Red Hat OpenShift Service on AWS

Red Hat® OpenShift® Service on AWS (ROSA) is a fully-managed turnkey application platform that allows you to focus on what matters most, delivering value to your customers by building and deploying applications. Red Hat and AWS site reliability engineering (SRE) experts manage the underlying platform so you don't have to worry about the complexity of infrastructure management.

ROSA provides seamless integration with a wide range of AWS compute, database, analytics, machine learning, networking, mobile, and other services to further accelerate the building and delivering of differentiating experiences to your customers.
The latest version of ROSA makes use of AWS Security Token Service (STS) to obtain credentials to manage infrastructure in your AWS account. AWS STS is a global web service that allows the creation of temporary credentials for identity and access management (IAM) users or federated users. ROSA uses this to assign IAM roles short-term, limited-privilege, security credentials. These credentials are associated with IAM roles that are specific to each component that makes AWS API calls. This better aligns with principles of least privilege and is much better aligned to secure practices in cloud service resource management. The ROSA CLI tool manages the STS credentials that are assigned for unique tasks and takes action upon AWS resources as part of OpenShift functionality.

What do you need before starting?

- Red Hat account
- AWS account

Requirements for using Red Hat OpenShift Service on AWS (ROSA)

There are currently two supported credential methods when creating a Red Hat® OpenShift® Service on AWS (ROSA) cluster. One method uses an identity and access management (IAM) user with the AdministratorAccess policy (only for the account using ROSA). The other, more recent, and recommended method uses AWS with STS. In this learning path, we will only be using the STS method.
The following steps will enable you to set up your accounts and environment so you can start deploying and managing your cluster.

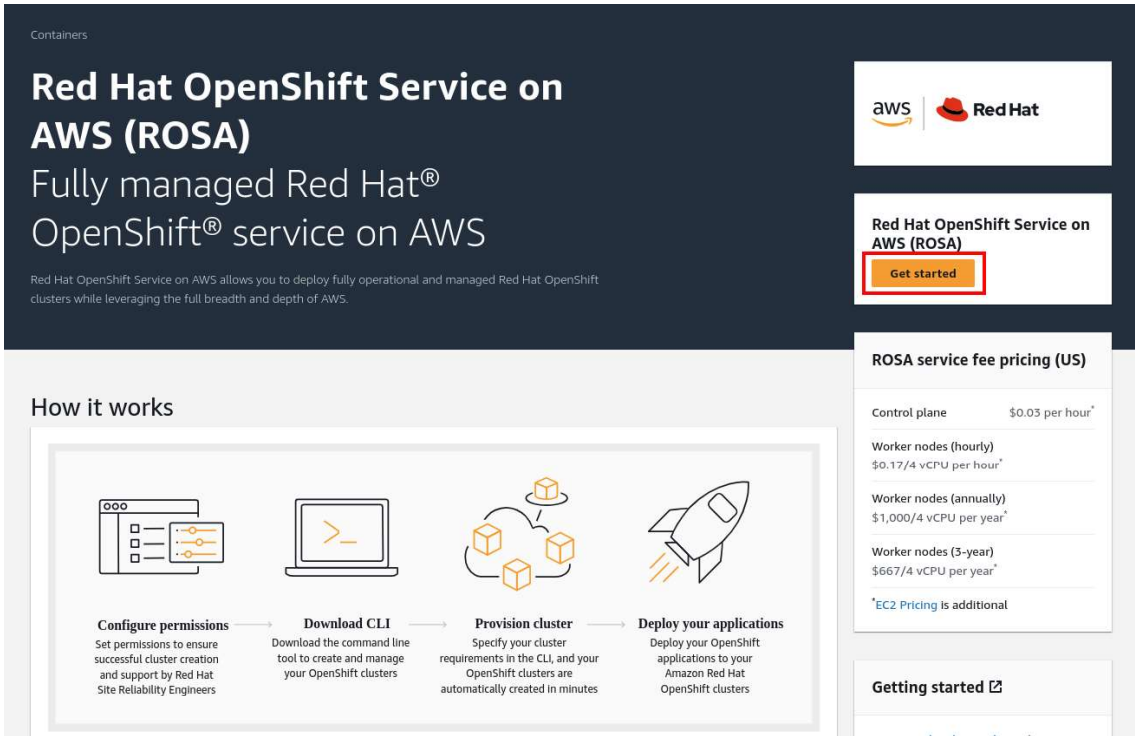
Setting up your accounts

1. Review the prerequisites before getting started.
2. You will need the following pieces of information from your AWS account:
    a. AWS IAM User
    b. AWS Access Key ID
    c. AWS Secret Access Key
3. If you do not have a Red Hat account, create one here. Accept the required terms and conditions. Then check your email for a verification link.

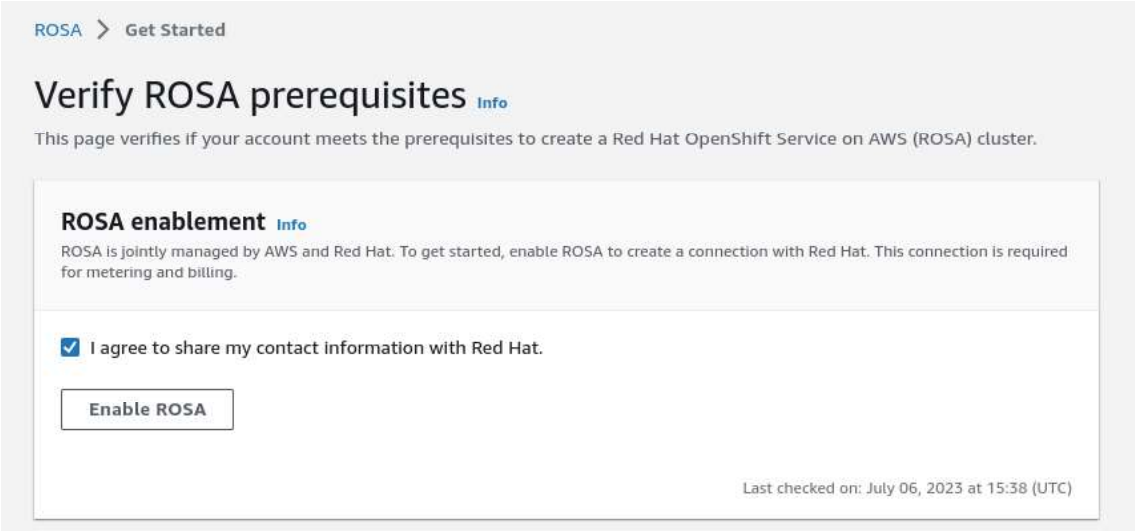
Installing the AWS command line interface (CLI)

1. Install the AWS CLI as per your operating system.

Enable the ROSA service

1. Enable your AWS account to use ROSA to use ROSA by clicking on the orange "Get started" button on the right. It will direct users to a new page.
    a. 

2. Once in the new page, click the checkbox to agree to terms and then click the "Enable ROSA" button.



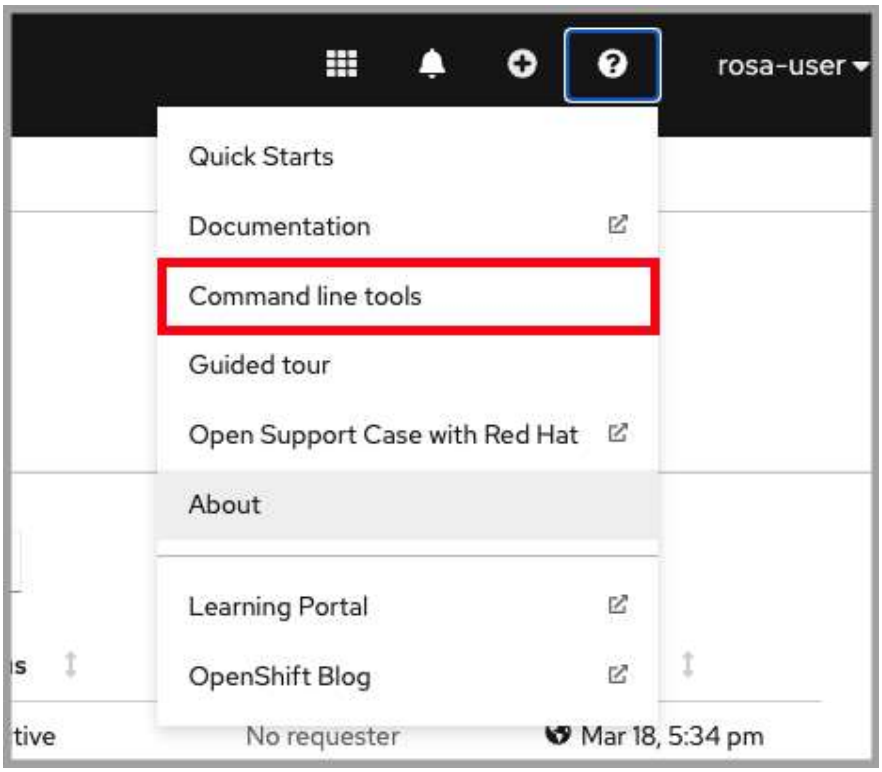
Install the ROSA CLI

1. Install the ROSA CLI as per your operating system.
2. Download and extract the relevant file for your operating system
    a. ex: tar -xvf rosa-linux.tar.gz
3. Save it to a location within your "PATH".
    a. ex: sudo mv rosa /usr/local/bin/rosa
4. Run rosa version to make sure it works and that it returns the version number.

Install the OpenShift CLI

There are a few ways to install the OpenShift (oc) CLI:

1. If you have the rosa CLI installed, the simplest way is to run rosa download oc. Once downloaded, untar (or unzip) the file and move the executables into a directory in your PATH.
2. Or, you can download and install the latest oc.
3. Or, if you already have an OpenShift cluster you can access the command line tools page by clicking on the Question mark > Command Line Tools. Then download the relevant tool for your operating system.



View of Red Hat console page with question mark icon clicked and menu highlighted to show 'command line tools' download option.

Configure the AWS CLI

If you've just installed the AWS CLI, or simply want to make sure it is using the correct AWS account, follow these steps in a terminal:

1. Enter aws configure in the terminal.
2. Enter the AWS Access Key ID and press enter.
3. Enter the AWS Secret Access Key and press enter.
4. Enter the default region you want to deploy into.
5. Enter the output format you want ("table" or "json"). For this guide you can choose "table" as it is easier to read but either is fine.

It should look like the following as an example:

$ aws configure

AWS Access Key ID: AKIA0000000000000000

AWS Secret Access Key: NGvmP00000000000000000000000000

Default region name: us-east-1

Default output format: table

Verify the configuration

Verify that the configuration is correct.

1. Run the following command to query the AWS API:
   aws sts get-caller-identity
2. You should see a table (or JSON if that's what you set it to above) like the one below. Verify that the account information is correct.

aws sts get-caller-identity

```
$ aws sts get-caller-identity
-----------------------------------------------------------------------------
|                              GetCallerIdentity                            |
+--------------+-------------------------------------+--------------------+
|    Account   |                 Arn                 |        UserId      |
+--------------+-------------------------------------+--------------------+
|  000000000000|  arn:aws:iam::00000000000:user/myuser  |  AIDA00000000000000|
+--------------+-------------------------------------+--------------------
```

Ensure the elastic load balancer (ELB) service role exists

Make sure that the service role for ELB already exists, otherwise the cluster deployment could fail. As such, run the following to check for the role and create it if it is missing.

aws iam get-role --role-name "AWSServiceRoleForElasticLoadBalancing" || aws iam create-service-linked-role --aws-service-name "elasticloadbalancing.amazonaws.com"

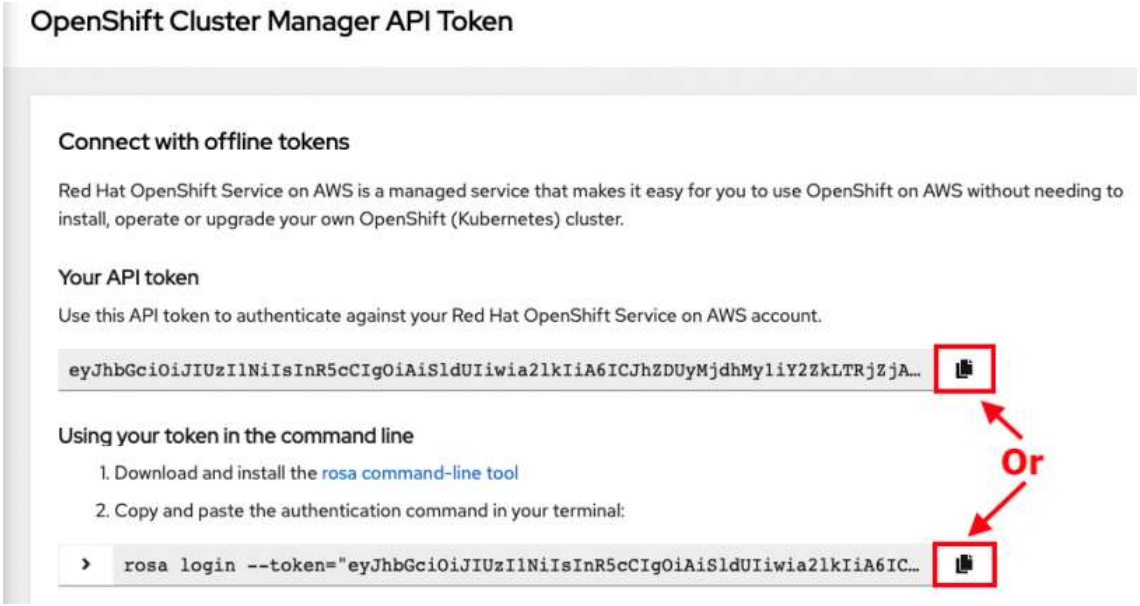
If you received the following error during cluster creation, then the above command should correct it.

Error: Error creating network Load Balancer: AccessDenied: User: arn:aws:sts::970xxxxxxxxx:assumed-role/ManagedOpenShift-Installer-Role/163xxxxxxxxxxxxxx is not authorized to perform: iam:CreateServiceLinkedRole on resource: arn:aws:iam::970xxxxxxxxx:role/aws-service-role/elasticloadbalancing.amazonaws.com/AWSServiceRoleForElasticLoadBalancing"

Log in to your Red Hat account

1. Enter rosa login in a terminal.
2. It will prompt you to open a web browser and go to https://console.redhat.com/openshift/token/rosa

3. Log in with your Red Hat account credentials.
4. Click the "Load token" button.
5. Copy the token and paste it back into the CLI prompt and press enter. Alternatively, you can just copy the full rosa login --token=abc... command and paste that in the terminal.



Where to find your API token

Verify credentials

Verify that all the credentials set up are correct.

1. Run:
   rosa whoami

   You should see an output like below:

```
AWS Account ID:               000000000000
AWS Default Region:           us-east-2
AWS ARN:                      arn:aws:iam::000000000000:user/myuser
OCM API:                      https://api.openshift.com
OCM Account ID:               1DzGIdIhqEWy000000000000000
OCM Account Name:             Your Name
OCM Account Username:         you@domain.com
OCM Account Email:            you@domain.com
OCM Organization ID:          1HopHfA2000000000000000000
OCM Organization Name:        Red Hat
OCM Organization External ID: 0000000
```

1. Please check all information for accuracy before proceeding.

Verify quota

Verify that your AWS account has ample quota in the region you will be deploying your cluster to.

1. Run:
   rosa verify quota

   It should return a response like:

```
I: Validating AWS quota...
I: AWS quota ok. If cluster installation fails, validate actual AWS resource usage
against https://docs.openshift.com/rosa/rosa_getting_started/rosa-required-aws-
service-quotas.html
```

Verify oc CLI

Verify that the oc CLI is installed correctly with:
rosa verify openshift-client
We have now successfully set up our account and environment and are ready to deploy our cluster using AWS's Security Token Service (STS), a method for granting short-lived, dynamic credentials to your users.

How to deploy a cluster with Red Hat OpenShift Service on AWS using the CLI

This resource will take you through the steps to deploy a Red Hat OpenShift® Service on AWS cluster using the ROSA command line interface (CLI).

What will you learn?

- How to deploy a ROSA cluster using the ROSA CLI

What do you need before starting?

- Met all prerequisites

You have two options for deploying your cluster: using the CLI or using the console user interface. In this resource, we'll look at the process for using the CLI. If you're searching for the steps to use the console user interface, please proceed to the next resource, which covers that approach.

Deploying a cluster with the CLI

There are two modes with which to deploy a ROSA with STS cluster. One is automatic, which is quicker and will do the manual work for you. The other is manual, which will require you to execute some extra commands, but will allow you to inspect the roles and policies being created.
This resource will document both options. If you just want to get your cluster created quickly, please use the automatic section, but if you would rather explore the objects being created, then feel free to use manual. This is achieved via the --mode flag in the relevant commands.
Valid options for --mode are:

- manual: Role and Policy documents will be created and saved in the current directory. You will need to manually run the commands that are provided as the next step. This will allow you to review the policy and roles before creating them.
- auto: Roles and policies will be created and applied automatically using the current AWS account, instead of having to manually run each command.

For the purposes of this resource, either method will work, though we do recommend auto mode as that is quicker and has less steps.

Deployment flow

The overall flow that we will follow boils down to:

1. rosa create account-roles - This is executed only once per account. Once created this does not need to be executed again for more clusters of the same y-stream version.
2. rosa create cluster
3. rosa create operator-roles (Manual mode only)
4. rosa create oidc-provider (Manual mode only)

For each succeeding cluster in the same account for the same y-stream version, only step 2 is needed (or 2-4 for manual mode).

Automatic mode (recommended)

As mentioned above, if you want the ROSA CLI to automate the creation of the roles and policies to create your cluster quickly, then use this method.

