitoring of your subscriptions, cluster inventory, chargeback reporting, and so on.
quay-registry.s3.amazonaws.com	443	Provides core container images.
cm-quay-production-s3.s3.amazonaws.com	443	Provides core container images.
cart-rhcos-ci.s3.amazonaws.com	443	Provides Red Hat Enterprise Linux CoreOS (RHCOS) images.
openshift.org	443	Provides Red Hat Enterprise Linux CoreOS (RHCOS) images.
registry.access.redhat.com	443	Provides access to the **odo** CLI tool that helps developers build on OpenShift and Kubernetes.
console.redhat.com	443, 80	Required. Allows interactions between the cluster and OpenShift Console Manager to enable functionality, such as scheduling upgrades.
sso.redhat.com	443	The **https://console.redhat.com/openshift** site uses authentication from **sso.redhat.com**.
pull.q1w2.quay.rhcloud.com	443	Provides core container images as a fallback when quay.io is not available.
.q1w2.quay.rhcloud.com	443	Provides core container images as a fallback when quay.io is not available.

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

CDN host names, such as **cdn01.quay.io**, are covered when you add a wildcard entry, such as **.quay.io**, in your allowlist.

2. Allowlist the following telemetry URLs:

Domain	Port	Function
cert-api.access.redhat.com	443	Required for telemetry.
api.access.redhat.com	443	Required for telemetry.
infogw.api.openshift.com	443	Required for telemetry.
console.redhat.com	443	Required for telemetry and Red Hat Insights.
observatorium.api.openshift.comm	443	Required for managed OpenShift-specific telemetry.

Managed clusters require enabling telemetry to allow Red Hat to react more quickly to problems, better support the customers, and better understand how product upgrades impact clusters. See About remote health monitoring for more information about how remote health monitoring data is used by Red Hat.

3. Allowlist the following Amazon Web Services (AWS) API URls:

Domain	Port	Function
.amazonaws.com	443	Required to access AWS services and resources.

Alternatively, if you choose to not use a wildcard for Amazon Web Services (AWS) APIs, you must allowlist the following URLs:

Domain	Port	Function
ec2.amazonaws.com	443	Used to install and manage clusters in an AWS environment.
events.amazonaws.com	443	Used to install and manage clusters in an AWS environment.
iam.amazonaws.com	443	Used to install and manage clusters in an AWS environment.
route53.amazonaws.com	443	Used to install and manage clusters in an AWS environment.
sts.amazonaws.com	443	Used to install and manage clusters in an AWS environment.

Domain	Port	Function
tagging.us-east-1.amazonaws.com	443	Used to install and manage clusters in an AWS environment. This endpoint is always us-east-1, regardless of the region the cluster is deployed in.
ec2.<aws_region>.amazonaws.com	443	Used to install and manage clusters in an AWS environment.
elasticloadbalancing.<aws_region>.amazonaws.com	443	Used to install and manage clusters in an AWS environment.
servicequotas.<aws region>.amazonaws.com	443, 80	Required. Used to confirm quotas for deploying the service.
tagging.<region>.amazonaws.com	443, 80	Allows the assignment of metadata about AWS resources in the form of tags.

4. Allowlist the following OpenShift URLs:

Domain	Port	Function
mirror.openshift.com	443	Used to access mirrored installation content and images. This site is also a source of release image signatures, although the Cluster Version Operator (CVO) needs only a single functioning source.
storage.googleapis.com/openshift-release (Recommended)	443	Alternative site to mirror.openshift.com/. Used to download platform release signatures that are used by the cluster to know what images to pull from quay.io.
api.openshift.com	443	Used to check if updates are available for the cluster.

5. Allowlist the following site reliability engineering (SRE) and management URLs:

Domain	Port	Function
api.pagerduty.com	443	This alerting service is used by the in-cluster alertmanager to send alerts notifying Red Hat SRE of an event to take action on.
events.pagerduty.com	443	This alerting service is used by the in-cluster alertmanager to send alerts notifying Red Hat SRE of an event to take action on.

