Red Hat OpenShift Service on AWS 4

Prepare your environment

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

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.
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. Access requirements ...................................................................................... 5
    1.2.3. Support requirements ..................................................................................... 5
    1.2.4. Security requirements .................................................................................... 5
  1.3. REQUIREMENTS FOR DEPLOYING A CLUSTER IN AN OPT-IN REGION ............... 5
    1.3.1. Setting the AWS security token version ......................................................... 6
  1.4. RED HAT MANAGED IAM REFERENCES FOR AWS ........................................... 6
  1.5. PROVISIONED AWS INFRASTRUCTURE ............................................................... 7
    1.5.1. EC2 instances .................................................................................................. 7
    1.5.2. AWS Elastic Block Store (EBS) storage ......................................................... 7
    1.5.3. Elastic load balancers ..................................................................................... 7
    1.5.4. S3 storage ........................................................................................................ 8
    1.5.5. VPC .................................................................................................................. 8
      1.5.5.1. Sample VPC Architecture ....................................................................... 8
    1.5.6. Security groups ............................................................................................... 8
  1.6. NEXT STEPS ......................................................................................................... 9
  1.7. ADDITIONAL RESOURCES .................................................................................. 9

CHAPTER 2. AWS PREREQUISITES FOR ROSA ............................................................... 10
  2.1. DEPLOYMENT PREREQUISITES ............................................................................. 10
  2.2. CUSTOMER REQUIREMENTS ................................................................................. 10
    2.2.1. Account ........................................................................................................... 10
    2.2.2. Access requirements ...................................................................................... 11
    2.2.3. Support requirements ..................................................................................... 11
    2.2.4. Security requirements .................................................................................... 11
  2.3. REQUIRED CUSTOMER PROCEDURE .................................................................. 12
    2.3.1. Minimum required service control policy (SCP) .......................................... 12
  2.4. RED HAT MANAGED IAM REFERENCES FOR AWS ......................................... 16
    2.4.1. IAM Policies ................................................................................................... 16
    2.4.2. IAM users ....................................................................................................... 16
  2.5. PROVISIONED AWS INFRASTRUCTURE .............................................................. 16
    2.5.1. EC2 instances .................................................................................................. 17
    2.5.2. AWS Elastic Block Store (EBS) storage ......................................................... 17
    2.5.3. Elastic load balancers ..................................................................................... 17
    2.5.4. S3 storage ........................................................................................................ 17
    2.5.5. VPC .................................................................................................................. 18
      2.5.5.1. Sample VPC Architecture ....................................................................... 18
    2.5.6. Security groups ............................................................................................... 18
  2.6. FIREWALL PREREQUISITES ................................................................................. 19
  2.7. NEXT STEPS ......................................................................................................... 23
  2.8. ADDITIONAL RESOURCES .................................................................................. 23

CHAPTER 3. LIMITS AND SCALABILITY ......................................................................... 25
  3.1. INITIAL PLANNING CONSIDERATIONS ............................................................... 25
  3.2. ROSA TESTED CLUSTER MAXIMUMS ................................................................. 25
  3.3. OPENSHIFT CONTAINER PLATFORM TESTING ENVIRONMENT AND CONFIGURATION 26

CHAPTER 4. PLANNING YOUR ENVIRONMENT ............................................................... 28
  4.1. PLANNING YOUR ENVIRONMENT BASED ON TESTED CLUSTER MAXIMUMS ...... 28
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.

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

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

- The customer ensures that the AWS limits are sufficient to support Red Hat OpenShift Service on AWS provisioned within the customer’s AWS account.
- If SCP policies are applied and enforced, these policies must not be more restrictive than the roles and policies required by the cluster.
- The customer’s AWS account should not be transferable to Red Hat.
- The customer 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.
- The customer may deploy native AWS services within the same AWS account.
The account must have a service-linked role set up as it is required for elastic load balancers (ELBs) to be configured.

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

1.2.2. Access requirements

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

1.2.3. 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.

1.2.4. Security requirements

- Volume snapshots will remain within the customer’s AWS account and customer-specified region.
- Red Hat must have ingress access to EC2 hosts and the API server from allow-listed IP addresses.
- Red Hat must have egress allowed to the documented domains.

