

Tao Hu Professor: Jun Li

CS 381

3/13/2023

1.Task 1: Launch a Kubernetes cluster. You will need to configure the cluster with one master and at least two workers. You can build your own cluster from scratch on virtual machines or use cloud-based Kubernetes services such as AWS EKS, Azure Kubernetes Service, or Google Kubernetes Engine.

Advantage of AWS EKS (Elastic Kubernetes Service):

- AWS manage master nodes
- necessary app(environment) pre-installed
- AWS mantaince for you

- 1.create as a Node group(groups of Nodes)
2. choose cluster it will attach to
3. define security group, select instance type, resource
4. define max and min number of Nodes
5. then you can deploy application by kubernetes command line tool (kubectl)

EKSCLI is a simple CLI tool for creating and managing clusters on EKS - Amazon's managed Kubernetes service for EC2. It is written in Go, uses CloudFormation, was created by Weaveworks and it welcomes contributions from the community.

Kubectl and eskctl installation

ref: <https://docs.aws.amazon.com/eks/latest/userguide/eksctl.html>

- a. Download and extract the latest release of eksctl by docs

```
tao727188712@DESKTOP-1BM478C:~$ curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_${uname -s}_amd64.tar.gz" | tar xz -C /tmp
tao727188712@DESKTOP-1BM478C:~$
tao727188712@DESKTOP-1BM478C:~$ sudo mv /tmp/eksctl /usr/local/bin
[sudo] password for tao727188712:
tao727188712@DESKTOP-1BM478C:~$ eksctl version
v0.113.0
tao727188712@DESKTOP-1BM478C:~$
```

- b. To install or update kubectl on Linux

Ref: <https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html>

Follow docs to finish installation

```
tao727188712@DESKTOP-1BM478C:~$ kubectl version --short --client
kubectl: command not found
tao727188712@DESKTOP-1BM478C:~$ curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.6/2023-01-30/bin/linux/amd64/kubectl
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 42.9M 100 42.9M 0 0 2711k 0 0:00:16 0:00:16 --:--:-- 7377k
tao727188712@DESKTOP-1BM478C:~$ curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.25.6/2023-01-30/bin/linux/amd64/kubectl.sha256
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 73 100 73 0 0 105 0 --:--:-- --:--:-- --:--:-- 104
tao727188712@DESKTOP-1BM478C:~$ sha256sum -c kubectl.sha256
kubectl: OK
tao727188712@DESKTOP-1BM478C:~$ openssl sha1 -sha256 kubectl
SHA256(kubectl)= 96946285e6ac3f44ea604754e301d95ee21cc8f8999b28361d2be04a71b6ee
tao727188712@DESKTOP-1BM478C:~$ openssl sha1 -sha256 kubectl
SHA256(kubectl)= 96946285e6ac3f44ea604754e301d95ee21cc8f8999b28361d2be04a71b6ee
tao727188712@DESKTOP-1BM478C:~$ chmod +x ./kubectl
tao727188712@DESKTOP-1BM478C:~$ mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin
tao727188712@DESKTOP-1BM478C:~$ echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
tao727188712@DESKTOP-1BM478C:~$ kubectl version --short --client
Flag --short has been deprecated, and will be removed in the future. The --short output will become the default.
Client Version: v1.25.6-eks-48e63af
Kustomize Version: v4.5.7
tao727188712@DESKTOP-1BM478C:~$
```