Domain	Port	Function
api.deadmanssnitch.com	443	Alerting service used by Red Hat OpenShift Service on AWS to send periodic pings that indicate whether the cluster is available and running.
nosnch.in	443	Alerting service used by Red Hat OpenShift Service on AWS to send periodic pings that indicate whether the cluster is available and running.
***.osdsecuritylogs.splunkcloud.com** OR **inputs1.osdsecuritylogs.splunkcloud.cominputs2.osdsecuritylogs.splunkcloud.cominputs4.osdsecuritylogs.splunkcloud.cominputs5.osdsecuritylogs.splunkcloud.cominputs6.osdsecuritylogs.splunkcloud.cominputs7.osdsecuritylogs.splunkcloud.cominputs8.osdsecuritylogs.splunkcloud.cominputs9.osdsecuritylogs.splunkcloud.cominputs10.osdsecuritylogs.splunkcloud.cominputs11.osdsecuritylogs.splunkcloud.cominputs12.osdsecuritylogs.splunkcloud.cominputs13.osdsecuritylogs.splunkcloud.cominputs14.osdsecuritylogs.splunkcloud.cominputs15.osdsecuritylogs.splunkcloud.com**	9997	Used by the **splunk-forwarder-operator** as a logging forwarding endpoint to be used by Red Hat SRE for log-based alerting.
http-inputs-osdsecuritylogs.splunkcloud.com	443	Required. Used by the **splunk-forwarder-operator** as a logging forwarding endpoint to be used by Red Hat SRE for log-based alerting.
sftp.access.redhat.com (Recommended)	22	The SFTP server used by **must-gather-operator** to upload diagnostic logs to help troubleshoot issues with the cluster.

6. If you did not allow a wildcard for Amazon Web Services (AWS) APIs, you must also allow the S3 bucket used for the internal OpenShift registry. To retrieve that endpoint, run the following command after the cluster is successfully provisioned:

```
$ oc -n openshift-image-registry get pod -l docker-registry=default -o json | jq
'.items[].spec.containers[].env[] | select(.name=="REGISTRY_STORAGE_S3_BUCKET")'
```

The S3 endpoint should be in the following format: '<cluster-name>-<random-string>-image-registry-<cluster-region>-<random-string>.s3.dualstack.<cluster-region>.amazonaws.com'.

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

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

9.1.7. Next steps

- Review the required AWS service quotas

9.1.8. Additional resources

- Limits and scalability
- SRE access to all Red Hat OpenShift Service on AWS clusters
- Understanding the ROSA deployment workflow

9.2. UNDERSTANDING THE ROSA DEPLOYMENT WORKFLOW

Before you create a Red Hat OpenShift Service on AWS (ROSA) cluster that uses the AWS Security Token Service (STS), you must complete the AWS prerequisites, verify that the required AWS service quotas are available, and set up your environment.

This document provides an overview of the ROSA with STS deployment workflow stages and refers to detailed resources for each stage.

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.

9.2.1. Overview of the ROSA deployment workflow

You can follow the workflow stages outlined in this section to set up and access a Red Hat OpenShift Service on AWS (ROSA) cluster.

1. Perform the AWS prerequisites. To deploy a ROSA cluster, your AWS account must meet the prerequisite requirements.

2. Review the required AWS service quotas . To prepare for your cluster deployment, review the AWS service quotas that are required to run a ROSA cluster.

3. Configure your AWS account . Before you create a ROSA cluster, you must enable ROSA in your AWS account, install and configure the AWS CLI (**aws**) tool, and verify the AWS CLI tool configuration.

4. Install the ROSA and OpenShift CLI tools and verify the AWS servce quotas . Install and configure the ROSA CLI (**rosa**) and the OpenShift CLI (**oc**). You can verify if the required AWS resource quotas are available by using the ROSA CLI.

5. Create a ROSA cluster or Create a ROSA cluster using AWS PrivateLink . Use the ROSA CLI (**rosa**) to create a cluster. You can optionally create a ROSA cluster with AWS PrivateLink.

6. Access a cluster . You can configure an identity provider and grant cluster administrator privileges to the identity provider users as required. You can also access a newly deployed cluster quickly by configuring a **cluster-admin** user.

7. Revoke access to a ROSA cluster for a user . You can revoke access to a ROSA cluster from a user by using the ROSA CLI or the web console.

8. Delete a ROSA cluster. You can delete a ROSA cluster by using the ROSA CLI (**rosa**).

9.2.2. Additional resources

- For information about using the ROSA deployment workflow to create a cluster that uses the AWS Security Token Service (STS), see Understanding the ROSA with STS deployment workflow.
- Configuring identity providers
- Deleting a cluster
- Deleting access to a cluster
- Command quick reference for creating clusters and users

9.3. REQUIRED AWS SERVICE QUOTAS

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

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.

9.3.1. Required AWS service quotas

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

NOTE

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

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

Quota name	Service code	Quota code	Minimum required value	Recommended value
Number of EIPs - VPC EIPs	ec2	L-0263D0A3	5	5

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

9.3.2. Next steps

- Configure your AWS account

9.3.3. Additional resources

- Understanding the ROSA deployment workflow

9.4. CONFIGURING YOUR AWS ACCOUNT

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

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.

9.4.1. Configuring your AWS account

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

Prerequisites

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

Procedure

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

   If you are using AWS Organizations and you need to have a service control policy (SCP) applied to the AWS account you plan to use, see AWS Prerequisites for details on the minimum required SCP.