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:

   ```bash
   $ 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:

   ```bash
   $ 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:

   ```bash
   $ 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 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.

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

**NOTE**

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

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

---

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. NEXT STEPS

Review the required AWS service quotas

1.7. ADDITIONAL RESOURCES

- See Intial Planning Considerations for guidance on worker node count.

- See SRE access to all Red Hat OpenShift Service on AWS clusters for information about how Red Hat site reliability engineering accesses ROSA clusters.
CHAPTER 2. AWS PREREQUISITES FOR ROSA

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

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

2.1. 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.

2.2. CUSTOMER REQUIREMENTS

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

NOTE

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

2.2.1. Account

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

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

NOTE

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

- The customer’s AWS account should not be transferable to Red Hat.
The customer may not impose AWS usage restrictions on Red Hat activities. Imposing restrictions will severely hinder Red Hat’s ability to respond to incidents.

The customer may deploy native AWS services within the same AWS account.

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

2.2.2. Access requirements

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

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

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

2.2.3. Support requirements

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

2.2.4. Security requirements

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

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

Procedure

1. If you, as the customer, are utilizing AWS Organizations, then you must use an AWS account within your organization or create a new one.

2. To ensure that Red Hat can perform necessary actions, you must either create a service control policy (SCP) or ensure that none is applied to the AWS account.

3. Attach the SCP to the AWS account.

4. Follow the ROSA procedures for setting up the environment.

2.3.1. Minimum required service control policy (SCP)

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

NOTE

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

<table>
<thead>
<tr>
<th>Service</th>
<th>Actions</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Amazon EC2 Auto Scaling</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Amazon S3</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Identity And Access Management</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Elastic Load Balancing</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Elastic Load Balancing V2</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Amazon CloudWatch</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Amazon CloudWatch Events</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>Service</td>
<td>Actions</td>
<td>Effect</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>---------------</td>
<td>--------</td>
</tr>
<tr>
<td>Amazon CloudWatch Logs</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>AWS Support</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>AWS Key Management Service</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>AWS Security Token Service</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>AWS Resource Tagging</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>AWS Route53 DNS</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>AWS Service Quotas</td>
<td>ListServices</td>
<td>Allow</td>
</tr>
<tr>
<td></td>
<td>GetRequestedServiceQuotaChange</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GetServiceQuota</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RequestServiceQuotaIncrease</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ListServiceQuotas</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AWS Billing</td>
<td>ViewAccount</td>
<td>Allow</td>
</tr>
<tr>
<td></td>
<td>Viewbilling</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ViewUsage</td>
<td></td>
</tr>
<tr>
<td>AWS Cost and Usage Report</td>
<td>All</td>
<td>Allow</td>
</tr>
<tr>
<td>AWS Cost Explorer Services</td>
<td>All</td>
<td>Allow</td>
</tr>
</tbody>
</table>

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

"Effect": "Allow",
"Action": [
  "s3:*"
],
"Resource": [
  "*"
]
},

"Effect": "Allow",
"Action": [
  "iam:*"
],
"Resource": [
  "*"
]
},

"Effect": "Allow",
"Action": [
  "elasticloadbalancing:*"
],
"Resource": [
  "*"
]
},

"Effect": "Allow",
"Action": [
  "cloudwatch:*"
],
"Resource": [
  "*"
]
},

"Effect": "Allow",
"Action": [
  "events:*"
],
"Resource": [
  "*"
]
}
{   "Effect": "Allow",   "Action": [     "logs:*"   ],   "Resource": [     "*"   ] }
},
{   "Effect": "Allow",   "Action": [     "support:*"   ],   "Resource": [     "*"   ] }
},
{   "Effect": "Allow",   "Action": [     "kms:*"   ],   "Resource": [     "*"   ] }
},
{   "Effect": "Allow",   "Action": [     "sts:*"   ],   "Resource": [     "*"   ] }
},
{   "Effect": "Allow",   "Action": [     "tag:*"   ],   "Resource": [     "*"   ] }
},
{   "Effect": "Allow",   "Action": [     "route53:*"   ],   "Resource": [     "*"   ] }
}
}
2.4. RED HAT MANAGED IAM REFERENCES FOR AWS

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

2.4.1. IAM Policies

**NOTE**

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

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

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

2.4.2. IAM users

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

2.5. PROVISIONED AWS INFRASTRUCTURE

This is an overview of the provisioned Amazon Web Services (AWS) components on a deployed Red Hat OpenShift Service on AWS (ROSA) cluster. For a more detailed listing of all provisioned AWS components, see the [OpenShift Container Platform documentation](#).
2.5.1. 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.

2.5.2. 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

2.5.3. 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](https://docs.aws.amazon.com/elasticloadbalancing/latest/).

2.5.4. S3 storage

The image registry and Elastic Block Store (EBS) volume snapshots are backed by AWS S3 storage. Pruning of resources is performed regularly to optimize S3 usage and cluster performance.
NOTE
Two buckets are required with a typical size of 2TB each.

2.5.5. 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.

2.5.5.1. Sample VPC Architecture

2.5.6. 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</td>
<td>icmp</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Group</td>
<td>tcp</td>
<td>22</td>
</tr>
</tbody>
</table>
### 2.6. FIREWALL PREREQUISITES

**IMPORTANT**

Only ROSA clusters deployed with PrivateLink may 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>Domain</td>
<td>Port</td>
<td>Function</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>------</td>
<td>--------------------------------------------------------------------------</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>
<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>
</tbody>
</table>

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>
</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 URIs:

<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 wish 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.</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>
</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><strong>mirror.openshift.com</strong></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><strong>storage.googleapis.com/openshift-release</strong> (Recommended)</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><strong>api.openshift.com</strong></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><strong>api.pagerduty.com</strong></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><strong>events.pagerduty.com</strong></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><strong>api.deadmanssnitch.com</strong></td>
<td>443</td>
<td>Alerting service used by OpenShift Dedicated to send periodic pings that indicate whether the cluster is available and running.</td>
</tr>
<tr>
<td><strong>nosnch.in</strong></td>
<td>443</td>
<td>Alerting service used by OpenShift Dedicated to send periodic pings that indicate whether the cluster is available and running.</td>
</tr>
</tbody>
</table>
### Domain | Port | Function
--- | --- | ---
*.osdsecuritylogs.splunkcloud.com OR inputs1.osdsecuritylogs.splunkcloud.com | 999 | Used by the **splunk-forwarder-operator** as a logging forwarding endpoint to be used by Red Hat SRE for log-based alerting.
inputs2.osdsecuritylogs.splunkcloud.com | 7 | 7.
inputs3.osdsecuritylogs.splunkcloud.com |  | OR
http-inputs-osdsecuritylogs.splunkcloud.com | 443 | Required. Used by the **splunk-forwarder-operator** as a logging forwarding endpoint to be used by Red Hat SRE for log-based alerting.
sftp.access.redhat.com (Recommended) | 22 | The SFTP server used by **must-gather-operator** to upload diagnostic logs to help troubleshoot issues with the cluster.

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