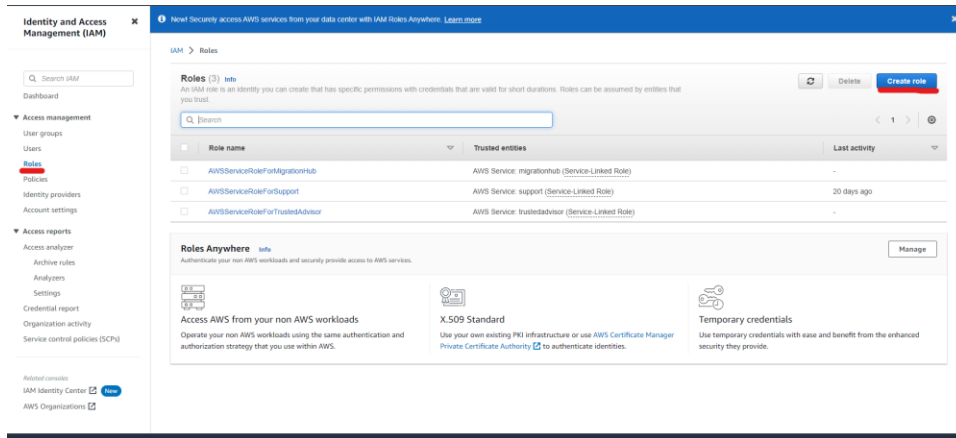
AWS EKS

Ref: <https://us-east-1.console.aws.amazon.com/eks/home?region=us-east-1>

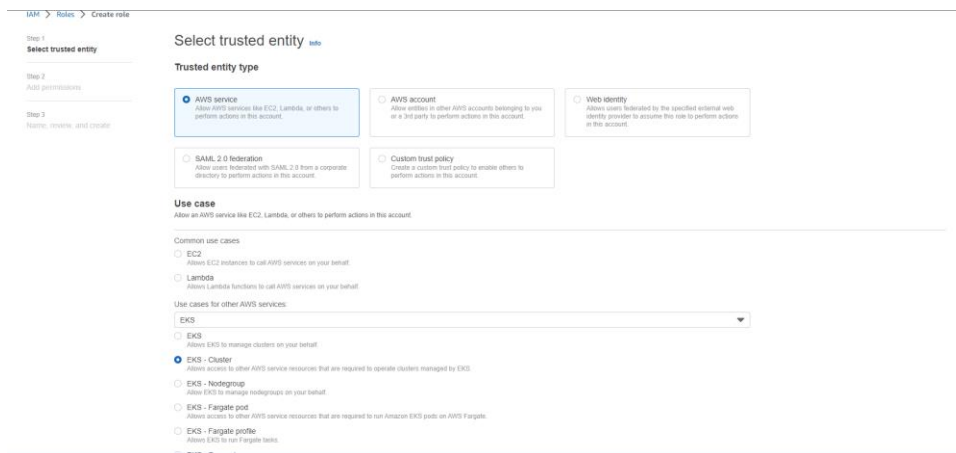
The screenshot displays the AWS Elastic Kubernetes Service (Amazon EKS) console page. The main heading is 'Elastic Kubernetes Service (Amazon EKS) Fully managed Kubernetes control plane'. A large orange button labeled 'Add cluster' is prominently displayed. Below the main heading, a 'How it works' diagram illustrates the process: 'Provision an EKS cluster' (using Amazon EKS API), 'Deploy service pods' (using Kubernetes API), 'Connect to EKS' (using kubectl), and 'Monitor cluster health' (using Amazon CloudWatch). To the right, a 'Pricing' table shows the costs for different components: EKS control plane (EKS pricing), Worker nodes (EC2 pricing), and Fargate pods (Fargate pricing). A 'Getting started' link is also visible.

- a. Name your cluster
- b. create or select cluster service role

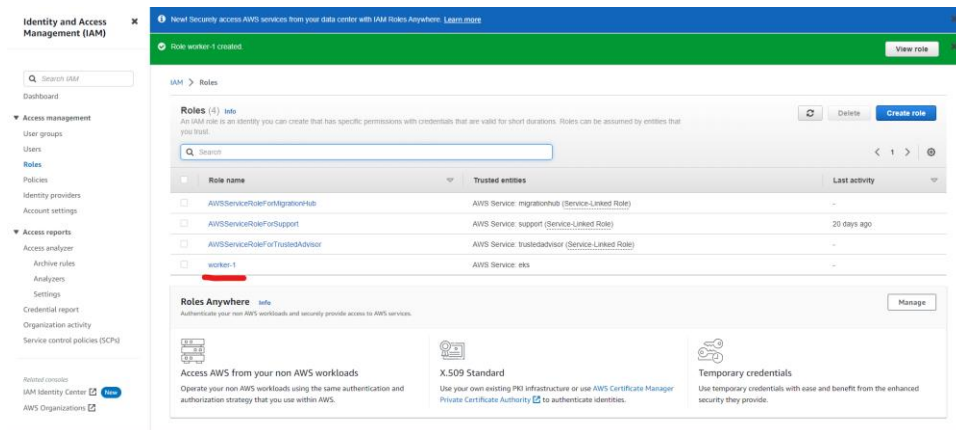
Ref: <https://us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/roles>



Select eks as follows:



As result:



In step 2 – 6:

choose all default.