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

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

2. Enable the ROSA service in the AWS Console.
   a. Sign in to your AWS account.
   b. To enable ROSA, go to the ROSA service and select **Enable OpenShift**.

3. Install and configure the AWS CLI.
   a. Follow the AWS command-line interface documentation to install and configure the AWS CLI for your operating system.
      Specify the correct **aws_access_key_id** and **aws_secret_access_key** in the **.aws/credentials** file. See AWS Configuration basics in the AWS documentation.
   b. Set a default AWS region.

      > NOTE
      >
      > It is recommended to set the default AWS region by using the environment variable.

      The ROSA service evaluates regions in the following priority order:

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

ii. The region set in the **AWS_DEFAULT_REGION** environment variable. See Environment variables to configure the AWS CLI in the AWS documentation.

iii. The default region set in your AWS configuration file. See Quick configuration with aws configure in the AWS documentation.

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

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

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

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

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

Example output

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

After completing these steps, install ROSA.

9.4.2. Next steps

- Installing the ROSA CLI

9.4.3. Additional resources

- AWS prerequisites
- Required AWS service quotas and requesting increases
- Understanding the ROSA deployment workflow

9.5. INSTALLING THE ROSA CLI

After you configure your AWS account, install and configure the ROSA CLI (**rosa**).

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.

9.5.1. Installing and configuring the ROSA CLI

Install and configure the ROSA CLI (**rosa**). You can also install the OpenShift CLI (**oc**) and verify if the required AWS resource quotas are available by using the ROSA CLI.

Prerequisites

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

Procedure

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

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

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

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

   #### Example

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

   d. Enter the following command to verify your installation:

   ```
   $ rosa
   ```

   #### Example output

   ```
   Command line tool for ROSA.

   Usage:
     rosa [command]

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

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

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

e. Optional: Generate the command completion scripts for the **rosa** CLI. The following example generates the Bash completion scripts for a Linux machine:

```
$ rosa completion bash | sudo tee /etc/bash_completion.d/rosa
```

f. Optional: Enable **rosa** command completion from your existing terminal. The following example enables Bash completion for **rosa** in an existing terminal on a Linux machine:

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

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

```
$ rosa verify permissions
```

Example output

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

NOTE

This command verifies permissions only for ROSA clusters that do not use the AWS Security Token Service (STS).

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

a. Enter the following command.

```
$ rosa login
```

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

Example output

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

Example output continued

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

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

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

Example output

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

NOTE

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

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

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

5. Prepare your AWS account for cluster deployment:

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

   ```
   $ rosa whoami
   ```

   ### Example output

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

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

   ```
   $ rosa init
   ```

   ### Example output

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

```
I: AWS quota ok
I: Ensuring cluster administrator user 'osdCcsAdmin'...
I: Admin user 'osdCcsAdmin' created successfully!
I: Verifying whether OpenShift command-line tool is available...
E: OpenShift command-line tool is not installed.
Run 'rosa download oc' to download the latest version, then add it to your PATH.
```

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

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

    ```
    $ rosa download oc
    ```

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

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

    ```
    $ rosa verify oc
    ```

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

9.5.2. Next steps

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

9.5.3. Additional resources

- AWS prerequisites
- Required AWS service quotas and requesting increases
- Understanding the ROSA deployment workflow

9.6. CREATING A ROSA CLUSTER WITHOUT AWS STS

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

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

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.

9.6.1. Creating your cluster

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

Prerequisites

You have installed Red Hat OpenShift Service on AWS.

NOTE

AWS Shared VPCs are not currently supported for ROSA installs.

Procedure

1. You can create a cluster using the default settings or by specifying custom settings using the interactive mode. To view other options when creating a cluster, enter **rosa create cluster --help**.
   Creating a cluster can take up to 40 minutes.

   NOTE

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

   - To create your cluster with the default cluster settings:

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

     Example output

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

   - To create a cluster using interactive prompts:

     ```
     $ rosa create cluster --interactive
     ```

   - To configure your networking IP ranges, you can use the following default ranges. For more information when using manual mode, use **rosa create cluster --help | grep cidr**. In interactive mode, you are prompted for the settings.
     - Node CIDR: 10.0.0.0/16
     - Service CIDR: 172.30.0.0/16
     - Pod CIDR: 10.128.0.0/14

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

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

   Example output

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

NOTE

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

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

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

9.6.2. Next steps

Configure identity providers

9.6.3. Additional resources

- Understanding the ROSA deployment workflow
- Deleting a ROSA cluster
- ROSA architecture

9.7. DELETING ACCESS TO A ROSA CLUSTER

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

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.

9.7.1. Revoking dedicated-admin access using the rosa CLI

You can revoke access for a **dedicated-admin** user if you are the user who created the cluster, the organization administrator user, or the super administrator user.

Prerequisites

Prerequisites

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

Procedure

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

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

2. Enter the following command to verify that your user no longer has **dedicated-admin** access. The output does not list the revoked user.

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

9.7.2. Revoking cluster-admin access using the rosa CLI

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

Prerequisites

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

Procedure

1. Enter the following command to revoke the **cluster-admin** access of a user:

   ```
   $ rosa revoke user cluster-admins --user=myusername --cluster=mycluster
   ```

2. Enter the following command to verify that the user no longer has **cluster-admin** access. The output does not list the revoked user.

   ```
   $ oc get groups cluster-admins
   ```

9.8. DELETING A ROSA CLUSTER

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

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.

9.8.1. Prerequisites

- If Red Hat OpenShift Service on AWS created a VPC, you must remove the following items from your cluster before you can successfully delete your cluster:
  - Network configurations, such as VPN configurations and VPC peering connections
  - Any additional services that were added to the VPC

If these configurations and services remain, the cluster does not delete properly.

9.8.2. Deleting a ROSA cluster and the cluster-specific IAM resources

You can delete a Red Hat OpenShift Service on AWS (ROSA) with AWS Security Token Service (STS) cluster by using the ROSA CLI (**rosa**) or Red Hat OpenShift Cluster Manager.

After deleting the cluster, you can clean up the cluster-specific Identity and Access Management (IAM) resources in your AWS account by using the ROSA CLI (**rosa**). The cluster-specific resources include the Operator roles and the OpenID Connect (OIDC) provider.

IMPORTANT

The cluster deletion must complete before you remove the IAM resources, because the resources are used in the cluster deletion and clean-up processes.

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

Prerequisites

- You have installed a ROSA cluster.
- You have installed and configured the latest ROSA CLI (**rosa**) on your installation host.

Procedure

1. Obtain the cluster ID, the Amazon Resource Names (ARNs) for the cluster-specific Operator roles and the endpoint URL for the OIDC provider:

```
$ rosa describe cluster --cluster=<cluster_name> 1
```

1 Replace **<cluster_name>** with the name of your cluster.

Example output

```
Name:                  mycluster
ID:                 1s3v4x39lhs8sm49m90mi0822o34544a 1
...
Operator IAM Roles: 2
 - arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-machine-api-aws-cloud-credentials
 - arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-cloud-credential-operator-cloud-crede
 - arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-image-registry-installer-cloud-creden
 - arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-ingress-operator-cloud-
credentials
 - arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-cluster-csi-drivers-ebs-
cloud-credent
 - arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-cloud-network-config-
controller-cloud
State:                    ready
Private:                  No
Created:                  May 13 2022 11:26:15 UTC
Details Page:
https://console.redhat.com/openshift/details/s/296kyEFwzoy1CREQicFRdZybrc0
OIDC Endpoint URL:        https://rh-oidc.s3.us-east-
1.amazonaws.com/1s5v4k39lhm8sm59m90mi0822o31844a 3
```

1. Lists the cluster ID.
2. Specifies the ARNs for the cluster-specific Operator roles. For example, in the sample output the ARN for the role required by the Machine Config Operator is **arn:aws:iam::<aws_account_id>:role/mycluster-x4q9-openshift-machine-api-aws-cloud-credentials**.
3. Specifies the endpoint URL for the cluster-specific OIDC provider.

IMPORTANT

You require the cluster ID to delete the cluster-specific STS resources using the ROSA CLI (**rosa**) after the cluster is deleted.

2. Delete the cluster:

- To delete the cluster by using Red Hat OpenShift Cluster Manager:

  a. Navigate to OpenShift Cluster Manager.

  b. Click the Options menu ⋮ next to your cluster and select **Delete cluster**.

  c. Type the name of your cluster at the prompt and click **Delete**.

- To delete the cluster using the ROSA CLI (**rosa**):

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

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

IMPORTANT

You must wait for the cluster deletion to complete before you remove the Operator roles and the OIDC provider. The cluster-specific Operator roles are required to clean-up the resources created by the OpenShift Operators. The Operators use the OIDC provider to authenticate.

3. Delete the OIDC provider that the cluster Operators use to authenticate:

```
$ rosa delete oidc-provider -c <cluster_id> --mode auto 1
```

1 Replace **<cluster_id>** with the ID of the cluster.

NOTE

You can use the **-y** option to automatically answer yes to the prompts.

4. Delete the cluster-specific Operator IAM roles:

```
$ rosa delete operator-roles -c <cluster_id> --mode auto 1
```

1 Replace **<cluster_id>** with the ID of the cluster.

9.9. COMMAND QUICK REFERENCE FOR CREATING CLUSTERS AND USERS

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.

9.9.1. Command quick reference list

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

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

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

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

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

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

9.9.2. Additional resources

- Understanding the ROSA deployment workflow