Red Hat OpenShift Service on AWS 4

Prepare your environment

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

Last Updated: 2022-08-29
Planning, limits, and scalability for Red Hat OpenShift Service on AWS
Legal Notice

Copyright © 2022 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at http://creativecommons.org/licenses/by-sa/3.0/. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation’s permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document provides planning considerations for Red Hat OpenShift Service on AWS (ROSA) cluster deployments, including information about cluster limits and scalability.
# Table of Contents

## CHAPTER 1. AWS PREREQUISITES FOR ROSA WITH STS ................................. 4
  1.1. DEPLOYMENT PREREQUISITES 4
  1.2. CUSTOMER REQUIREMENTS WHEN USING STS FOR DEPLOYMENT 4
    1.2.1. Account 4
    1.2.2. AWS account association 5
    1.2.3. Linking your AWS account 5
    1.2.4. Associating multiple AWS accounts with your Red Hat organization 6
    1.2.5. Access requirements 7
    1.2.6. Support requirements 7
    1.2.7. Security requirements 8
  1.3. REQUIREMENTS FOR DEPLOYING A CLUSTER IN AN OPT-IN REGION 8
    1.3.1. Setting the AWS security token version 8
  1.4. RED HAT MANAGED IAM REFERENCES FOR AWS 9
  1.5. PROVISIONED AWS INFRASTRUCTURE 9
    1.5.1. EC2 instances 9
    1.5.2. AWS Elastic Block Store (EBS) storage 10
    1.5.3. Elastic load balancers 10
    1.5.4. S3 storage 10
    1.5.5. VPC
      1.5.5.1. Sample VPC Architecture 11
      1.5.6. Security groups 11
  1.6. AWS FIREWALL PREREQUISITES 12
  1.7. NEXT STEPS 16
  1.8. ADDITIONAL RESOURCES 16

## CHAPTER 2. OPENSHIFT CLUSTER MANAGER IAM ROLE RESOURCES ..................... 17
  2.1. ABOUT THE OCM-ROLE IAM RESOURCE 17
    2.1.1. Creating an OpenShift Cluster Manager IAM role 18
  2.2. ABOUT THE USER-ROLE IAM ROLE 19
    2.2.1. Creating an user-role IAM role 20
  2.3. AWS ACCOUNT ASSOCIATION
    2.3.1. Linking your AWS account 21
    2.3.2. Associating multiple AWS accounts with your Red Hat organization 22
  2.4. ADDITIONAL RESOURCES 22

## CHAPTER 3. LIMITS AND SCALABILITY .......................................................... 24
  3.1. ROSA TESTED CLUSTER MAXIMUMS 24
  3.2. OPENSHIFT CONTAINER PLATFORM TESTING ENVIRONMENT AND CONFIGURATION 25
  3.3. CONTROL PLANE AND INFRASTRUCTURE NODE SIZING AND SCALING 26
    3.3.1. Node sizing during installation 26
    3.3.2. Node scaling after installation 26
    3.3.3. Sizing considerations for larger clusters 27
  3.4. NEXT STEPS 27
  3.5. ADDITIONAL RESOURCES 27

## CHAPTER 4. PLANNING YOUR ENVIRONMENT .................................................... 28
  4.1. PLANNING YOUR ENVIRONMENT BASED ON TESTED CLUSTER MAXIMUMS 28
  4.2. PLANNING YOUR ENVIRONMENT BASED ON APPLICATION REQUIREMENTS 28

## CHAPTER 5. REQUIRED AWS SERVICE QUOTAS ............................................... 32
  5.1. REQUIRED AWS SERVICE QUOTAS 32
  5.2. NEXT STEPS 33
CHAPTER 1. AWS PREREQUISITES FOR ROSA WITH 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.

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.

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.

1.2. 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).