Create account roles

If this is the first time you are deploying ROSA in this account and have not yet created the account roles, then create the account-wide roles and policies, including Operator policies.
Run the following command to create the account-wide roles:
rosa create account-roles --mode auto --yes
You will see an output like the following:

```
I: Creating roles using 'arn:aws:iam::000000000000:user/rosa-user'
I: Created role 'ManagedOpenShift-ControlPlane-Role' with ARN
'arn:aws:iam::000000000000:role/ManagedOpenShift-ControlPlane-Role'
I: Created role 'ManagedOpenShift-Worker-Role' with ARN
'arn:aws:iam::000000000000:role/ManagedOpenShift-Worker-Role'
I: Created role 'ManagedOpenShift-Support-Role' with ARN
'arn:aws:iam::000000000000:role/ManagedOpenShift-Support-Role'
I: Created role 'ManagedOpenShift-Installer-Role' with ARN
'arn:aws:iam::000000000000:role/ManagedOpenShift-Installer-Role'
I: Created policy with ARN 'arn:aws:iam::000000000000:policy/ManagedOpenShift-
openshift-machine-api-aws-cloud-credentials'
I: Created policy with ARN 'arn:aws:iam::000000000000:policy/ManagedOpenShift-
openshift-cloud-credential-operator-cloud-crede'
I: Created policy with ARN 'arn:aws:iam::000000000000:policy/ManagedOpenShift-
openshift-image-registry-installer-cloud-creden'
I: Created policy with ARN 'arn:aws:iam::000000000000:policy/ManagedOpenShift-
```

Create the cluster

Run the following command to create a cluster with all the default options:

rosa create cluster --cluster-name <cluster-name> --sts --mode auto --yes

You should see a response like the following:

```
...
I: Creating cluster 'my-rosa-cluster'
I: To view a list of clusters and their status, run 'rosa list clusters'
I: Cluster 'my-rosa-cluster' has been created.
I: Once the cluster is installed you will need to add an Identity Provider before you
can login into the cluster. See 'rosa create idp --help' for more information.
I: To determine when your cluster is Ready, run 'rosa describe cluster -c my-rosa-
cluster'.
I: To watch your cluster installation logs, run 'rosa logs install -c my-rosa-cluster
--watch'.
Name:                       my-rosa-cluster
ID:                         1mlhulb3bo0l54ojd0ji000000000000
```

NOTE: This will also create the required operator roles and OIDC provider. If you want to see all available options for your cluster use the --help flag or for interactive mode you can use --interactive.

The default settings are as follows:

- 3 Master Nodes, 2 Infra Nodes, 2 Worker Nodes
    - See here for more details.
- Region: As configured for the AWS CLI
- Networking IP ranges:
    - Machine CIDR: 10.0.0.0/16
    - Service CIDR: 172.30.0.0/16
    - Pod CIDR: 10.128.0.0/14
- The most recent version of OpenShift available to rosa
- A single availability zone

- Public cluster

Check installation status

You can run the following command to check the detailed status of the cluster:
rosa describe cluster --cluster <cluster-name>
Or you can run the following for an abridged view of the status:
rosa list clusters
You should notice the state change from “waiting” to “installing” to "ready". This will take about 40 minutes to run.
Once the state changes to “ready” your cluster is now installed.

Manual mode

As mentioned above if you want to be able to review the roles and policies created before applying them, you can use this manual method. Though it will require running a few extra commands to create the roles and policies.
In this section we will make use of the --interactive mode so that it will be easier to follow along, though feel free to use the default cluster creation command above if you'd like. See here for a description of the fields in this section.

Create account roles

1. If this is the first time you are deploying ROSA in this account and have not yet created the account roles, then create the account-wide roles and policies, including Operator policies. This command will create the needed JSON files for the required roles and policies for your account in the current directory. This will also output the aws commands you need to run in order to create these objects.
   Run the following command to create the needed files and output the commands you need to run:
   rosa create account-roles --mode manual
   You will see an output like the following:

```
I: All policy files saved to the current directory
I: Run the following commands to create the account roles and policies:

aws iam create-role \
--role-name ManagedOpenShift-Worker-Role \
--assume-role-policy-document file://sts_instance_worker_trust_policy.json \
--tags Key=rosa_openshift_version,Value=4.8
Key=rosa_role_prefix,Value=ManagedOpenShift
Key=rosa_role_type,Value=instance_worker

aws iam put-role-policy \
--role-name ManagedOpenShift-Worker-Role \
--policy-name ManagedOpenShift-Worker-Role-Policy \
--policy-document file://sts_instance_worker_permission_policy.json
```

If you look at the contents of your current directory you will see the new files created. We will be using the aws CLI to create each of these objects.

```
$ ls
openshift_cloud_credential_operator_cloud_credential_operator_iam_ro_creds_policy.j
son  sts_instance_controlplane_permission_policy.json
openshift_cluster_csi_drivers_ebs_cloud_credentials_policy.json
sts_instance_controlplane_trust_policy.json
openshift_image_registry_installer_cloud_credentials_policy.json
sts_instance_worker_permission_policy.json
openshift_ingress_operator_cloud_credentials_policy.json
sts_instance_worker_trust_policy.json
openshift_machine_api_aws_cloud_credentials_policy.json
sts_support_permission_policy.json
sts_installer_permission_policy.json
sts_support_trust_policy.json
sts_installer_trust_policy.json
```

(Optional) If you'd like, you may open the files to review what you will be creating. For example if we open the sts_installer_permission_policy.json we can see:

```
$ cat sts_installer_permission_policy.json
{
"Version": "2012-10-17",
"Statement": [
{
    "Effect": "Allow",
    "Action": [
        "autoscaling:DescribeAutoScalingGroups",
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AssociateDhcpOptions",
        "ec2:AssociateRouteTable",
        "ec2:AttachInternetGateway",
        "ec2:AttachNetworkInterface",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        [...]
```

Execute the aws commands presented from the above step. You can copy and paste as long as you are in the same directory as the json files created.

Create the cluster

After all the aws commands have been executed successfully run the following command to begin the ROSA cluster creation in interactive mode:
rosa create cluster --interactive --sts
See here for a description of the fields below.
For the purpose of this tutorial please select the following values.
Cluster name: my-rosa-cluster
OpenShift version: <choose version>
External ID (optional): <leave blank>
Operator roles prefix: <accept default>
Multiple availability zones: No
AWS region: <choose region>
PrivateLink cluster: No
Install into an existing VPC: No
Enable Customer Managed key: No
Compute nodes instance type: m5.xlarge
Enable autoscaling: No
Compute nodes: 2
Machine CIDR: <accept default>
Service CIDR: <accept default>
Pod CIDR: <accept default>
Host prefix: <accept default>

Encrypt etcd data (optional): No
Disable Workload monitoring: No
You will see the following response along with the command to create this cluster in the future so that you don't need to go through the interactive mode again.

```
I: Creating cluster 'my-rosa-cluster'
I: To create this cluster again in the future, you can run:
rosa create cluster --cluster-name my-rosa-cluster --role-arn
arn:aws:iam::000000000000:role/ManagedOpenShift-Installer-Role --support-role-arn
arn:aws:iam::000000000000:role/ManagedOpenShift-Support-Role --master-iam-role
arn:aws:iam::000000000000:role/ManagedOpenShift-ControlPlane-Role --worker-iam-role
arn:aws:iam::000000000000:role/ManagedOpenShift-Worker-Role --operator-roles-prefix
my-rosa-cluster --region us-west-2 --version 4.8.13 --compute-nodes 2 --machine-cidr
10.0.0.0/16 --service-cidr 172.30.0.0/16 --pod-cidr 10.128.0.0/14 --host-prefix 23
I: To view a list of clusters and their status, run 'rosa list clusters'
I: Cluster 'my-rosa-cluster' has been created.
I: Once the cluster is installed you will need to add an Identity Provider before you
can login into the cluster. See 'rosa create idp --help' for more information.
Name:                       my-rosa-cluster
ID:                         1t6i7c0dbum4mslts6c000000000000
```

NOTE: The state will stay in "waiting" until the next two steps below are completed.

Create operator roles

We can see at the end of the output from the above step we are told exactly what we need to run next. These roles need to be created once per cluster. To create the roles run the following:
rosa create operator-roles --mode manual --cluster <cluster-name>
Run each of the aws commands presented.
You will see an output like the following with all the commands that need to be executed.

```
I: Run the following commands to create the operator roles:

aws iam create-role \
    --role-name my-rosa-cluster-openshift-image-registry-installer-cloud-credentials \
    --assume-role-policy-document
file://operator_image_registry_installer_cloud_credentials_policy.json \
    --tags Key=rosa_cluster_id,Value=1mkesci269png3tck000000000000000
Key=rosa_openshift_version,Value=4.8 Key=rosa_role_prefix,Value=
Key=operator_namespace,Value=openshift-image-registry
Key=operator_name,Value=installer-cloud-credentials

aws iam attach-role-policy \
    --role-name my-rosa-cluster-openshift-image-registry-installer-cloud-credentials \
    --policy-arn arn:aws:iam::000000000000:policy/ManagedOpenShift-openshift-image-
registry-installer-cloud-creden
[...]
```

Create the OIDC provider

Run the following to create the OIDC provider:
rosa create oidc-provider --mode manual --cluster <cluster-name>
This will display the aws commands that you need to run. Run the commands like below:
I: Run the following commands to create the OIDC provider:

```
$ aws iam create-open-id-connect-provider \
--url https://rh-oidc.s3.us-east-1.amazonaws.com/1mkesci269png3tckknhh0rfs2da5fj9 \
--client-id-list openshift sts.amazonaws.com \
--thumbprint-list a9d53002e97e00e043244f3d170d000000000000

$ aws iam create-open-id-connect-provider \
--url https://rh-oidc.s3.us-east-1.amazonaws.com/1mkesci269png3tckknhh0rfs2da5fj9 \
--client-id-list openshift sts.amazonaws.com \
--thumbprint-list a9d53002e97e00e043244f3d170d000000000000
```

Your cluster will then continue the installation process.

Check installation status

You can run the following command to check the detailed status of the cluster:
rosa describe cluster --cluster <cluster-name>
Or you can run the following for an abridged view of the status:
rosa list clusters
You should notice the state change from “waiting” to “installing” to "ready". This will take about 40 minutes to run.
Once the state changes to “ready” your cluster is now installed.

Obtain the console URL

To get the console URL, run:
rosa describe cluster -c <cluster-name> | grep Console

The cluster has now been successfully deployed.

- Want to experiment within a cluster yourself? Dive into our Developer Sandbox with a 30-day no-cost trial, where you can practice building within OpenShift.

In the next resource, we’ll cover the other option for deploying a cluster: using the console user-interface.

How to deploy a cluster with Red Hat OpenShift Service on AWS using the console UI

This resource will take you through the steps to deploy a Red Hat OpenShift® Service on AWS cluster using the OpenShift® Cluster Manager (OCM) user interface (UI).

What will you learn?

- How to deploy a ROSA cluster using the console interface

What do you need before starting?

- Met all prerequisites

Now we'll look at the second option for deploying your cluster: using the console user interface.

Deploying a cluster with the UI

Follow these steps to deploy a ROSA cluster using the OCM UI.

Deployment flow

The overall flow that we will follow is below. Step 1 only needs to be performed the first time you are deploying into an AWS account. Step 2 only needs to be performed the first time you are using the user interface. So for each successive cluster of the same y-stream version, you would just create the cluster.

1. Create the account wide roles and policies
2. Associate your AWS account with your Red Hat account
    a. Create and link OCM role
    b. Create and link User role
3. Create the cluster

Create account-wide roles

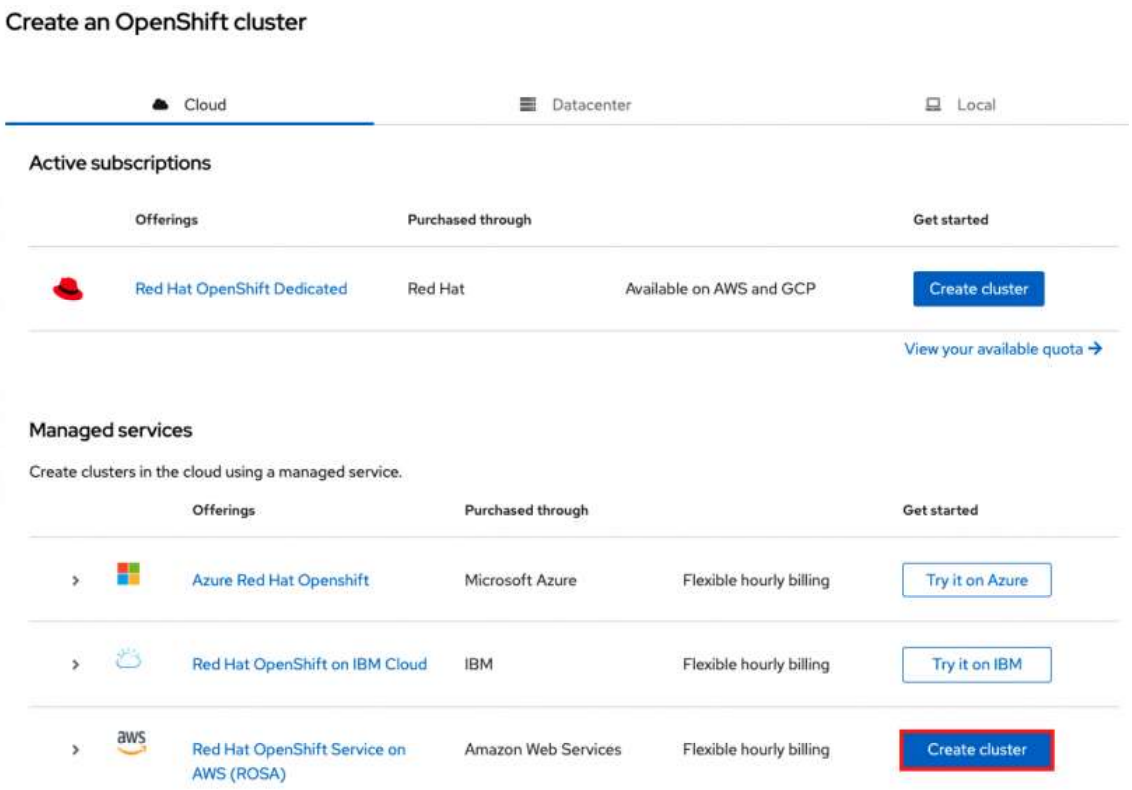
NOTE: If you already have account roles (possibly from an earlier deployment) then skip this step. You will see that the UI will detect your existing roles after you select an associated AWS account.

If this is the first time you are deploying ROSA in this account and have not yet created the account roles, then create the account-wide roles and policies, including Operator policies.
In your terminal run the following command to create the account-wide roles:
rosa create account-roles --mode auto --yes
You will see an output like the following:

```
I: Creating roles using 'arn:aws:iam::000000000000:user/rosa-user'
I: Created role 'ManagedOpenShift-ControlPlane-Role' with ARN
'arn:aws:iam::000000000000:role/ManagedOpenShift-ControlPlane-Role'
I: Created role 'ManagedOpenShift-Worker-Role' with ARN
'arn:aws:iam::000000000000:role/ManagedOpenShift-Worker-Role'
I: Created role 'ManagedOpenShift-Support-Role' with ARN
'arn:aws:iam::000000000000:role/ManagedOpenShift-Support-Role'
I: Created role 'ManagedOpenShift-Installer-Role' with ARN
'arn:aws:iam::000000000000:role/ManagedOpenShift-Installer-Role'
I: Created policy with ARN 'arn:aws:iam::000000000000:policy/ManagedOpenShift-
openshift-machine-api-aws-cloud-credentials'
I: Created policy with ARN 'arn:aws:iam::000000000000:policy/ManagedOpenShift-
openshift-cloud-credential-operator-cloud-crede'
I: Created policy with ARN 'arn:aws:iam::000000000000:policy/ManagedOpenShift-
openshift-image-registry-installer-cloud-creden'
I: Created policy with ARN 'arn:aws:iam::000000000000:policy/ManagedOpenShift-
openshift-ingress-operator-cloud-credentials'
I: Created policy with ARN 'arn:aws:iam::000000000000:policy/ManagedOpenShift-
```

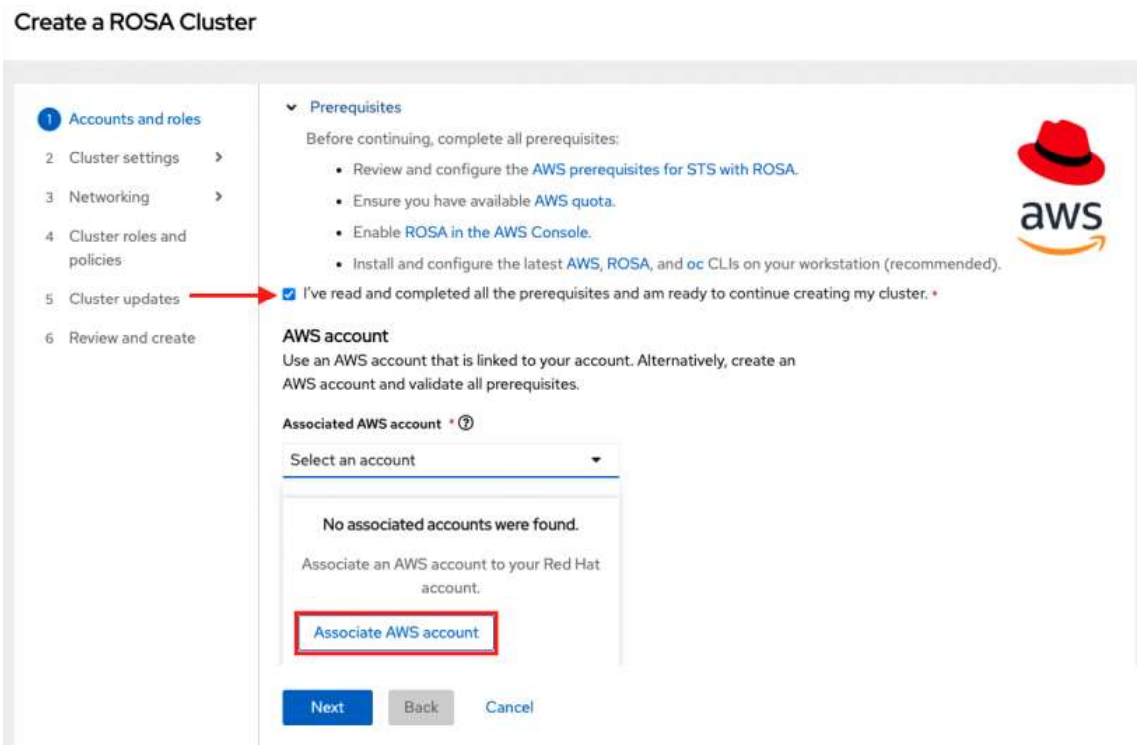
Associate your AWS account with your Red Hat account

NOTE: If you have already associated AWS accounts that you want to use, please skip this step.
The next step is to tell OCM what is/are your AWS account(s) that you want to use for deploying ROSA into.
Open OCM by visiting https://console.redhat.com/openshift and log in to your Red Hat account.
Click on the "Create Cluster" button.
Then in the ROSA row (about midway down the page, under "Managed services") click on the "Create Cluster" button.

