



Red Hat OpenShift Service on AWS 4

ROSA CLI

Learning how to use the command-line tools for Red Hat OpenShift Service on AWS

Red Hat OpenShift Service on AWS 4 ROSA CLI

Learning how to use the command-line tools for Red Hat OpenShift Service on AWS

Legal Notice

Copyright © 2024 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 information about installing, configuring, and using the command-line tools for Red Hat OpenShift Service on AWS (ROSA). It also contains a reference of CLI commands and examples.

Table of Contents

CHAPTER 1. GETTING STARTED WITH THE RED HAT OPENSIFT SERVICE ON AWS (ROSA) CLI, ROSA	4
1.1. ABOUT THE RED HAT OPENSIFT SERVICE ON AWS (ROSA) CLI, ROSA	4
1.2. SETTING UP THE ROSA CLI	4
1.3. CONFIGURING THE ROSA CLI	5
1.3.1. login	5
1.3.2. logout	6
1.3.3. verify permissions	7
1.3.4. verify quota	8
1.3.5. download rosa	8
1.3.6. download oc	9
1.3.7. verify oc	9
1.4. INITIALIZING RED HAT OPENSIFT SERVICE ON AWS	10
1.4.1. init	10
1.5. USING A BASH SCRIPT	11
1.6. UPDATING THE ROSA CLI	12
CHAPTER 2. MANAGING OBJECTS WITH THE ROSA CLI	14
2.1. COMMON COMMANDS AND ARGUMENTS	14
2.1.1. debug	14
2.1.2. download	14
2.1.3. help	14
2.1.4. interactive	14
2.1.5. profile	15
2.1.6. version	15
2.2. PARENT COMMANDS	15
2.2.1. create	15
2.2.2. edit	15
2.2.3. delete	15
2.2.4. list	16
2.2.5. describe	16
2.3. CREATE OBJECTS	16
2.3.1. create account-roles	16
2.3.2. create admin	17
2.3.3. create cluster	18
2.3.4. create idp	21
2.3.5. create ingress	23
2.3.6. create machinepool	24
2.3.7. create ocm-role	26
2.3.8. create user-role	27
2.4. ADDITIONAL RESOURCES	27
2.5. EDIT OBJECTS	28
2.5.1. edit cluster	28
2.5.2. edit ingress	28
2.5.3. edit machinepool	29
2.6. DELETE OBJECTS	31
2.6.1. delete admin	31
2.6.2. delete cluster	32
2.6.3. delete idp	32
2.6.4. delete ingress	33
2.6.5. delete machinepool	34
2.7. INSTALL AND UNINSTALL ADD-ONS	35

2.7.1. install addon	35
2.7.2. uninstall addon	36
2.8. LIST AND DESCRIBE OBJECTS	36
2.8.1. list addon	36
2.8.2. list clusters	37
2.8.3. list idps	37
2.8.4. list ingresses	38
2.8.5. list instance-types	39
2.8.6. list machinepools	39
2.8.7. list regions	40
2.8.8. list upgrades	40
2.8.9. list users	41
2.8.10. list versions	42
2.8.11. describe admin	42
2.8.12. describe addon	43
2.8.13. describe cluster	43
2.9. UPGRADE AND DELETE UPGRADE FOR CLUSTERS	44
2.9.1. upgrade cluster	44
2.9.2. delete upgrade	45
CHAPTER 3. CHECKING ACCOUNT AND VERSION INFORMATION WITH THE ROSA CLI	47
3.1. CHECKING ACCOUNT AND VERSION INFORMATION WITH THE ROSA CLI	47
3.1.1. whoami	47
3.1.2. version	47
CHAPTER 4. CHECKING LOGS WITH THE ROSA CLI	49
4.1. CHECKING LOGS WITH THE ROSA CLI	49
4.1.1. logs install	49
4.1.2. logs uninstall	49

CHAPTER 1. GETTING STARTED WITH THE RED HAT OPENSIFT SERVICE ON AWS (ROSA) CLI, ROSA

Setup and basic usage of the Red Hat OpenShift Service on AWS (ROSA) CLI, **rosa**.

1.1. ABOUT THE RED HAT OPENSIFT SERVICE ON AWS (ROSA) CLI, ROSA

Use the **rosa** command-line utility for Red Hat OpenShift Service on AWS (ROSA) to create, update, manage, and delete Red Hat OpenShift Service on AWS clusters and resources.

1.2. SETTING UP THE ROSA CLI

Use the following steps to install and configure the Red Hat OpenShift Service on AWS (ROSA) CLI, **rosa**, on your installation host.

Procedure

1. Download the latest version of the ROSA CLI (**rosa**) for your operating system from the [Downloads](#) page on OpenShift Cluster Manager.
2. Extract the **rosa** binary file from the downloaded archive. The following example extracts the binary from a Linux tar archive:

```
$ tar xvf rosa-linux.tar.gz
```

3. Add **rosa** to your path. In the following example, the **/usr/local/bin** directory is included in the path of the user:

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

4. Verify if the ROSA CLI is installed correctly by querying the **rosa** version:

```
$ rosa version
```

Example output

```
1.2.15  
Your ROSA CLI is up to date.
```

5. Optional: Enable tab completion for the ROSA CLI. With tab completion enabled, you can press the **Tab** key twice to automatically complete subcommands and receive command suggestions:
 - To enable persistent tab completion for Bash on a Linux host:
 - a. Generate a **rosa** tab completion configuration file for Bash and save it to your **/etc/bash_completion.d/** directory:

```
# rosa completion bash > /etc/bash_completion.d/rosa
```

- b. Open a new terminal to activate the configuration.

- To enable persistent tab completion for Bash on a macOS host:
 - a. Generate a **rosa** tab completion configuration file for Bash and save it to your `/usr/local/etc/bash_completion.d/` directory:

```
$ rosa completion bash > /usr/local/etc/bash_completion.d/rosa
```

- b. Open a new terminal to activate the configuration.
- To enable persistent tab completion for Zsh:
 - a. If tab completion is not enabled for your Zsh environment, enable it by running the following command:

```
$ echo "autoload -U compinit; compinit" >> ~/.zshrc
```

- b. Generate a **rosa** tab completion configuration file for Zsh and save it to the first directory in your functions path:

```
$ rosa completion zsh > "${fpath[1]}/_rosa"
```

- c. Open a new terminal to activate the configuration.
- To enable persistent tab completion for fish:
 - a. Generate a **rosa** tab completion configuration file for fish and save it to your `~/.config/fish/completions/` directory:

```
$ rosa completion fish > ~/.config/fish/completions/rosa.fish
```

- b. Open a new terminal to activate the configuration.
- To enable persistent tab completion for PowerShell:
 - a. Generate a **rosa** tab completion configuration file for PowerShell and save it to a file named **rosa.ps1**:

```
PS> rosa completion powershell | Out-String | Invoke-Expression
```

- b. Source the **rosa.ps1** file from your PowerShell profile.