1.2.1. Account

- You must ensure that the AWS limits are sufficient to support Red Hat OpenShift Service on AWS provisioned within your AWS account. Running `rosa verify quota` 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 balancers (ELBs) to be configured. See the "Creating the service role for the elastic load balancer (ELB)" 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 service role for the elastic load balancer (ELB)

1.2.2. AWS account association

Red Hat OpenShift Service on AWS (ROSA) cluster-provisioning tasks require linking ocm-role and user-role OpenShift Cluster Manager IAM resources 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.

1.2.3. Linking your AWS account

You link your AWS account using the rosa CLI.

Prerequisites

- You have an AWS account.
- You are using OpenShift Cluster Manager 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**

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>'
   
   Example output
   ```

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>
   
   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>'
   
   Example output
   ```

### 1.2.4. 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 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:
  
  ```bash
  $ rosa create --profile <aws_profile> ocm-role
  ```

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

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

  **NOTE**

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

### 1.2.5. 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 cluster.
- Actions available in the rosa CLI utility or OpenShift Cluster Manager 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

### 1.2.6. 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.

1.2.7. 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

1.3. 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.

1.3.1. 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.

### 1.4. 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.

### 1.5. PROVISIONED IAM 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.

#### 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.
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

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.

1.5.4. 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.

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.

### 1.5.5.1. Sample VPC Architecture

![Sample VPC Architecture Diagram]

### 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.

<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>
## 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:

<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>cm-quay-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</td>
<td>443</td>
<td>Provides access to the odo CLI tool that helps developers build on OpenShift and Kubernetes.</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>
</tbody>
</table>

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:

<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>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. 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:

<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.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>sts.amazonaws.com</td>
<td>443</td>
<td>Used to install and manage clusters in an AWS environment.</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;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>Domain</td>
<td>Port</td>
<td>Function</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------</td>
<td>--------------------------------------------------------------------------</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.deadmanssnitch.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>
</tbody>
</table>
The SFTP server used by `must-gather-operator` to upload diagnostic logs to help troubleshoot issues with the cluster.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Port</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sftp.access.redhat.com</code></td>
<td>22</td>
<td>The SFTP server used by <code>must-gather-operator</code> 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:

```
$ 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.

### 1.7. NEXT STEPS

- Review the required AWS service quotas

### 1.8. ADDITIONAL RESOURCES

- SRE access to all Red Hat OpenShift Service on AWS clusters
- Configuring custom domains for applications
CHAPTER 2. OPENShift CLUSTER MANAGER IAM ROLE RESOURCES

Red Hat OpenShift Service on AWS (ROSA) web UI needs you to have some permissions on your AWS account that create a trust relationship to provide the end-user experience at OpenShift Cluster Manager and for the rosa command line interface (CLI).

This trust relationship is achieved through the creation and association of two AWS IAM roles:

1. `ocm-role`
2. `user-role`

If you use the `rosa` CLI, the `rosa` tool creates a number of these required permissions for you. This creation is available since your user account authenticates for both Red Hat and AWS. In the OpenShift Cluster Manager web UI, you need to create these roles.

**NOTE**

Role creation does not request your AWS access or secret keys. This is because it uses an AWS Secure Token Service (STS) as the basis of its workflow.

2.1. ABOUT THE OCM-ROLE IAM RESOURCE

You must create the `ocm-role` IAM resource to enable a Red Hat organization of users to create ROSA clusters.

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

- You have only one `ocm-role` per AWS account in a Red Hat organization. You can have multiple `ocm-role` IAM roles in a Red Hat organization as long as these roles are for different AWS accounts.
- 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.

- Only one `ocm-role` IAM resource can be created per AWS account per Red Hat organization.
- See the associated permission tables for a list of the AWS permissions policy for the basic and admin `ocm-role` IAM resources.

Using the `rosa` CLI, you can link your IAM resource when you create it.
"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::<ARN>: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.

2.1.1. Creating an OpenShift Cluster Manager IAM role

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

Prerequisites

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

Procedure

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

  $ rosa create ocm-role

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

  $ rosa create ocm-role --admin

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

Example output

I: Creating ocm role
? Role prefix: ManagedOpenShift 1
? Enable admin capabilities for the OCM role (optional): No 2
? Permissions boundary ARN (optional): 3
? Role creation mode: auto 4
I: Creating role using 'arn:aws:iam::<ARN>:user/<UserName>'
? Create the 'ManagedOpenShift-OCM-Role-182' role? Yes 5
A prefix value for all of the created AWS resources. In this example, ManagedOpenShift prepends all of the AWS resources.

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

**NOTE**

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

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

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

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

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

Links the created role with your AWS organization.

### 2.2. ABOUT THE USER-ROLE IAM ROLE

You need to create the `user-role` IAM role to enable a Red Hat organization of 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.

**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::<ARN>:role/RH-Managed-OpenShift-Installer` resource.
2.2.1. Creating an user-role IAM role

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

Prerequisites

- You have an AWS account.
- You have installed and configured the latest AWS (aws) and ROSA (rosa) CLIs on your installation host.

Procedure

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

  $ rosa create user-role

  This command allows you create the role by specifying specific attributes. The following example output shows the “auto mode” selected, which lets the rosa CLI to create your Operator roles and policies. See “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 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. Choose the method of how to create your AWS roles. Using auto, the rosa CLI tool generates and links the role to your AWS account. In the auto mode, you receive some different prompts to create the AWS roles.
4. The auto method asks if you want to create a specific user-role using your prefix.
5. Links the created role with your AWS organization.

2.3. AWS ACCOUNT ASSOCIATION
Red Hat OpenShift Service on AWS (ROSA) cluster-provisioning tasks require linking `ocm-role` and `user-role` OpenShift Cluster Manager IAM resources 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.

### 2.3.1. Linking your AWS account

You link your AWS account using the `rosa` CLI.

#### Prerequisites

- You have an AWS account.
- You are using OpenShift Cluster Manager 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

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>
   ```

