Red Hat OpenShift Service on AWS 4

Prepare your environment

Planning, limits, and scalability for Red Hat OpenShift Service on AWS
Planning, limits, and scalability for Red Hat OpenShift Service on AWS
Abstract

This document provides planning considerations for Red Hat OpenShift Service on AWS (ROSA) cluster deployments, including information about cluster limits and scalability.
CHAPTER 1. PREREQUISITES CHECKLIST FOR DEPLOYING ROSA USING STS

This is a checklist of prerequisites needed to create a Red Hat OpenShift Service on AWS (ROSA) classic cluster with STS.

NOTE
This is a high level checklist and your implementation can vary.

Before running the installation process, verify that you deploy this from a machine that has access to:

- The API services for the cloud to which you provision.
- The hosts on the network that you provision.
- The internet to obtain installation media.

1.1. ACCOUNTS AND CLIS PREREQUISITES
Accounts and CLIs you must install to deploy the cluster.

1.1.1. AWS account
Gather the following details:

- AWS IAM User
- AWS Access Key ID
- AWS Secret Access Key

Ensure that you have the right permissions as detailed AWS managed IAM policies for ROSA and About IAM resources for ROSA clusters that use STS.

See Account for more details.

1.1.2. AWS CLI (aws)

- Install from AWS Command Line Interface if you have not already.

- Configure the CLI:
  1. Enter **aws configure** in the terminal:
     
```bash
$ aws configure
```
  2. Enter the AWS Access Key ID and press **enter**.
  3. Enter the AWS Secret Access Key and press **enter**.
  4. Enter the default region you want to deploy into.
5. Enter the output format you want, “table” or “json”.

6. Verify the output by running:

   $ aws sts get-caller-identity

7. Ensure that the service role for ELB already exists by running:

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

   a. If it does not exist, run:

   $ aws iam create-service-linked-role --aws-service-name "elasticloadbalancing.amazonaws.com"

1.1.3. Red Hat account

- Create a Red Hat Hybrid Cloud Console account if you have not already.

1.1.4. ROSA CLI (rosa)

1. Enable ROSA from your AWS account on the AWS console if you have not already.

2. Install the CLI from Installing the Red Hat OpenShift Service on AWS (ROSA) CLI, rosa or from the OpenShift console AWS console.

3. Enter rosa login in a terminal, and this will prompt you to go to the token page through the console:

   $ rosa login

4. Log in with your Red Hat account credentials.

5. Click the Load token button.

6. Copy the token and paste it back into the CLI prompt and press enter.

   - Alternatively, you can copy the full $ rosa login --token=abc... command and paste that in the terminal:

   $ rosa login --token=<abc..>

7. Verify your credentials by running:

   $ rosa whoami

8. Ensure you have sufficient quota by running:

   $ rosa verify quota

   - See Provisioned AWS Infrastructure for more details on AWS services provisioned for ROSA cluster.
1.1.5. OpenShift CLI (oc)

1. Install from Getting started with the OpenShift CLI or from the OpenShift console Command-line interface (CLI) tools.

2. Verify that the OpenShift CLI has been installed correctly by running:

   ```
   $ rosa verify openshift-client
   ```

Once you have the above prerequisites installed and enabled, proceed to the next steps.

1.2. SCP PREREQUISITES

ROSA clusters are 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.

- Ensure that your organization’s SCPs are not more restrictive than the roles and policies required by the cluster.

- Ensure that your SCP is configured to allow the required `aws-marketplace:Subscribe` permission when you choose Enable ROSA from the console, and see AWS Organizations service control policy (SCP) is denying required AWS Marketplace permissions for more details.

- When you create a ROSA classic cluster, an associated AWS OpenID Connect (OIDC) identity provider is created.
  - This OIDC provider configuration relies on a public key that is located in the us-east-1 AWS region.
  - Customers with AWS SCPs must allow the use of the us-east-1 AWS region, even if these clusters are deployed in a different region.

1.3. NETWORKING PREREQUISITES

Prerequisites needed from a networking standpoint.

1.3.1. Firewall

- Configure your firewall to allow access to the domains and ports listed in AWS firewall prerequisites.

1.3.2. Additional custom security groups

When you create a cluster using an existing non-managed VPC, you can add additional custom security groups during cluster creation. Complete these prerequisites before you create the cluster:

- Create the custom security groups in AWS before you create the cluster.

- Associate the custom security groups with the VPC that you are using to create the cluster. Do not associate the custom security groups with any other VPC.
• You may need to request additional AWS quota for Security groups per network interface. For more details see the detailed requirements for Security groups.

1.3.3. Custom DNS

• If you want to use custom DNS, then the ROSA installer must be able to use VPC DNS with default DHCP options so it can resolve hosts locally.
  • To do so, run `aws ec2 describe-dhcp-options` and see if the VPC is using VPC Resolver:
    ```
    $ aws ec2 describe-dhcp-options
    ```
  • Otherwise, the upstream DNS will need to forward the cluster scope to this VPC so the cluster can resolve internal IPs and services.

1.4. PRIVATELINK PREREQUISITES

If you choose to deploy a PrivateLink cluster, then be sure to deploy the cluster in the pre-existing BYO VPC:

• Create a public and private subnet for each AZ that your cluster uses.
  • Alternatively, implement transit gateway for internet and egress with appropriate routes.

• The VPC’s CIDR block must contain the `Networking.MachineCIDR` range, which is the IP address for cluster machines.
  • The subnet CIDR blocks must belong to the machine CIDR that you specify.

• Set both `enableDnsHostnames` and `enableDnsSupport` to true.
  • That way, the cluster can use the Route 53 zones that are attached to the VPC to resolve cluster internal DNS records.

• Verify route tables by running:
  ```
  ----
  $ aws ec2 describe-route-tables --filters "Name=vpc-id,Values=<vpc-id>"
  ----
  • Ensure that the cluster can egress either through NAT gateway in public subnet or through transit gateway.
  • Ensure whatever UDR you would like to follow is set up.

• You can also configure a cluster-wide proxy during or after install. Configuring a cluster-wide proxy for more details.

NOTE

You can install a non-PrivateLink ROSA cluster in a pre-existing BYO VPC.
CHAPTER 2. DETAILED REQUIREMENTS FOR DEPLOYING ROSA USING STS

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.

**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.

Ensure that the following AWS prerequisites are met before installing ROSA with STS.

**IMPORTANT**

When you create a ROSA cluster using AWS STS, an associated AWS OpenID Connect (OIDC) identity provider is created as well. This OIDC provider configuration relies on a public key that is located in the **us-east-1** AWS region. Customers with AWS SCPs must allow the use of the **us-east-1** AWS region, even if these clusters are deployed in a different region.

**2.1. CUSTOMER REQUIREMENTS WHEN USING STS FOR DEPLOYMENT**

The following prerequisites must be complete before you deploy a Red Hat OpenShift Service on AWS (ROSA) cluster that uses the AWS Security Token Service (STS).

**2.1.1. Account**

- You must ensure that the AWS limits are sufficient to support Red Hat OpenShift Service on AWS provisioned within your AWS account. Running the `rosa verify quota` command in the CLI validates that you have the required quota to run a cluster.

  **NOTE**

  Quota verification checks your AWS quota, but it does not compare your consumption to your AWS quota. See the "Limits and scalability" link in Additional resources for more information.

- If SCP policies are applied and enforced, these policies must not be more restrictive than the roles and policies required by the cluster.

- Your AWS account should not be transferable to Red Hat.

- You should not impose additional AWS usage restrictions beyond the defined roles and policies on Red Hat activities. Imposing restrictions will severely hinder Red Hat’s ability to respond to incidents.

- You may deploy native AWS services within the same AWS account.

- Your account must have a service-linked role set up as it is required for Elastic Load Balancing (ELB) to be configured. See the "Creating the Elastic Load Balancing (ELB) service-linked role"
link in the Additional resources for information about creating a service-linked role for your ELB if you have not created a load balancer in your AWS account previously.

NOTE
You are encouraged, but not required, 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.

Additional resources

- Limits and scalability
- Creating the Elastic Load Balancing (ELB) service-linked role

2.1.2. Access requirements

- Red Hat must have AWS console access to the customer-provided AWS account. Red Hat protects and manages this access.

- You must not use the AWS account to elevate your permissions within the Red Hat OpenShift Service on AWS (ROSA) cluster.

- Actions available in the ROSA CLI (rosa) or OpenShift Cluster Manager Hybrid Cloud Console console must not be directly performed in your AWS account.

- You do not need to have a preconfigured domain to deploy ROSA clusters. If you wish to use a custom domain, see the Additional resources for information.

Additional resources

- See Configuring custom domains for applications

2.1.3. Support requirements

- Red Hat recommends that the customer have at least Business Support from AWS.

- Red Hat may have permission from the customer to request AWS support on their behalf.

- Red Hat may have permission from the customer to request AWS resource limit increases on the customer’s account.

- Red Hat manages the restrictions, limitations, expectations, and defaults for all Red Hat OpenShift Service on AWS clusters in the same manner, unless otherwise specified in this requirements section.

2.1.4. Security requirements

- 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 the documented domains. See the “AWS firewall prerequisites” section for the designated domains.
Additional resources

- AWS firewall prerequisites

2.1.5. Requirements for using OpenShift Cluster Manager

The following sections describe requirements for OpenShift Cluster Manager Hybrid Cloud Console. If you use the CLI tools exclusively, then you can disregard the requirements.

To use OpenShift Cluster Manager, you must link your AWS accounts. This linking concept is also known as account association.

2.1.5.1. AWS account association

Red Hat OpenShift Service on AWS (ROSA) cluster-provisioning tasks require linking ocm-role and user-role IAM roles to your AWS account using your Amazon Resource Name (ARN).

The ocm-role ARN is stored as a label in your Red Hat organization while the user-role ARN is stored as a label inside your Red Hat user account. Red Hat uses these ARN labels to confirm that the user is a valid account holder and that the correct permissions are available to perform the necessary tasks in the AWS account.

2.1.5.2. Linking your AWS account

You can link your AWS account to existing IAM roles by using the Red Hat OpenShift Service on AWS (ROSA) CLI, rosa.

Prerequisites

- You have an AWS account.
- You are using OpenShift Cluster Manager Hybrid Cloud Console to create clusters.
- You have the permissions required to install AWS account-wide roles. See the "Additional resources" of this section for more information.
- You have installed and configured the latest AWS (aws) and ROSA (rosa) CLIs on your installation host.
- You have created your ocm-role and user-role IAM roles, but have not yet linked them to your AWS account. You can check whether your IAM roles are already linked by running the following commands:

  $ rosa list ocm-role
  $ rosa list user-role

  If Yes is displayed in the Linked column for both roles, you have already linked the roles to an AWS account.

Procedure

1. From the CLI, link your ocm-role resource to your Red Hat organization by using your Amazon Resource Name (ARN):
NOTE

You must have Red Hat Organization Administrator privileges to run the `rosa link` command. After you link the `ocm-role` resource with your AWS account, it is visible for all users in the organization.

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

Example output