NOTE

For more information about configuring **rosa** tab completion, see the help menu by running the **rosa completion --help** command.

1.3. CONFIGURING THE ROSA CLI

Use the following commands to configure the Red Hat OpenShift Service on AWS (ROSA) CLI, **rosa**.

1.3.1. login

Log in to your Red Hat account, saving the credentials to the **rosa** configuration file. You must provide a token when logging in. You can copy your token from [the Red Hat OpenShift Service on AWS token page](#).

The ROSA CLI (**rosa**) looks for a token in the following priority order:

1. Command-line arguments
2. The **ROSA_TOKEN** environment variable
3. The **rosa** configuration file
4. Interactively from a command-line prompt

Syntax

```
$ rosa login [arguments]
```

Table 1.1. Arguments

Option	Definition
--client-id	The OpenID client identifier (string). Default: cloud-services
--client-secret	The OpenID client secret (string).
--insecure	Enables insecure communication with the server. This disables verification of TLS certificates and host names.
--scope	The OpenID scope (string). If this option is used, it replaces the default scopes. This can be repeated multiple times to specify multiple scopes. Default: openid
--token	Accesses or refreshes the token (string).
--token-url	The OpenID token URL (string). Default: https://sso.redhat.com/auth/realms/redhat-external/protocol/openid-connect/token

Table 1.2. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

1.3.2. logout

Log out of **rosa**. Logging out also removes the **rosa** configuration file.

Syntax

```
$ rosa logout [arguments]
```

Table 1.3. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

1.3.3. verify permissions

Verify that the AWS permissions required to create a ROSA cluster are configured correctly:

Syntax

```
$ rosa verify permissions [arguments]
```



NOTE

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

Table 1.4. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--region	The AWS region (string) in which to run the command. This value overrides the AWS_REGION environment variable.
--profile	Specifies an AWS profile (string) from your credentials file.

Examples

Verify that the AWS permissions are configured correctly:

```
$ rosa verify permissions
```

Verify that the AWS permissions are configured correctly in a specific region:

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

1.3.4. verify quota

Verifies that AWS quotas are configured correctly for your default region.

Syntax

```
$ rosa verify quota [arguments]
```

Table 1.5. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--region	The AWS region (string) in which to run the command. This value overrides the AWS_REGION environment variable.
--profile	Specifies an AWS profile (string) from your credentials file.

Examples

Verify that the AWS quotas are configured correctly for the default region:

```
$ rosa verify quota
```

Verify that the AWS quotas are configured correctly in a specific region:

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

1.3.5. download rosa

Download the latest compatible version of the ROSA CLI.

After you download **rosa**, extract the contents of the archive and add it to your path. See [Setting up the ROSA CLI](#) for more details.

Syntax

```
$ rosa download rosa [arguments]
```

Table 1.6. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.

Option	Definition
--debug	Enables debug mode.

1.3.6. download oc

Download the latest compatible version of the OpenShift Container Platform CLI (**oc**).

After you download **oc**, you must extract the contents of the archive and add it to your path.

Syntax

```
$ rosa download oc [arguments]
```

Table 1.7. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.

Example

Download **oc** client tools:

```
$ rosa download oc
```

1.3.7. verify oc

Verifies that the OpenShift Container Platform CLI (**oc**) is installed correctly.

Syntax

```
$ rosa verify oc [arguments]
```

Table 1.8. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.

Example

Verify **oc** client tools:

```
$ rosa verify oc
```

1.4. INITIALIZING RED HAT OPENSIFT SERVICE ON AWS

Use the **init** command to initialize Red Hat OpenShift Service on AWS (ROSA) only if you are using non-STS.

1.4.1. init

Perform a series of checks to verify that you are ready to deploy an Red Hat OpenShift Service on AWS cluster.

The list of checks includes the following:

- Checks to see that you have logged in (see **login**)
- Checks that your AWS credentials are valid
- Checks that your AWS permissions are valid (see **verify permissions**)
- Checks that your AWS quota levels are high enough (see **verify quota**)
- Runs a cluster simulation to ensure cluster creation will perform as expected
- Checks that the **osdCcsAdmin** user has been created in your AWS account
- Checks that the OpenShift Container Platform command-line tool is available on your system

Syntax

```
$ rosa init [arguments]
```

Table 1.9. Arguments

Option	Definition
<code>--region</code>	The AWS region (string) in which to verify quota and permissions. This value overrides the AWS_REGION environment variable only when running the init command, but it does not change your AWS CLI configuration.
<code>--delete</code>	Deletes the stack template that is applied to your AWS account during the init command.
<code>--client-id</code>	The OpenID client identifier (string). Default: cloud-services
<code>--client-secret</code>	The OpenID client secret (string).
<code>--insecure</code>	Enables insecure communication with the server. This disables verification of TLS certificates and host names.

Option	Definition
--scope	The OpenID scope (string). If this option is used, it completely replaces the default scopes. This can be repeated multiple times to specify multiple scopes. Default: openid
--token	Accesses or refreshes the token (string).
--token-url	The OpenID token URL (string). Default: https://sso.redhat.com/auth/realms/redhat-external/protocol/openid-connect/token

Table 1.10. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Examples

Configure your AWS account to allow ROSA clusters:

```
$ rosa init
```

Configure a new AWS account using pre-existing OpenShift Cluster Manager credentials:

```
$ rosa init --token=$OFFLINE_ACCESS_TOKEN
```

1.5. USING A BASH SCRIPT

This is an example workflow of how to use a Bash script with the Red Hat OpenShift Service on AWS (ROSA) CLI, **rosa**.

