**Assisted Practice: 5.4 Deploy your Application**

This section will guide you to:

* Configure Kubernetes cluster to push a random application to AWS EKS

This lab has mainly four subsections, namely:

5.4.1 Setting up EKS CTL command line and dependencies

5.4.2 Creating an EKS cluster using eksctl command line

5.4.3 Deploying an application to AWS EKS cluster

5.4.4 Pushing the code to GitHub repositories

**Step 5.4.1:** Setting up EKS CTL command line and dependencies

**Please Note:** Amazon EKS clusters require **kubectl**, **kubelet** binaries, and AWS IAM Authenticator for Kubernetes to allow IAM authentication for Kubernetes cluster.

* 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**



* 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**



* 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**

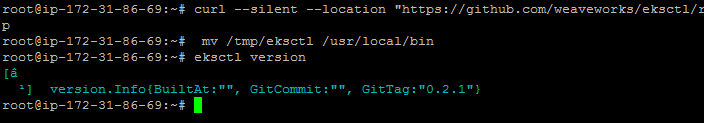


* Install **EKS CTL command line** to create an EKS cluster.

**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**



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

**apt install python-pip**

**pip install awscli**

**aws --version**

* Configure AWS CLI. We need to create **Access Keys** in AWS IAM Console.



* Click on **Create Access key** and keep the keys safe with you.



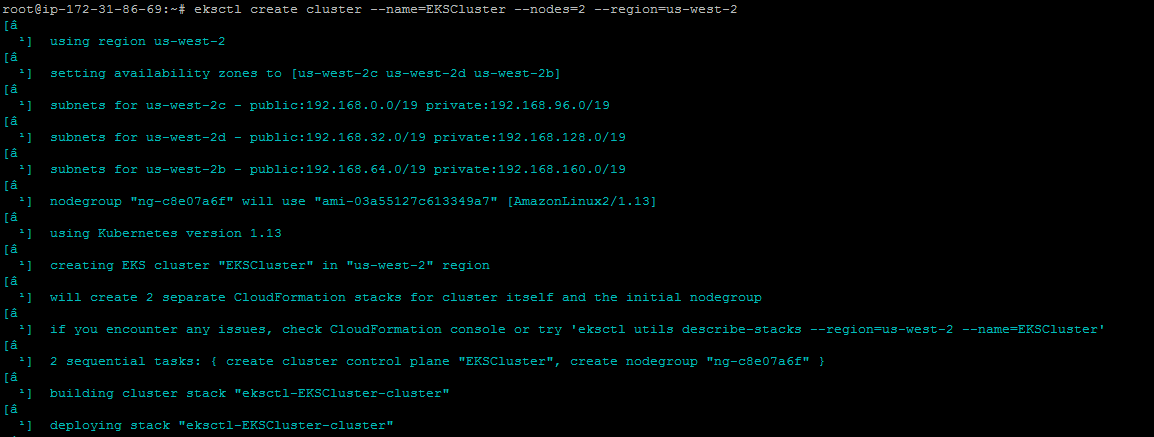
* Configure AWS CLI and provide **Access Keys and Secret Access Keys** while configuring AWS CLI.

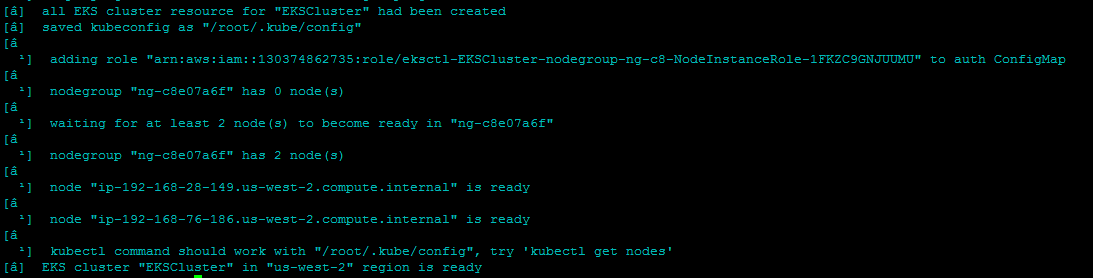


**Step 5.4.2:** Creating an EKS cluster using eksctl command line

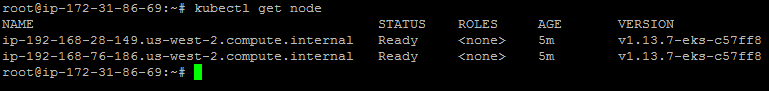
* Create an EKS Cluster using the command below:

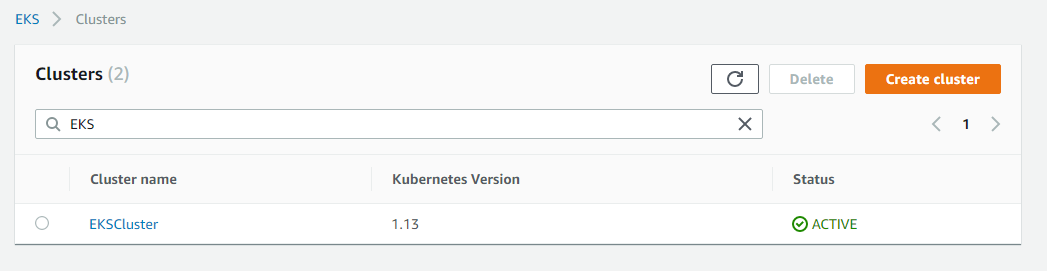
**eksctl create cluster --name=EKSCluster --nodes=2 --region=us-west-2**





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



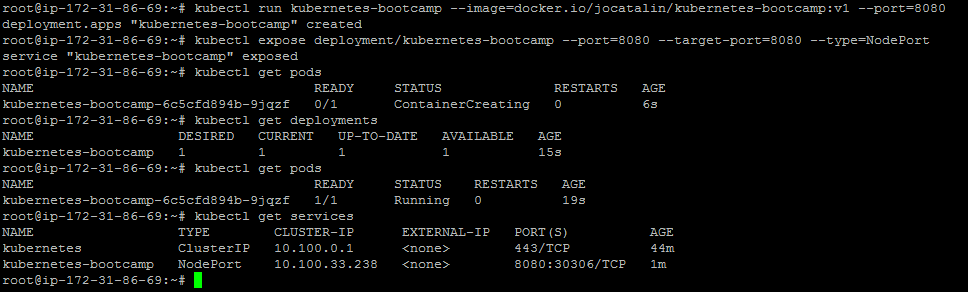


**Step 5.4.3:** Deploying an application to AWS EKS cluster

* 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**



**Step 5.4.4:** Pushing the code to your GitHub repositories

* Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path>**

* Initialize your repository using the following command:

**git init**

* Add all the files to your git repository using the following command:

**git add .**

* Commit the changes using the following command:

**git commit . -m “Changes have been committed.”**

* Push the files to the folder you initially created using the following command:

**git push -u origin master**