```
I: Linking OCM role
? Link the `<AWS ACCOUNT ID>` role with organization `<ORG ID>`? Yes
I: Successfully linked role-arn `<AWS ACCOUNT ID>` with organization account `<ORG ID>`
```

2. From the CLI, link your `user-role` resource to your Red Hat user account by using your Amazon Resource Name (ARN):

```bash
$ rosa link user-role --role-arn <arn>
```

Example output

```
I: Linking User role
? Link the `arn:aws:iam::<ARN>:role/ManagedOpenShift-User-Role-125` role with organization `<AWS ID>`? Yes
I: Successfully linked role-arn `arn:aws:iam::<ARN>:role/ManagedOpenShift-User-Role-125` with organization account `<AWS ID>`
```

Additional resources

- See [Account-wide IAM role and policy reference](#) for a list of IAM roles needed for cluster creation.

2.1.5.3. Associating multiple AWS accounts with your Red Hat organization

You can associate multiple AWS accounts with your Red Hat organization. Associating multiple accounts lets you create Red Hat OpenShift Service on AWS (ROSA) clusters on any of the associated AWS accounts from your Red Hat organization.

With this feature, you can create clusters in different AWS regions by using multiple AWS profiles as region-bound environments.

**Prerequisites**

- You have an AWS account.
- You are using OpenShift Cluster Manager Hybrid Cloud Console to create clusters.
- You have the permissions required to install AWS account-wide roles.
- You have installed and configured the latest AWS (`aws`) and ROSA (`rosa`) CLIs on your installation host.
- You have created your `ocm-role` and `user-role` IAM roles.
Procedure

To associate an additional AWS account, first create a profile in your local AWS configuration. Then, associate the account with your Red Hat organization by creating the `ocm-role`, user, and account roles in the additional AWS account.

To create the roles in an additional region, specify the `--profile <aws-profile>` parameter when running the `rosa create` commands and replace `<aws_profile>` with the additional account profile name:

- To specify an AWS account profile when creating an OpenShift Cluster Manager role:
  
  ```sh
  $ rosa create --profile <aws_profile> ocm-role
  ```

- To specify an AWS account profile when creating a user role:
  
  ```sh
  $ rosa create --profile <aws_profile> user-role
  ```

- To specify an AWS account profile when creating the account roles:
  
  ```sh
  $ rosa create --profile <aws_profile> account-roles
  ```

  **NOTE**

  If you do not specify a profile, the default AWS profile is used.

### 2.2. REQUIREMENTS FOR DEPLOYING A CLUSTER IN AN OPT-IN REGION

An AWS opt-in region is a region that is not enabled by default. If you want to deploy a Red Hat OpenShift Service on AWS (ROSA) cluster that uses the AWS Security Token Service (STS) in an opt-in region, you must meet the following requirements:

- The region must be enabled in your AWS account. For more information about enabling opt-in regions, see Managing AWS Regions in the AWS documentation.

- The security token version in your AWS account must be set to version 2. You cannot use version 1 security tokens for opt-in regions.

  **IMPORTANT**

  Updating to security token version 2 can impact the systems that store the tokens, due to the increased token length. For more information, see the AWS documentation on setting STS preferences.

#### 2.2.1. Setting the AWS security token version

If you want to create a Red Hat OpenShift Service on AWS (ROSA) cluster with the AWS Security Token Service (STS) in an AWS opt-in region, you must set the security token version to version 2 in your AWS account.

**Prerequisites**

- You have installed and configured the latest AWS CLI on your installation host.
Procedure

1. List the ID of the AWS account that is defined in your AWS CLI configuration:

   $ aws sts get-caller-identity --query Account --output json

   Ensure that the output matches the ID of the relevant AWS account.

2. List the security token version that is set in your AWS account:

   $ aws iam get-account-summary --query SummaryMap.GlobalEndpointTokenVersion --output json

   Example output

   1

3. To update the security token version to version 2 for all regions in your AWS account, run the following command:

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

   **IMPORTANT**

   Updating to security token version 2 can impact the systems that store the tokens, due to the increased token length. For more information, see the AWS documentation on setting STS preferences.

2.3. RED HAT MANAGED IAM REFERENCES FOR AWS

With the STS deployment model, Red Hat is no longer responsible for creating and managing Amazon Web Services (AWS) IAM policies, IAM users, or IAM roles. For information on creating these roles and policies, see the following sections on IAM roles.

- To use the `ocm` CLI, you must have an `ocm-role` and `user-role` resource. See OpenShift Cluster Manager IAM role resources.

- If you have a single cluster, see Account-wide IAM role and policy reference.

- For every cluster, you must have the necessary operator roles. See Cluster-specific Operator IAM role reference.

2.4. PROVISIONED AWS INFRASTRUCTURE

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

2.4.1. 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 information about initial planning considerations in the "Limits and scalability" topic listed in the "Additional resources" section of this page.

### 2.4.2. Amazon Elastic Block Store storage

Amazon Elastic Block Store (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: gp3
  - Input/Output Operations Per Second: 900

- **Worker Volume**
  - Size: 300GB
  - Type: gp3
  - Input/Output Operations Per Second: 900

**NOTE**

Clusters deployed before the release of OpenShift Container Platform 4.11 use gp2 type storage by default.

### 2.4.3. Elastic Load Balancing

Up to two Network Load Balancers for API and up to two Classic Load Balancers for application router. For more information, see the [ELB documentation for AWS](https://aws.amazon.com/documentation/elastic-load-balancing/).

### 2.4.4. S3 storage
The image registry is backed by AWS S3 storage. Pruning of resources is performed regularly to optimize S3 usage and cluster performance.

**NOTE**

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

### 2.4.5. 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.

  **NOTE**

  A public subnet connects directly to the internet through an internet gateway. A private subnet connects to the internet through a network address translation (NAT) gateway.

- **Route tables**: One route 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.

**Figure 2.1. Sample VPC Architecture**

![Sample VPC Architecture Diagram]

### 2.4.6. Security groups

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

### Table 2.1. Required ports for default security groups

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

#### 2.4.6.1. Additional custom security groups

When you create a cluster using an existing non-managed VPC, you can add additional custom security groups during cluster creation. Custom security groups are subject to the following limitations:

- You must create the custom security groups in AWS before you create the cluster. For more information, see Amazon EC2 security groups for Linux instances.

- You must associate the custom security groups with the VPC that the cluster will be installed into. Your custom security groups cannot be associated with another VPC.

- You might need to request additional quota for your VPC if you are adding additional custom security groups. For information on AWS quota requirements for ROSA, see Required AWS service quotas in Prepare your environment. For information on requesting an AWS quota increase, see Requesting a quota increase.

#### 2.5. 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:

<table>
<thead>
<tr>
<th>Domain</th>
<th>Port</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>registry.redhat.io</td>
<td>443</td>
<td>Provides core container images.</td>
</tr>
<tr>
<td>quay.io</td>
<td>443</td>
<td>Provides core container images.</td>
</tr>
<tr>
<td>*.quay.io</td>
<td>443</td>
<td>Provides core container images.</td>
</tr>
<tr>
<td>sso.redhat.com</td>
<td>443,80</td>