Prerequisites

Make sure that AWS credentials are available as one of the following options:

- AWS profile
- Environment variables (**AWS_ACCESS_KEY_ID**, **AWS_SECRET_ACCESS_KEY**)

Procedure

1. Initialize **rosa** using an Red Hat OpenShift Cluster Manager offline token [from Red Hat](#):

```
$ rosa init --token=<token>
```

2. Create the Red Hat OpenShift Service on AWS (ROSA) cluster:

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

3. Add an identity provider (IDP):

```
$ rosa create idp --cluster=<cluster_name> --type=<identity_provider> [arguments]
```

4. Add a **dedicated-admin** user:

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

1.6. UPDATING THE ROSA CLI

Update to the latest compatible version of the Red Hat OpenShift Service on AWS (ROSA) CLI, **rosa**.

Procedure

1. Confirm that a new version of the ROSA CLI (**rosa**) is available:

```
$ rosa version
```

Example output

```
1.2.12  
There is a newer release version '1.2.15', please consider updating:  
https://mirror.openshift.com/pub/openshift-v4/clients/rosa/latest/
```

2. Download the latest compatible version of the ROSA CLI:

```
$ rosa download rosa
```

This command downloads an archive called **rosa-*.tar.gz** into the current directory. The exact name of the file depends on your operating system and system architecture.

3. Extract the contents of the archive:

```
$ tar -xzf rosa-linux.tar.gz
```

4. Install the new version of the ROSA CLI by moving the extracted file into your path. In the following example, the **/usr/local/bin** directory is included in the path of the user:

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

Verification

- Verify that the new version of ROSA is installed.

```
$ rosa version
```

Example output

1.2.15

Your ROSA CLI is up to date.

CHAPTER 2. MANAGING OBJECTS WITH THE ROSA CLI

Managing objects with the Red Hat OpenShift Service on AWS (ROSA) CLI, **rosa**, such as adding **dedicated-admin** users, managing clusters, and scheduling cluster upgrades.

2.1. COMMON COMMANDS AND ARGUMENTS

These common commands and arguments are available for the Red Hat OpenShift Service on AWS (ROSA) CLI, **rosa**.

2.1.1. debug

Enables debug mode for the parent command to help with troubleshooting.

Example

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

2.1.2. download

Downloads the latest compatible version of the specified software to the current directory in an archive file. Extract the contents of the archive and add the contents to your path to use the software. To download the latest Red Hat OpenShift Service on AWS CLI, specify **rosa**. To download the latest OpenShift CLI, specify **oc**.

Example

```
$ rosa download <software>
```

2.1.3. help

Displays general help information for the ROSA CLI (**rosa**) and a list of available commands. This option can also be used as an argument to display help information for a parent command, such as **version** or **create**.

Examples

Displays general help for the ROSA CLI.

```
$ rosa --help
```

Displays general help for **version**.

```
$ rosa version --help
```

2.1.4. interactive

Enables interactive mode.

Example

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

-

2.1.5. profile

Specifies an AWS profile from your credential file.

Example

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

2.1.6. version

Displays the **rosa** version and checks whether a newer version is available.

Example

```
$ rosa version [arguments]
```

Example output

Displayed when a newer version of the ROSA CLI is available.

```
1.2.12  
There is a newer release version '1.2.15', please consider updating:  
https://mirror.openshift.com/pub/openshift-v4/clients/rosa/latest/
```

2.2. PARENT COMMANDS

The Red Hat OpenShift Service on AWS (ROSA) CLI, **rosa**, uses parent commands with child commands to manage objects. The parent commands are **create**, **edit**, **delete**, **list**, and **describe**. Not all parent commands can be used with all child commands. For more information, see the specific reference topics that describes the child commands.

2.2.1. create

Creates an object or resource when paired with a child command.

Example

```
$ rosa create cluster --cluster-name=mycluster
```

2.2.2. edit

Edits options for an object, such as making a cluster private.

Example

```
$ rosa edit cluster --cluster=mycluster --private
```

2.2.3. delete

Deletes an object or resource when paired with a child command.

Example

```
$ rosa delete ingress --cluster=mycluster
```

2.2.4. list

Lists clusters or resources for a specific cluster.

Example

```
$ rosa list users --cluster=mycluster
```

2.2.5. describe

Shows the details for a cluster.

Example

```
$ rosa describe cluster --cluster=mycluster
```

2.3. CREATE OBJECTS

This section describes the **create** commands for clusters and resources.

2.3.1. create account-roles

Create the required account-wide role and policy resources for your cluster.

Syntax

```
$ rosa create account-roles [flags]
```

Table 2.1. Flags

Option	Definition
--debug	Enable debug mode.
-i, --interactive	Enable interactive mode.
-m, --mode string	How to perform the operation. Valid options are: auto Resource changes will be automatically applied using the current AWS account. manual Commands necessary to modify AWS resources will be output to be run manually.

Option	Definition
<code>--path string</code>	The ARN path for the account-wide roles and policies, including the Operator policies.
<code>--permissions-boundary string</code>	The ARN of the policy that is used to set the permissions boundary for the account roles.
<code>--prefix string</code>	User-defined prefix for all generated AWS resources. The default is ManagedOpenShift .
<code>--profile string</code>	Use a specific AWS profile from your credential file.
<code>-y, --yes</code>	Automatically answer yes to confirm operations.

2.3.2. create admin

Create a cluster administrator with an automatically generated password that can log in to a cluster.

Syntax

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

Table 2.2. Arguments

Option	Definition
<code>--cluster <cluster_name> <cluster_id></code>	Required. The name or ID (string) of the cluster to add to the identity provider (IDP).

Table 2.3. Optional arguments inherited from parent commands

Option	Definition
<code>--help</code>	Shows help for this command.
<code>--debug</code>	Enables debug mode.
<code>--interactive</code>	Enables interactive mode.
<code>--profile string</code>	Specifies an AWS profile from your credentials file.

Example

Create a cluster administrator that can log in to a cluster named **mycluster**.

```
$ rosa create admin --cluster=mycluster
```

2.3.3. create cluster

Create a new cluster.

