Deploy your Application:

wget https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/linux/amd64/kubectl

chmod +x kubectl

./kubectl

```
root@ip-172-31-17-73:~# wget https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/linux/amd64/kubectl
--2019-07-28 02:03:07--  https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/linux/amd64/kubectl
Resolving amazon-eks.s3-us-west-2.amazonaws.com (amazon-eks.s3-us-west-2.amazonaws.com)... 52.218.253.65
Connecting to amazon-eks.s3-us-west-2.amazonaws.com (amazon-eks.s3-us-west-2.amazonaws.com)|52.218.253.65|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 54146532 (52M) [binary/octet-stream]
Saving to: âkubectlâ

kubectl                             100%[=======================================================================================>]  51.64M  7.89MB/s

2019-07-28 02:03:14 (7.41 MB/s) - âkubectlâ saved [54146532/54146532]

root@ip-172-31-17-73:~# ./kubectl
-su: ./kubectl: Permission denied
root@ip-172-31-17-73:~# chmod +x kubectl
root@ip-172-31-17-73:~# ./kubectl
kubectl controls the Kubernetes cluster manager.

Find more information at: https://kubernetes.io/docs/reference/kubectl/overview/
```

mkdir bin

cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH

kubectl version

kubectl version --short --client

```
root@ip-172-31-17-73:~# mkdir bin
root@ip-172-31-17-73:~# cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
root@ip-172-31-17-73:~# kubectl version
Client Version: version.Info{Major:"1", Minor:"10", GitVersion:"v1.10.3", GitCommit:"2bba0
-26T20:40:11Z", GoVersion:"go1.9.3", Compiler:"gc", Platform:"linux/amd64"}
```

wget https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/linux/amd64/aws-iam-authenticator

chmod +x ./aws-iam-authenticator

cp ./aws-iam-authenticator $HOME/bin/aws-iam-authenticator && export PATH=$HOME/bin:$PATH

aws-iam-authenticator help

```
root@ip-172-31-17-73:~# wget https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/linux/amd64/aws-iam-authenticator
--2019-07-28 02:11:02--  https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/linux/amd64/aws-iam-authenticator
Resolving amazon-eks.s3-us-west-2.amazonaws.com (amazon-eks.s3-us-west-2.amazonaws.com)... 52.218.193.153
Connecting to amazon-eks.s3-us-west-2.amazonaws.com (amazon-eks.s3-us-west-2.amazonaws.com)|52.218.193.153|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26349462 (25M) [binary/octet-stream]
Saving to: âaws-iam-authenticatorâ

aws-iam-authenticator               100%[=======================================================================================>]

2019-07-28 02:11:05 (9.03 MB/s) - âaws-iam-authenticatorâ saved [26349462/26349462]

root@ip-172-31-17-73:~# chmod +x ./aws-iam-authenticator
root@ip-172-31-17-73:~# cp ./aws-iam-authenticator $HOME/bin/aws-iam-authenticator && export PATH=$HOME/bin:$PATH
root@ip-172-31-17-73:~# aws-iam-authenticator help
A tool to authenticate to Kubernetes using AWS IAM credentials
```

curl --silent --location "https://github.com/weaveworks/eksctl/releases/download/latest_release/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp

mv /tmp/eksctl /usr/local/bin

eksctl version

```
root@ip-172-31-86-69:~# curl --silent --location "https://github.com/weaveworks/eksctl/r
p
root@ip-172-31-86-69:~#  mv /tmp/eksctl /usr/local/bin
root@ip-172-31-86-69:~# eksctl version
[â
  ¹]  version.Info{BuiltAt:"", GitCommit:"", GitTag:"0.2.1"}
root@ip-172-31-86-69:~#
```

Access keys

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. Learn more

Create access key

Access key ID	Created	Last used	Status
AKIAVORWYFFGC3WVPNWC	2019-07-24 08:28 UTC+0530	2019-07-26 13:51 UTC+0530 with sts in us-east-1	Active | Make inactive ✖