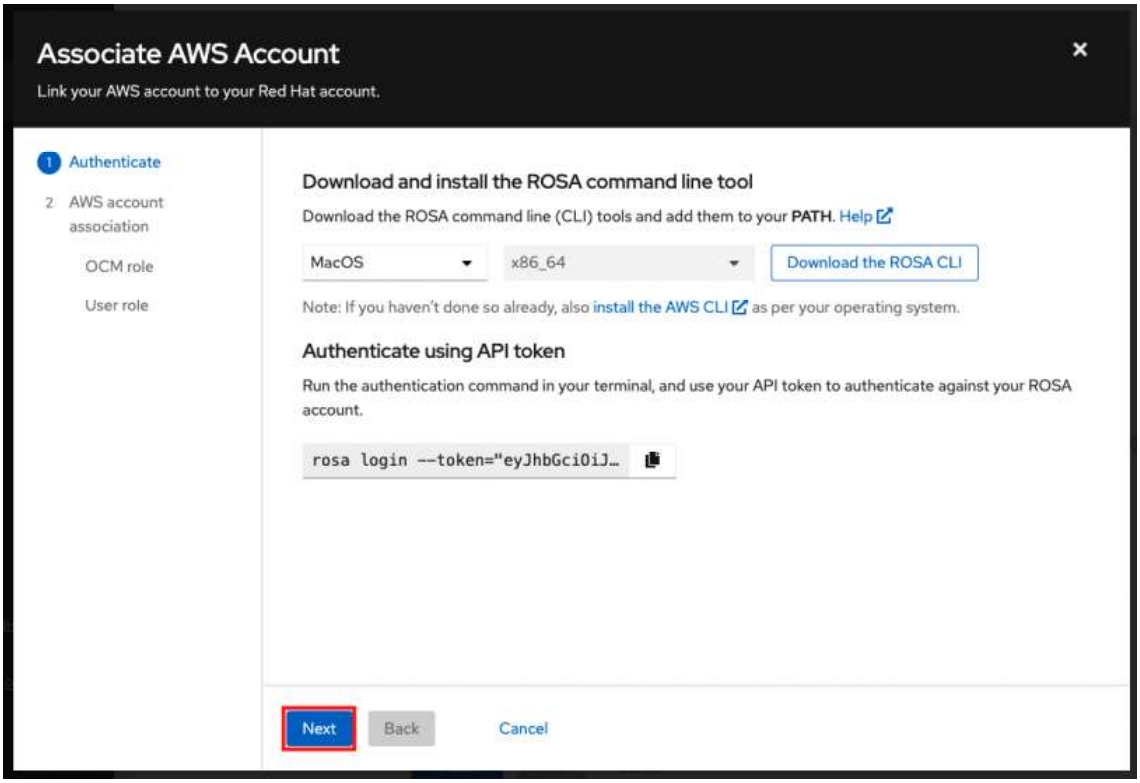


Managed services screen and the “Create cluster” button

Check the box stating that you have read and completed all the prerequisites. Then click the dropbox under "Associated AWS account". You may see that there are no associated accounts. This is expected since we have not associated any AWS accounts yet. Click on the box that says "Associate AWS account."



The box stating that you have read and completed all the prerequisites.
A pop up window will open instructing you to download the ROSA CLI, AWS CLI, and to log into your Red Hat account. If you have been following this learning path, we already did this in a previous section, so just click "Next".



Pop-up for downloading CLIs or to click “Next”

On the next page you will see the commands to create the OCM role for the level of permissions that this role will have. You can create:

- Basic OCM role: Allows OCM to have read-only access to the account in order to check if the roles and policies that are required by ROSA are present before creating a cluster. You will need to manually create the required roles, policies and OIDC provider using the CLI.
- Admin OCM role: Grants OCM additional permissions in order to create the required roles, policies, and OIDC provider for ROSA. Using this makes the deployment of a ROSA cluster quicker since OCM will be able to create the required resources for you avoiding the need for you to manually create them.

To read more about these roles, please visit the OpenShift Cluster Manager roles and permissions section of the documentation.
For the purposes of this workshop, we'll use the Admin OCM role since we want the simplest and quickest approach.

Create and associate an OCM role

You can copy the command for the Admin OCM role from that window which will launch interactive mode. Or for simplicity switch to your terminal and execute:
rosa create ocm-role --mode auto --admin --yes

```
I: Creating ocm role
I: Creating role using 'arn:aws:iam::000000000000:user/rosa-user'
I: Created role 'ManagedOpenShift-OCM-Role-12561000' with ARN
'arn:aws:iam::000000000000:role/ManagedOpenShift-OCM-Role-12561000'
I: Linking OCM role
I: Successfully linked role-arn 'arn:aws:iam::000000000000:role/ManagedOpenShift-OCM-
Role-12561000' with organization account '1MpZfntsZeUdjWHg7XRgP000000'
```

This will create the OCM roles for you and associate them with your Red Hat account.
NOTE: As an alternative, you can define --mode manual if you'd prefer to execute the AWS CLI commands yourself. The AWS commands will be outputted to the CLI and the relevant JSON files will be created in the current directory. Also make sure to link the role as well which is the last command output. Also, if you insist on creating a Basic OCM role, then just remove --admin from the command above.
Then, click "Next".

Create an OCM User role

As defined in the documentation, the user role needs to be created so that the ROSA service can verify your AWS identity. This role has no permissions, and it is only used to create a trust relationship between the installer account and your OCM role resources.
Run the following to create the User Role and to link it to your Red Hat account.

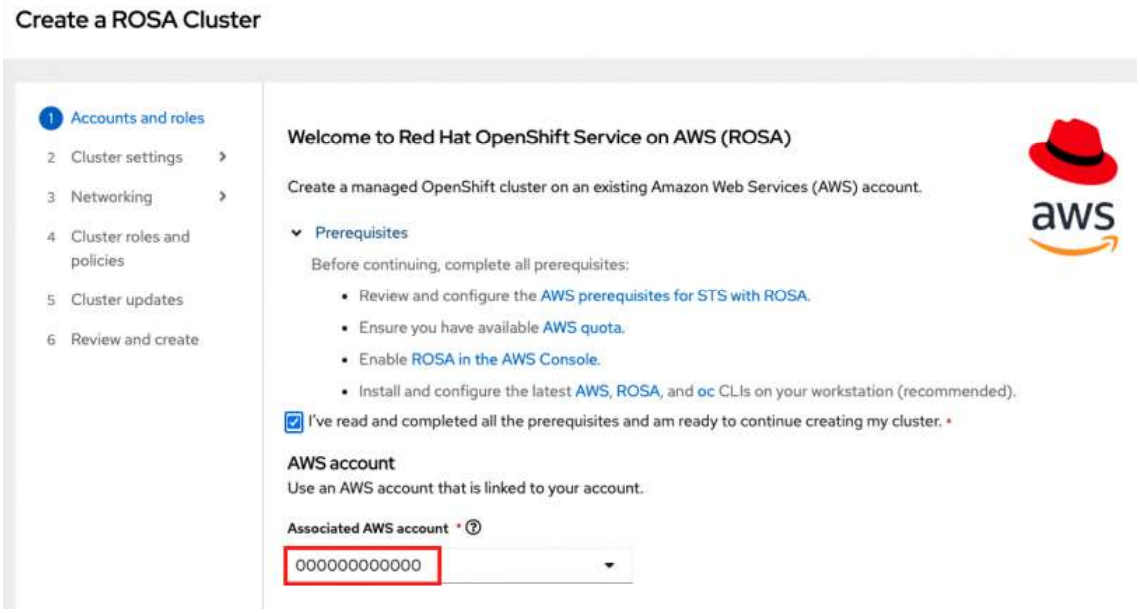
rosa create user-role --mode auto --yes
You will see a response like:

```
I: Creating User role
I: Creating ocm user role using 'arn:aws:iam::000000000000:user/rosa-user'
I: Created role 'ManagedOpenShift-User-rosa-user-Role' with ARN
'arn:aws:iam::000000000000:role/ManagedOpenShift-User-rosa-user-Role'
I: Linking User role
I: Successfully linked role ARN 'arn:aws:iam::000000000000:role/ManagedOpenShift-User-
rosa-user-Role' with account '1rbOQez0z5j1YolInhcXY000000'
```

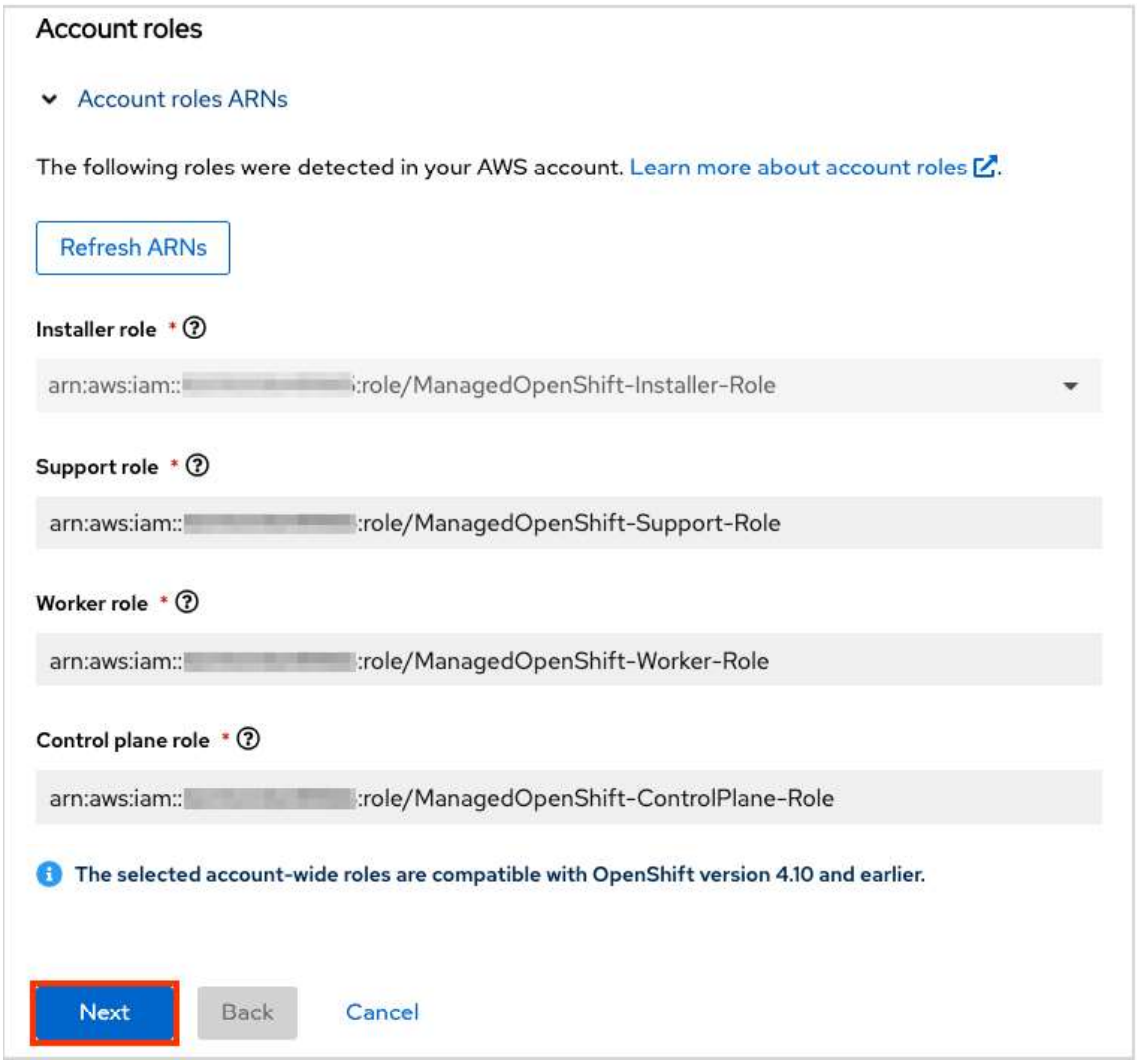
Click “Ok.”

Confirm successful association

You will be brought back to the original window in which you should see your AWS account that you associated above in the drop down. If you see your account there, it was successful.
Select the account.



Screen where user selects their account
You will then see the account role ARNs (created earlier) populated below. Then click “Next”.



List of populated roles

Create the cluster

For the purposes of this learning path, make the following selections.

Cluster settings

Details:

- Cluster name: <pick a name>
- Version: <select latest version>
- Region: <select desired region>
- Availability: Single zone
- Enable user workload monitoring: leave checked
- Enable additional etcd encryption: leave unchecked
- Encrypt persistent volumes with customer keys: leave unchecked

Click "Next".

Machine pool (leave the defaults which are):

- Compute node instance type: m5.xlarge - 4 vCPU 16 GiB RAM
- Enable autoscaling: unchecked
- Compute node count: 2
- Leave node labels blank

Click "Next".

Networking

Configuration - Leave all default values
Click "Next".
CIDR ranges - Leave all default values
Click "Next".

Cluster roles and policies

For the purposes of this workshop leave "Auto" selected and it will make the cluster deployment process simpler and quicker.
NOTE: If you selected a Basic OCM role earlier you can only use manual mode and you must manually create the operator roles and OIDC provider. See "For Basic OCM roles only" section below after you've completed the "Cluster updates" section and started the cluster creation.

Cluster updates

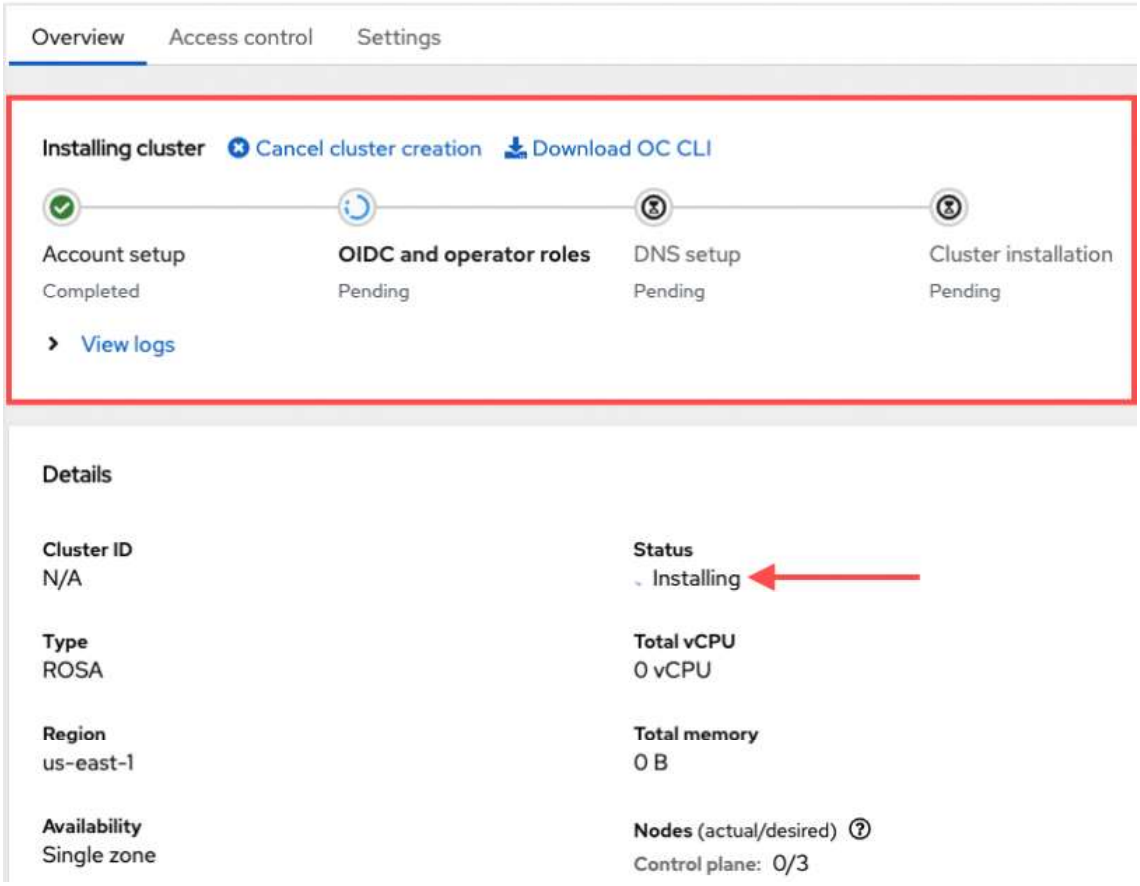
Leave all the default options.

Review and create

Review the content for the cluster configuration and click "Create cluster".

Monitor installation progress

Stay at the current page to monitor the installation progress.



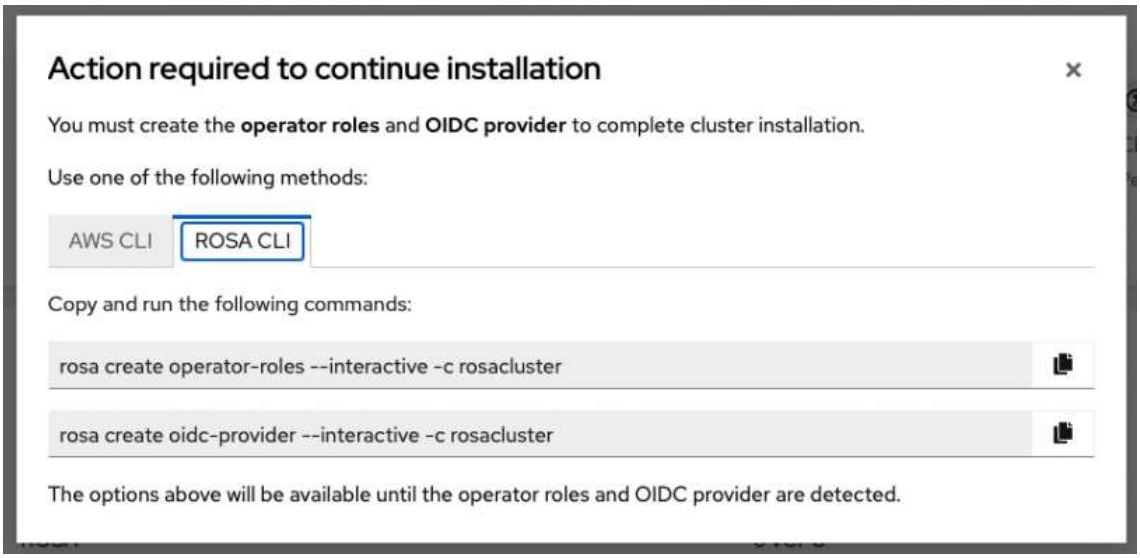
Monitoring installation progress

For Basic OCM role only

NOTE: If you created an Admin OCM role as directed above please ignore this section since OCM will create the role for you.

Create operator roles

If you created a Basic OCM Role earlier, you will need to manually create 2 more elements before the cluster installation can continue.
Operator roles
OIDC provider
NOTE: To understand what these do, please see the "What is STS?" resource.
There will be a pop up window that will show you the commands to run.



Pop up for creating operator roles and OIDC provider.