<td>Required. The <a href="https://console.redhat.com/openshift">https://console.redhat.com/openshift</a> 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.</td>
</tr>
<tr>
<td>quay-registry.s3.amazonaws.com</td>
<td>443</td>
<td>Provides core container images.</td>
</tr>
<tr>
<td>ocm-quay-production-s3.s3.amazonaws.com</td>
<td>443</td>
<td>Provides core container images.</td>
</tr>
<tr>
<td>quayio-production-s3.s3.amazonaws.com</td>
<td>443</td>
<td>Provides core container images.</td>
</tr>
<tr>
<td>cart-rhcos-ci.s3.amazonaws.com</td>
<td>443</td>
<td>Provides Red Hat Enterprise Linux CoreOS (RHCOS) images.</td>
</tr>
<tr>
<td>openshift.org</td>
<td>443</td>
<td>Provides Red Hat Enterprise Linux CoreOS (RHCOS) images.</td>
</tr>
<tr>
<td>registry.access.redhat.com[1]</td>
<td>443</td>
<td>Hosts all the container images that are stored on the Red Hat Ecosystem Catalog. Additionally, the registry provides access to the odo CLI tool that helps developers build on OpenShift and Kubernetes.</td>
</tr>
<tr>
<td>registry.connect.redhat.com</td>
<td>443,80</td>
<td>Required for all third-party images and certified Operators.</td>
</tr>
<tr>
<td>console.redhat.com</td>
<td>443,80</td>
<td>Required. Allows interactions between the cluster and OpenShift Console Manager to enable functionality, such as scheduling upgrades.</td>
</tr>
<tr>
<td>Domain</td>
<td>Port</td>
<td>Function</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>--------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pull.q1w2.quay.rhcloud.com</td>
<td>443</td>
<td>Provides core container images as a fallback when quay.io is not available.</td>
</tr>
<tr>
<td>.q1w2.quay.rhcloud.com</td>
<td>443</td>
<td>Provides core container images as a fallback when quay.io is not available.</td>
</tr>
<tr>
<td><a href="http://www.okd.io">www.okd.io</a></td>
<td>443</td>
<td>The <a href="https://openshift.org">openshift.org</a> site redirects through <a href="http://www.okd.io">www.okd.io</a>.</td>
</tr>
<tr>
<td>aws.amazon.com</td>
<td>443</td>
<td>The <a href="https://iam.amazonaws.com">iam.amazonaws.com</a> and <a href="https://sts.amazonaws.com">sts.amazonaws.com</a> sites redirect through aws.amazon.com.</td>
</tr>
<tr>
<td>catalog.redhat.com</td>
<td>443</td>
<td>The <a href="https://registry.access.redhat.com">registry.access.redhat.com</a> and <a href="https://registry.redhat.io">https://registry.redhat.io</a> sites redirect through catalog.redhat.com.</td>
</tr>
<tr>
<td>time-a-g.nist.gov</td>
<td>123</td>
<td>Allows NTP traffic for FedRAMP. [2]</td>
</tr>
</tbody>
</table>

1. In a firewall environment, ensure that the **access.redhat.com** resource is on the allowlist. This resource hosts a signature store that a container client requires for verifying images when pulling them from [registry.access.redhat.com](https://registry.access.redhat.com).

2. Both TCP and UDP ports.

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:
<table>
<thead>
<tr>
<th>Domain</th>
<th>Port</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>cert-api.access.redhat.com</td>
<td>443</td>
<td>Required for telemetry.</td>
</tr>
<tr>
<td>api.access.redhat.com</td>
<td>443</td>
<td>Required for telemetry.</td>
</tr>
<tr>
<td>infogw.api.openshift.com</td>
<td>443</td>
<td>Required for telemetry.</td>
</tr>
<tr>
<td>console.redhat.com</td>
<td>443</td>
<td>Required for telemetry and Red Hat Insights.</td>
</tr>
<tr>
<td>cloud.redhat.com/api/ingress</td>
<td>443</td>
<td>Required for telemetry and Red Hat Insights.</td>
</tr>
<tr>
<td>observatorium-mst.api.openshift.com</td>
<td>443</td>
<td>Required for managed OpenShift-specific telemetry.</td>
</tr>
<tr>
<td>observatorium.api.openshift.com</td>
<td>443</td>
<td>Required for managed OpenShift-specific telemetry.</td>
</tr>
</tbody>
</table>

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. For more information about how remote health monitoring data is used by Red Hat, see About remote health monitoring in the Additional resources section.

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

<table>
<thead>
<tr>
<th>Domain</th>
<th>Port</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>.amazonaws.com</td>
<td>443</td>
<td>Required to access AWS services and resources.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Domain</th>
<th>Port</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ec2.amazonaws.com</td>
<td>443</td>
<td>Used to install and manage clusters in an AWS environment.</td>
</tr>
<tr>
<td>events. &lt;aws_region&gt;.amazonaws.com</td>
<td>443</td>
<td>Used to install and manage clusters in an AWS environment.</td>
</tr>
<tr>
<td>iam.amazonaws.com</td>
<td>443</td>
<td>Used to install and manage clusters in an AWS environment.</td>
</tr>
<tr>
<td>route53.amazonaws.com</td>
<td>443</td>
<td>Used to install and manage clusters in an AWS environment.</td>
</tr>
<tr>
<td>Domain</td>
<td>Port</td>
<td>Function</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sts.amazonaws.com</td>
<td>443</td>
<td>Used to install and manage clusters in an AWS environment, for clusters configured to use the global endpoint for AWS STS.</td>
</tr>
<tr>
<td>sts.&lt;aws_region&gt;.amazonaws.com</td>
<td>443</td>
<td>Used to install and manage clusters in an AWS environment, for clusters configured to use regionalized endpoints for AWS STS. See <a href="#">AWS STS regionalized endpoints</a> for more information.</td>
</tr>
<tr>
<td>tagging.us-east-1.amazonaws.com</td>
<td>443</td>
<td>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.</td>
</tr>
<tr>
<td>ec2.&lt;aws_region&gt;.amazonaws.com</td>
<td>443</td>
<td>Used to install and manage clusters in an AWS environment.</td>
</tr>
<tr>
<td>elasticloadbalancing.&lt;aws_region&gt;.amazonaws.com</td>
<td>443</td>
<td>Used to install and manage clusters in an AWS environment.</td>
</tr>
<tr>
<td>servicequotas.&lt;aws_region&gt;.amazonaws.com</td>
<td>443, 80</td>
<td>Required. Used to confirm quotas for deploying the service.</td>
</tr>
<tr>
<td>tagging.&lt;aws_region&gt;.amazonaws.com</td>
<td>443, 80</td>
<td>Allows the assignment of metadata about AWS resources in the form of tags.</td>
</tr>
</tbody>
</table>

4. Allowlist the following OpenShift URLs:

<table>
<thead>
<tr>
<th>Domain</th>
<th>Port</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>mirror.openshift.com</td>
<td>443</td>
<td>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.</td>
</tr>
<tr>
<td>storage.googleapis.com/openshift-release</td>
<td>443</td>
<td>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.</td>
</tr>
<tr>
<td>api.openshift.com</td>
<td>443</td>
<td>Used to check if updates are available for the cluster.</td>
</tr>
</tbody>
</table>

5. Allowlist the following site reliability engineering (SRE) and management URLs:
<table>
<thead>
<tr>
<th>Domain</th>
<th>Port</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>api.pagerduty.com</td>
<td>443</td>
<td>This alerting service is used by the in-cluster alertmanager to send alerts notifying Red Hat SRE of an event to take action on.</td>
</tr>
<tr>
<td>events.pagerduty.com</td>
<td>443</td>
<td>This alerting service is used by the in-cluster alertmanager to send alerts notifying Red Hat SRE of an event to take action on.</td>
</tr>
<tr>
<td>api.deadmansnitch.com</td>
<td>443</td>
<td>Alerting service used by Red Hat OpenShift Service on AWS to send periodic pings that indicate whether the cluster is available and running.</td>
</tr>
<tr>
<td>nosnch.in</td>
<td>443</td>
<td>Alerting service used by Red Hat OpenShift Service on AWS to send periodic pings that indicate whether the cluster is available and running.</td>
</tr>
<tr>
<td>http-inputs-osdsecuritylogs.splunkcloud.com</td>
<td>443</td>
<td>Required. Used by the <strong>splunk-forwarder-operator</strong> as a logging forwarding endpoint to be used by Red Hat SRE for log-based alerting.</td>
</tr>
<tr>
<td>sftp.access.redhat.com (Recommended)</td>
<td>22</td>
<td>The SFTP server used by <strong>must-gather-operator</strong> to upload diagnostic logs to help troubleshoot issues with the cluster.</td>
</tr>
</tbody>
</table>

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:
The S3 endpoint should be in the following format:

```
$ 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.