Create access key

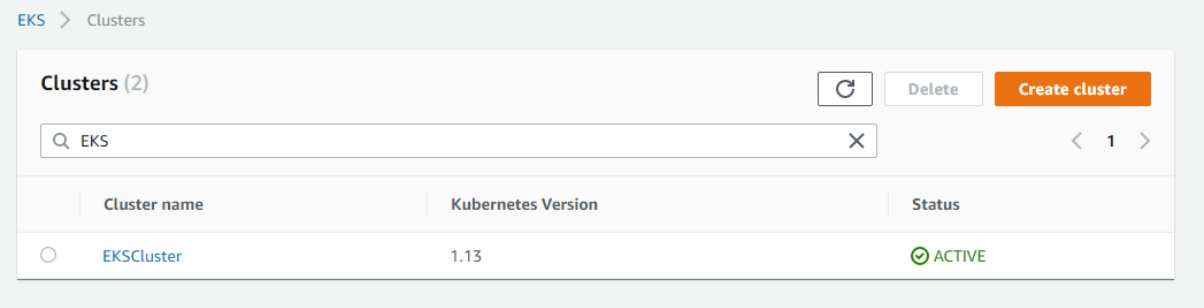
Access key ID	Created	Last used	Status
AKIAVORWYFFGC3WVPNWC	2019-07-24 08:28 UTC+0530	2019-07-26 13:51 UTC+0530 with sts in us-east-1	Active | Make inactive ✖
AKIAVORWYFFGE3YTFZFZ	2019-07-28 07:49 UTC+0530	N/A	Active | Make inactive ✖

```
root@ip-172-31-17-73:~# aws configure
AWS Access Key ID [None]: AKIAVORWYFFGE3YTFZFZ
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
root@ip-172-31-17-73:~#
```

```
root@ip-172-31-86-69:~# eksctl create cluster --name=EKSCluster --nodes=2 --region=us-west-2
[â
  ¹]  using region us-west-2
[â
  ¹]  setting availability zones to [us-west-2c us-west-2d us-west-2b]
[â
  ¹]  subnets for us-west-2c - public:192.168.0.0/19 private:192.168.96.0/19
[â
  ¹]  subnets for us-west-2d - public:192.168.32.0/19 private:192.168.128.0/19
[â
  ¹]  subnets for us-west-2b - public:192.168.64.0/19 private:192.168.160.0/19
[â
  ¹]  nodegroup "ng-c8e07a6f" will use "ami-03a55127c613349a7" [AmazonLinux2/1.13]
[â
  ¹]  using Kubernetes version 1.13
[â
  ¹]  creating EKS cluster "EKSCluster" in "us-west-2" region
[â
  ¹]  will create 2 separate CloudFormation stacks for cluster itself and the initial nodegroup
[â
  ¹]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-west-2 --name=EKSCluster'
[â
  ¹]  2 sequential tasks: { create cluster control plane "EKSCluster", create nodegroup "ng-c8e07a6f" }
[â
  ¹]  building cluster stack "eksctl-EKSCluster-cluster"
[â
  ¹]  deploying stack "eksctl-EKSCluster-cluster"
```

```
[á]  all EKS cluster resource for "EKSCluster" had been created
[á]  saved kubeconfig as "/root/.kube/config"
[á
  ¹]  adding role "arn:aws:iam::130374862735:role/eksctl-EKSCluster-nodegroup-ng-c8-NodeInstanceRole-1FKZC9GNJUUMU" to auth ConfigMap
[á
  ¹]  nodegroup "ng-c8e07a6f" has 0 node(s)
[á
  ¹]  waiting for at least 2 node(s) to become ready in "ng-c8e07a6f"
[á
  ¹]  nodegroup "ng-c8e07a6f" has 2 node(s)
[á
  ¹]  node "ip-192-168-28-149.us-west-2.compute.internal" is ready
[á
  ¹]  node "ip-192-168-76-186.us-west-2.compute.internal" is ready
[á
  ¹]  kubectl command should work with "/root/.kube/config", try 'kubectl get nodes'
[á]  EKS cluster "EKSCluster" in "us-west-2" region is ready
```

```
root@ip-172-31-86-69:~# kubectl get node
NAME                                           STATUS    ROLES     AGE       VERSION
ip-192-168-28-149.us-west-2.compute.internal   Ready     <none>    5m        v1.13.7-eks-c57ff8
ip-192-168-76-186.us-west-2.compute.internal   Ready     <none>    5m        v1.13.7-eks-c57ff8
root@ip-172-31-86-69:~#
```




```
root@ip-172-31-86-69:~# kubectl run kubernetes-bootcamp --image=docker.io/jocatalin/kubernetes-bootcamp:v1 --port=8080
deployment.apps "kubernetes-bootcamp" created
root@ip-172-31-86-69:~# kubectl expose deployment/kubernetes-bootcamp --port=8080 --target-port=8080 --type=NodePort
service "kubernetes-bootcamp" exposed
root@ip-172-31-86-69:~# kubectl get pods
NAME                                  READY     STATUS              RESTARTS   AGE
kubernetes-bootcamp-6c5cfd894b-9jqzf   0/1       ContainerCreating   0          6s
root@ip-172-31-86-69:~# kubectl get deployments
NAME                  DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   1         1         1            1           15s
root@ip-172-31-86-69:~# kubectl get pods
NAME                                  READY     STATUS    RESTARTS   AGE
kubernetes-bootcamp-6c5cfd894b-9jqzf   1/1       Running   0          19s
root@ip-172-31-86-69:~# kubectl get services
NAME                  TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes            ClusterIP   10.100.0.1      <none>        443/TCP          44m
kubernetes-bootcamp   NodePort    10.100.33.238   <none>        8080:30306/TCP   1m
root@ip-172-31-86-69:~#
```