In your terminal, you may run the commands from the window which will launch interactive mode. For simplicity, though, run the following to create the Operator roles:

rosa create operator-roles --mode auto --cluster <cluster-name> --yes

You will see a response like:

```
I: Creating roles using 'arn:aws:iam::000000000000:user/rosauser'
I: Created role 'rosacluster-b736-openshift-ingress-operator-cloud-credentials' with
ARN 'arn:aws:iam::000000000000:role/rosacluster-b736-openshift-ingress-operator-cloud-
credentials'
I: Created role 'rosacluster-b736-openshift-cluster-csi-drivers-ebs-cloud-credent'
with ARN 'arn:aws:iam::000000000000:role/rosacluster-b736-openshift-cluster-csi-
drivers-ebs-cloud-credent'
I: Created role 'rosacluster-b736-openshift-cloud-network-config-controller-cloud'
with ARN 'arn:aws:iam::000000000000:role/rosacluster-b736-openshift-cloud-network-
config-controller-cloud'
I: Created role 'rosacluster-b736-openshift-machine-api-aws-cloud-credentials' with
ARN 'arn:aws:iam::000000000000:role/rosacluster-b736-openshift-machine-api-aws-cloud-
credentials'
I: Created role 'rosacluster-b736-openshift-cloud-credential-operator-cloud-crede'
with ARN 'arn:aws:iam::000000000000:role/rosacluster-b736-openshift-cloud-credential-
operator-cloud-crede'
I: Created role 'rosacluster-b736-openshift-image-registry-installer-cloud-creden'
with ARN 'arn:aws:iam::000000000000:role/rosacluster-b736-openshift-image-registry-
installer-cloud-creden'
```

Create OIDC provider

In your terminal run the following to create the Operator roles:

rosa create oidc-provider --mode auto --cluster <cluster-name> --yes

You will see a response like:

```
I: Creating OIDC provider using 'arn:aws:iam::000000000000:user/rosauser'
I: Created OIDC provider with ARN 'arn:aws:iam::000000000000:oidc-provider/rh-
oidc.s3.us-east-1.amazonaws.com/1tt4kvrr2kha2rgs8gjfvf0000000000'
```

You are now ready to move on to the next resource, where you'll learn how to create an admin user.

- Want to experiment within a cluster yourself? Dive into our Developer Sandbox with a 30-day no-cost trial, where you can practice building within OpenShift.

Creating ROSA with HCP clusters using the default options

Red Hat® OpenShift® Service on AWS (ROSA) with hosted control planes (HCP) offers a more efficient and reliable architecture for creating ROSA clusters. With ROSA with HCP, each cluster has a dedicated control plane that is isolated in a ROSA service account.

Create a ROSA with HCP cluster quickly by using the default options and automatic AWS Identity and Access Management (IAM) resource creation. You can deploy your cluster by using the ROSA CLI (rosa).

Note: Since it is not possible to upgrade or convert existing ROSA clusters to a hosted control planes architecture, you must create a new cluster to use ROSA with HCP functionality.

Note: Sharing VPCs across multiple AWS accounts is not currently supported for ROSA with HCP. Do not install a ROSA with HCP cluster into subnets shared from another AWS account. See "Are multiple ROSA clusters in a single VPC supported?" for more information.

After reading this resource, you will know:

- How to create a Virtual Private Cloud for your ROSA with HCP clusters
- How to create a ROSA with HCP cluster using the CLI
- Other options for creating ROSA with HCP clusters

You can quickly create a ROSA with HCP cluster with the Security Token Service (STS) by using the default installation options. View this summary to describe the default cluster specifications.

ROSA with HCP prerequisites

To create a ROSA with HCP cluster, you must have the following items:

- A configured virtual private cloud (VPC)
- Account-wide roles
- An OIDC configuration
- Operator roles

Creating a Virtual Private Cloud for your ROSA with HCP clusters

You must have a Virtual Private Cloud (VPC) to create ROSA with HCP cluster. You can use the following methods to create a VPC:

- Create a VPC by using a Terraform template
- Manually create the VPC resources in the AWS console

Note: The Terraform instructions are for testing and demonstration purposes. Your own installation requires some modifications to the VPC for your own use. You should also ensure that when you use this Terraform script it is in the same region that you intend to install your cluster. In these examples, use us-east-2.

Creating a Virtual Private Cloud using Terraform

Terraform is a tool that allows you to create various resources using an established template. The following process uses the default options as required to create a ROSA with HCP cluster. For more information about using Terraform, see the additional resources.

Prerequisites:

- You have installed Terraform version 1.4.0 or newer on your machine.
- You have installed Git on your machine.

Steps:

1. Open a shell prompt and clone the Terraform VPC repository by running the following command:
   git clone https://github.com/openshift-cs/terraform-vpc-example
2. Navigate to the created directory by running the following command:
   cd terraform-vpc-example
3. Initiate the Terraform file by running the following command:
   terraform init
   A message confirming the initialization appears when this process completes.
4. To build your VPC Terraform plan based on the existing Terraform template, run the plan command. You must include your AWS region. You can choose to specify a cluster name. A rosa.tfplan file is added to the hypershift-tf directory after the terraform plan completes. For more detailed options, see the Terraform VPC repository's README file.
   terraform plan -out rosa.tfplan -var region=<region>
5. Apply this plan file to build your VPC by running the following command:
   terraform apply rosa.tfplan
   a. Optional: You can capture the values of the Terraform-provisioned private, public, and machinepool subnet IDs as environment variables to use when creating your ROSA with HCP cluster by running the following commands:
      export SUBNET_IDS=$(terraform output -raw cluster-subnets-string)

b. Verify that the variables were correctly set with the following command:
   echo $SUBNET_IDS

Note: See the Terraform VPC repository for a detailed list of all options available when customizing the VPC for your needs.

Creating a Virtual Private Cloud manually

If you choose to manually create your Virtual Private Cloud (VPC) instead of using Terraform, go to the VPC page in the AWS console.

Before you can use your VPC to create a ROSA with HCP cluster, you must tag your VPC subnets. Automated service preflight checks verify that these resources are tagged correctly before you can use these resources. The following table shows how your resources should be tagged as the following:

Resource	Key	Value
Public subnet	kubernetes.io/role/elb	1 or no value
Public subnet	kubernetes.io/role/internal-elb	1 or no value

Note: You must tag at least one private subnet and, if applicable, and one public subnet.

Prerequisites:

- You have created a VPC.
- You have installed the aws CLI.

Steps:

1. Tag your resources in your terminal by running the following commands:
   a. For public subnets, run:
      aws ec2 create-tags --resources <public-subnet-id> --tags Key=kubernetes.io/role/elb,Value=1
   b. For private subnets, run:
      aws ec2 create-tags --resources <private-subnet-id> --tags Key=kubernetes.io/role/internal-elb,Value=1
2. Verify that the tag is correctly applied by running the following command:
   aws ec2 describe-tags --filters "Name=resource-id,Values=<subnet_id>"
3. Example output:
   TAGS Name <subnet-id> subnet <prefix>-subnet-public1-us-east-1a
   TAGS kubernetes.io/role/elb <subnet-id> subnet 1

Creating the account-wide STS roles and policies

Prerequisites:

- You have completed the AWS prerequisites for ROSA with HCP.
- You have available AWS service quotas.
- You have enabled the ROSA service in the AWS Console.

- You have installed and configured the latest ROSA CLI (rosa) on your installation host.
- You have logged in to your Red Hat account by using the ROSA CLI.

Steps:

1. If they do not exist in your AWS account, create the required account-wide STS roles and attach the policies by running the following command:
   rosa create account-roles --hosted-cp
2. Optional: Set your prefix as an environmental variable by running the following command:
   export ACCOUNT_ROLES_PREFIX=<account_role_prefix>
   a. View the value of the variable by running the following command:
      echo $ACCOUNT_ROLES_PREFIX

For more information regarding AWS managed IAM policies for ROSA, see AWS managed IAM policies for ROSA.

Creating an OpenID Connect configuration

When using a ROSA with HCP cluster, you must create the OpenID Connect (OIDC) configuration prior to creating your cluster. This configuration is registered to be used with OpenShift Cluster Manager.

Prerequisites:

- You have completed the AWS prerequisites for ROSA with HCP.
- You have completed the AWS prerequisites for Red Hat OpenShift Service on AWS.
- You have installed and configured the latest Red Hat OpenShift Service on AWS (ROSA) CLI, rosa, on your installation host.

Steps:

1. To create your OIDC configuration alongside the AWS resources, run the following command:
   rosa create oidc-config --mode=auto --yes
   a. When creating your cluster, you must supply the OIDC config ID. The CLI output provides this value for --mode auto, otherwise you must determine these values based on aws CLI output for --mode manual.
2. Optional: you can save the OIDC configuration ID as a variable to use later. Run the following command to save the variable:
   export OIDC_ID=<oidc_config_id>
   a. In the example output above, the OIDC configuration ID is 13cdr6b.
   b. View the value of the variable by running the following command:
      echo $OIDC_ID
3. You can list the possible OIDC configurations available for your clusters that are associated with your user organization. Run the following command:
   rosa list oidc-config

Creating Operator roles and policies

When using a ROSA with HCP cluster, you must create the Operator IAM roles that are required for Red Hat OpenShift Service on AWS (ROSA) with hosted control planes (HCP) deployments. The cluster Operators use the Operator roles to obtain the temporary permissions required to carry out cluster operations, such as managing back-end storage, cloud provider credentials, and external access to a cluster.

Prerequisites:

- You have completed the AWS prerequisites for ROSA with HCP.
- You have installed and configured the latest Red Hat OpenShift Service on AWS ROSA CLI (rosa), on your installation host.
- You created the account-wide AWS roles.

Steps:

1. Set your prefix name to an environment variable using the following command:
   export OPERATOR_ROLES_PREFIX=<prefix_name>
2. To create your Operator roles, run the following command:
   rosa create operator-roles --hosted-cp
   The following breakdown provides options for the Operator role creation.
   rosa create operator-roles --hosted-cp --prefix=$OPERATOR_ROLES_PREFIX --oidc-config-id=$OIDC_ID --installer-role-arn arn:aws:iam::${AWS_ACCOUNT_ID}:role/${ACCOUNT_ROLES_PREFIX}-HCP-ROSA-Installer-Role
   Notes:
   You must supply a prefix when creating these Operator roles. Failing to do so produces an error. See the Additional resources of this section for information on the Operator prefix.
   This value is the OIDC configuration ID that you created for your ROSA with HCP cluster.
   This value is the installer role ARN that you created when you created the ROSA account roles.
   You must include the --hosted-cp parameter to create the correct roles for ROSA with HCP clusters. This command returns the following information.

   The Operator roles are now created and ready to use for creating your ROSA with HCP cluster.
3. You can list the Operator roles associated with your ROSA account. Run the following command:
   rosa list operator-roles

After the command runs, it displays all the prefixes associated with your AWS account and notes how many roles are associated with this prefix. If you need to

see all of these roles and their details, enter "Yes" on the detail prompt to have these roles listed out with specifics.

Creating a ROSA with HCP cluster using the CLI

When using the Red Hat OpenShift Service on AWS (ROSA) CLI, rosa, to create a cluster, you can select the default options to create the cluster quickly.
Prerequisites:

- You have completed the AWS prerequisites for ROSA with HCP.
- You have available AWS service quotas.
- You have enabled the ROSA service in the AWS Console.
- You have installed and configured the latest ROSA CLI (rosa) on your installation host. Run rosa version to see your currently installed version of the ROSA CLI. If a newer version is available, the CLI provides a link to download this upgrade.
- You have logged in to your Red Hat account by using the ROSA CLI.
- You have created an OIDC configuration.
- You have verified that the AWS Elastic Load Balancing (ELB) service role exists in your AWS account.

Steps:

1. Use one of the following commands to create your ROSA with HCP cluster:
   Note: When creating a ROSA with HCP cluster, the default machine Classless Inter-Domain Routing (CIDR) is 10.0.0.0/16. If this does not correspond to the CIDR range for your VPC subnets, add --machine-cidr <address_block> to the following commands. To learn more about the default CIDR ranges for Red Hat OpenShift Service on AWS, see CIDR range definitions.
   a. If you did not set environmental variables, run the following command:
      rosa create cluster --cluster-name=<cluster_name> \ <.>
      <.> Specify the name of your cluster. If your cluster name is longer than 15 characters, it will contain an autogenerated domain prefix as a subdomain for your provisioned cluster on openshiftapps.com. To customize the subdomain, use the --domain-prefix flag. The domain prefix cannot be longer than 15 characters, must be unique, and cannot be changed after cluster creation. <.> Optional: The --private argument is used to create private ROSA with HCP clusters. If you use this argument, ensure that you only use your private subnet ID for --subnet-ids. <.> By default, the cluster-specific Operator role names are prefixed with the cluster name and a random 4-digit hash. You can optionally specify a custom prefix to replace <cluster_name>-<hash> in the role names. The prefix is applied when you create the cluster-specific Operator IAM roles. For information about the prefix, see About custom Operator IAM role prefixes.
      Note: If you specified custom ARN paths when you created the

associated account-wide roles, the custom path is automatically detected. The custom path is applied to the cluster-specific Operator roles when you create them in a later step.

b. If you set the environmental variables, create a cluster with a single, initial machine pool, using either a publicly or privately available API, and a publicly or privately available Ingress by running the following command:
   rosa create cluster --private --cluster-name=<cluster_name> \
     --mode=auto --hosted-cp
   --operator-roles-prefix=$OPERATOR_ROLES_PREFIX \
     --oidc-config-id=$OIDC_ID --subnet-ids=$SUBNET_IDS

c. If you set the environmental variables, create a cluster with a single, initial machine pool, a publicly available API, and a publicly available Ingress by running the following command:
   rosa create cluster --cluster-name=<cluster_name> --mode=auto \
     --hosted-cp --operator-roles-prefix=$OPERATOR_ROLES_PREFIX \
     --oidc-config-id=$OIDC_ID --subnet-ids=$SUBNET_IDS

2. Check the status of your cluster by running the following command:
   rosa describe cluster --cluster=<cluster_name>
   The following State field changes are listed in the output as the cluster installation progresses: pending (Preparing account), installing (DNS setup in progress), installing, Ready.

   Note: If the installation fails or the State field does not change to ready after more than 10 minutes, check the installation troubleshooting documentation for details. For more information, see Troubleshooting installations. For steps to contact Red Hat Support for assistance, see Getting support for Red Hat OpenShift Service on AWS.
3. Track the progress of the cluster creation by watching the Red Hat OpenShift Service on AWS installation program logs. To check the logs, run the following command:
   rosa logs install --cluster=<cluster_name> --watch \ <.>
   <.> Optional: To watch for new log messages as the installation progresses, use the --watch argument.

Creating an admin user for quick access on Red Hat OpenShift Service on AWS

5 mins

If you want to be able to access your cluster immediately through a cluster-admin user, you can follow these steps. This is good if you need quick access to the

cluster, though the recommended approach is to use a formal identity provider to access the cluster (and then grant that user admin privileges, if desired).

What will you learn?

- How to access your cluster through a cluster-admin user

What do you need before starting?

- Met all prerequisites
- Deployed a cluster

Create an admin user

1. Run this command to create the admin user:
   rosa create admin --cluster=<cluster-name>

   You will see a response like the following:

W: It is recommended to add an identity provider to login to this cluster. See 'rosa create idp --help' for more information.
I: Admin account has been added to cluster 'my-rosa-cluster'. It may take up to a minute for the account to become active.
I: To login, run the following command:
oc login https://api.my-rosa-cluster.abcd.p1.openshiftapps.com:6443 \
--username cluster-admin \

2. --password FWGYL-2mkJI-00000-00000
3. Copy the login command returned to you in the previous step and paste that into your terminal. This should log you into the cluster via the CLI so you can start using the cluster.

$ oc login https://api.my-rosa-cluster.abcd.p1.openshiftapps.com:6443 \
> --username cluster-admin \
> --password FWGYL-2mkJI-00000-00000
Login successful.
You have access to 79 projects, the list has been suppressed. You can list all projects with ' projects'

4. Using project "default".
5. To check that you are logged in as the admin user you can run: oc whoami
6. You can also confirm by running the following command. Only a cluster-admin user can run this without errors: oc get all -n openshift-apiserver

7. You can now use the cluster as an admin user, though it is highly recommended to set up an identity provider (IdP).

You are now ready to set up an IdP.

Setting up an IdP for Red Hat OpenShift Service on AWS

5 mins

To log in to your cluster, we recommend that you set up an identity provider (IdP). The following procedure uses GitHub as an example IdP. See the full list of supported IdPs by Red Hat® OpenShift® Service on AWS (ROSA).

What will you learn?

- How to use an IdP to log in to your cluster

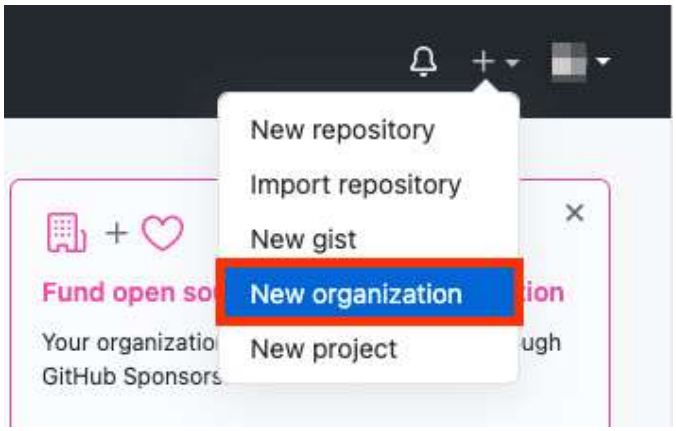
What do you need before starting?