   **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
   I: Linking OCM role
   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: Linking User role
   ```
2.3.2. 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 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
  ```

**NOTE**

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

2.4. ADDITIONAL RESOURCES
• See Troubleshooting IAM roles
CHAPTER 3. 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.

3.1. ROSA TESTED 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 3.1. Tested cluster maximums

<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.

3.2. 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/etcd</td>
<td>m5.4xlarge</td>
<td>16</td>
<td>64</td>
<td>io1</td>
<td>350 / 1,000</td>
<td>3</td>
<td>us-west-2</td>
</tr>
<tr>
<td>Infrastructure nodes</td>
<td>r5.2xlarge</td>
<td>8</td>
<td>64</td>
<td>gp2</td>
<td>300 / 900</td>
<td>3</td>
<td>us-west-2</td>
</tr>
<tr>
<td>Workload</td>
<td>m5.2xlarge</td>
<td>8</td>
<td>32</td>
<td>gp2</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>gp2</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 because etcd is I/O intensive and latency sensitive. A greater number of IOPS can be required, depending on usage.

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.
3.3. 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.

3.3.1. 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.

3.3.2. 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

Resizing alerts are triggered for the control plane nodes in a cluster when either of the following scenarios are true:

- Each control plane node has less than 16GiB RAM, and there are more than 25 and less than 101 compute nodes.
- Each control plane node has less than 32GiB RAM, and there are more than 100 compute nodes.

**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 either of the following scenarios are true:

- Each infrastructure node has less than 16GiB RAM or less than 5 CPUs, and there are more than 25 and less than 101 compute nodes.
- Each infrastructure node has less than 32GiB RAM or less than 9 CPUs, and there are more than 100 compute nodes.

**NOTE**

The maximum number of compute nodes on ROSA is 180.

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*.