Syntax

```
$ rosa create cluster --cluster-name=<cluster_name> [arguments]
```

Table 2.4. Arguments

Option	Definition
<code>--cluster-name <cluster_name></code>	Required. The name of the cluster. When used with the create cluster command, this argument is used to set the cluster name and to generate a sub-domain for your cluster on openshiftapps.com . The value for this argument must be unique within your organization.
<code>--compute-machine-type <instance_type></code>	The instance type for compute nodes in the cluster. This determines the amount of memory and vCPU that is allocated to each compute node. For more information on valid instance types, see <i>AWS Instance types in Red Hat OpenShift Service on AWS service definition</i> .
<code>--compute-nodes n</code>	The number of worker nodes to provision per availability zone. Single-zone clusters require at least 2 nodes. Multi-zone clusters require at least 3 nodes. Default: 2 for single-zone clusters; 3 for multi-zone clusters.
<code>--controlplane-iam-role <arn></code>	The Amazon Resource Name (ARN) of the IAM role to attach to control plane instances.
<code>--disable-scp-checks</code>	Indicates whether cloud permission checks are disabled when attempting to install a cluster.
<code>--dry-run</code>	Simulates creating the cluster.
<code>--enable-autoscaling</code>	Enables autoscaling of compute nodes. By default, autoscaling is set to 2 nodes. To set non-default node limits, use this argument with the --min-replicas and --max-replicas arguments.
<code>--host-prefix <subnet></code>	The subnet prefix length to assign to each individual node, as an integer. For example, if host prefix is set to 23 , then each node is assigned a /23 subnet out of the given CIDR.

Option	Definition
<code>--machine-cidr</code> <address_block>	<p>Block of IP addresses (ipNet) used by Red Hat OpenShift Service on AWS while installing the cluster, for example, 10.0.0.0/16.</p> <div style="display: flex; align-items: flex-start;">  <div> <p>IMPORTANT</p> <p>OVN-Kubernetes, the default network provider in Red Hat OpenShift Service on AWS 4.11 and later, uses the 100.64.0.0/16 IP address range internally. If your cluster uses OVN-Kubernetes, do not include the 100.64.0.0/16 IP address range in any other CIDR definitions in your cluster.</p> </div> </div>
<code>--max-replicas</code> <number_of_nodes>	Specifies the maximum number of compute nodes when enabling autoscaling. Default: 2
<code>--min-replicas</code> <number_of_nodes>	Specifies the minimum number of compute nodes when enabling autoscaling. Default: 2
<code>--multi-az</code>	Deploys to multiple data centers.
<code>--operator-roles-prefix</code> <string>	Prefix to use for all IAM roles used by the operators needed in the OpenShift installer. A prefix is generated automatically if you do not specify one.
<code>--pod-cidr</code> <address_block>	<p>Block of IP addresses (ipNet) from which pod IP addresses are allocated, for example, 10.128.0.0/14.</p> <div style="display: flex; align-items: flex-start;">  <div> <p>IMPORTANT</p> <p>OVN-Kubernetes, the default network provider in Red Hat OpenShift Service on AWS 4.11 and later, uses the 100.64.0.0/16 IP address range internally. If your cluster uses OVN-Kubernetes, do not include the 100.64.0.0/16 IP address range in any other CIDR definitions in your cluster.</p> </div> </div>
<code>--private</code>	Restricts primary API endpoint and application routes to direct, private connectivity.
<code>--private-link</code>	Specifies to use AWS PrivateLink to provide private connectivity between VPCs and services. The --subnet-ids argument is required when using --private-link .
<code>--region</code> <region_name>	The name of the AWS region where your worker pool will be located, for example, us-east-1 . This argument overrides the AWS_REGION environment variable.
<code>--role-arn</code> <arn>	The Amazon Resource Name (ARN) of the installer role that OpenShift Cluster Manager uses to create the cluster. This is required if you have not already created account roles.


Option	Definition
<code>--service-cidr</code> <code><address_block></code>	<p>Block of IP addresses (ipNet) for services, for example, 172.30.0.0/16.</p> <div style="display: flex; align-items: flex-start;">  <div> <p>IMPORTANT</p> <p>OVN-Kubernetes, the default network provider in Red Hat OpenShift Service on AWS 4.11 and later, uses the 100.64.0.0/16 IP address range internally. If your cluster uses OVN-Kubernetes, do not include the 100.64.0.0/16 IP address range in any other CIDR definitions in your cluster.</p> </div> </div>
<code>--sts</code> <code>--non-sts</code>	Specifies whether to use AWS Security Token Service (STS) or IAM credentials (non-STs) to deploy your cluster.
<code>--subnet-ids</code> <code><aws_subnet_id></code>	<p>The AWS subnet IDs to use when installing the cluster, for example, subnet-01abc234d5678ef9a. Subnet IDs must be in pairs with one private subnet ID and one public subnet ID per availability zone. Subnets are comma-delimited, for example, --subnet-ids=subnet-1,subnet-2. Leave the value empty for installer-provisioned subnet IDs.</p> <p>When using --private-link, the --subnet-ids argument is required and only one private subnet is allowed per zone.</p>
<code>--support-role-arn</code> string	The Amazon Resource Name (ARN) of the role used by Red Hat Site Reliability Engineers (SREs) to enable access to the cluster account to provide support.
<code>--version</code> string	The version of Red Hat OpenShift Service on AWS that will be used to install the cluster or cluster resources. For cluster use an X.Y.Z format, for example, 4.12.9 . For account-role use an X.Y format, for example, 4.12 .
<code>--worker-iam-role</code> string	The Amazon Resource Name (ARN) of the IAM role that will be attached to compute instances.

Table 2.5. Optional arguments inherited from parent commands

Option	Definition
<code>--help</code>	Shows help for this command.
<code>--debug</code>	Enables debug mode.
<code>--interactive</code>	Enables interactive mode.
<code>--profile</code>	Specifies an AWS profile (string) from your credentials file.

Examples

Create a cluster named **mycluster**.


```
$ rosa create cluster --cluster-name=mycluster
```

Create a cluster with a specific AWS region.

```
$ rosa create cluster --cluster-name=mycluster --region=us-east-2
```

Create a cluster with autoscaling enabled on the default worker machine pool.

```
$ rosa create cluster --cluster-name=mycluster -region=us-east-1 --enable-autoscaling --min-replicas=2 --max-replicas=5
```

2.3.4. create idp

Add an identity provider (IDP) to define how users log in to a cluster.