- Met all prerequisites
- Created an admin user

Set up an IdP with GitHub

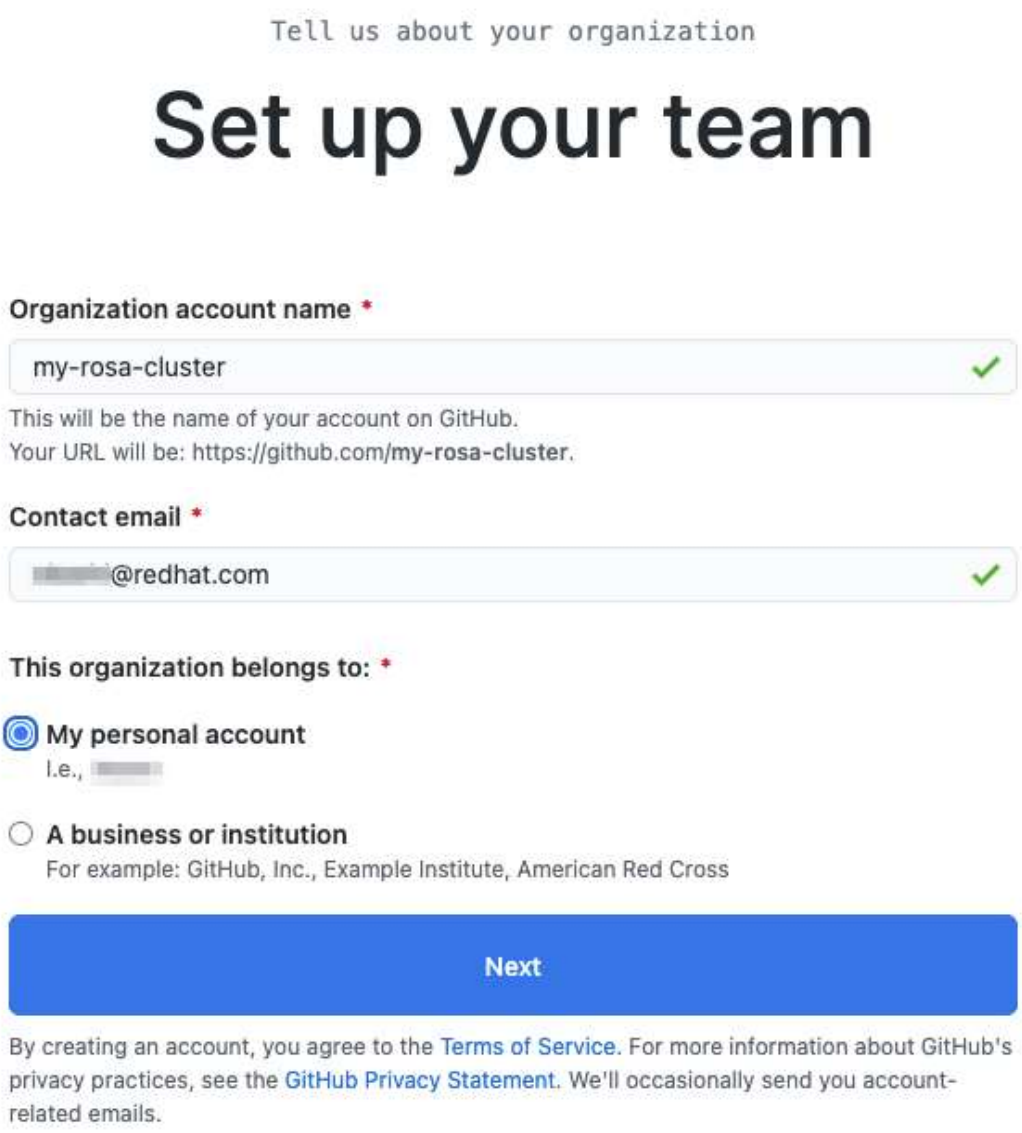
NOTE: To view all options run: rosa create idp --help

1. Log into your GitHub account.
2. You can either use an existing Organization that you're an admin of, or create a new one. If you already have one that you want to use, skip to step 7. Here we will create a new Organization for use with our new ROSA cluster. Click on the "+" icon in the top then click on "New Organization".



3. If you are asked to "Pick a plan for your team," choose the most applicable to you, or just click "Join for free" on the bottom left.

4. Choose a name for the organization, an email, and whether it is personal or business. Click Next.



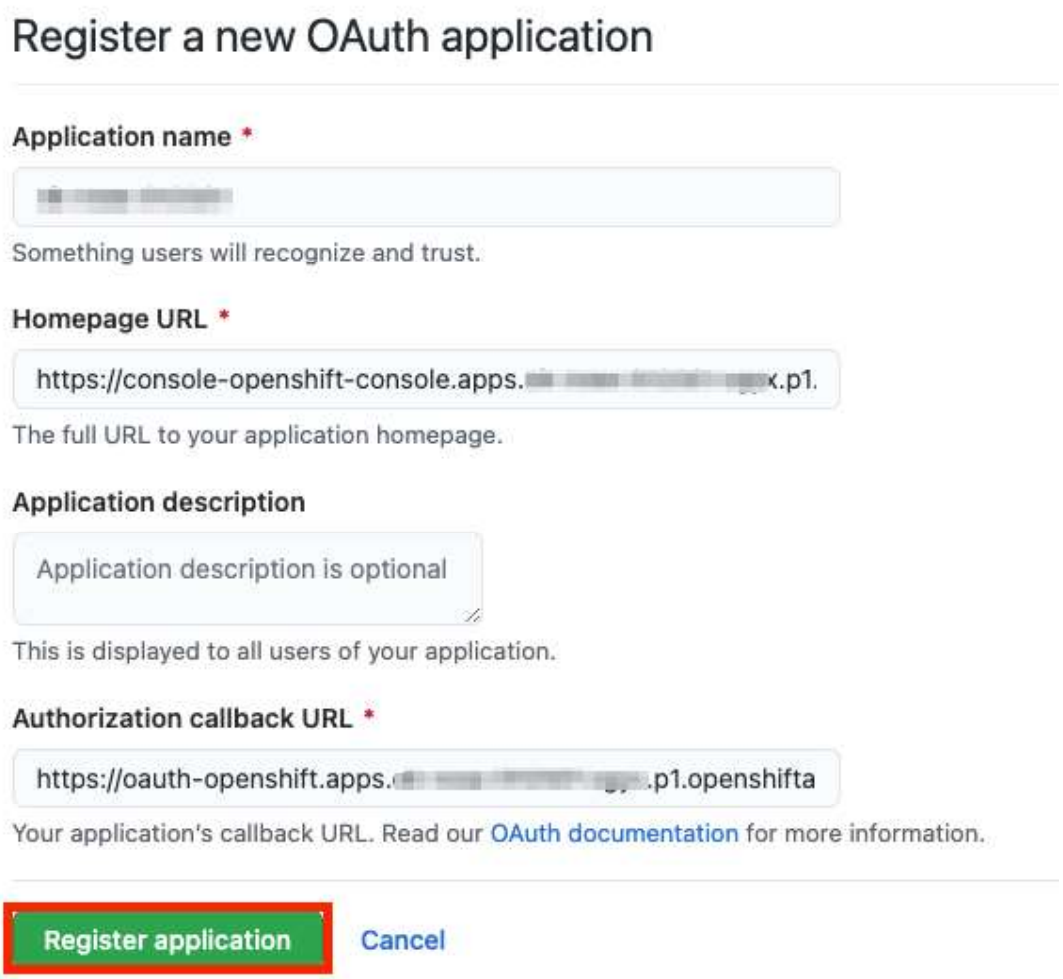
5. If you have other users that you want to grant access to your ROSA cluster you can add their GitHub IDs to the organization or you can add them later. We will click “Complete Setup” without adding anyone else.
6. You can fill in the requested information on the following page or just click “Submit” at the bottom.
7. Go back to the terminal and enter the following command to set up the GitHub IdP.
   rosa create idp --cluster=<cluster-name> --interactive
8. Enter the following values that are in bold below:
   - Type of identity provider: github

- Identity Provider Name: rosa-github (Or this can be any name you choose)
- Restrict to members of: organizations
- GitHub organizations: my-rosa-cluster (or enter the name of your org)

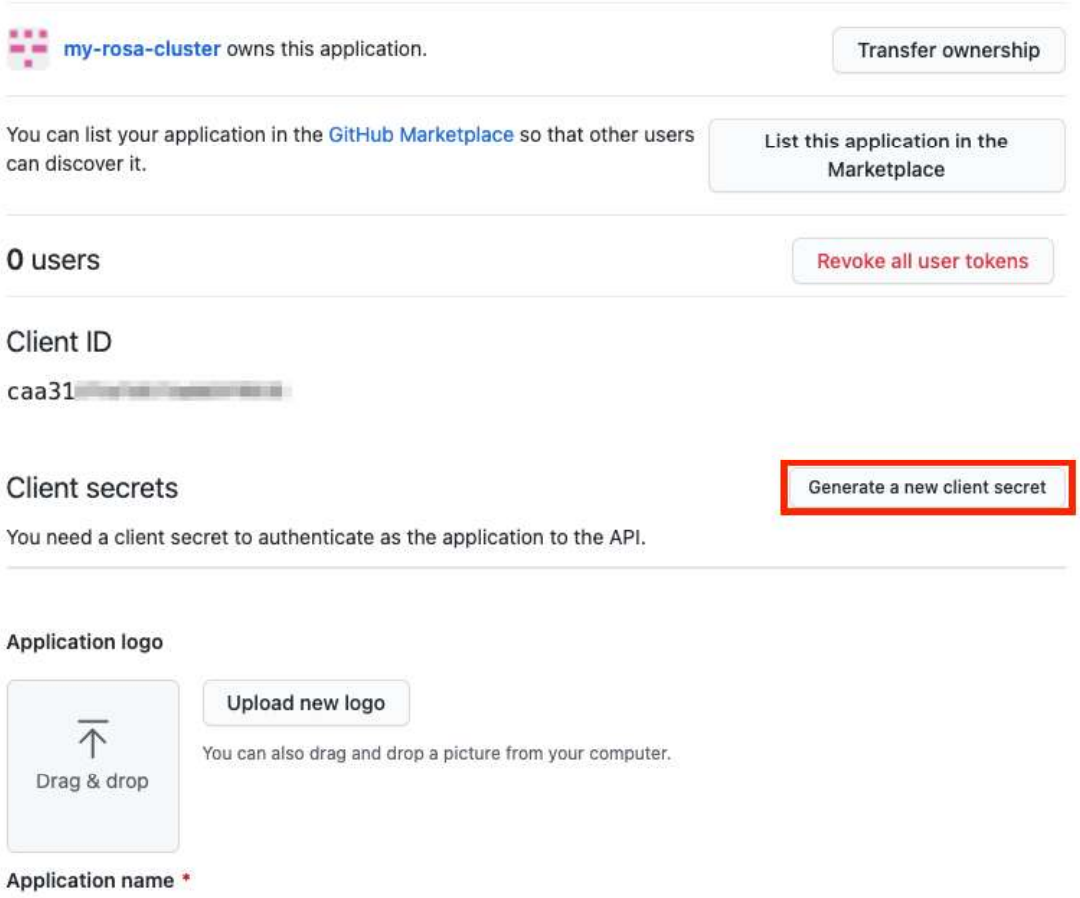
9. The command line interface (CLI) will provide you with a link. Copy and paste that into a browser and press enter. This will pre-fill the required information for you in order to register this application for OAuth. You don't need to modify any of the information.

```
I: Interactive mode enabled.
Any optional fields can be left empty and a default will be selected.
? Type of identity provider: github
[? Identity provider name: rosa-github
? Restrict to members of: organizations
[? GitHub organizations: my-rosa-cluster
? To use GitHub as an identity provider, you must first register the application:
  - Open the following URL:
    https://github.com/organizations/my-rosa-cluster/settings/applications/new?oauth_application%5Bcallback_url%5D=h
uth2callback%2Frosa-github&oauth_application%5Bname%5D=                    &oauth_application%5Burl%5D=https%3A%2F%2Fconso
  - Click on 'Register application'
? Client ID: [? for help]
```

10. Click "Register application."



11. On the next page it will show you a “Client ID.” Copy this and paste it back into the terminal where it asks for “Client ID.” DO NOT CLOSE THIS TAB.
12. The CLI now asks for a “Client Secret,” so go back in your browser and click on “Generate a new client secret” near the middle of the page towards the right.



13. A secret will be generated for you. Make sure to copy it as it will never be visible again.
14. Paste it into the terminal where the CLI is asking for the Client Secret and press enter.
15. Leave "GitHub Enterprise Hostname" blank.
16. Select “claim.” (For more details see Identity provider parameters)
17. Then the IdP will be created but can take up to 1 minute for the configuration to land onto your cluster.

Your inputs should look similar to the following:

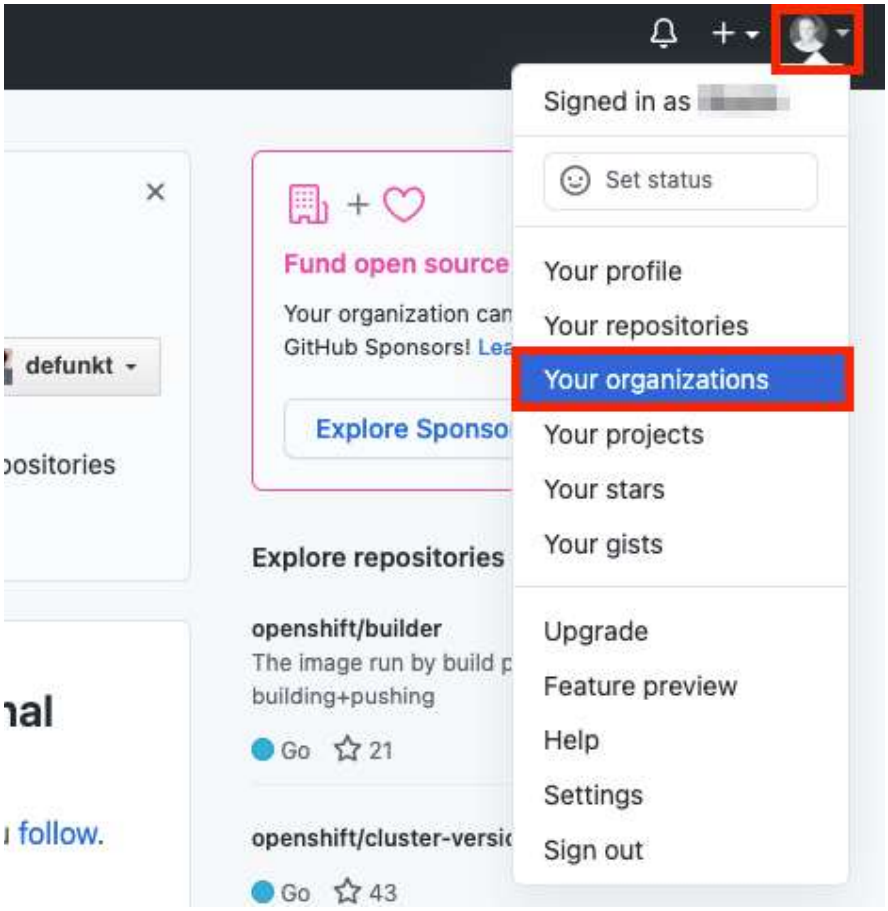
```
I: Interactive mode enabled.
Any optional fields can be left empty and a default will be selected.
? Type of identity provider: github
? Identity provider name: rosa-github
? Restrict to members of: organizations
? GitHub organizations: my-rosa-cluster
? To use GitHub as an identity provider, you must first register the application:
  - Open the following URL:
    https://github.com/organizations/my-rosa-cluster/settings/applications/new?oauth_application%5Bcallback_url%5D=
https%3A%2F%2Foauth-openshift.apps.[redacted].p1.openshiftapps.com%2Foauth2callback%2Frosa-github&oauth_ap
plication%5Bname%5D=[redacted]L&oauth_application%5Burl%5D=https%3A%2F%2Fconsole-openshift-console.apps.[redacted]
[redacted].p1.openshiftapps.com
  - Click on 'Register application'
? Client ID: caa31[redacted]
? Client Secret: [? for help] ****************************************
? GitHub Enterprise Hostname (optional):
? Mapping method: claim
I: Configuring IDP for cluster '[redacted]'
I: Identity Provider 'rosa-github' has been created.
   It will take up to 1 minute for this configuration to be enabled.
   To add cluster administrators, see 'rosa create user --help'.
   To login into the console, open https://console-openshift-console.apps.[redacted].p1.openshiftapps.com
and click on rosa-github.
```

18. Copy and paste the link returned at the end into your browser and you should see the IdP we just set up available. If you've followed this tutorial, it is called “rosa-github”. You can click on this and use your GitHub credentials to access the cluster.

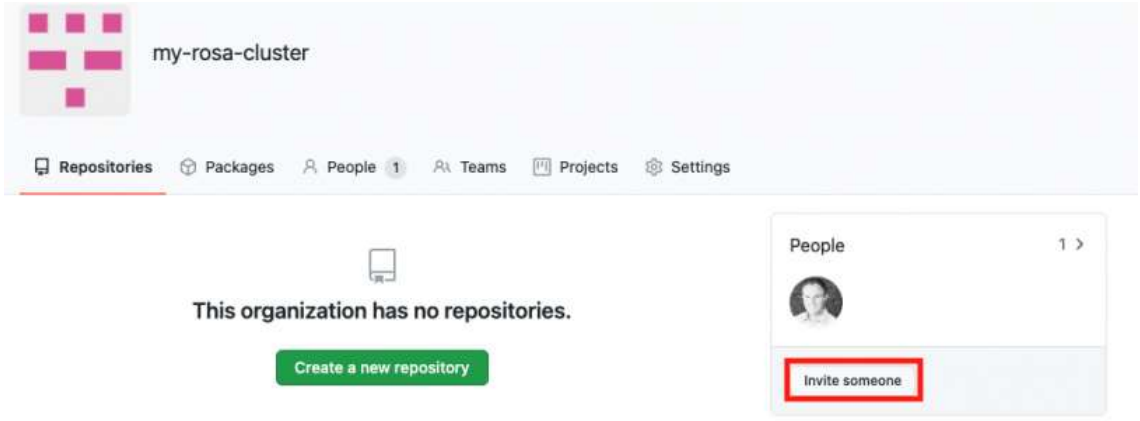
Grant access to the cluster

1. In order to grant access to other users of your cluster, you will need to add their GitHub user ID to the GitHub Organization used for this cluster. If you are following the tutorial, go to “Your organizations” page.

2. Click on your profile icon > Your organizations > {your organization name}. In our case, the organization name is “my-rosa-cluster”.



3. Click on the “Invite someone” button.



4. Enter their GitHub ID, select the correct user, and click “Invite.”
5. Once the other user accepts the invitation, they will be able to log into the ROSA cluster via the console link and use their GitHub credentials.

You are now ready to grant admin rights.

Granting cluster admin rights to users in Red Hat OpenShift Service on AWS

5 mins

Cluster-admin rights are not automatically granted to users that you add to the cluster. If there are users that you want to grant this level of privilege to, you will need to manually add cluster-admin rights to each user.

What will you learn?

- How to grant cluster admin rights to other users

What do you need before starting?

- Met all prerequisites
- Deployed cluster
- Created admin user

Grant cluster-admin rights

Let's start off with granting cluster-admin rights to ourselves using the GitHub username we created for the cluster in the Set up an IdP resource. There are two ways to do this; either from the Red Hat® OpenShift® on AWS command line interface (CLI) or the OpenShift Cluster Manager (OCM) web user interface (UI).