**Additional resources**

- About remote health monitoring

**2.6. NEXT STEPS**

- Review the required AWS service quotas

**2.7. ADDITIONAL RESOURCES**

- SRE access to all Red Hat OpenShift Service on AWS clusters
- Configuring custom domains for applications
- Instance types
Red Hat OpenShift Service on AWS (ROSA) web UI requires that you have specific permissions on your AWS account that create a trust relationship to provide the end-user experience at OpenShift Cluster Manager Hybrid Cloud Console and for the `rosa` command line interface (CLI).

This trust relationship is achieved through the creation and association of the `ocm-role` AWS IAM role. This role has a trust policy with the AWS installer that links your Red Hat account to your AWS account. In addition, you also need a `user-role` AWS IAM role for each web UI user, which serves to identify these users. This `user-role` AWS IAM role has no permissions.

The AWS IAM roles required to use OpenShift Cluster Manager are:

- `ocm-role`
- `user-role`

Whether you manage your clusters using the ROSA CLI (`rosa`) or OpenShift Cluster Manager web UI, you must create the account-wide roles, known as `account-roles` in the ROSA CLI, by using the ROSA CLI. These account roles are necessary for your first cluster, and these roles can be used across multiple clusters. These required account roles are:

- Worker-Role
- Support-Role
- Installer-Role
- ControlPlane-Role

**NOTE**

Role creation does not request your AWS access or secret keys. AWS Security Token Service (STS) is used as the basis of this workflow. AWS STS uses temporary, limited-privilege credentials to provide authentication.

For more information about creating these roles, see [Account-wide IAM role and policy reference](#).

Cluster-specific Operator roles, known as `operator-roles` in the ROSA CLI, obtain the temporary permissions required to carry out cluster operations, such as managing back-end storage, ingress, and registry. These roles are required by the cluster that you create. These required Operator roles are:

- `<cluster_name>-<hash>-openshift-cluster-csi-drivers-ebs-cloud-credentials`
- `<cluster_name>-<hash>-openshift-cloud-network-config-controller-credentials`
- `<cluster_name>-<hash>-openshift-machine-api-aws-cloud-credentials`
- `<cluster_name>-<hash>-openshift-cloud-credential-operator-cloud-credentials`
- `<cluster_name>-<hash>-openshift-image-registry-installer-cloud-credentials`
- `<cluster_name>-<hash>-openshift-ingress-operator-cloud-credentials`

For more information on creating these roles, see [Cluster-specific Operator IAM role reference](#).
3.1. ABOUT THE OCM-ROLE IAM RESOURCE

You must create the `ocm-role` IAM resource to enable a Red Hat organization of users to create Red Hat OpenShift Service on AWS (ROSA) clusters. Within the context of linking to AWS, a Red Hat organization is a single user within OpenShift Cluster Manager.

Some considerations for your `ocm-role` IAM resource are:

- Only one `ocm-role` IAM role can be linked per Red Hat organization; however, you can have any number of `ocm-role` IAM roles per AWS account. The web UI requires that only one of these roles can be linked at a time.

- Any user in a Red Hat organization may create and link an `ocm-role` IAM resource.

- Only the Red Hat Organization Administrator can unlink an `ocm-role` IAM resource. This limitation is to protect other Red Hat organization members from disturbing the interface capabilities of other users.

**NOTE**

If you just created a Red Hat account that is not part of an existing organization, this account is also the Red Hat Organization Administrator.

- See "Understanding the OpenShift Cluster Manager role" in the Additional resources of this section for a list of the AWS permissions policies for the basic and admin `ocm-role` IAM resources.

Using the ROSA CLI (`rosa`), you can link your IAM resource when you create it.

**NOTE**

"Linking" or "associating" your IAM resources with your AWS account means creating a trust-policy with your `ocm-role` IAM role and the Red Hat OpenShift Cluster Manager AWS role. After creating and linking your IAM resource, you see a trust relationship from your `ocm-role` IAM resource in AWS with the `arn:aws:iam::7333:role/RH-Managed-OpenShift-Installer` resource.

After a Red Hat Organization Administrator has created and linked an `ocm-role` IAM resource, all organization members may want to create and link their own `user-role` IAM role. This IAM resource only needs to be created and linked only once per user. If another user in your Red Hat organization has already created and linked an `ocm-role` IAM resource, you need to ensure you have created and linked your own `user-role` IAM role.

**Additional resources**

- See Understanding the OpenShift Cluster Manager role

3.1.1. Creating an `ocm-role` IAM role

You create your `ocm-role` IAM roles by using the command-line interface (CLI).

**Prerequisites**

- You have an AWS account.
You have Red Hat Organization Administrator privileges in the OpenShift Cluster Manager organization.

You have the permissions required to install AWS account-wide roles.

You have installed and configured the latest Red Hat OpenShift Service on AWS (ROSA) CLI, `rosa`, on your installation host.

**Procedure**

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

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

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

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

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

**Example output**

```
I: Creating ocm role
? Role prefix: ManagedOpenShift 1
? Enable admin capabilities for the OCM role (optional): No 2
? Permissions boundary ARN (optional): 3
? Role Path (optional): 4
? Role creation mode: auto 5
I: Creating role using 'arn:aws:iam::<ARN>:user/<UserName>'
? Create the 'ManagedOpenShift-OCM-Role-182' role? Yes 6
I: Created role 'ManagedOpenShift-OCM-Role-182' with ARN  'arn:aws:iam::<ARN>:role/ManagedOpenShift-OCM-Role-182'
I: Linking OCM role
? OCM Role ARN: arn:aws:iam::<ARN>:role/ManagedOpenShift-OCM-Role-182 7
? Link the 'arn:aws:iam::<ARN>:role/ManagedOpenShift-OCM-Role-182' role with organization 'AWS ARN'? Yes 8
I: Successfully linked role-arn 'arn:aws:iam::<ARN>:role/ManagedOpenShift-OCM-Role-182' with organization account '<AWS ARN>'
```

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

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

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

4. **NOTE**

   You do not see this prompt if you used the `--admin` option.
4 Specify an IAM path for the user name.

5 Choose the method to create your AWS roles. Using auto, the ROSA CLI generates and links the roles and policies. In the auto mode, you receive some different prompts to create the AWS roles.

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

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

8 Links the created role with your AWS organization.

3.2. ABOUT THE USER-ROLE IAM ROLE

You need to create a user-role IAM role per web UI user to enable those users to create ROSA clusters.

Some considerations for your user-role IAM role are:

- You only need one user-role IAM role per Red Hat user account, but your Red Hat organization can have many of these IAM resources.
- Any user in a Red Hat organization may create and link an user-role IAM role.
- There can be numerous user-role IAM roles per AWS account per Red Hat organization.
- Red Hat uses the user-role IAM role to identify the user. This IAM resource has no AWS account permissions.
- Your AWS account can have multiple user-role IAM roles, but you must link each IAM role to each user in your Red Hat organization. No user can have more than one linked user-role IAM role.

**NOTE**

“Linking” or “associating” your IAM resources with your AWS account means creating a trust-policy with your user-role IAM role and the Red Hat OpenShift Cluster Manager AWS role. After creating and linking this IAM resource, you see a trust relationship from your user-role IAM role in AWS with the `arn:aws:iam::710019948333:role/RH-Managed-OpenShift-Installer` resource.

3.2.1. Creating a user-role IAM role

You can create your user-role IAM roles by using the command-line interface (CLI).

**Prerequisites**

- You have an AWS account.
- You have installed and configured the latest Red Hat OpenShift Service on AWS (ROSA) CLI, rosa, on your installation host.

**Procedure**

- To create a user-role IAM role with basic privileges, run the following command:
This command allows you to create the role by specifying specific attributes. The following example output shows the "auto mode" selected, which lets the ROSA CLI (`rosa`) to create your Operator roles and policies. See "Understanding the auto and manual deployment modes" in the Additional resources for more information.