Syntax

```
$ rosa create idp --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.6. Arguments

Option	Definition
<code>--cluster <cluster_name> <cluster_id></code>	Required. The name or ID of the cluster to which the IDP will be added.
<code>--ca <path_to_file></code>	The path to the PEM-encoded certificate file to use when making requests to the server, for example, /usr/share/cert.pem .
<code>--client-id</code>	The client ID (string) from the registered application.
<code>--client-secret</code>	The client secret (string) from the registered application.
<code>--mapping-method</code>	Specifies how new identities (string) are mapped to users when they log in. Default: claim
<code>--name</code>	The name (string) for the identity provider.
<code>--type</code>	The type (string) of identity provider. Options: github, gitlab, google, ldap, openid

Table 2.7. GitHub arguments

Option	Definition
<code>--hostname</code>	The optional domain (string) to use with a hosted instance of GitHub Enterprise.

Option	Definition
--organizations	Specifies the organizations for login access. Only users that are members of at least one of the listed organizations (string) are allowed to log in.
--teams	Specifies the teams for login access. Only users that are members of at least one of the listed teams (string) are allowed to log in. The format is <org>/<team> .

Table 2.8. GitLab arguments

Option	Definition
--host-url	The host URL (string) of a GitLab provider. Default: https://gitlab.com

Table 2.9. Google arguments

Option	Definition
--hosted-domain	Restricts users to a Google Apps domain (string).

Table 2.10. LDAP arguments

Option	Definition
--bind-dn	The domain name (string) to bind with during the search phase.
--bind-password	The password (string) to bind with during the search phase.
--email-attributes	The list (string) of attributes whose values should be used as the email address.
--id-attributes	The list (string) of attributes whose values should be used as the user ID. Default: dn
--insecure	Does not make TLS connections to the server.
--name-attributes	The list (string) of attributes whose values should be used as the display name. Default: cn
--url	An RFC 2255 URL (string) which specifies the LDAP search parameters to use.
--username-attributes	The list (string) of attributes whose values should be used as the preferred username. Default: uid

Table 2.11. OpenID arguments

Option	Definition
<code>--email-claims</code>	The list (string) of claims to use as the email address.
<code>--extra-scopes</code>	The list (string) of scopes to request, in addition to the openid scope, during the authorization token request.
<code>--issuer-url</code>	The URL (string) that the OpenID provider asserts as the issuer identifier. It must use the HTTPS scheme with no URL query parameters or fragment.
<code>--name-claims</code>	The list (string) of claims to use as the display name.
<code>--username-claims</code>	The list (string) of claims to use as the preferred username when provisioning a user.

Table 2.12. Optional arguments inherited from parent commands

Option	Definition
<code>--help</code>	Shows help for this command.
<code>--debug</code>	Enables debug mode.
<code>--interactive</code>	Enables interactive mode.
<code>--profile</code>	Specifies an AWS profile (string) from your credentials file.

Examples

Add a GitHub identity provider to a cluster named **mycluster**.

```
$ rosa create idp --type=github --cluster=mycluster
```

Add an identity provider following interactive prompts.

```
$ rosa create idp --cluster=mycluster --interactive
```

2.3.5. create ingress

Add an ingress endpoint to enable API access to the cluster.

Syntax

```
$ rosa create ingress --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.13. Arguments

Option	Definition
<code>--cluster <cluster_name> <cluster_id></code>	Required: The name or ID of the cluster to which the ingress will be added.
<code>--label-match</code>	The label match (string) for ingress. The format must be a comma-delimited list of key=value pairs. If no label is specified, all routes are exposed on both routers.
<code>--private</code>	Restricts application route to direct, private connectivity.

Table 2.14. Optional arguments inherited from parent commands

Option	Definition
<code>--help</code>	Shows help for this command.
<code>--debug</code>	Enables debug mode.
<code>--interactive</code>	Enables interactive mode.
<code>--profile</code>	Specifies an AWS profile (string) from your credentials file.

Examples

Add an internal ingress to a cluster named **mycluster**.

```
$ rosa create ingress --private --cluster=mycluster
```

Add a public ingress to a cluster named **mycluster**.

```
$ rosa create ingress --cluster=mycluster
```

Add an ingress with a route selector label match.

```
$ rosa create ingress --cluster=mycluster --label-match=foo=bar,bar=baz
```

2.3.6. create machinepool

Add a machine pool to an existing cluster.

Syntax

```
$ rosa create machinepool --cluster=<cluster_name> | <cluster_id> --replicas=<number> --name=<machinepool_name> [arguments]
```

Table 2.15. Arguments

Option	Definition
<code>--cluster <cluster_name> <cluster_id></code>	Required: The name or ID of the cluster to which the machine pool will be added.
<code>--enable-autoscaling</code>	Enable or disable autoscaling of compute nodes. To enable autoscaling, use this argument with the <code>--min-replicas</code> and <code>--max-replicas</code> arguments. To disable autoscaling, use <code>--enable-autoscaling=false</code> with the <code>--replicas</code> argument.
<code>--instance-type</code>	The instance type (string) that should be used. Default: m5.xlarge
<code>--labels</code>	The labels (string) for the machine pool. The format must be a comma-delimited list of key=value pairs. This list overwrites any modifications made to node labels on an ongoing basis.
<code>--max-replicas</code>	Specifies the maximum number of compute nodes when enabling autoscaling.
<code>--min-replicas</code>	Specifies the minimum number of compute nodes when enabling autoscaling.
<code>--name</code>	Required: The name (string) for the machine pool.
<code>--replicas</code>	Required when autoscaling is not configured. The number (integer) of machines for this machine pool.
<code>--taints</code>	Taints for the machine pool. This string value should be formatted as a comma-separated list of key=value:ScheduleType . This list will overwrite any modifications made to Node taints on an ongoing basis.

Table 2.16. Optional arguments inherited from parent commands

Option	Definition
<code>--help</code>	Shows help for this command.
<code>--debug</code>	Enables debug mode.
<code>--interactive</code>	Enables interactive mode.
<code>--profile</code>	Specifies an AWS profile (string) from your credentials file.

Examples

Interactively add a machine pool to a cluster named **mycluster**.

```
$ rosa create machinepool --cluster=mycluster --interactive
```

Add a machine pool that is named **mp-1** to a cluster with autoscaling enabled.

```
$ rosa create machinepool --cluster=mycluster --enable-autoscaling --min-replicas=2 --max-replicas=5 --name=mp-1
```

Add a machine pool that is named **mp-1** with 3 replicas of **m5.xlarge** to a cluster.

```
$ rosa create machinepool --cluster=mycluster --replicas=3 --instance-type=m5.xlarge --name=mp-1
```

Add a machine pool with labels to a cluster.

```
$ rosa create machinepool --cluster=mycluster --replicas=2 --instance-type=r5.2xlarge --labels=foo=bar,bar=baz --name=mp-1
```

2.3.7. create ocm-role

Create the required ocm-role resources for your cluster.