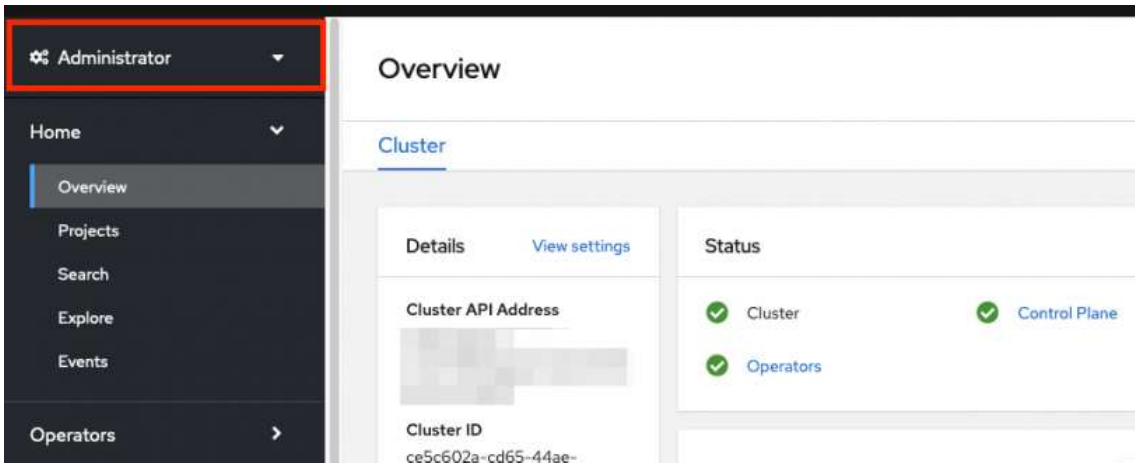
1. Via rosa CLI
    a. Assuming you are the user who created the cluster, you can grant cluster-admin to a user (or our GitHub user) by running:
    rosa grant user cluster-admin --user <idp_user_name> --cluster=<cluster-name>
    b. Verify that we were added as a cluster-admin by running:
    rosa list users --cluster=<cluster-name>
    You should see your GitHub ID of the user listed.

rosa list users --cluster=my-rosa-cluster
ID GROUPS

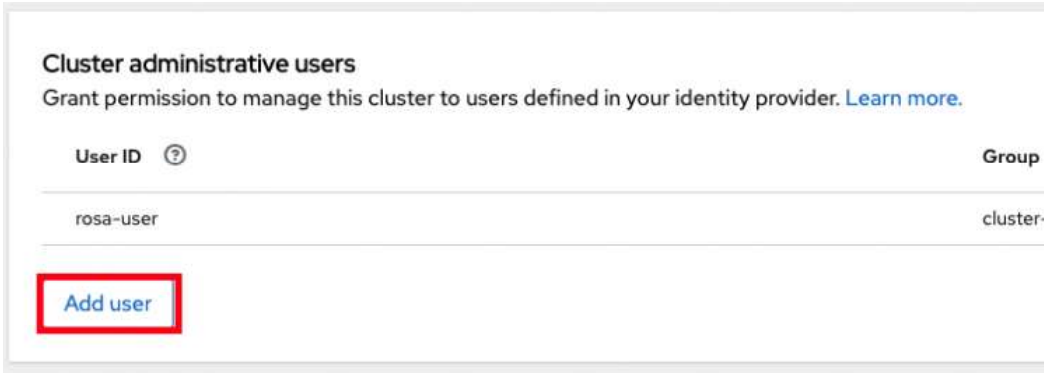
rosa-user cluster-admin

    a. Logout and log back into the cluster to see a new perspective with the “Administrator Panel”. (You might need to try an

Incognito/Private window).

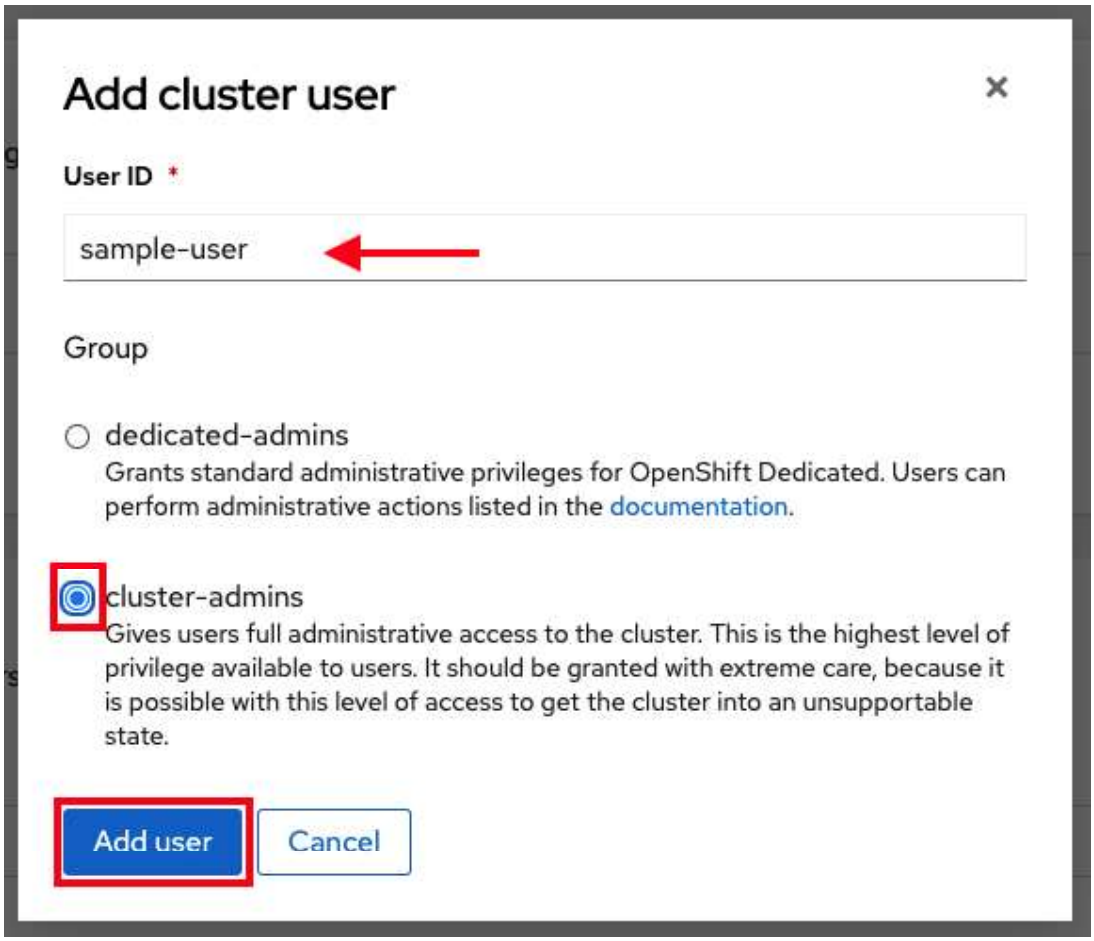


b. You can also test this by running the following command. Only a cluster-admin user can run this without errors:
   oc get all -n openshift-apiserver
c. Via OCM UI
   i. Log into OCM from https://console.redhat.com/openshift
   ii. Select your cluster.
   iii. Click on the "Access Control" tab.
   iv. Towards the bottom in the "Cluster Administrative Users" section click on "Add User."



   v. On the pop-up screen enter the person's user ID (in our example the GitHub ID).

Select whether you want to grant them cluster-admin or dedicated-admin.



Granting dedicated-admin

ROSA has the option to set a “dedicated-admin” role, which means to create an admin user that can complete most administrative tasks but is slightly limited to prevent anything damaging. It is best practice to use dedicated-admin when elevated privileges are needed. You can read more about it here.

1. Enter the following command to promote your user to a dedicated-admin:
   rosa grant user dedicated-admin --user <idp_user_name> --cluster=<cluster-name>
2. Enter the following command to verify that your user now has dedicated-admin access
   oc get groups dedicated-admins
3. You can also grant dedicated-admin rights via the OCM UI as described in the cluster-admin section, but just select the “dedicated-admins” radio button instead.

You are now ready to access your cluster.

Revoking access

In the event that you need to revoke cluster-admin or dedicated-admin access, it can be done through the following steps:

1. From the ROSA CLI, enter in the following command to remove cluster-admin users:

rosa revoke user cluster-admin --user=<idp_user_name> --cluster=<cluster_name>

   a. Replace <idp_user_name> and <cluster_name> with the name of the identity provider user and your cluster name.

2. To remove dedicated-admin users, use this command instead:

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

Once submitted, check the admin list to ensure the user is no longer listed using the following command:

rosa list users --cluster=<cluster_name>

Accessing a Red Hat OpenShift Service on AWS cluster using CLI or web console

There are multiple ways to interact with your cluster. You can connect to it via the command-line-interface (CLI) or via the Web Console. We will review both options below.

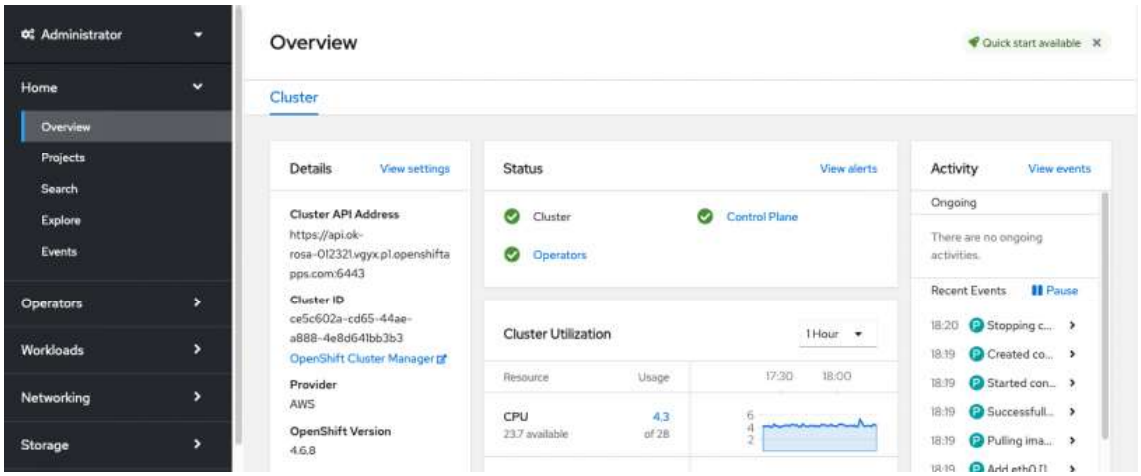
What will you learn?

- How to access your cluster via the Web Console
- How to access your cluster via the CLI

What do you need before starting?

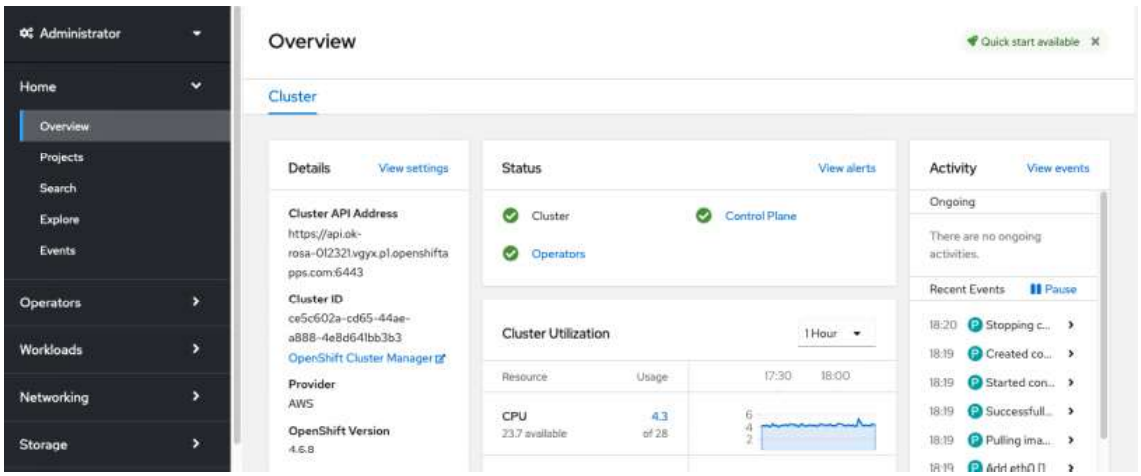
- Met prerequisites
- Deployed cluster
- Created admin user

Access the cluster via the console



1. Enter the console URL into your web browser. If you need to retrieve it you can run:
   rosa describe cluster -c <cluster-name> | grep Console
2. Click on your identity provider (IdP). In this learning path, we created “rosa-github.”

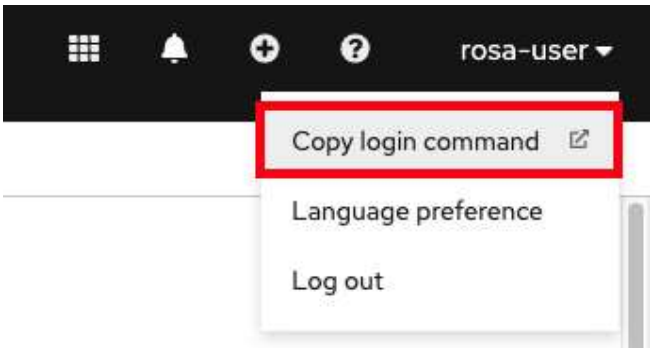
3. Enter your GitHub user credentials (or other credentials if not using GitHub).
4. You should now be logged in. If you've followed this learning path in order, you will be a cluster-admin and should see a web console like the following with the "Administrator" panel visible.



Access the cluster via the CLI

In order to access the cluster via the CLI you must have the oc CLI installed, which was completed in the prerequisites for this learning path.

1. Log into the web console as stated above.
2. Click on your username in the top right corner
3. Click on “Copy Login Command”



4. This will open a new tab with a choice of IdP. Click on the IdP you want to use, in our case, “rosa-github.”
5. A new tab will open. Click on “Display token.”
6. You will see a page like the following:

7. oc login command and paste it into your terminal. Press enter.
8. We can confirm that we are now the user we logged in with by running:

oc whoami

Managing worker nodes in Red Hat OpenShift Service on AWS clusters

10 mins

When using your cluster, there may be times when you need to change aspects of your worker nodes like scaling, changing the type, adding labels or taints, to name a few.

Most of these things are done through the use of machine pools in Red Hat® OpenShift® Service on AWS (ROSA). Think of a machine pool as a “template” for

the kinds of machines that make up the worker nodes of your cluster. A machine pool allows users to manage many machines as a single entity. Every ROSA cluster has a "Default" machine pool created when the cluster is created. If you'd like to learn more see About machine pools and autoscaling.

What will you learn?

- How to create a machine pool using the rosa command line interface (CLI)
- How to create a machine pool using the OpenShift Cluster Management (OCM) user interface (UI)
- How to scale worker nodes
- How to add node labels
- How to mix different node types

What do you need before starting?

- Deployed cluster
- Granted admin rights
- Accessed cluster

Creating a machine pool using the rosa CLI

1. Run:
2. rosa create machinepool --cluster=<cluster-name> --name=<machinepool-name> --replicas=<number-nodes>
3. For example:

```
$ rosa create machinepool --cluster=my-rosa-cluster --name=new-mp --replicas=2
I: Machine pool 'new-mp' created successfully on cluster 'my-rosa-cluster'
```

4. I: To view all machine pools, run 'rosa list machinepools -c my-rosa-cluster'

5. Sometimes it is beneficial to add node label(s) and/or taints. One use case is to target certain workloads to specific nodes. Let's say we want to run our database on specific nodes. We can add node labels to the worker nodes when we create a new machine pool using the CLI.
6. rosa create machinepool --cluster=<cluster-name> --name=<machinepool-name> --replicas=<number-nodes> --labels='<key=pair>'

7. For example:

```
$ rosa create machinepool --cluster=my-rosa-cluster --name=db-nodes-mp
--replicas=2 --labels='app=db','tier=backend'
```

8. I: Machine pool 'db-nodes-mp' created successfully on cluster 'my-rosa-cluster'

9. This will create an additional 2 nodes that you can manage as one unit and also assign them the labels shown.
10. Now run the following to see the new machine pool created along with the labels we gave.
11. rosa list machinepools --cluster=<cluster-name>

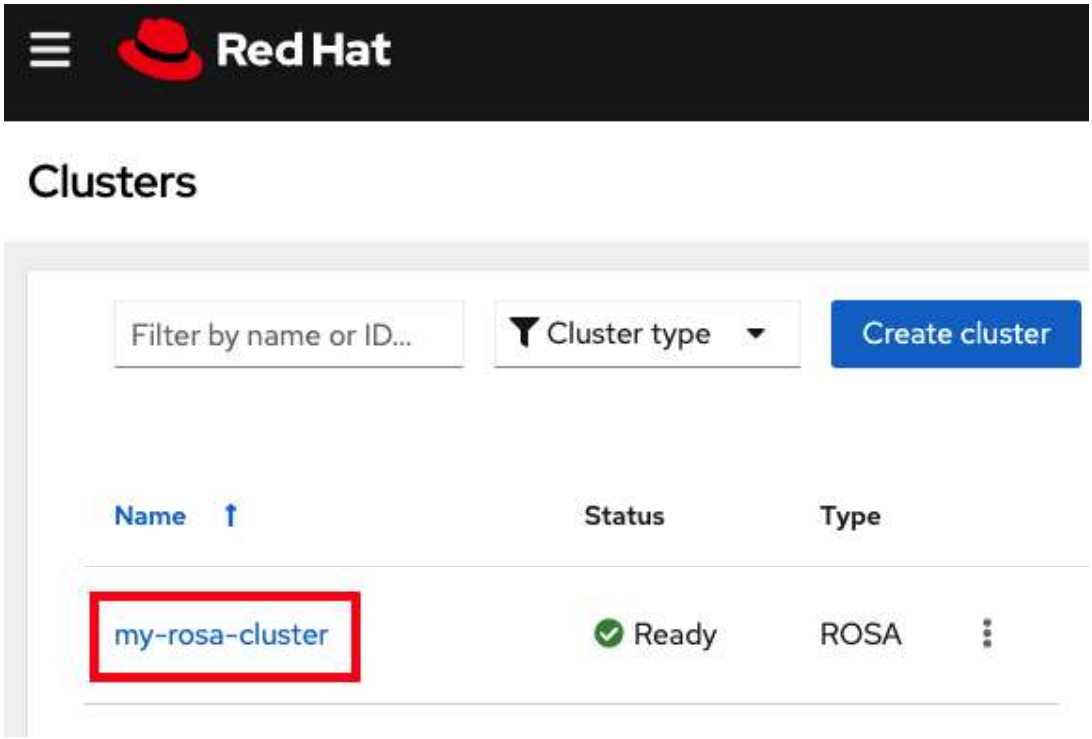
12. You will see a list of the machine pools like this:

```
ID                AUTOSCALING REPLICAS INSTANCE TYPE LABELS TAINTS AVAILABILITY ZONES
Default           No          2        m5.xlarge                   us-east-1a
new-mp            No          2        m5.xlarge                   us-east-1a
```

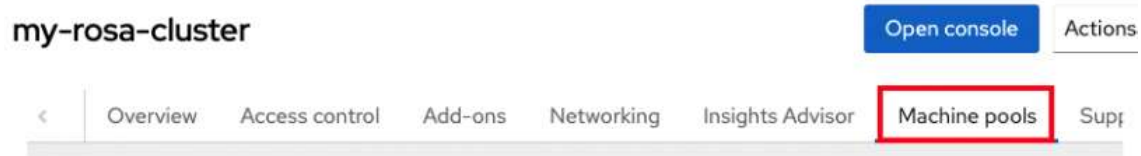
13. db-nodes-mp No 2 m5.xlarge app=db, tier=backend us-east-1a
14. ...

