

Setting up EKS CTL command line and dependencies

- Download the Amazon EKS-vended kubectl binary from Amazon S3:

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

- Follow the steps shown below in the screenshot.

```
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
--2018-07-28 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)... 52.210.253.65
Connecting to amazon-eks.s3-us-west-2.amazonaws.com (amazon-eks.s3-us-west-2.amazonaws.com)|52.210.253.65|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 54146332 (52M) [binary/octet-stream]
Saving to: 'kubectl'

kubectl                               100%[=====] 52.44M  7.83MB/s
2018-07-28 02:03:14 (7.41 MB/s) - 'kubectl' saved [54146332/54146332]

root@ip-172-31-17-73:~# ./kubectl
-mv ./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/
```

- Configure **kubectl** in PATH variable to call **kubectl** command globally. Follow the set of commands given below to configure PATH variable:

```
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"}
```

- Configure AWS CLI and aws-iam-authenticator. Follow the set of commands given below to install these command lines. Download the Amazon EKS-vended aws-iam-authenticator binary from Amazon S3:

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

```
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
--2019-07-26 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: 'aws-iam-authenticator'

aws-iam-authenticator 100%[=====>]
2019-07-26 02:11:06 (9.03 MB/s) - 'aws-iam-authenticator' 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
```

- Install **EKS CTL command line** to create an EKS cluster.

```
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/releases/download/latest_release/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
root@ip-172-31-86-69:~# mv /tmp/eksctl /usr/local/bin
root@ip-172-31-86-69:~# eksctl version
[â
  ']' version.Info{BuiltAt:"", GitCommit:"", GitTag:"0.2.1"}
root@ip-172-31-86-69:~# █
```

- Install AWS CLI using the sequence of commands given below.

```
apt install python-pip
```

```
pip install awscli
```

```
aws --version
```

- Configure AWS CLI and provide **Access Keys and Secret Access Keys** while configuring AWS CLI.

```
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:~#
```

Step 5.4.2: Creating an EKS cluster using eksctl command line

- Create an EKS Cluster using the command below:

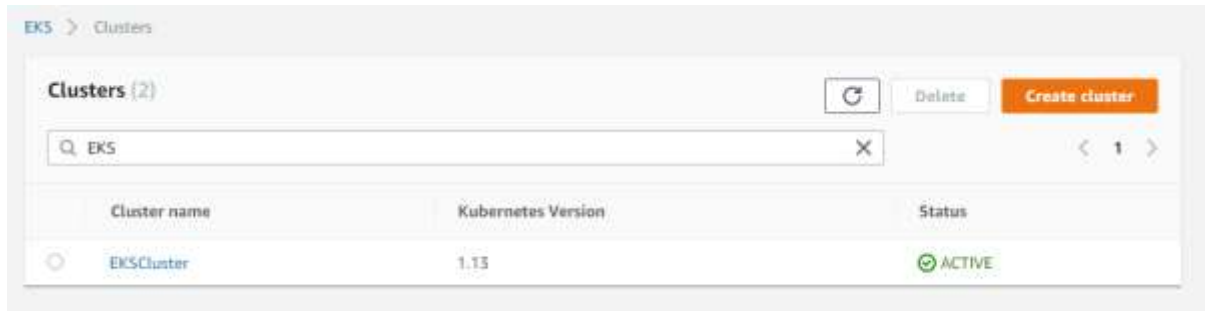
```
eksctl create cluster --name=EKScluster --nodes=2 --region=us-west-2
```

```
root@ip-172-31-17-73:~# eksctl create cluster --name=EKScluster --nodes=2 --region=us-west-2
[6] using region us-west-2
[6] setting availability zones to [us-west-2c us-west-2d us-west-2h]
[6] subnets for us-west-2c - public:192.168.0.0/19 private:192.168.96.0/19
[6] subnets for us-west-2d - public:192.168.32.0/19 private:192.168.128.0/19
[6] subnets for us-west-2h - public:192.168.64.0/19 private:192.168.160.0/19
[6] nodegroup "ng-c8e07aef" will use "ami-00a55127c61394a7" [AmazonLinux2/1.19]
[6] using Kubernetes version 1.19
[6] creating EKS cluster "EKScluster" in "us-west-2" region
[6] will create 2 separate CloudFormation stacks for cluster itself and the initial nodegroup.
[6] if you encounter any issues, check CloudFormation console or try "eksctl utils describe-stacks --region=us-west-2 --name=EKScluster"
[6] 2 sequential tasks: ( create cluster control plane "EKScluster", create nodegroup "ng-c8e07aef" )
[6] building cluster stack "eksctl-EKScluster-cluster"
[6] deploying stack "eksctl-EKScluster-cluster"
```

```
[6] all EKS cluster resource for "EKScluster" had been created
[6] saved kubeconfig as "/root/.kube/config"
[6] adding role "arn:aws:iam::130174862735:role/eksctl-EKScluster-nodegroup-ng-c8-NodeInstanceRole-1FZIC5G8WUUMU" to auth ConfigMap
[6] nodegroup "ng-c8e07aef" has 0 node(s)
[6] waiting for at least 2 node(s) to become ready in "ng-c8e07aef"
[6] nodegroup "ng-c8e07aef" has 2 node(s)
[6] node "ip-192-168-3d-148.us-west-2.compute.internal" is ready
[6] node "ip-192-168-74-166.us-west-2.compute.internal" is ready
[6] kubectl command should work with "/root/.kube/config", try 'kubectl get nodes'
[6] EKS cluster "EKScluster" in "us-west-2" region is ready
```

- Validate the cluster using **kubectl get node** command through AWS Console.

```
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:~#
```



Deploying an application to AWS EKS cluster

- Create Kubernetes deployment and service using the set of commands mentioned below:

```
kubectl run kubernetes-bootcamp --image=docker.io/jocatalin/kubernetes-bootcamp:v1 --port=8080
```

```
kubectl expose deployment/kubernetes-bootcamp --port=8080 --target-port=8080 --type=NodePort
```

```
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-9jqzr 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-9jqzr 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:~#
```