Review:

The screenshot shows the 'Review and create' step of the 'Create EKS cluster' process. On the left, a sidebar lists six steps: Step 1 (Configure cluster), Step 2 (Specify networking), Step 3 (Configure logging), Step 4 (Select add-ons), Step 5 (Configure selected add-ons settings), and Step 6 (Review and create). The main content area is titled 'Review and create' and contains two sections: 'Step 1: Cluster' and 'Step 2: Networking'. 'Step 1: Cluster' includes a 'Cluster configuration' table with 'Name' (Project-1-Kubernetes-Cluster) and 'Kubernetes version' (1.25), and a 'Cluster service role' (arn:aws:iam::288544653200:role/worker-1). Below this is a 'Tags (0)' section with a search bar and a table showing no tags. 'Step 2: Networking' is partially visible at the bottom.

EKS > Clusters > Create EKS cluster

Step 1
Configure cluster

Step 2
Specify networking

Step 3
Configure logging

Step 4
Select add-ons

Step 5
Configure selected add-ons settings

Step 6
Review and create

Review and create

Step 1: Cluster

Cluster configuration

Name	Project-1-Kubernetes-Cluster	Kubernetes version	1.25
------	------------------------------	--------------------	------

Cluster service role
arn:aws:iam::288544653200:role/worker-1

Tags (0)

A label that you assign to an AWS resource. Each tag consists of a key and an optional value.

Filter by key or value

Key	Value
No tags	
This cluster does not have any tags.	

Step 2: Networking

Networking

Then, wait till creating process finished.

Create node group under compute section:

Name node group, create or select Node IAM role, select EC2 under Common use cases

Check

AmazonEKSWorkerNodePolicy

AmazonEKS_CNI_Policy

AmazonEC2ContainerRegistryReadOnly

The screenshot shows the 'Step 2: Add permissions' section. It has a title bar with 'Step 2: Add permissions' and an 'Edit' button. Below is a 'Permissions policy summary' table with three columns: 'Policy name', 'Type', and 'Attached as'. The table lists three policies: AmazonEC2ContainerRegistryReadOnly, AmazonEKS_CNI_Policy, and AmazonEKSWorkerNodePolicy, all of which are 'AWS managed' and attached as 'Permissions policy'.

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonEC2ContainerRegistryReadOnly	AWS managed	Permissions policy
AmazonEKS_CNI_Policy	AWS managed	Permissions policy
AmazonEKSWorkerNodePolicy	AWS managed	Permissions policy

Back to node group

Configure node group [info](#)

A node group is a group of EC2 instances that supply compute capacity to your Amazon EKS cluster. You can add multiple node groups to your cluster.

Node group configuration

These properties cannot be changed after the node group is created.

Name

Assign a unique name for this node group.

The node group name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 63.

Node IAM role [info](#)

Select the IAM role that will be used by the nodes. To create a new role, go to the [IAM console](#).



The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.

[Learn more](#)

Result:

The screenshot shows the AWS EKS console interface for a cluster named "Project-1". The "Cluster info" section shows the Kubernetes version as 1.25, the status as "Active", and the provider as EKS. The "Compute" tab is selected, showing a list of nodes. There are two nodes, both of type "t3.medium" and belonging to the "Group-1" node group. Both nodes are in a "Ready" state. Below the nodes list, the "Node groups (1)" section shows the configuration for "Group-1", which has a desired size of 2, uses AMI release version 1.25.6-20230304, and is in an "Active" state.