Example output

```
I: Creating User role
? Role prefix: ManagedOpenShift
? Permissions boundary ARN (optional):
? Role Path (optional):
? Role creation mode: auto
I: Creating ocm user role using 'arn:aws:iam::2066:user'
? Create the 'ManagedOpenShift-User.osdocs-Role' role? Yes
I: Created role 'ManagedOpenShift-User.osdocs-Role' with ARN 'arn:aws:iam::2066:role/ManagedOpenShift-User.osdocs-Role'
I: Linking User role
? User Role ARN: arn:aws:iam::2066:role/ManagedOpenShift-User.osdocs-Role
? Link the 'arn:aws:iam::2066:role/ManagedOpenShift-User.osdocs-Role' role with account '1AGE'? Yes
I: Successfully linked role ARN 'arn:aws:iam::2066:role/ManagedOpenShift-User.osdocs-Role' with account '1AGE'
```

1. A prefix value for all of the created AWS resources. In this example, ManagedOpenShift prepends all of the AWS resources.
2. The Amazon Resource Name (ARN) of the policy to set permission boundaries.
3. Specify an IAM path for the user name.
4. Choose the method to create your AWS roles. Using `auto`, the ROSA CLI generates and links the roles and policies. In the `auto` mode, you receive some different prompts to create the AWS roles.
5. The `auto` method asks if you want to create a specific `user-role` using your prefix.
6. Links the created role with your AWS organization.

**IMPORTANT**

If you unlink or delete your `user-role` IAM role prior to deleting your cluster, an error prevents you from deleting your cluster. You must create or relink this role to proceed with the deletion process. See Repairing a cluster that cannot be deleted for more information.

### 3.3. AWS ACCOUNT ASSOCIATION

Red Hat OpenShift Service on AWS (ROSA) cluster-provisioning tasks require linking `ocm-role` and `user-role` IAM roles to your AWS account using your Amazon Resource Name (ARN).

The `ocm-role` ARN is stored as a label in your Red Hat organization while the `user-role` ARN is stored as a label inside your Red Hat user account. Red Hat uses these ARN labels to confirm that the user is a
valid account holder and that the correct permissions are available to perform the necessary tasks in the AWS account.

### 3.3.1. Linking your AWS account

You can link your AWS account to existing IAM roles by using the Red Hat OpenShift Service on AWS (ROSA) CLI, **rosa**.

**Prerequisites**

- You have an AWS account.
- You are using [OpenShift Cluster Manager Hybrid Cloud Console](https://console.redhat.com) to create clusters.
- You have the permissions required to install AWS account-wide roles. See the "Additional resources" of this section for more information.
- You have installed and configured the latest AWS (**aws**) and ROSA (**rosa**) CLIs on your installation host.
- You have created your **ocm-role** and **user-role** IAM roles, but have not yet linked them to your AWS account. You can check whether your IAM roles are already linked by running the following commands:

  ```bash
  $ rosa list ocm-role
  $ rosa list user-role
  
  If **Yes** is displayed in the **Linked** column for both roles, you have already linked the roles to an AWS account.
  ```

**Procedure**

1. From the CLI, link your **ocm-role** resource to your Red Hat organization by using your Amazon Resource Name (ARN):

   ```bash
   $ rosa link ocm-role --role-arn <arn>
   
   I: Linking OCM role
   ? Link the `<AWS ACCOUNT ID>` role with organization `<ORG ID>`? Yes
   I: Successfully linked role-arn `<AWS ACCOUNT ID>` with organization account `<ORG ID>`
   ```

2. From the CLI, link your **user-role** resource to your Red Hat user account by using your Amazon Resource Name (ARN):

   ```bash
   $ rosa link user-role --role-arn <arn>
   ```

   **Example output**

   ```bash
   I: Successfully linked role-arn `<AWS ACCOUNT ID>` with organization account `<ORG ID>`
   ```
3.3.2. Associating multiple AWS accounts with your Red Hat organization

You can associate multiple AWS accounts with your Red Hat organization. Associating multiple accounts lets you create Red Hat OpenShift Service on AWS (ROSA) clusters on any of the associated AWS accounts from your Red Hat organization.

With this feature, you can create clusters in different AWS regions by using multiple AWS profiles as region-bound environments.

Prerequisites

- You have an AWS account.
- You are using OpenShift Cluster Manager Hybrid Cloud Console to create clusters.
- You have the permissions required to install AWS account-wide roles.
- You have installed and configured the latest AWS (aws) and ROSA (rosa) CLIs on your installation host.
- You have created your ocm-role and user-role IAM roles.

Procedure

To associate an additional AWS account, first create a profile in your local AWS configuration. Then, associate the account with your Red Hat organization by creating the ocm-role, user, and account roles in the additional AWS account.

To create the roles in an additional region, specify the --profile <aws-profile> parameter when running the rosa create commands and replace <aws_profile> with the additional account profile name:

- To specify an AWS account profile when creating an OpenShift Cluster Manager role:

  $ rosa create --profile <aws_profile> ocm-role

- To specify an AWS account profile when creating a user role:

  $ rosa create --profile <aws_profile> user-role

- To specify an AWS account profile when creating the account roles:

  $ rosa create --profile <aws_profile> account-roles

Example output

I: Linking User role
? Link the 'arn:aws:iam::<ARN>:role/ManagedOpenShift-User-Role-125' role with organization '<AWS ID>'? Yes
I: Successfully linked role-arn 'arn:aws:iam::<ARN>:role/ManagedOpenShift-User-Role-125' with organization account '<AWS ID>'
NOTE

If you do not specify a profile, the default AWS profile is used.

3.4. ADDITIONAL RESOURCES

- See Troubleshooting IAM roles
- See Account-wide IAM role and policy reference for a list of IAM roles needed for cluster creation.
CHAPTER 4. LIMITS AND SCALABILITY

This document details the tested cluster maximums for Red Hat OpenShift Service on AWS (ROSA) clusters, along with information about the test environment and configuration used to test the maximums. Information about control plane and infrastructure node sizing and scaling is also provided.

4.1. CLUSTER MAXIMUMS

Consider the following tested object maximums when you plan a Red Hat OpenShift Service on AWS (ROSA) cluster installation. The table specifies the maximum limits for each tested type in a (ROSA) cluster.

These guidelines are based on a cluster of 102 compute (also known as worker) nodes in a multiple availability zone configuration. For smaller clusters, the maximums are lower.

**NOTE**

The OpenShift Container Platform version used in all of the tests is OCP 4.8.0.

<table>
<thead>
<tr>
<th>Maximum type</th>
<th>4.8 tested maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>102</td>
</tr>
<tr>
<td>Number of pods [1]</td>
<td>20,400</td>
</tr>
<tr>
<td>Number of pods per node</td>
<td>250</td>
</tr>
<tr>
<td>Number of pods per core</td>
<td>There is no default value</td>
</tr>
<tr>
<td>Number of namespaces [2]</td>
<td>3,400</td>
</tr>
<tr>
<td>Number of pods per namespace [3]</td>
<td>20,400</td>
</tr>
<tr>
<td>Number of services [4]</td>
<td>10,000</td>
</tr>
<tr>
<td>Number of services per namespace</td>
<td>10,000</td>
</tr>
<tr>
<td>Number of back ends per service</td>
<td>10,000</td>
</tr>
<tr>
<td>Number of deployments per namespace [3]</td>
<td>1,000</td>
</tr>
</tbody>
</table>

1. The pod count displayed here is the number of test pods. The actual number of pods depends on the application’s memory, CPU, and storage requirements.

2. When there are a large number of active projects, etcd can suffer from poor performance if the keyspace grows excessively large and exceeds the space quota. Periodic maintenance of etcd, including defragmentation, is highly recommended to make etcd storage available.
3. There are a number of control loops in the system that must iterate over all objects in a given namespace as a reaction to some changes in state. Having a large number of objects of a type, in a single namespace, can make those loops expensive and slow down processing the state changes. The limit assumes that the system has enough CPU, memory, and disk to satisfy the application requirements.

4. Each service port and each service back end has a corresponding entry in iptables. The number of back ends of a given service impacts the size of the endpoints objects, which then impacts the size of data that is sent throughout the system.

In OpenShift Container Platform 4.8, half of a CPU core (500 millicore) is reserved by the system compared to previous versions of OpenShift Container Platform.

### 4.2. OPENSHIFT CONTAINER PLATFORM TESTING ENVIRONMENT AND CONFIGURATION

The following table lists the OpenShift Container Platform environment and configuration on which the cluster maximums are tested for the AWS cloud platform.

<table>
<thead>
<tr>
<th>Node</th>
<th>Type</th>
<th>vCPU</th>
<th>RAM(GiB)</th>
<th>Disk type</th>
<th>Disk size(GiB)/IOPS</th>
<th>Count</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control plane/etc.d [1]</td>
<td>m5.4xlarge</td>
<td>16</td>
<td>64</td>
<td>gp3</td>
<td>350 / 1,000</td>
<td>3</td>
<td>us-west-2</td>
</tr>
<tr>
<td>Infrastructure nodes [2]</td>
<td>r5.2xlarge</td>
<td>8</td>
<td>64</td>
<td>gp3</td>
<td>300 / 900</td>
<td>3</td>
<td>us-west-2</td>
</tr>
<tr>
<td>Workload [3]</td>
<td>m5.2xlarge</td>
<td>8</td>
<td>32</td>
<td>gp3</td>
<td>350 / 900</td>
<td>3</td>
<td>us-west-2</td>
</tr>
<tr>
<td>Compute nodes</td>
<td>m5.2xlarge</td>
<td>8</td>
<td>32</td>
<td>gp3</td>
<td>350 / 900</td>
<td>102</td>
<td>us-west-2</td>
</tr>
</tbody>
</table>

1. io1 disks are used for control plane/etcd nodes in all versions prior to 4.10.