Syntax

```
$ rosa create ocm-role [flags]
```

Table 2.17. Flags

Option	Definition
--admin	Enable admin capabilities for the role.
--debug	Enable debug mode.
-i, --interactive	Enable interactive mode.
-m, --mode string	How to perform the operation. Valid options are: <ul style="list-style-type: none"> ● auto: Resource changes will be automatically applied using the current AWS account ● manual: Commands necessary to modify AWS resources will be output to be run manually
--path string	The ARN path for the OCM role and policies.
--permissions-boundary string	The ARN of the policy that is used to set the permissions boundary for the OCM role.
--prefix string	User-defined prefix for all generated AWS resources. The default is ManagedOpenShift .
--profile string	Use a specific AWS profile from your credential file.

Option	Definition
-y, --yes	Automatically answer yes to confirm operation.

For more information about the OCM role created with the **rosa create ocm-role** command, see *Account-wide IAM role and policy reference*.

2.3.8. create user-role

Create the required user-role resources for your cluster.

Syntax

```
$ rosa create user-role [flags]
```

Table 2.18. Flags

Option	Definition
--debug	Enable debug mode.
-i, --interactive	Enable interactive mode.
-m, --mode string	How to perform the operation. Valid options are: <ul style="list-style-type: none"> ● auto: Resource changes will be automatically applied using the current AWS account ● manual: Commands necessary to modify AWS resources will be output to be run manually
--path string	The ARN path for the user role and policies.
--permissions-boundary string	The ARN of the policy that is used to set the permissions boundary for the user role.
--prefix string	User-defined prefix for all generated AWS resources The default is ManagedOpenShift .
--profile string	Use a specific AWS profile from your credential file.
-y, --yes	Automatically answer yes to confirm operation.

For more information about the user role created with the **rosa create user-role** command, see *Understanding AWS account association*.

2.4. ADDITIONAL RESOURCES

- See [AWS Instance types](#) for a list of supported instance types.

- See [Account-wide IAM role and policy reference](#) for a list of IAM roles needed for cluster creation.
- See [Understanding AWS account association](#) for more information about the OCM role and user role.

2.5. EDIT OBJECTS

This section describes the **edit** commands for clusters and resources.

2.5.1. edit cluster

Allows edits to an existing cluster.

Syntax

```
$ rosa edit cluster --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.19. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster to edit.
--private	Restricts a primary API endpoint to direct, private connectivity.

Table 2.20. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--interactive	Enables interactive mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Examples

Edit a cluster named **mycluster** to make it private.

```
$ rosa edit cluster --cluster=mycluster --private
```

Edit all cluster options interactively on a cluster named **mycluster**.

```
$ rosa edit cluster --cluster=mycluster --interactive
```

2.5.2. edit ingress

Edits the additional non-default application router for a cluster.

Syntax

```
$ rosa edit ingress --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.21. Arguments

Option	Definition
<code>--cluster</code>	Required: The name or ID (string) of the cluster to which the ingress will be added.
<code>--label-match</code>	The label match (string) for ingress. The format must be a comma-delimited list of key=value pairs. If no label is specified, all routes are exposed on both routers.
<code>--private</code>	Restricts the application route to direct, private connectivity.

Table 2.22. Optional arguments inherited from parent commands

Option	Definition
<code>--help</code>	Shows help for this command.
<code>--debug</code>	Enables debug mode.
<code>--interactive</code>	Enables interactive mode.
<code>--profile</code>	Specifies an AWS profile (string) from your credentials file.

Examples

Make an additional ingress with the ID **a1b2** as a private connection on a cluster named **mycluster**.

```
$ rosa edit ingress --private --cluster=mycluster a1b2
```

Update the router selectors for the additional ingress with the ID **a1b2** on a cluster named **mycluster**.

```
$ rosa edit ingress --label-match=foo=bar --cluster=mycluster a1b2
```

Update the default ingress using the sub-domain identifier **apps** on a cluster named **mycluster**.

```
$ rosa edit ingress --private=false --cluster=mycluster apps
```

2.5.3. edit machinepool

Allows edits to the machine pool in a cluster.

Syntax

```
$ rosa edit machinepool --cluster=<cluster_name> | <cluster_id> <machinepool_ID> [arguments]
```

Table 2.23. Arguments

Option	Definition
<code>--cluster</code>	Required: The name or ID (string) of the cluster to edit on which the additional machine pool will be edited.
<code>--enable-autoscaling</code>	Enable or disable autoscaling of compute nodes. To enable autoscaling, use this argument with the --min-replicas and --max-replicas arguments. To disable autoscaling, use --enable-autoscaling=false with the --replicas argument.
<code>--labels</code>	The labels (string) for the machine pool. The format must be a comma-delimited list of key=value pairs. Editing this value only affects newly created nodes of the machine pool, which are created by increasing the node number, and does not affect the existing nodes. This list overwrites any modifications made to node labels on an ongoing basis.
<code>--max-replicas</code>	Specifies the maximum number of compute nodes when enabling autoscaling.
<code>--min-replicas</code>	Specifies the minimum number of compute nodes when enabling autoscaling.
<code>--replicas</code>	Required when autoscaling is not configured. The number (integer) of machines for this machine pool.
<code>--taints</code>	Taints for the machine pool. This string value should be formatted as a comma-separated list of key=value:ScheduleType . Editing this value only affect newly created nodes of the machine pool, which are created by increasing the node number, and does not affect the existing nodes. This list overwrites any modifications made to Node taints on an ongoing basis.

Table 2.24. Optional arguments inherited from parent commands

Option	Definition
<code>--help</code>	Shows help for this command.
<code>--debug</code>	Enables debug mode.
<code>--interactive</code>	Enables interactive mode.
<code>--profile</code>	Specifies an AWS profile (string) from your credentials file.

Examples

Set 4 replicas on a machine pool named **mp1** on a cluster named **mycluster**.

```
$ rosa edit machinepool --cluster=mycluster --replicas=4 --name=mp1
```

Enable autoscaling on a machine pool named **mp1** on a cluster named **mycluster**.

```
$ rosa edit machinepool --cluster=mycluster --enable-autoscaling --min-replicas=3 --max-replicas=5 --name=mp1
```

Disable autoscaling on a machine pool named **mp1** on a cluster named **mycluster**.

```
$ rosa edit machinepool --cluster=mycluster --enable-autoscaling=false --replicas=3 --name=mp1
```

Modify the autoscaling range on a machine pool named **mp1** on a cluster named **mycluster**.

```
$ rosa edit machinepool --max-replicas=9 --cluster=mycluster --name=mp1
```

2.6. DELETE OBJECTS

This section describes the **delete** commands for clusters and resources.