Node name	Instance type	Node group	Created	Status
ip-172-31-10-128.ec2.internal	t3.medium	Group-1	Created 5 minutes ago	Ready
ip-172-31-32-50.ec2.internal	t3.medium	Group-1	Created 5 minutes ago	Ready

Group name	Desired size	AMI release version	Launch template	Status
Group-1	2	1.25.6-20230304	-	Active

Ref: <https://us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/roles>

Follow docs to creating or updating a kubeconfig file for an Amazon EKS cluster:

Run following code in Linux one by one:

aws sts get-caller-identity

aws eks update-kubeconfig --region region-code --name my-cluster

```
aws sts get-caller-identity
tao727188712@DESKTOP-IBM4J8C:~$ aws sts get-caller-identity
{
  "UserId": "288544653200",
  "Account": "288544653200",
  "Arn": "arn:aws:iam::288544653200:root"
}
tao727188712@DESKTOP-IBM4J8C:~$ --aws --version
--aws: command not found
tao727188712@DESKTOP-IBM4J8C:~$ aws --version
aws-cli/2.10.1 Python/3.9.11 Linux/5.10.16.3-microsoft-standard-WSL2 exe/x86_64.ubuntu.20 prompt/off
tao727188712@DESKTOP-IBM4J8C:~$ aws eks update-kubeconfig us-east-1 Project-1

usage: aws [options] <command> [<subcommand> ...] [parameters]
To see help text, you can run:

    aws help
    aws <command> help
    aws <command> <subcommand> help

aws: error: the following arguments are required: --name
tao727188712@DESKTOP-IBM4J8C:~$ aws eks update-kubeconfig --region us-east-1 --name project-1

An error occurred (ResourceNotFoundException) when calling the DescribeCluster operation: No cluster found for name: project-1.
tao727188712@DESKTOP-IBM4J8C:~$ aws eks update-kubeconfig --region us-east-1 --name Project-1
Added new context arn:aws:eks:us-east-1:288544653200:cluster/Project-1 to /home/tao727188712/.kube/config
```

```
tao727188712@DESKTOP-IBM4J8C:~$ kubectl get svc
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes          ClusterIP   10.100.0.1    <none>         443/TCP    50m
tao727188712@DESKTOP-IBM4J8C:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
ip-172-31-10-128.ec2.internal       Ready    <none>    26m    v1.25.6-eks-48e63af
ip-172-31-32-50.ec2.internal        Ready    <none>    26m    v1.25.6-eks-48e63af
tao727188712@DESKTOP-IBM4J8C:~$
```

Ref: <https://docs.aws.amazon.com/eks/latest/userguide/create-kubeconfig.html>

Task 2 and 3: Deploy a containerized application which run multiple instances of the same container.

Scale the pod to more container instances.

The docs ref is :

<https://kubernetes.io/docs/tutorials/kubernetes-basics/deploy-app/deploy-interactive/>

<https://kubernetes.io/docs/tutorials/kubernetes-basics/scale/scale-interactive/>

We need to provide the deployment name and app image location (include the full repository url for images hosted outside Docker hub).

Command I used:

kubectl create deployment kubernetes-bootcamp --image=gcr.io/google-samples/kubernetes-bootcamp:v1

kubectl create -f https://raw.githubusercontent.com/javahometech/kubernetes/master/pods/pods.yml

The process is:

- searched for a suitable node where an instance of the application could be run (we have only 1 available node)
- scheduled the application to run on that Node
- configured the cluster to reschedule the instance on a new Node when needed

To list your deployments use the get deployments command:

kubectl get deployments

We see that there is 1 deployment running a single instance of your app. The instance is running inside a Docker container on your node.

To scale up pods we use:

kubectl scale deployments/kubernetes-bootcamp --replicas=4

We can replace desired number after --replicas=

```
tao727188712@EKS-TP-1B043BC:~$ kubectl get pods -A -o wide
NAMESPACE   NAME             READY   STATUS    RESTARTS   AGE   IP              NODE                                     NOMINATED NODE   READINESS GATES
default      nodeapp          1/1     Running   0           2m52s  172.31.36.72    ip-172-31-32-50.ec2.internal           <none>            <none>
kube-system  aws-node-baglm   1/1     Running   0           37m    172.31.10.128   ip-172-31-10-128.ec2.internal           <none>            <none>
kube-system  aws-node-fnzrz   1/1     Running   0           37m    172.31.32.50    ip-172-31-32-50.ec2.internal           <none>            <none>
kube-system  coredns-7975d6fbb-cmgw  1/1     Running   0           61m    172.31.12.92    ip-172-31-10-128.ec2.internal           <none>            <none>
kube-system  coredns-7975d6fbb-tk2m  1/1     Running   0           61m    172.31.1.183    ip-172-31-10-128.ec2.internal           <none>            <none>
kube-system  kube-proxy-pnz1b   1/1     Running   0           37m    172.31.32.50    ip-172-31-32-50.ec2.internal           <none>            <none>
kube-system  kube-proxy-rwzcf   1/1     Running   0           37m    172.31.10.128   ip-172-31-10-128.ec2.internal           <none>            <none>
tao727188712@EKS-TP-1B043BC:~$ In the previous modules we created a Deployment, and then exposed it publicly via a Service. The Deployment created only one Pod for
he application to keep up with user demand.

Scaling is accomplished by changing the number of replicas in a DeploymentIn: command not found
tao727188712@EKS-TP-1B043BC:~$
tao727188712@EKS-TP-1B043BC:~$ kubectl get deployments
No resources found in default namespace.
tao727188712@EKS-TP-1B043BC:~$ kubectl get pods -o wide
NAME             READY   STATUS    RESTARTS   AGE   IP              NODE                                     NOMINATED NODE   READINESS GATES
nodeapp          1/1     Running   0           13m    172.31.36.72    ip-172-31-32-50.ec2.internal           <none>            <none>
tao727188712@EKS-TP-1B043BC:~$ kubectl get pods
NAME             READY   STATUS    RESTARTS   AGE
nodeapp          1/1     Running   0           14m
tao727188712@EKS-TP-1B043BC:~$ kubectl get rs
No resources found in default namespace.
tao727188712@EKS-TP-1B043BC:~$ kubectl get deployments
No resources found in default namespace.
tao727188712@EKS-TP-1B043BC:~$ kubectl get nodes
NAME             STATUS    ROLES    AGE   VERSION
ip-172-31-10-128.ec2.internal Ready      <none>   53m   v1.25.6-eks-48e63af
ip-172-31-32-50.ec2.internal Ready      <none>   52m   v1.25.6-eks-48e63af
tao727188712@EKS-TP-1B043BC:~$ kubectl create deployment kubernetes-bootcamp --image=gcr.io/google-samples/kubernetes-bootcamp:v1
deployment.apps/kubernetes-bootcamp created
tao727188712@EKS-TP-1B043BC:~$ kubectl get deployments
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp 1/1      1             1           38s
tao727188712@EKS-TP-1B043BC:~$ kubectl create namespace eks-sample-app
namespace/eks-sample-app created
tao727188712@EKS-TP-1B043BC:~$ kubectl apply -f eks-sample-deployment.yaml
error: the path "eks-sample-deployment.yaml" does not exist
tao727188712@EKS-TP-1B043BC:~$ kubectl get deployments
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp 1/1      1             1           10m
tao727188712@EKS-TP-1B043BC:~$ kubectl get rs
NAME             DESIRED   CURRENT   READY   AGE
kubernetes-bootcamp-75c5d958ff 1           1           1       10m
tao727188712@EKS-TP-1B043BC:~$ kubectl scale deployments/kubernetes-bootcamp --replicas=4
deployment.apps/kubernetes-bootcamp scaled
tao727188712@EKS-TP-1B043BC:~$ kubectl get deployments
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp 4/4      4             4           11m
tao727188712@EKS-TP-1B043BC:~$ kubectl get pods -o wide
NAME             READY   STATUS    RESTARTS   AGE   IP              NODE                                     NOMINATED NODE   READINESS GATES
kubernetes-bootcamp-75c5d958ff-2gcbc 1/1     Running   0           12s    172.31.9.215    ip-172-31-10-128.ec2.internal           <none>            <none>
kubernetes-bootcamp-75c5d958ff-qshjh 1/1     Running   0           12s    172.31.13.156   ip-172-31-10-128.ec2.internal           <none>            <none>
kubernetes-bootcamp-75c5d958ff-s5dzn 1/1     Running   0           12s    172.31.36.171   ip-172-31-32-50.ec2.internal           <none>            <none>
kubernetes-bootcamp-75c5d958ff-xlhbh 1/1     Running   0           11m    172.31.44.289   ip-172-31-32-50.ec2.internal           <none>            <none>
nodeapp          1/1     Running   0           31m    172.31.36.72    ip-172-31-32-50.ec2.internal           <none>            <none>
```