### 3.3.3. 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.

### 3.4. NEXT STEPS

- Planning your environment

### 3.5. ADDITIONAL RESOURCES

- Accessing the service logs for ROSA clusters
CHAPTER 4. PLANNING YOUR ENVIRONMENT

4.1. 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:

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

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:

\[
\frac{2200}{250} = 8.8
\]

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

\[
\frac{2200}{20} = 110
\]

Where:

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

4.2. 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:
<table>
<thead>
<tr>
<th>Pod type</th>
<th>Pod quantity</th>
<th>Max memory</th>
<th>CPU cores</th>
<th>Persistent storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>apache</td>
<td>100</td>
<td>500 MB</td>
<td>0.5</td>
<td>1 GB</td>
</tr>
<tr>
<td>node.js</td>
<td>200</td>
<td>1 GB</td>
<td>1</td>
<td>1 GB</td>
</tr>
<tr>
<td>postgresql</td>
<td>100</td>
<td>1 GB</td>
<td>2</td>
<td>10 GB</td>
</tr>
<tr>
<td>JBoss EAP</td>
<td>100</td>
<td>1 GB</td>
<td>1</td>
<td>1 GB</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Node type</th>
<th>Quantity</th>
<th>CPUs</th>
<th>RAM (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes (option 1)</td>
<td>100</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Nodes (option 2)</td>
<td>50</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>Nodes (option 3)</td>
<td>25</td>
<td>16</td>
<td>64</td>
</tr>
</tbody>
</table>

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 cannot 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:
```
This template will create a deploymentConfig with 1 replica, 4 env vars and a service.

```yaml
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 KiB 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_PORT=<PORT>`
- `<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 5. 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.

5.1. REQUIRED AWS SERVICE QUOTAS

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

NOTE

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

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

<table>
<thead>
<tr>
<th>Quota name</th>
<th>Service code</th>
<th>Quota code</th>
<th>Minimum required value</th>
<th>Recommended value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of EIPs - VPC EIPs</td>
<td>ec2</td>
<td>L-0263D0A3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances</td>
<td>ec2</td>
<td>L-1216C47A</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>VPCs per Region</td>
<td>vpc</td>
<td>L-F678F1CE</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Internet gateways per Region</td>
<td>vpc</td>
<td>L-A4707A72</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Network interfaces per Region</td>
<td>vpc</td>
<td>L-DF5E4CA3</td>
<td>5,000</td>
<td>5,000</td>
</tr>
<tr>
<td>General Purpose SSD (gp2) volume storage</td>
<td>ebs</td>
<td>L-D18FCD1D</td>
<td>50</td>
<td>300</td>
</tr>
<tr>
<td>Number of EBS snapshots</td>
<td>ebs</td>
<td>L-309BACF6</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Provisioned IOPS</td>
<td>ebs</td>
<td>L-B3A130E6</td>
<td>300,000</td>
<td>300,000</td>
</tr>
<tr>
<td>Provisioned IOPS SSD (io1) volume storage</td>
<td>ebs</td>
<td>L-FD252861</td>
<td>50</td>
<td>300</td>
</tr>
<tr>
<td>Quota name</td>
<td>Service code</td>
<td>Quota code</td>
<td>Minimum required value</td>
<td>Recommended value</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>------------------</td>
<td>----------------</td>
<td>------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Application Load Balancers per Region</td>
<td>elasticloadbalancing</td>
<td>L-53DA6B97</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Classic Load Balancers per Region</td>
<td>elasticloadbalancing</td>
<td>L-E9E983ID</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

5.2. NEXT STEPS

- Set up the environment and install ROSA
CHAPTER 6. 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.

6.1. 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:
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
   ```

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.

   ```
   $ chmod +x rosa
   ```

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

   ```
   $ 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
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. 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:

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

5. Log in to your Red Hat account with the `rosa` CLI.

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'

6. Verify that your AWS account has the necessary quota to deploy a ROSA cluster.

```
$ rosa verify quota [--region=<region>]
```

**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 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:               1DzGldlhqEWyt8UUXQhSoWaaaaa
      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. Enter this command to download the latest version of the oc CLI:
      
      $ rosa download openshift-client

   c. Enter this command to verify that the oc CLI is installed correctly:
      
      $ rosa verify openshift-client

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

Create roles

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

6.2. NEXT STEPS

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

6.3. ADDITIONAL RESOURCES
• AWS Prerequisites

• Required AWS service quotas and increase requests