2. Infrastructure nodes are used to host monitoring components because Prometheus can claim a large amount of memory, depending on usage patterns.

3. Workload nodes are dedicated to run performance and scalability workload generators.

Larger cluster sizes and higher object counts might be reachable. However, the sizing of the infrastructure nodes limits the amount of memory that is available to Prometheus. When creating, modifying, or deleting objects, Prometheus stores the metrics in its memory for roughly 3 hours prior to persisting the metrics on disk. If the rate of creation, modification, or deletion of objects is too high, Prometheus can become overwhelmed and fail due to the lack of memory resources.
4.3. CONTROL PLANE AND INFRASTRUCTURE NODE SIZING AND SCALING

When you install a Red Hat OpenShift Service on AWS (ROSA) cluster, the sizing of the control plane and infrastructure nodes are automatically determined by the compute node count.

If you change the number of compute nodes in your cluster after installation, the Red Hat Site Reliability Engineering (SRE) team scales the control plane and infrastructure nodes as required to maintain cluster stability.

4.3.1. Node sizing during installation

During the installation process, the sizing of the control plane and infrastructure nodes are dynamically calculated. The sizing calculation is based on the number of compute nodes in a cluster.

The following table lists the control plane and infrastructure node sizing that is applied during installation.

<table>
<thead>
<tr>
<th>Number of compute nodes</th>
<th>Control plane size</th>
<th>Infrastructure node size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 25</td>
<td>m5.2xlarge</td>
<td>r5.xlarge</td>
</tr>
<tr>
<td>26 to 100</td>
<td>m5.4xlarge</td>
<td>r5.2xlarge</td>
</tr>
<tr>
<td>101 to 180 [1]</td>
<td>m5.8xlarge</td>
<td>r5.4xlarge</td>
</tr>
</tbody>
</table>

1. The maximum number of compute nodes on ROSA is 180.

4.3.2. Node scaling after installation

If you change the number of compute nodes after installation, the control plane and infrastructure nodes are scaled by the Red Hat Site Reliability Engineering (SRE) team as required. The nodes are scaled to maintain platform stability.

Post-installation scaling requirements for control plane and infrastructure nodes are assessed on a case-by-case basis. Node resource consumption and received alerts are taken into consideration.

Rules for control plane node resizing alerts

The resizing alert is triggered for the control plane nodes in a cluster when the following occurs:

- Control plane nodes sustain over 66% utilization on average in a classic ROSA cluster.

NOTE

The maximum number of compute nodes on ROSA is 180.

Rules for infrastructure node resizing alerts

Resizing alerts are triggered for the infrastructure nodes in a cluster when it has high-sustained CPU or memory utilization. This high-sustained utilization status is:
- Infrastructure nodes sustain over 50% utilization on average in a classic ROSA cluster with a single availability zone using 2 infrastructure nodes.

- Infrastructure nodes sustain over 66% utilization on average in a classic ROSA cluster with multiple availability zones using 3 infrastructure nodes.

**NOTE**

The maximum number of compute nodes on ROSA is 180.

The resizing alerts only appear after sustained periods of high utilization. Short usage spikes, such as a node temporarily going down causing the other node to scale up, do not trigger these alerts.

The SRE team might scale the control plane and infrastructure nodes for additional reasons, for example to manage an increase in resource consumption on the nodes.

When scaling is applied, the customer is notified through a service log entry. For more information about the service log, see [Accessing the service logs for ROSA clusters](#).

### 4.3.3. Sizing considerations for larger clusters

For larger clusters, infrastructure node sizing can become a significant impacting factor to scalability. There are many factors that influence the stated thresholds, including the etcd version or storage data format.

Exceeding these limits does not necessarily mean that the cluster will fail. In most cases, exceeding these numbers results in lower overall performance.

### 4.4. NEXT STEPS

- Planning your environment

### 4.5. ADDITIONAL RESOURCES

- Accessing the service logs for ROSA clusters
CHAPTER 5. PLANNING YOUR ENVIRONMENT

5.1. PLANNING YOUR ENVIRONMENT BASED ON TESTED CLUSTER MAXIMUMS

This document describes how to plan your Red Hat OpenShift Service on AWS environment based on the tested cluster maximums.

Oversubscribing the physical resources on a node affects resource guarantees the Kubernetes scheduler makes during pod placement. Learn what measures you can take to avoid memory swapping.

Some of the tested maximums are stretched only in a single dimension. They will vary when many objects are running on the cluster.

The numbers noted in this documentation are based on Red Hat testing methodology, setup, configuration, and tunings. These numbers can vary based on your own individual setup and environments.

While planning your environment, determine how many pods are expected to fit per node using the following formula:

\[
\text{required pods per cluster} / \text{pods per node} = \text{total number of nodes needed}
\]

The current maximum number of pods per node is 250. However, the number of pods that fit on a node is dependent on the application itself. Consider the application’s memory, CPU, and storage requirements, as described in Planning your environment based on application requirements.

Example scenario

If you want to scope your cluster for 2200 pods per cluster, you would need at least nine nodes, assuming that there are 250 maximum pods per node:

\[
2200 / 250 = 8.8
\]

If you increase the number of nodes to 20, then the pod distribution changes to 110 pods per node:

\[
2200 / 20 = 110
\]

Where:

\[
\text{required pods per cluster} / \text{total number of nodes} = \text{expected pods per node}
\]

5.2. PLANNING YOUR ENVIRONMENT BASED ON APPLICATION REQUIREMENTS

This document describes how to plan your Red Hat OpenShift Service on AWS environment based on your application requirements.

Consider an example application environment:
### Extrapolated requirements

Extrapolated requirements: 550 CPU cores, 450 GB RAM, and 1.4 TB storage.

### Instance size for nodes can be modulated up or down, depending on your preference. Nodes are often resource overcommitted. In this deployment scenario, you can choose to run additional smaller nodes or fewer larger nodes to provide the same amount of resources. Factors such as operational agility and cost-per-instance should be considered.

### Some applications lend themselves well to overcommitted environments, and some do not. Most Java applications and applications that use huge pages are examples of applications that would not allow for overcommitment. That memory can not be used for other applications. In the example above, the environment would be roughly 30 percent overcommitted, a common ratio.

### The application pods can access a service either by using environment variables or DNS. If using environment variables, for each active service the variables are injected by the kubelet when a pod is run on a node. A cluster-aware DNS server watches the Kubernetes API for new services and creates a set of DNS records for each one. If DNS is enabled throughout your cluster, then all pods should automatically be able to resolve services by their DNS name. Service discovery using DNS can be used in case you must go beyond 5000 services. When using environment variables for service discovery, if the argument list exceeds the allowed length after 5000 services in a namespace, then the pods and deployments will start failing.

Disable the service links in the deployment’s service specification file to overcome this:

**Example**

```yaml
Kind: Template
apiVersion: template.openshift.io/v1
metadata:
  name: deploymentConfigTemplate
creationTimestamp: annotations:
```
**description:** This template will create a deploymentConfig with 1 replica, 4 env vars and a service.

**tags:** "

**objects:**
- kind: DeploymentConfig
  apiVersion: apps.openshift.io/v1
  metadata:
    name: deploymentconfig${IDENTIFIER}
  spec:
    template:
      metadata:
        labels:
          name: replicationcontroller${IDENTIFIER}
      spec:
        enableServiceLinks: false
        containers:
          - name: pause${IDENTIFIER}
            image: "${IMAGE}"
            ports:
              - containerPort: 8080
                protocol: TCP
            env:
              - name: ENVVAR1_${IDENTIFIER}
                value: "${ENV_VALUE}"
              - name: ENVVAR2_${IDENTIFIER}
                value: "${ENV_VALUE}"
              - name: ENVVAR3_${IDENTIFIER}
                value: "${ENV_VALUE}"
              - name: ENVVAR4_${IDENTIFIER}
                value: "${ENV_VALUE}"
        resources: {}
        imagePullPolicy: IfNotPresent
        capabilities: {}
        securityContext:
          capabilities: {}
          privileged: false
        restartPolicy: Always
        serviceAccount: "
        replicas: 1
        selector:
          name: replicationcontroller${IDENTIFIER}
        triggers:
          - type: ConfigChange
            strategy:
              type: Rolling
  - kind: Service
    apiVersion: v1
    metadata:
      name: service${IDENTIFIER}
    spec:
      selector:
        name: replicationcontroller${IDENTIFIER}
      ports:
        - name: serviceport${IDENTIFIER}
          protocol: TCP
          port: 80
          targetPort: 8080
The number of application pods that can run in a namespace is dependent on the number of services and the length of the service name when the environment variables are used for service discovery. **ARG_MAX** on the system defines the maximum argument length for a new process and it is set to 2097152 bytes (2 MiB) by default. The kubelet injects environment variables in to each pod scheduled to run in the namespace including:

- `<SERVICE_NAME>_SERVICE_HOST=<IP>`
- `<SERVICE_NAME>_SERVICE_PORT=<PORT>`
- `<SERVICE_NAME>_PORT=tcp://<IP>:<PORT>`
- `<SERVICE_NAME>_PORT_<PORT>_TCP=tcp://<IP>:<PORT>`
- `<SERVICE_NAME>_PORT_<PORT>_TCP_PROTO=tcp`
- `<SERVICE_NAME>_PORT_<PORT>_TCP_ADDR=<ADDR>`

The pods in the namespace start to fail if the argument length exceeds the allowed value and if the number of characters in a service name impacts it.
CHAPTER 6. 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.