```bash
$ 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.

### 2.7. NEXT STEPS

Review the required AWS service quotas

### 2.8. ADDITIONAL RESOURCES
- See *Initial Planning Considerations* for guidance on worker node count.

- See *SRE access to all Red Hat OpenShift Service on AWS clusters* for information about how Red Hat site reliability engineering accesses ROSA clusters.

- *Understanding the ROSA deployment workflow*
CHAPTER 3. LIMITS AND SCALABILITY

3.1. INITIAL PLANNING CONSIDERATIONS

Consider the following tested object maximums when you plan your Red Hat OpenShift Service on AWS cluster.

These guidelines are based on a cluster of 102 workers in a multi-availability zone configuration. For smaller clusters, the maximums are lower.

The sizing of the control plane and infrastructure nodes is dynamically calculated during the installation process, based on the number of worker nodes. If you change the number of worker nodes after the installation, control plane and infra nodes must be resized manually. Infra nodes are resized by the Red Hat SRE team, and you can open a ticket in the Customer Portal to request the infra node resizing.

The following table lists the size of control plane and infrastructure nodes that are assigned during installation.

<table>
<thead>
<tr>
<th>Number of worker 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</td>
<td>m5.8xlarge</td>
<td>r5.4xlarge</td>
</tr>
</tbody>
</table>

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

For larger clusters, infrastructure node sizing can become a large 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.

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

3.2. ROSA TESTED CLUSTER MAXIMUMS

The following table specifies the maximum limits for each tested type in a Red Hat OpenShift Service on AWS cluster.

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>Maximum type</td>
<td>4.8 tested maximum</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>---------------------</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.

### 3.3. 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) / IOS</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>io1</td>
<td>350 / 1,000</td>
<td>3</td>
<td>us-west-2</td>
</tr>
<tr>
<td>Node Type</td>
<td>Node Type</td>
<td>vCPU</td>
<td>RAM(GiB)</td>
<td>Disk type</td>
<td>Disk size(GiB) /IOM</td>
<td>Count</td>
<td>Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
<td>------</td>
<td>----------</td>
<td>-----------</td>
<td>---------------------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>Infrastructure nodes [2]</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 nodes [3]</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>Worker 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.
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:
### Pod Specifications

<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: null
annotations: null
```
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 into 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.