Creating a machine pool with the OCM UI

1. In the OCM user interface, click on your cluster.

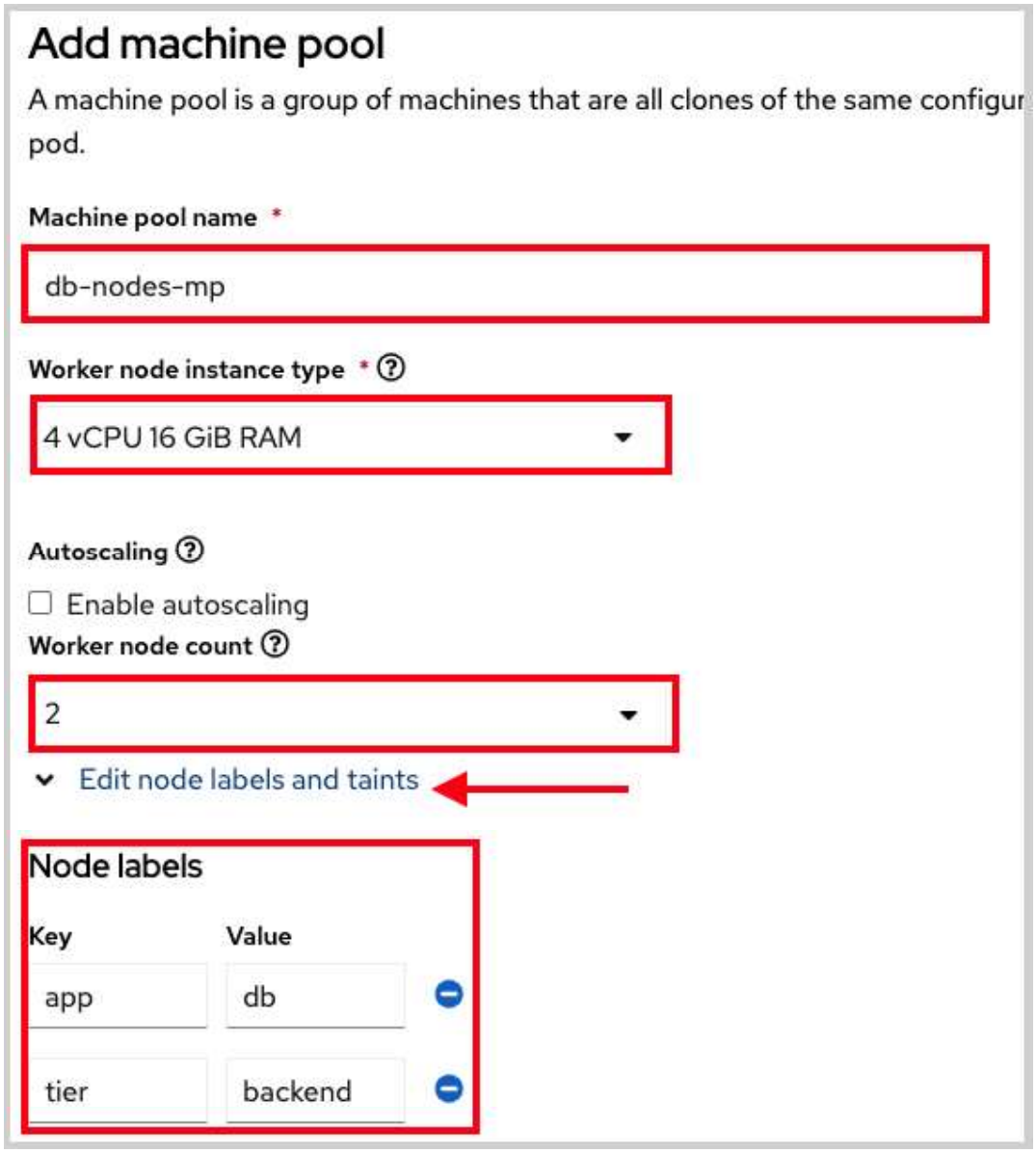


2. Then click on the "Machine Pools" tab.



3. Click the "Add Machine pool" button.

4. Fill in the desired configuration. At this point you can also expand the "Edit node labels and taints" section to add node labels and taints to the nodes in this machine pool.



5. You will now see the new machine pool you created in the UI.

Scaling worker nodes

1. To scale the number of worker nodes, we need to edit the machine pool they belong to. The default machine pool is called “Default” which is created with every cluster. We could also use the one we created before.
2. We should see the “Default” pool that is created with each cluster.

rosa list machinepools --cluster=<cluster-name>

...

You will see a response like:
ID AUTOSCALING REPLICAS INSTANCE TYPE LABELS TAINTS AVAILABILITY ZONES
Default No 2 m5.xlarge us-east-1a
...

3. To scale this out to 3 nodes run

rosa edit machinepool --cluster=<cluster-name> --replicas=<number-nodes> <machinepool-name>
...
For example:
rosa edit machinepool --cluster=my-rosa-cluster --replicas 3 Default
...

4. Run the following to see that it has taken effect:

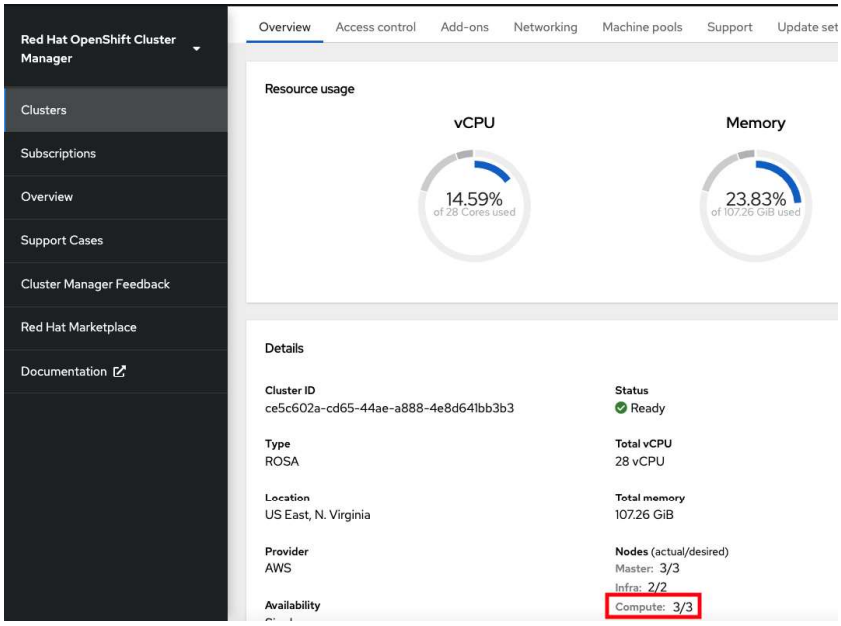
rosa describe cluster --cluster=<cluster-name> | grep Compute

...
You will see a response showing 3 compute nodes:
$ rosa describe cluster --cluster=my-rosa-cluster | grep Compute
- Compute: 3 (m5.xlarge)
...

5. (Optional) One can also scale the cluster from the OCM UI by clicking on the "three dots" on the right of the machine pool you want to edit and clicking "scale".

6. We can also confirm this by accessing OCM and selecting the cluster.

7. On the overview tab, scroll down to the middle section under details you will see Compute listing "3/3".



Adding node labels

NOTE: Labels or taints cannot be added to the "Default" machine pool (yet).

1. Adding node label(s) can be achieved by the following command

rosa edit machinepool --cluster=<cluster-name> --replicas=<number-nodes> --labels='key=value' <machinepool-name>

...

For example if we wanted to add 2 labels to the new machine pool we created:

rosa edit machinepool --cluster=my-rosa-cluster --replicas=2 --labels 'foo=bar','baz=one' new-mp

...

2. This command replaces all machine pool configurations with what is defined. If you just want to add another label and keep the old, you must state all the labels, otherwise it will replace anything existing with the one you had wanted to add. Similarly, if you want to delete a label only state the ones you want, excluding the one you want to delete.

Mixing different node types

1. You can also mix different worker node machine types in the same cluster by using new machine pools. You cannot change the node type of a machine pool once created, but we can create a new machine pool with different nodes by adding the --instance-type flag.
2. If we take the use case above (database nodes) but instead wanted to have a different node type when creating it, we would have run:

rosa create machinepool --cluster=<cluster-name> --name=<mp-name> --replicas=<number-nodes> --labels='<key-pair>' --instance-type=<type>

…

For example:

rosa create machinepool --cluster=my-rosa-cluster --name=db-nodes-large-mp --replicas=2 --labels='app=db','tier=backend' --instance-type=m5.2xlarge

…

3. If you'd like to see all the instance types available you can run:

rosa list instance-types

…

4. Or to make the decisions step-by-step, then use the --instance-type

rosa create machinepool -c <cluster-name> --interactive

…

```
[? Machine pool name: large-nodes-pool
[? Enable autoscaling (optional): No
[? Replicas: 3
 ? Instance type:  [Use arrows to move, type to filter, ? for more help]
 > m5.xlarge
   r5.xlarge
   r5.2xlarge
   m5.2xlarge
   c5.2xlarge
   r5.4xlarge
   m5.4xlarge
```

5. List the machine pools to see the new larger instance type.

rosa list machinepools -c <cluster-name>

…

```
ID                AUTOSCALING  REPLICAS  INSTANCE TYPE  LABELS                 TAINTS    AVAILABILITY ZONES
default           No           3         m5.xlarge                                       us-east-1a
db-nodes-large--mp No          2         m5.2xlarge     app=db, tier=backend             us-east-1a
db-nodes-mp       No_          2         m5.xlarge      app=db, tier=backend             us-east-1a
```

You are now ready to set up cluster autoscaling.

Enable cluster autoscaling in Red Hat OpenShift Service on AWS

Autoscaling can refer to one of the following:

- Horizontal pod autoscaler - whereby Kubernetes will automatically create more or remove pods of an application to handle an increase/decrease in workload, though total resources available to the cluster will remain unchanged.
- Cluster autoscaler - This is where more worker nodes will be added or removed from the cluster based on pods failing due to insufficient resources thereby affecting the total number of resources available.

We will focus on the second definition as it relates to Red Hat® OpenShift® Service on AWS (ROSA).

NOTE: Cluster autoscaling can also be enabled at cluster creation time using the --enable-autoscaling flag. This will enable autoscaling on the "Default" machine pool. It can also be enabled when creating a machine pool.

What will you learn?

- How to add or remove worker nodes

What do you need before starting?

- Deployed your cluster
- Granted admin rights
- Accessed cluster

Setting up cluster autoscaling

1. Autoscaling is set per machine pool definition. To find out which machine pools are available in our cluster run:
   rosa list machinepools -c <cluster-name>
   You will see a response like:

1. Now run the following to add autoscaling to that machine pool.
   rosa edit machinepool -c <cluster-name> --enable-autoscaling <machinepool-name> --min-replicas=<num> --max-replicas=<num>
   For example:
   rosa edit machinepool -c my-rosa-cluster --enable-autoscaling Default --min-replicas=2 --max-replicas=4
   This will create an autoscaler for the worker nodes to scale between 2 and 4 nodes depending on the resources.
   The cluster autoscaler increases the size of the cluster when there are pods that failed to schedule on any of the current nodes due to insufficient resources or when another node is necessary to meet deployment needs. The cluster autoscaler does not increase the cluster resources beyond the limits that you specify. The cluster autoscaler decreases the size of the cluster when some nodes are consistently not needed for a significant period, such as when it has low resource use and all of its important pods can fit on other nodes.
2. This can also be done via the OpenShift® Cluster Manager (OCM) user interface (UI). You might have noticed a checkbox at machine pool creation time for "Enable autoscaling".
3. Lastly, autoscaling can also be added to an existing machine pool via the UI by going to the "Machine pools" tab in OCM for your cluster > clicking on the "three dots" at the right for the machine pool > "Scale" > "Enable autoscaling".

4. Run the following to confirm that autoscaling was added.
   rosa list machinepools -c <cluster-name>
   You will see a response like:

ID AUTOSCALING REPLICAS INSTANCE TYPE LABELS TAINTS AVAILABILITY ZONES

5. Default Yes 2-4 m5.xlarge us-east-1a
6. Copy snippet

You are now ready to upgrade your cluster via the CLI, OCM user interface, or through automated upgrades.

Upgrading a Red Hat OpenShift Service on AWS cluster using CLI or console

All upgrades are fully executed by Red Hat® OpenShift® Service on AWS (ROSA) for your cluster, meaning that you won't need to execute any commands or make changes to the cluster. You do have choices for scheduling them.
There are 3 ways to upgrade your cluster.

- Manually via the command line interface (CLI) - Start a one-time immediate upgrade or schedule a one-time upgrade for a future date/time.
- Manually via the OpenShift Cluster Manager (OCM) user interface (UI) - Start a one-time immediate upgrade or schedule a one-time upgrade for a future date/time.
- Automated upgrades - Set an upgrade window for recurring y-stream upgrades whenever a new version is available without needing to manually schedule it (Ex: Saturday at 06:00 UTC). Minor versions have to be manually scheduled.

We will go through each of these three scenarios.
You can always use --help for more details like:
rosa upgrade cluster --help

What will you learn?

- How to schedule a one-time immediate or future upgrade using the CLI
- How to schedule a one-time immediate or future upgrade using the OCM UI
- How to set an automated y-stream upgrade window for major versions

What do you need before starting?

- Deployed your cluster
- Granted admin rights
- Accessed cluster

Upgrade manually via the CLI

Check if there is an upgrade available by running the following command:
Inline - rosa list upgrade -c <cluster-name>
You will get a list that shows the available version and the current version of your cluster. For example:

1.

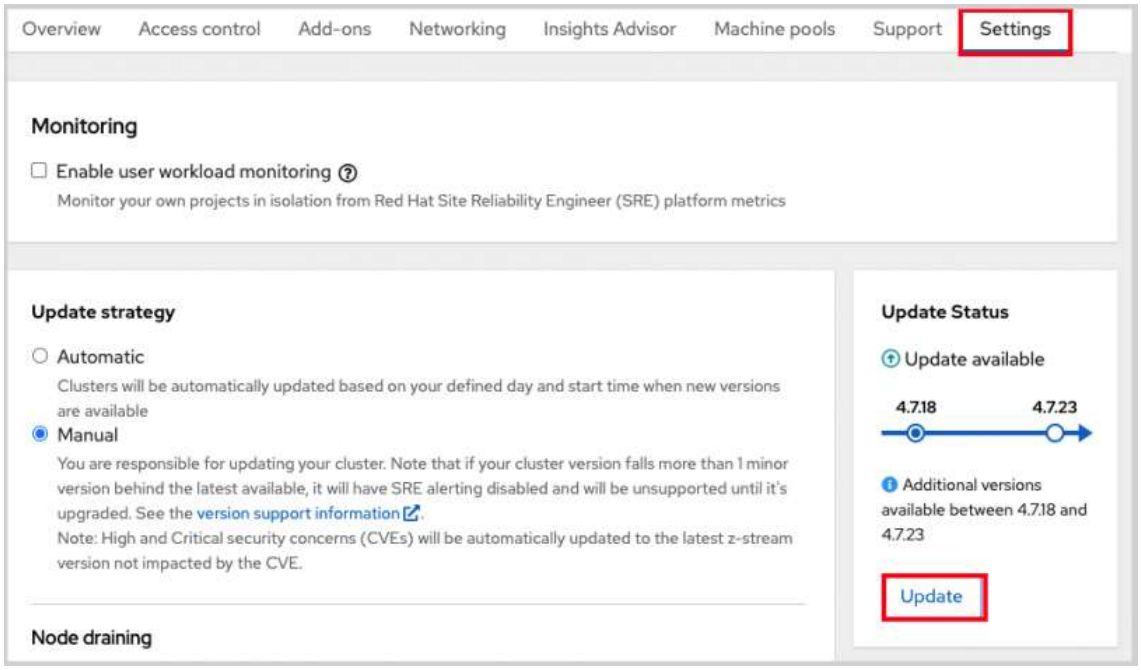
```
$ rosa list upgrade -c <cluster-name>

    VERSION  NOTES

    4.10.20   recommended

    4.10.18
```

2. In our example we see that version 4.10.18 is available and so is 4.10.20.
3. Upgrade the cluster to the latest version by running:
   rosa upgrade cluster -c my-rosa-cluster --version 4.10.20
4. This will schedule the cluster for upgrade within the hour. It will take some time to complete.
5. You can also schedule the upgrade for a later date/time by using the <--schedule-date and --schedule-time flags.

Upgrade manually via OCM UI

1. To perform an upgrade via the UI, log into OCM and select the cluster you want to upgrade.
2. Click on the “Settings” tab.

3. You will see if an upgrade is available and if so, click on the “Update” button.



4. This will open a window allowing you to select the version to upgrade to.
5. You can then schedule a time for the upgrade or begin it immediately.

Set up automatic updates

1. To set up your cluster for recurring upgrades, log into OCM and select the cluster you want to upgrade.
2. Click on the “Update Settings” tab.
3. Under “Update Strategy” click the “Automatic” radio button.
4. This will open up options for a day of the week and time that you can set for the update to occur.
5. Select a “Grace period” for allowing the nodes to gracefully drain before forcing the pod eviction under “Node draining”.
6. Click Save.

You have officially set up, deployed, and customized your first cluster. In the next resource, we’ll show you how to delete a cluster if the need arises.

All upgrades are fully executed by Red Hat® OpenShift® Service on AWS (ROSA) for your cluster, meaning that you won't need to execute any commands or make changes to the cluster. You do have choices for scheduling them.

There are 3 ways to upgrade your cluster.

- Manually via the command line interface (CLI) - Start a one-time immediate upgrade or schedule a one-time upgrade for a future date/time.
- Manually via the OpenShift Cluster Manager (OCM) user interface (UI) - Start a one-time immediate upgrade or schedule a one-time upgrade for a future date/time.

- Automated upgrades - Set an upgrade window for recurring y-stream upgrades whenever a new version is available without needing to manually schedule it (Ex: Saturday at 06:00 UTC). Minor versions have to be manually scheduled.

We will go through each of these three scenarios.
You can always use --help for more details like:
rosa upgrade cluster --help

What will you learn?

- How to schedule a one-time immediate or future upgrade using the CLI
- How to schedule a one-time immediate or future upgrade using the OCM UI
- How to set an automated y-stream upgrade window for major versions

What do you need before starting?

- Deployed your cluster
- Granted admin rights
- Accessed cluster

Upgrade manually via the CLI

Check if there is an upgrade available by running the following command:
Inline - rosa list upgrade -c <cluster-name>
You will get a list that shows the available version and the current version of your cluster. For example:

1.