6.1. REQUIRED AWS SERVICE QUOTAS

The table below describes the AWS service quotas and levels required to create and run one Red Hat OpenShift Service on AWS cluster. Although most default values are suitable for most workloads, you might need to request additional quota for the following cases:

- ROSA clusters require at least 100 vCPUs, but the default maximum value for vCPUs assigned to Running On-Demand Standard Amazon EC2 instances is 5. Therefore if you have not created a ROSA cluster using the same AWS account previously, you must request additional EC2 quota for Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances.

- Some optional cluster configuration features, such as custom security groups, might require you to request additional quota. For example, because ROSA associates 1 security group with network interfaces in worker machine pools by default, and the default quota for Security groups per network interface is 5, if you want to add 5 custom security groups, you must request additional quota, because this would bring the total number of security groups on worker network interfaces to 6.

**NOTE**

The AWS SDK allows ROSA to check quotas, but the AWS SDK calculation does not account for 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. Large quota requests are submitted to Amazon Support for review, and take some time to be approved. If your quota request is urgent, contact AWS Support.

Table 6.1. ROSA-required service quota

<table>
<thead>
<tr>
<th>Quota name</th>
<th>Service code</th>
<th>Quota code</th>
<th>AWS default</th>
<th>Minimum required</th>
<th>Description</th>
</tr>
</thead>
</table>


<table>
<thead>
<tr>
<th>Quota name</th>
<th>Service code</th>
<th>Quota code</th>
<th>AWS default</th>
<th>Minimum required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances</td>
<td>ec2</td>
<td>L-1216C47A</td>
<td>5</td>
<td>100</td>
<td>Maximum number of vCPUs assigned to the Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances. The default value of 5 vCPUs is not sufficient to create ROSA clusters. ROSA has a minimum requirement of 100 vCPUs for cluster creation.</td>
</tr>
<tr>
<td>Storage for General Purpose SSD (gp2) volume storage in TiB</td>
<td>ebs</td>
<td>L-D18FCD1D</td>
<td>50</td>
<td>300</td>
<td>The maximum aggregated amount of storage, in TiB, that can be provisioned across General Purpose SSD (gp2) volumes in this Region.</td>
</tr>
<tr>
<td>Storage for General Purpose SSD (gp3) volume storage in TiB</td>
<td>ebs</td>
<td>L-7A658B76</td>
<td>50</td>
<td>300</td>
<td>The maximum aggregated amount of storage, in TiB, that can be provisioned across General Purpose SSD (gp3) volumes in this Region. 300 TiB of storage is the required minimum for optimal performance.</td>
</tr>
</tbody>
</table>

300 TiB of storage is the required minimum for optimal performance.
<table>
<thead>
<tr>
<th>Quota name</th>
<th>Service code</th>
<th>Quota code</th>
<th>AWS default</th>
<th>Minimum required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage for Provisioned IOPS SSD (io1) volumes in TiB</td>
<td>ebs</td>
<td>L-FD252861</td>
<td>50</td>
<td>300</td>
<td>The maximum aggregated amount of storage, in TiB, that can be provisioned across Provisioned IOPS SSD (io1) volumes in this Region. 300 TiB of storage is the required minimum for optimal performance.</td>
</tr>
</tbody>
</table>

Table 6.2. General AWS service quotas

<table>
<thead>
<tr>
<th>Quota name</th>
<th>Service code</th>
<th>Quota code</th>
<th>AWS default</th>
<th>Minimum required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC2-VPC Elastic IPs</td>
<td>ec2</td>
<td>L-0263D0A3</td>
<td>5</td>
<td>5</td>
<td>The maximum number of Elastic IP addresses that you can allocate for EC2-VPC in this Region.</td>
</tr>
<tr>
<td>VPCs per Region</td>
<td>vpc</td>
<td>L-F678F1CE</td>
<td>5</td>
<td>5</td>
<td>The maximum number of VPCs per Region. This quota is directly tied to the maximum number of internet gateways per Region.</td>
</tr>
<tr>
<td>Quota name</td>
<td>Service code</td>
<td>Quota code</td>
<td>AWS default</td>
<td>Minimum required</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Internet gateways per Region</td>
<td>vpc</td>
<td>L-A4707A72</td>
<td>5</td>
<td>5</td>
<td>The maximum number of internet gateways per Region. This quota is directly tied to the maximum number of VPCs per Region. To increase this quota, increase the number of VPCs per Region.</td>
</tr>
<tr>
<td>Network interfaces per Region</td>
<td>vpc</td>
<td>L-DF5E4CA3</td>
<td>5,000</td>
<td>5,000</td>
<td>The maximum number of network interfaces per Region.</td>
</tr>
<tr>
<td>Security groups per network interface</td>
<td>vpc</td>
<td>L-2AFB9258</td>
<td>5</td>
<td>5</td>
<td>The maximum number of security groups per network interface. This quota, multiplied by the quota for rules per security group, cannot exceed 1000.</td>
</tr>
<tr>
<td>Snapshots per Region</td>
<td>ebs</td>
<td>L-309BACF6</td>
<td>10,000</td>
<td>10,000</td>
<td>The maximum number of snapshots per Region</td>
</tr>
<tr>
<td>Quota name</td>
<td>Service code</td>
<td>Quota code</td>
<td>AWS default</td>
<td>Minimum required</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>------------</td>
<td>-------------</td>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IOPS for Provisioned IOPS SSD (io1) volumes</td>
<td>ebs</td>
<td>L-B3A130E6</td>
<td>300,000</td>
<td>300,000</td>
<td>The maximum aggregated number of IOPS that can be provisioned across Provisioned IOPS SSD (io1) volumes in this Region.</td>
</tr>
<tr>
<td>Application Load Balancers per Region</td>
<td>elasticloadbalancing</td>
<td>L-53DA6B97</td>
<td>50</td>
<td>50</td>
<td>The maximum number of Application Load Balancers that can exist in each region.</td>
</tr>
<tr>
<td>Classic Load Balancers per Region</td>
<td>elasticloadbalancing</td>
<td>L-E9E9831D</td>
<td>20</td>
<td>20</td>
<td>The maximum number of Classic Load Balancers that can exist in each region.</td>
</tr>
</tbody>
</table>

### 6.1.1. Additional resources

- How can I request, view, and manage service quota increase requests using AWS CLI commands?
- ROSA service quotas
- Request a quota increase

### 6.2. NEXT STEPS

- Set up the environment and install ROSA
CHAPTER 7. SETTING UP THE ENVIRONMENT FOR USING STS

After you meet the AWS prerequisites, set up your environment and install Red Hat OpenShift Service on AWS (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.

7.1. SETTING UP THE ENVIRONMENT FOR STS

Before you create a Red Hat OpenShift Service on AWS (ROSA) cluster that uses the AWS Security Token Service (STS), complete the following steps to set up your environment.

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. It is recommended to use a dedicated AWS account to run production clusters. If you are using AWS Organizations, you can use an AWS account within your organization or create a new one.

   If you are using AWS Organizations and you need to have a service control policy (SCP) applied to the AWS account you plan to use, these policies must not be more restrictive than the roles and policies required by the cluster.

2. Enable the ROSA service in the AWS Management 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

   You can use the environment variable to set the default AWS region.

   The ROSA service evaluates regions in the following priority order:
The region specified when running the `rosa` command with the `--region` flag.

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

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 the `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:

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

4. Install the latest version of the ROSA CLI (`rosa`).

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

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

   ```bash
   $ chmod +x rosa
   ```

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

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

   d. Enter the following command to verify your installation:

   ```bash
   $ rosa
   ```

**Example output**

Command line tool for Red Hat OpenShift Service on AWS.
For further documentation visit https://access.redhat.com/documentation/en-us/red_hat_openshift_service_on_aws

Usage:
`rosa [command]`

Available Commands:
- `completion` Generates completion scripts
- `create` Create a resource from stdin
- `delete` Delete a specific resource
- `describe` Show details of a specific resource
- `download` Download necessary tools for using your cluster
- `edit` Edit a specific resource
- `grant` Grant role to a specific resource
e. Generate the command completion scripts for the ROSA CLI. The following example generates the Bash completion scripts for a Linux machine:

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

f. Source the scripts to enable `rosa` command completion from your existing terminal. The following example sources the Bash completion scripts for `rosa` on a Linux machine:

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

5. Log in to your Red Hat account with the ROSA CLI.
   
   a. Enter the following command.