Under the AWS EKS web we can see:

EKS > Clusters > Project-1

Project-1

🔄 Delete cluster

▼ Cluster info [Info](#)

Kubernetes version [Info](#)
1.25

Status
🟢 Active

Provider
EKS

Overview

Resources

Compute

Networking

Add-ons

Authentication

Logging

Update history

Tags

Nodes (2) [Info](#)

< 1 >

Node name	Instance type	Node group	Created	Status
ip-172-31-10-128.ec2.internal	t3.medium	Group-1	Created 🕒 an hour ago	🟢 Ready
ip-172-31-32-50.ec2.internal	t3.medium	Group-1	Created 🕒 an hour ago	🟢 Ready

Node groups (1) [Info](#)

Edit

Delete

Add node group

Group name	Desired size	AMI release version	Launch template	Status
Group-1	2	1.25.6-20230304	-	🟢 Active

Node 1:

17 Pods

Used
6 pods, 35%

Available
11 pods, 65%

Pods (6) [Info](#)

< 1 >

Name	Status	Created	IP
kubernetes-bootcamp-75c5d958ff-2gcbc	🟢 Running	7 minutes ago	172.31.9.215
kubernetes-bootcamp-75c5d958ff-qshjh	🟢 Running	7 minutes ago	172.31.13.156
aws-node-btglm	🟢 Running	an hour ago	172.31.10.128
coredns-7975d6fb9b-cmgwn	🟢 Running	2 hours ago	172.31.12.92
coredns-7975d6fb9b-4k2rm	🟢 Running	2 hours ago	172.31.1.183
kube-proxy-rw2c7	🟢 Running	an hour ago	172.31.10.128

Node 2:

17 Pods

Used
5 pods, 29%

Available
12 pods, 71%

Pods (5) [Info](#)

< 1 >

Name	Status	Created	IP
kubernetes-bootcamp-75c5d958ff-s5d2n	🟢 Running	9 minutes ago	172.31.36.171
kubernetes-bootcamp-75c5d958ff-xlbhb	🟢 Running	20 minutes ago	172.31.44.209
nodeapp	🟢 Running	40 minutes ago	172.31.36.72
aws-node-fnz5	🟢 Running	an hour ago	172.31.32.50
kube-proxy-pnzlb	🟢 Running	an hour ago	172.31.32.50

Task 4: Update the application with a new software version

Users expect applications to be available all the time and developers are expected to deploy new versions of them several times a day. In Kubernetes this is done with rolling updates. Rolling updates allow Deployments' update to take place with zero downtime by incrementally updating Pods instances with new ones. The new Pods will be scheduled on Nodes with available resources.

By default, the maximum number of Pods that can be unavailable during the update and the maximum number of new Pods that can be created, is one. Both options can be configured to either numbers or percentages (of Pods). In Kubernetes, updates are versioned and any Deployment update can be reverted to a previous (stable) version

Similar to application Scaling, if a Deployment is exposed publicly, the Service will load-balance the traffic only to available Pods during the update. An available Pod is an instance that is available to the users of the application.

Rolling updates allow the following actions:

Promote an application from one environment to another (via container image updates)