2.6.1. delete admin

Deletes a cluster administrator from a specified cluster.

Syntax

```
$ rosa delete admin --cluster=<cluster_name> | <cluster_id>
```

Table 2.25. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster to add to the identity provider (IDP).

Table 2.26. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--interactive	Enables interactive mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Example

Delete a cluster administrator from a cluster named **mycluster**.

```
$ rosa delete admin --cluster=mycluster
```

2.6.2. delete cluster

Deletes a cluster.

Syntax

```
$ rosa delete cluster --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.27. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster to delete.
--watch	Watches the cluster uninstallation logs.

Table 2.28. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--interactive	Enables interactive mode.
--profile	Specifies an AWS profile (string) from your credentials file.
--yes	Automatically answers yes to confirm the operation.

Examples

Delete a cluster named **mycluster**.

```
$ rosa delete cluster --cluster=mycluster
```

2.6.3. delete idp

Deletes a specific identity provider (IDP) from a cluster.

Syntax

```
$ rosa delete idp --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.29. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster from which the IDP will be deleted.

Table 2.30. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--interactive	Enables interactive mode.
--profile	Specifies an AWS profile (string) from your credentials file.
--yes	Automatically answers yes to confirm the operation.

Example

Delete an identity provider named **github** from a cluster named **mycluster**.

```
$ rosa delete idp github --cluster=mycluster
```

2.6.4. delete ingress

Deletes a non-default application router (ingress) from a cluster.

Syntax

```
$ rosa delete ingress --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.31. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster from which the ingress will be deleted.

Table 2.32. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.

Option	Definition
--debug	Enables debug mode.
--interactive	Enables interactive mode.
--profile	Specifies an AWS profile (string) from your credentials file.
--yes	Automatically answers yes to confirm the operation.

Examples

Delete an ingress with the ID **a1b2** from a cluster named **mycluster**.

```
$ rosa delete ingress --cluster=mycluster a1b2
```

Delete a secondary ingress with the subdomain name **apps2** from a cluster named **mycluster**.

```
$ rosa delete ingress --cluster=mycluster apps2
```

2.6.5. delete machinepool

Deletes a machine pool from a cluster.

Syntax

```
$ rosa delete machinepool --cluster=<cluster_name> | <cluster_id> <machine_pool_id>
```

Table 2.33. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster that the machine pool will be deleted from.

Table 2.34. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--interactive	Enables interactive mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Option	Definition
<code>--yes</code>	Automatically answers yes to confirm the operation.

Example

Delete the machine pool with the ID **mp-1** from a cluster named **mycluster**.

```
$ rosa delete machinepool --cluster=mycluster mp-1
```

2.7. INSTALL AND UNINSTALL ADD-ONS

This section describes how to install and uninstall Red Hat managed service add-ons to a cluster.

2.7.1. install addon

Installs a managed service add-on on a cluster.

Syntax

```
$ rosa install addon --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.35. Arguments

Option	Definition
<code>--cluster</code>	Required: The name or ID (string) of the cluster where the add-on will be installed.

Table 2.36. Optional arguments inherited from parent commands

Option	Definition
<code>--help</code>	Shows help for this command.
<code>--debug</code>	Enables debug mode.
<code>--profile</code>	Uses a specific AWS profile (string) from your credentials file.
<code>--yes</code>	Automatically answers yes to confirm the operation.

Example

Add the **dbaas-operator** add-on installation to a cluster named **mycluster**.

```
$ rosa install addon --cluster=mycluster dbaas-operator
```

2.7.2. uninstall addon

Uninstalls a managed service add-on from a cluster.

Syntax

```
$ rosa uninstall addon --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.37. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster that the add-on will be uninstalled from.

Table 2.38. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Uses a specific AWS profile (string) from your credentials file.
--yes	Automatically answers yes to confirm the operation.

Example

Remove the **dbaas-operator** add-on installation from a cluster named **mycluster**.

```
$ rosa uninstall addon --cluster=mycluster dbaas-operator
```

2.8. LIST AND DESCRIBE OBJECTS

This section describes the **list** and **describe** commands for clusters and resources.

2.8.1. list addon

List the managed service add-on installations.

Syntax

```
$ rosa list addons --cluster=<cluster_name> | <cluster_id>
```

Table 2.39. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster to list the add-ons for.

Table 2.40. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

2.8.2. list clusters

List all of your clusters.

Syntax

```
$ rosa list clusters [arguments]
```

Table 2.41. Arguments

Option	Definition
--count	The number (integer) of clusters to display. Default: 100

Table 2.42. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

2.8.3. list idps

List all of the identity providers (IDPs) for a cluster.

Syntax

```
$ rosa list idps --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.43. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster that the IDPs will be listed for.

Table 2.44. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Example

List all identity providers (IDPs) for a cluster named **mycluster**.

```
$ rosa list idps --cluster=mycluster
```

2.8.4. list ingresses

List all of the API and ingress endpoints for a cluster.

Syntax

```
$ rosa list ingresses --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.45. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster that the IDPs will be listed for.

Table 2.46. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Example

List all API and ingress endpoints for a cluster named **mycluster**.

```
$ rosa list ingresses --cluster=mycluster
```

2.8.5. list instance-types

List all of the available instance types for use with Red Hat OpenShift Service on AWS. Availability is based on the account's AWS quota.

Syntax

```
$ rosa list instance-types [arguments]
```

Table 2.47. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--output	The output format. Allowed formats are json or yaml .
--profile	Specifies an AWS profile (string) from your credentials file.

Example

List all instance types.

```
$ rosa list instance-types
```

2.8.6. list machinepools

List the machine pools configured on a cluster.

Syntax

```
$ rosa list machinepools --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.48. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster that the machine pools will be listed for.

Table 2.49. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Example

List all of the machine pools on a cluster named **mycluster**.

```
$ rosa list machinepools --cluster=mycluster
```

2.8.7. list regions

List all of the available regions for the current AWS account.

Syntax

```
$ rosa list regions [arguments]
```

Table 2.50. Arguments

Option	Definition
--multi-az	Lists regions that provide support for multiple availability zones.

Table 2.51. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Example

List all of the available regions.

```
$ rosa list regions
```

2.8.8. list upgrades

List all available and scheduled cluster version upgrades.