   ```bash
   $ 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**

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

6. Verify that your AWS account has the necessary quota to deploy a ROSA cluster.
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 the AWS Management Console and request a quota increase for the service that failed.

After the quota check succeeds, proceed to the next step.

7. 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 OpenShift Cluster Manager for now.

   ```
   $ rosa whoami
   ```

   **Example output**

   ```
   AWS Account ID:               000000000000
   AWS Default Region:           us-east-1
   AWS ARN:                      arn:aws:iam::000000000000:user/hello
   OCM API:                      https://api.openshift.com
   OCM Account ID:               1DzGldIhqEWyt8UXQhSoWaaaaa
   OCM Account Name:             Your Name
   OCM Account Username:         you@domain.com
   OCM Account Email:            you@domain.com
   OCM Organization ID:          1HopHfA2hmhup5gCr2uH5aaaaa
   OCM Organization Name:        Red Hat
   OCM Organization External ID: 0000000
   ```

   8. Install the OpenShift CLI (`oc`), version 4.7.9 or greater, from the ROSA (`rosa`) CLI.
      a. Enter this command to download the latest version of the `oc` CLI:

      ```
      $ rosa download openshift-client
      ```

      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 openshift-client
      ```
After completing these steps, you are ready to set up IAM and OIDC access-based roles.

### 7.2. NEXT STEPS

- Create a ROSA cluster with STS quickly or create a cluster using customizations.

### 7.3. ADDITIONAL RESOURCES

- AWS Prerequisites
- Required AWS service quotas and increase requests
CHAPTER 8. PREPARING TERRAFORM TO INSTALL ROSA CLUSTERS

Terraform is an infrastructure-as-code tool that provides a way to configure your resources once and replicate those resources as desired. Terraform accomplishes the creation tasks by using declarative language. You declare what you want the final state of the infrastructure resource to be, and Terraform creates these resources to your specifications.

8.1. PREREQUISITES FOR TERRAFORM

To use the Red Hat Cloud Services provider inside your Terraform configuration, you must meet the following prerequisites:

- You have installed the Red Hat OpenShift Service on AWS (ROSA) command-line interface (CLI) tool. See the Additional resources for further installation instructions.

- You have your offline Red Hat OpenShift Cluster Manager token. This token is generated through the Red Hat Hybrid Cloud Console. It is unique to your account and should not be shared. The token is generated based off your account access and permissions.

- You have installed Terraform version 1.4.6 or newer. You must have Terraform configured for your local system. The Terraform website contains installation options for MacOS, Windows, and Linux.

- You have an AWS account and associated credentials that allow you to create resources. The credentials are configured for the AWS provider. See the Authentication and Configuration section in AWS Terraform provider documentation.

- You have, at minimum, the following permissions in your AWS IAM role policy that is operating Terraform. Check for these permissions in the AWS console.

Example 8.1. Minimum AWS permissions for Terraform

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "iam:GetPolicyVersion",
                "iam:DeletePolicyVersion",
                "iam:CreatePolicyVersion",
                "iam:UpdateAssumeRolePolicy",
                "secretsmanager:DescribeSecret",
                "iam:ListRoleTags",
                "secretsmanager:PutSecretValue",
                "secretsmanager:CreateSecret",
                "iam:TagRole",
                "secretsmanager:DeleteSecret",
                "iam:UpdateOpenIDConnectProviderThumbprint",
                "iam:DeletePolicy",
                "iam:CreateRole",
            ]
        }
    ]
}
```
Additional resources

- See Prerequisites checklist for deploying ROSA using STS for a list of requirements that must be met before you can create ROSA clusters by using STS.

- See About IAM resources for ROSA clusters that use STS for information about the AWS account roles.

- See Getting started with the ROSA CLI for information about installing the ROSA CLI.

- See Hashicorp’s Terraform documentation for a comprehensive guide to Terraform.
8.2. ACCOUNT ROLES TERRAFORM EXAMPLE

The following example shows how Terraform can be used to create your Amazon Web Services (AWS) Identity and Access Management (IAM) account roles for ROSA.

**NOTE**

If you want to edit the Terraform files, you can use any text editor. You must re-run the `terraform init` and `terraform apply` commands if you change any values in the files.

Procedure

1. Check your AWS account for existing roles and policies by running the following command:

   `$ rosa list account-roles`

2. In your terminal, run the following command to export your Red Hat OpenShift Cluster Manager token. This value must include the full OpenShift Cluster Manager token:

   `$ export RHCS_TOKEN="<your_offline_token>"`

   You can verify that your token is saved by running the following command:

   `$ echo $RHCS_TOKEN`

   You see your token in the command line.

3. Optional: You can specify your own account-role prefix that prepends the roles you create by running the following command:

   **NOTE**

   If you do not specify an account-role prefix, a prefix is generated in the format of `account-role-` followed by a string of four random characters.

   `$ export TF_VAR_account_role_prefix=<account_role_prefix>`

4. Create the Terraform files locally by using the following code templates:

   **NOTE**

   These files are created in your current directory. Ensure that you are in the directory where you want to run Terraform.

   a. The `main.tf` file calls the Red Hat Cloud Services Terraform provider, which allows you to use OpenShift services with Terraform. Run the following command to create the `main.tf` file:

   `$ cat<<-EOF>main.tf
   #`
b. You define the account role prefix structure in the `output.tf` file. This output definition allows you to specify how the various generated roles are constructed. Run the following command to create your `output.tf` file:

```bash
$ cat<<EOF>output.tf
output "account_role_prefix" {
```

The variables.tf allows you to specify values you want for select variables. If you exported a variable for the account_role_prefix earlier, leave this variable’s default value blank. Setting the variable in both places with different values can produce unexpected results.

Run the following command to create your variables.tf file:

```bash
$ cat<<-EOF>variables.tf

variable "openshift_version" {
    type = string
    default = "4.13"
    description = "Enter the desired OpenShift version as X.Y. This version should match what you intend for your ROSA cluster. For example, if you plan to create a ROSA cluster using '4.13.10', then this version should be '4.13'. You can see the supported versions of OpenShift by running 'rosa list version'."
}

variable "account_role_prefix" {
    type = string
    default = ""
    description = "Your account roles are prepended with whatever value you enter here. The default value in the ROSA CLI is 'ManagedOpenshift-' before all of your account roles."
}

variable "tags" {
    type = map
    default = {}  
    description = "(Optional) List of AWS resource tags to apply."
}

EOF
```

The tags parameter uses a map of strings variable. The format that it takes looks like the following example:

```bash
variable "tags" {
    type = "map"
    default = {
        "us-east-1" = "image-1234"
        "us-west-2" = "image-4567"
    }
}
```

5. In the directory where you saved these Terraform files, run the following command to set up Terraform to create these resources:
Optional: Run the following command to confirm that the Terraform code you copied is correct:

```bash
$ terraform validate
```

**Sample output**

Success! The configuration is valid.

7. Optional: Test your Terraform template and create a reusable Terraform plan file by running the following command:

```bash
$ terraform plan -out account-roles.tfplan
```

8. Run the following command to build your account-wide IAM roles with Terraform:

```bash
$ terraform apply "account-roles.tfplan"
```

**NOTE**

If you used the `terraform plan` command first, you can provide your created `account-roles.tf` file here. Otherwise, Terraform temporarily creates this plan before it applies your desired outcome.

**Verification**

- Run the following command to verify that your account-roles have been created:

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

**Sample output**

```
I: Fetching account roles
ROLE NAME                    ROLE TYPE      ROLE ARN
OPENSSHIFT VERSION  AWS Managed
account-role-6kn4-ControlPlane-Role  Control plane
arn:aws:iam::269733383066:role/account-role-6kn4-ControlPlane-Role  4.13               No
account-role-6kn4-Installer-Role   Installer      arn:aws:iam::269733383066:role/account-role-6kn4-Installer-Role  4.13               No
account-role-6kn4-Support-Role    Support        arn:aws:iam::269733383066:role/account-role-6kn4-Support-Role  4.13               No
account-role-6kn4-Worker-Role     Worker         arn:aws:iam::269733383066:role/account-role-6kn4-Worker-Role  4.13               No
```

**Clean up**

When you are finished using the resources that you created using Terraform, you should purge these resources with the following command:

```bash
$ terraform destroy
```
8.3. NEXT STEPS

- Planning your environment

8.4. ADDITIONAL RESOURCES

- Accessing the service logs for ROSA clusters