$ rosa list upgrade -c <cluster-name>

VERSION NOTES

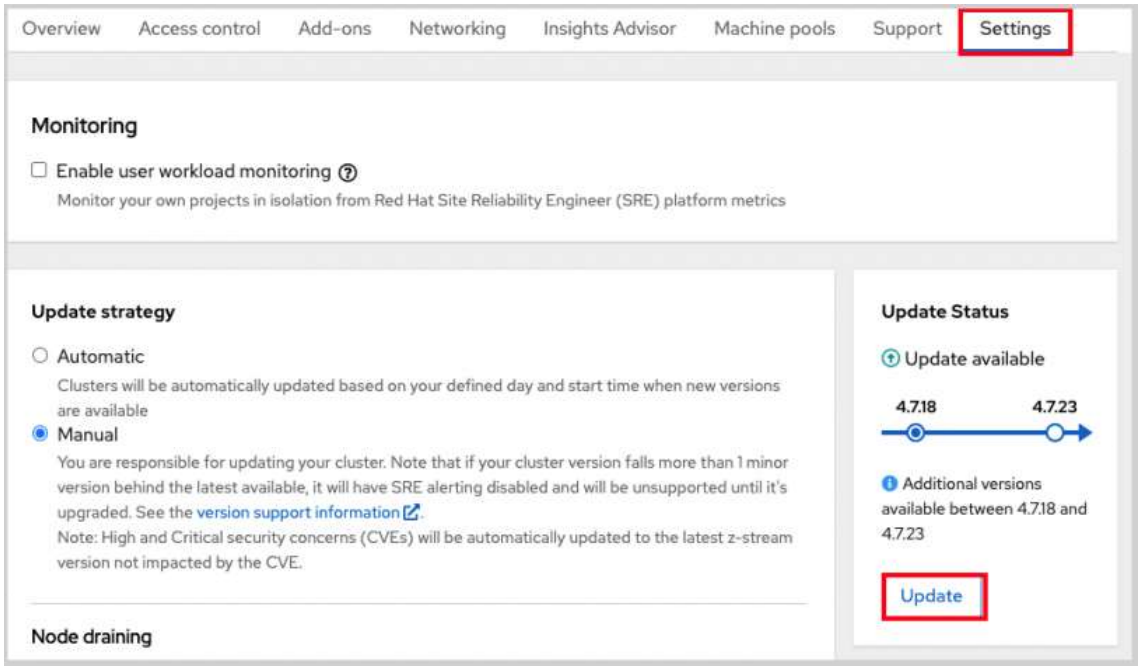
4.10.20 recommended

4.10.18

2. In our example we see that version 4.10.18 is available and so is 4.10.20.
3. Upgrade the cluster to the latest version by running:
   rosa upgrade cluster -c my-rosa-cluster --version 4.10.20
4. This will schedule the cluster for upgrade within the hour. It will take some time to complete.
5. You can also schedule the upgrade for a later date/time by using the <--schedule-date and --schedule-time flags.

Upgrade manually via OCM UI

1. To perform an upgrade via the UI, log into OCM and select the cluster you want to upgrade.
2. Click on the “Settings” tab.
3. You will see if an upgrade is available and if so, click on the “Update” button.



4. This will open a window allowing you to select the version to upgrade to.
5. You can then schedule a time for the upgrade or begin it immediately.

Set up automatic updates

1. To set up your cluster for recurring upgrades, log into OCM and select the cluster you want to upgrade.
2. Click on the “Update Settings” tab.
3. Under “Update Strategy” click the “Automatic” radio button.
4. This will open up options for a day of the week and time that you can set for the update to occur.
5. Select a “Grace period” for allowing the nodes to gracefully drain before forcing the pod eviction under “Node draining”.
6. Click Save.

You have officially set up, deployed, and customized your first cluster. In the next resource, we’ll show you how to delete a cluster if the need arises.

All upgrades are fully executed by Red Hat® OpenShift® Service on AWS (ROSA) for your cluster, meaning that you won't need to execute any commands or make changes to the cluster. You do have choices for scheduling them.

There are 3 ways to upgrade your cluster.

- Manually via the command line interface (CLI) - Start a one-time immediate upgrade or schedule a one-time upgrade for a future date/time.
- Manually via the OpenShift Cluster Manager (OCM) user interface (UI) - Start a one-time immediate upgrade or schedule a one-time upgrade for a future date/time.
- Automated upgrades - Set an upgrade window for recurring y-stream upgrades whenever a new version is available without needing to manually schedule it (Ex: Saturday at 06:00 UTC). Minor versions have to be manually scheduled.

We will go through each of these three scenarios.
You can always use --help for more details like:
rosa upgrade cluster --help

What will you learn?

- How to schedule a one-time immediate or future upgrade using the CLI
- How to schedule a one-time immediate or future upgrade using the OCM UI
- How to set an automated y-stream upgrade window for major versions

What do you need before starting?

- Deployed your cluster
- Granted admin rights
- Accessed cluster

Upgrade manually via the CLI

Check if there is an upgrade available by running the following command:
Inline - rosa list upgrade -c <cluster-name>
You will get a list that shows the available version and the current version of your cluster. For example:

1.

$ rosa list upgrade -c <cluster-name>

VERSION NOTES

4.10.20 recommended

4.10.18

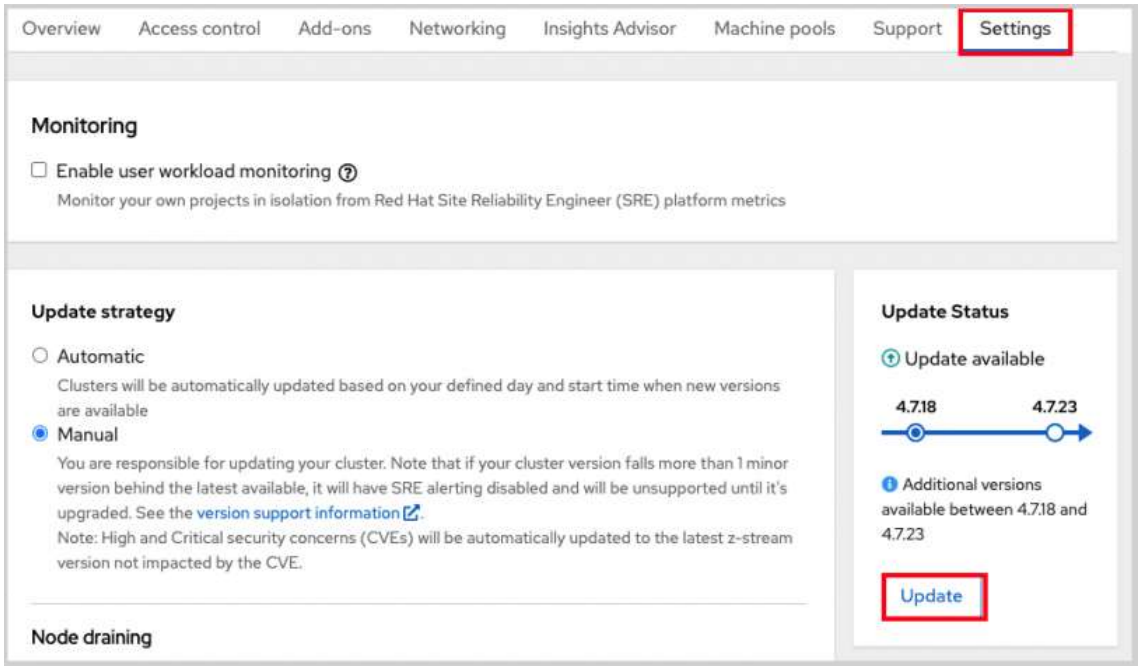
2. ...

3. In our example we see that version 4.10.18 is available and so is 4.10.20.

4. Upgrade the cluster to the latest version by running:
   rosa upgrade cluster -c my-rosa-cluster --version 4.10.20
5. This will schedule the cluster for upgrade within the hour. It will take some time to complete.
6. You can also schedule the upgrade for a later date/time by using the <--schedule-date and --schedule-time flags.

Upgrade manually via OCM UI

1. To perform an upgrade via the UI, log into OCM and select the cluster you want to upgrade.
2. Click on the “Settings” tab.
3. You will see if an upgrade is available and if so, click on the “Update” button.



4. This will open a window allowing you to select the version to upgrade to.
5. You can then schedule a time for the upgrade or begin it immediately.

Set up automatic updates

1. To set up your cluster for recurring upgrades, log into OCM and select the cluster you want to upgrade.
2. Click on the “Update Settings” tab.
3. Under “Update Strategy” click the “Automatic” radio button.
4. This will open up options for a day of the week and time that you can set for the update to occur.
5. Select a “Grace period” for allowing the nodes to gracefully drain before forcing the pod eviction under “Node draining”.

6. Click Save.

You have officially set up, deployed, and customized your first cluster. In the next resource, we'll show you how to delete a cluster if the need arises.

All upgrades are fully executed by Red Hat® OpenShift® Service on AWS (ROSA) for your cluster, meaning that you won't need to execute any commands or make changes to the cluster. You do have choices for scheduling them.

There are 3 ways to upgrade your cluster.

- Manually via the command line interface (CLI) - Start a one-time immediate upgrade or schedule a one-time upgrade for a future date/time.
- Manually via the OpenShift Cluster Manager (OCM) user interface (UI) - Start a one-time immediate upgrade or schedule a one-time upgrade for a future date/time.
- Automated upgrades - Set an upgrade window for recurring y-stream upgrades whenever a new version is available without needing to manually schedule it (Ex: Saturday at 06:00 UTC). Minor versions have to be manually scheduled.

We will go through each of these three scenarios.

You can always use --help for more details like:

rosa upgrade cluster --help

What will you learn?

- How to schedule a one-time immediate or future upgrade using the CLI
- How to schedule a one-time immediate or future upgrade using the OCM UI
- How to set an automated y-stream upgrade window for major versions

What do you need before starting?

- Deployed your cluster
- Granted admin rights
- Accessed cluster

Upgrade manually via the CLI

Check if there is an upgrade available by running the following command:

Inline - rosa list upgrade -c <cluster-name>

You will get a list that shows the available version and the current version of your cluster. For example:

1.

```
$ rosa list upgrade -c <cluster-name>

      VERSION  NOTES
```

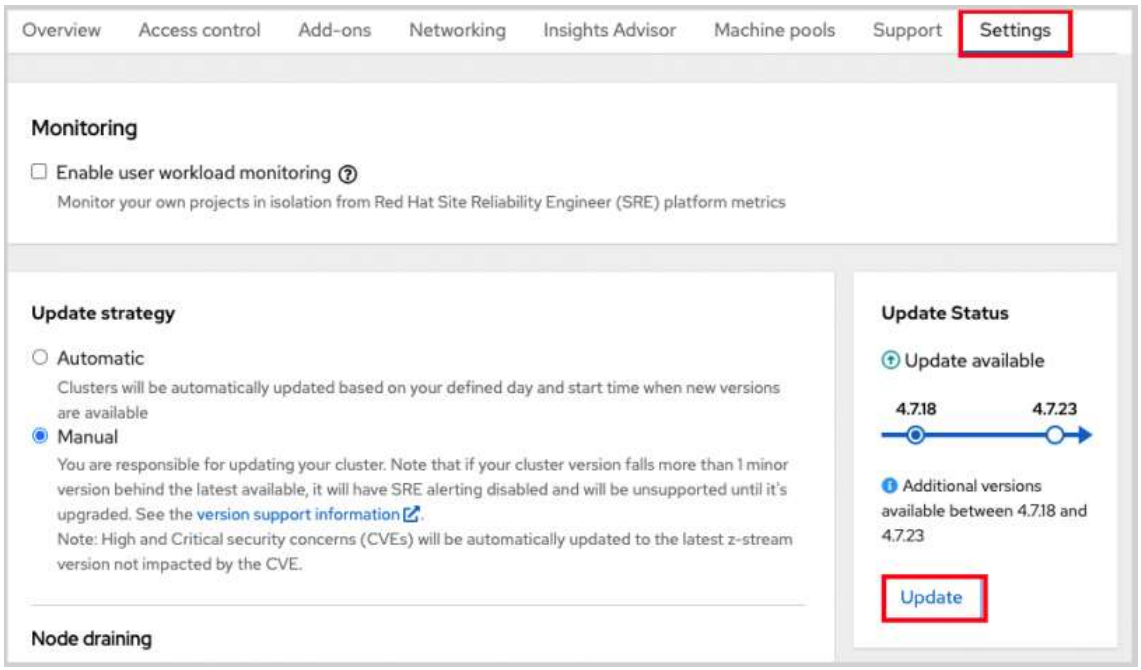
4.10.20 recommended

4.10.18

2. In our example we see that version 4.10.18 is available and so is 4.10.20.
3. Upgrade the cluster to the latest version by running:
   rosa upgrade cluster -c my-rosa-cluster --version 4.10.20
4. This will schedule the cluster for upgrade within the hour. It will take some time to complete.
5. You can also schedule the upgrade for a later date/time by using the <--schedule-date and --schedule-time flags.

Upgrade manually via OCM UI

1. To perform an upgrade via the UI, log into OCM and select the cluster you want to upgrade.
2. Click on the “Settings” tab.
3. You will see if an upgrade is available and if so, click on the “Update” button.



4. This will open a window allowing you to select the version to upgrade to.
5. You can then schedule a time for the upgrade or begin it immediately.

Set up automatic updates

1. To set up your cluster for recurring upgrades, log into OCM and select the cluster you want to upgrade.

2. Click on the “Update Settings” tab.
3. Under “Update Strategy” click the “Automatic” radio button.
4. This will open up options for a day of the week and time that you can set for the update to occur.
5. Select a “Grace period” for allowing the nodes to gracefully drain before forcing the pod eviction under “Node draining”.
6. Click Save.

You have officially set up, deployed, and customized your first cluster. In the next resource, we’ll show you how to delete a cluster if the need arises.
All upgrades are fully executed by Red Hat® OpenShift® Service on AWS (ROSA) for your cluster, meaning that you won't need to execute any commands or make changes to the cluster. You do have choices for scheduling them.
There are 3 ways to upgrade your cluster.

- Manually via the command line interface (CLI) - Start a one-time immediate upgrade or schedule a one-time upgrade for a future date/time.
- Manually via the OpenShift Cluster Manager (OCM) user interface (UI) - Start a one-time immediate upgrade or schedule a one-time upgrade for a future date/time.
- Automated upgrades - Set an upgrade window for recurring y-stream upgrades whenever a new version is available without needing to manually schedule it (Ex: Saturday at 06:00 UTC). Minor versions have to be manually scheduled.

We will go through each of these three scenarios.
You can always use --help for more details like:
rosa upgrade cluster --help

What will you learn?

- How to schedule a one-time immediate or future upgrade using the CLI
- How to schedule a one-time immediate or future upgrade using the OCM UI
- How to set an automated y-stream upgrade window for major versions

What do you need before starting?

- Deployed your cluster
- Granted admin rights
- Accessed cluster

Upgrade manually via the CLI

Check if there is an upgrade available by running the following command:
Inline - rosa list upgrade -c <cluster-name>
You will get a list that shows the available version and the current version of your

cluster. For example:

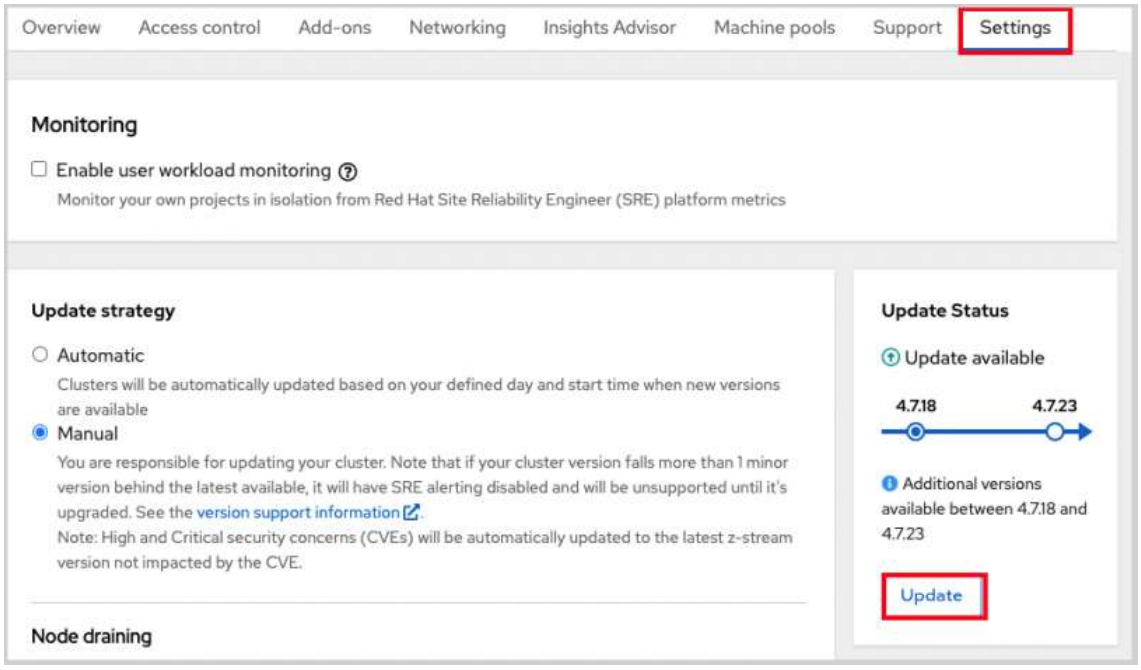
1.

```
$ rosa list upgrade -c <cluster-name>

    VERSION  NOTES

    4.10.20  recommended

    4.10.18
```

2. In our example we see that version 4.10.18 is available and so is 4.10.20.
3. Upgrade the cluster to the latest version by running:
   rosa upgrade cluster -c my-rosa-cluster --version 4.10.20
4. This will schedule the cluster for upgrade within the hour. It will take some time to complete.
5. You can also schedule the upgrade for a later date/time by using the <--schedule-date and --schedule-time flags.

Upgrade manually via OCM UI

1. To perform an upgrade via the UI, log into OCM and select the cluster you want to upgrade.
2. Click on the “Settings” tab.
3. You will see if an upgrade is available and if so, click on the “Update” button.



4. This will open a window allowing you to select the version to upgrade to.
5. You can then schedule a time for the upgrade or begin it immediately.

Set up automatic updates

1. To set up your cluster for recurring upgrades, log into OCM and select the cluster you want to upgrade.
2. Click on the “Update Settings” tab.
3. Under “Update Strategy” click the “Automatic” radio button.
4. This will open up options for a day of the week and time that you can set for the update to occur.
5. Select a “Grace period” for allowing the nodes to gracefully drain before forcing the pod eviction under “Node draining”.
6. Click Save.

You have officially set up, deployed, and customized your first cluster. In the next resource, we’ll show you how to delete a cluster if the need arises.