Syntax

```
$ rosa list upgrades --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.52. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster that the available upgrades will be listed for.

Table 2.53. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Example

List all of the available upgrades for a cluster named **mycluster**.

```
$ rosa list upgrades --cluster=mycluster
```

2.8.9. list users

List the cluster administrator and dedicated administrator users for a specified cluster.

Syntax

```
$ rosa list users --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.54. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster that the cluster administrators will be listed for.

Table 2.55. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.

Option	Definition
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Example

List all of the cluster administrators and dedicated administrators for a cluster named **mycluster**.

```
$ rosa list users --cluster=mycluster
```

2.8.10. list versions

List all of the OpenShift versions that are available for creating a cluster.

Syntax

```
$ rosa list versions [arguments]
```

Table 2.56. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Example

List all of the OpenShift Container Platform versions.

```
$ rosa list versions
```

2.8.11. describe admin

Show the details of a specified **cluster-admin** user and a command to log in to the cluster.

Syntax

```
$ rosa describe admin --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.57. Arguments

Option	Definition
<code>--cluster</code>	Required: The name or ID (string) of the cluster to which the cluster-admin belongs.

Table 2.58. Optional arguments inherited from parent commands

Option	Definition
<code>--help</code>	Shows help for this command.
<code>--debug</code>	Enables debug mode.
<code>--profile</code>	Specifies an AWS profile (string) from your credentials file.

Example

Describe the **cluster-admin** user for a cluster named **mycluster**.

```
$ rosa describe admin --cluster=mycluster
```

2.8.12. describe addon

Show the details of a managed service add-on.

Syntax

```
$ rosa describe addon <addon_id> | <addon_name> [arguments]
```

Table 2.59. Optional arguments inherited from parent commands

Option	Definition
<code>--help</code>	Shows help for this command.
<code>--debug</code>	Enables debug mode.
<code>--profile</code>	Specifies an AWS profile (string) from your credentials file.

Example

Describe an add-on named **dbaas-operator**.

```
$ rosa describe addon dbaas-operator
```

2.8.13. describe cluster

Shows the details for a cluster.

Syntax

```
$ rosa describe cluster --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.60. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster.

Table 2.61. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Example

Describe a cluster named **mycluster**.

```
$ rosa describe cluster --cluster=mycluster
```

2.9. UPGRADE AND DELETE UPGRADE FOR CLUSTERS

This section describes the **upgrade** command usage for clusters.

2.9.1. upgrade cluster

Schedule a cluster upgrade.

Syntax

```
$ rosa upgrade cluster --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 2.62. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster that the upgrade will be scheduled for.
--interactive	Enables interactive mode.

Option	Definition
<code>--version</code>	The version (string) of OpenShift Container Platform that the cluster will be upgraded to.
<code>--schedule-date</code>	The next date (string) when the upgrade will run at the specified time. Format: yyyy-mm-dd
<code>--schedule-time</code>	The next time the upgrade will run on the specified date. Format: HH:mm
<code>--node-drain-grace-period</code>	Sets a grace period (string) for how long the pod disruption budget-protected workloads are respected during upgrades. After this grace period, any workloads protected by pod disruption budgets that have not been successfully drained from a node will be forcibly evicted. Default: 1 hour

Table 2.63. Optional arguments inherited from parent commands

Option	Definition
<code>--help</code>	Shows help for this command.

Examples

Interactively schedule an upgrade on a cluster named **mycluster**.

```
$ rosa upgrade cluster --cluster=mycluster --interactive
```

Schedule a cluster upgrade within the hour on a cluster named **mycluster**.

```
$ rosa upgrade cluster --cluster=mycluster --version 4.5.20
```

2.9.2. delete upgrade

Cancel a scheduled cluster upgrade.

Syntax

```
$ rosa delete upgrade --cluster=<cluster_name> | <cluster_id>
```

Table 2.64. Arguments

Option	Definition
<code>--cluster</code>	Required: The name or ID (string) of the cluster that the upgrade will be cancelled for.

Table 2.65. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--yes	Automatically answers yes to confirm the operation.

CHAPTER 3. CHECKING ACCOUNT AND VERSION INFORMATION WITH THE ROSA CLI

3.1. CHECKING ACCOUNT AND VERSION INFORMATION WITH THE ROSA CLI

Use the following commands to check your account and version information with the Red Hat OpenShift Service on AWS (ROSA) CLI, **rosa**.

3.1.1. whoami

Display information about your AWS and Red Hat accounts.

Syntax

```
$ rosa whoami [arguments]
```

Table 3.1. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Example

```
$ rosa whoami
```

3.1.2. version

Display the version of your Red Hat OpenShift Service on AWS (ROSA) CLI, **rosa**.

Syntax

```
$ rosa version [arguments]
```

Table 3.2. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.

Option	Definition
--profile	Specifies an AWS profile (string) from your credentials file.

Example

```
$ rosa version
```

CHAPTER 4. CHECKING LOGS WITH THE ROSA CLI

4.1. CHECKING LOGS WITH THE ROSA CLI

You can check logs with the Red Hat OpenShift Service on AWS (ROSA) CLI, **rosa**. Use the following commands to check your install and uninstall logs.

4.1.1. logs install

Show the cluster install logs.

Syntax

```
$ rosa logs install --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 4.1. Arguments

Option	Definition
--cluster	Required: The name or ID (string) of the cluster to get logs for.
--tail	The number (integer) of lines to get from the end of the log. Default: 2000
--watch	Watches for changes after getting the logs.

Table 4.2. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Examples

Show the last 100 install log lines for a cluster named **mycluster**:

```
$ rosa logs install mycluster --tail=100
```

Show the install logs for a cluster named **mycluster**:

```
$ rosa logs install --cluster=mycluster
```

4.1.2. logs uninstall

Show the cluster uninstall logs.

Syntax

```
$ rosa logs uninstall --cluster=<cluster_name> | <cluster_id> [arguments]
```

Table 4.3. Arguments

Option	Definition
--cluster	The name or ID (string) of the cluster to get logs for.
--tail	The number (integer) of lines to get from the end of the log. Default: 2000
--watch	Watches for changes after getting the logs.

Table 4.4. Optional arguments inherited from parent commands

Option	Definition
--help	Shows help for this command.
--debug	Enables debug mode.
--profile	Specifies an AWS profile (string) from your credentials file.

Example

Show the last 100 uninstall logs for a cluster named **mycluster**:

```
$ rosa logs uninstall --cluster=mycluster --tail=100
```