Rollback to previous versions

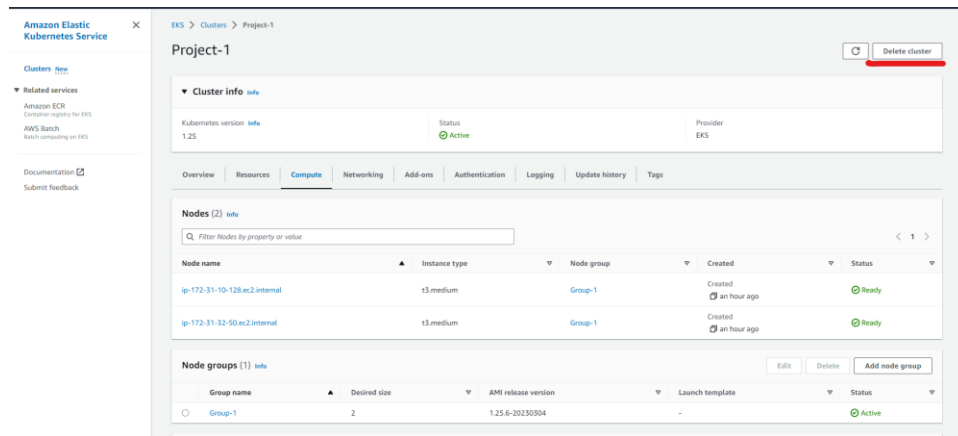
Continuous Integration and Continuous Delivery of applications with zero downtime

```
tao727188712@DESKTOP-18M438C:~$ kubectl set image deployments/kubernetes-bootcamp kubernetes-bootcamp-gcr.io/google-samples/kubernetes-bootcamp:v10
deployment.apps/kubernetes-bootcamp image updated
tao727188712@DESKTOP-18M438C:~$ kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp 3/4      2             3           44m
tao727188712@DESKTOP-18M438C:~$ kubectl get pods
NAME                READY   STATUS              RESTARTS   AGE
kubernetes-bootcamp-5cd9cf5899-fhm9m  0/1     ImagePullBackOff    0           16s
kubernetes-bootcamp-5cd9cf5899-przz4  0/1     ErrImagePull        0           16s
kubernetes-bootcamp-75c5d958ff-2gcbc  1/1     Running              0           33m
kubernetes-bootcamp-75c5d958ff-qshjh  1/1     Terminating       0           33m
kubernetes-bootcamp-75c5d958ff-s5d2n  1/1     Running              0           33m
kubernetes-bootcamp-75c5d958ff-x1bhb  1/1     Running              0           44m
nodeapp              1/1     Running              0           64m
tao727188712@DESKTOP-18M438C:~$
```

We can see our app is successful updated

Task 5: Delete the application and stop the Kubernetes cluster.

There are two ways to do this step: one is delete cluster on the web



Another is using command:

eksctl delete cluster --name prod

```
tao727188712@DESKTOP-1BMAJ8C:~$ eksctl delete cluster Project-1
2023-03-13 01:00:41 [i] deleting EKS cluster "Project-1"
2023-03-13 01:00:42 [i] deleted 0 Fargate profile(s)
2023-03-13 01:00:42 [i] cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
2023-03-13 01:00:43 [i] 1 task: { delete unowned nodegroup Group-1 }
2023-03-13 01:01:03 [i] waiting for all non eksctl-owned nodegroups to be deleted
2023-03-13 01:01:23 [i] waiting for all non eksctl-owned nodegroups to be deleted
2023-03-13 01:01:43 [i] waiting for all non eksctl-owned nodegroups to be deleted
2023-03-13 01:02:04 [i] waiting for all non eksctl-owned nodegroups to be deleted
2023-03-13 01:02:24 [i] waiting for all non eksctl-owned nodegroups to be deleted
2023-03-13 01:02:44 [i] waiting for all non eksctl-owned nodegroups to be deleted
2023-03-13 01:03:04 [i] waiting for all non eksctl-owned nodegroups to be deleted
2023-03-13 01:03:24 [i] waiting for all non eksctl-owned nodegroups to be deleted
2023-03-13 01:03:44 [i] waiting for all non eksctl-owned nodegroups to be deleted
2023-03-13 01:04:04 [i] waiting for all non eksctl-owned nodegroups to be deleted
2023-03-13 01:04:24 [i] waiting for all non eksctl-owned nodegroups to be deleted
2023-03-13 01:04:44 [i] waiting for all non eksctl-owned nodegroups to be deleted
2023-03-13 01:05:05 [i] waiting for all non eksctl-owned nodegroups to be deleted
2023-03-13 01:05:25 [i] no IAM and OIR resources were found for "Project-1"
2023-03-13 01:05:26 [i] Initiated deletion of cluster "Project-1"
2023-03-13 01:05:26 [i] to see the status of the deletion run 'eksctl get cluster --name Project-1 --region us-east-1'
2023-03-13 01:05:26 [i] all cluster resources were deleted
tao727188712@DESKTOP-1BMAJ8C:~$
```

Cost is \$0.10 per hour for each Amazon EKS cluster that you create, we need delete cluster on aws eks to prevent unwanted charge