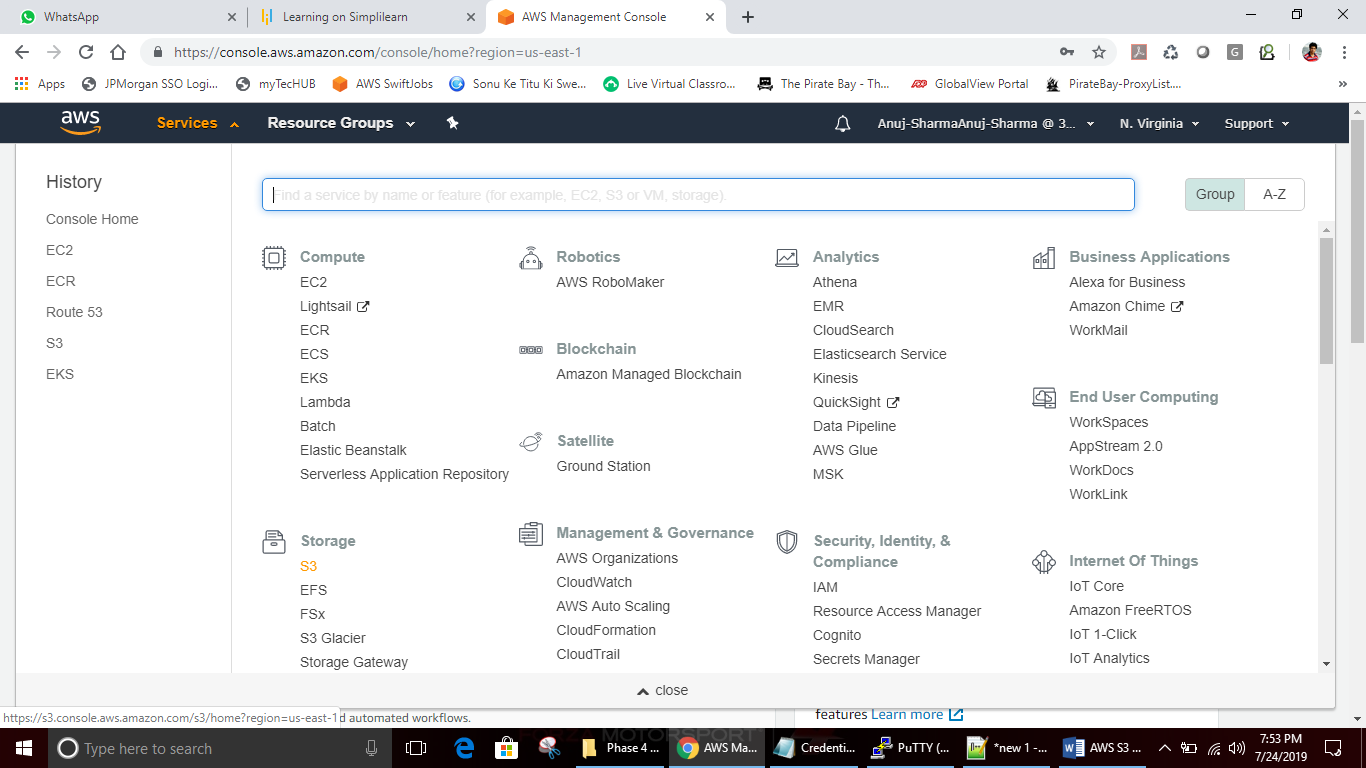
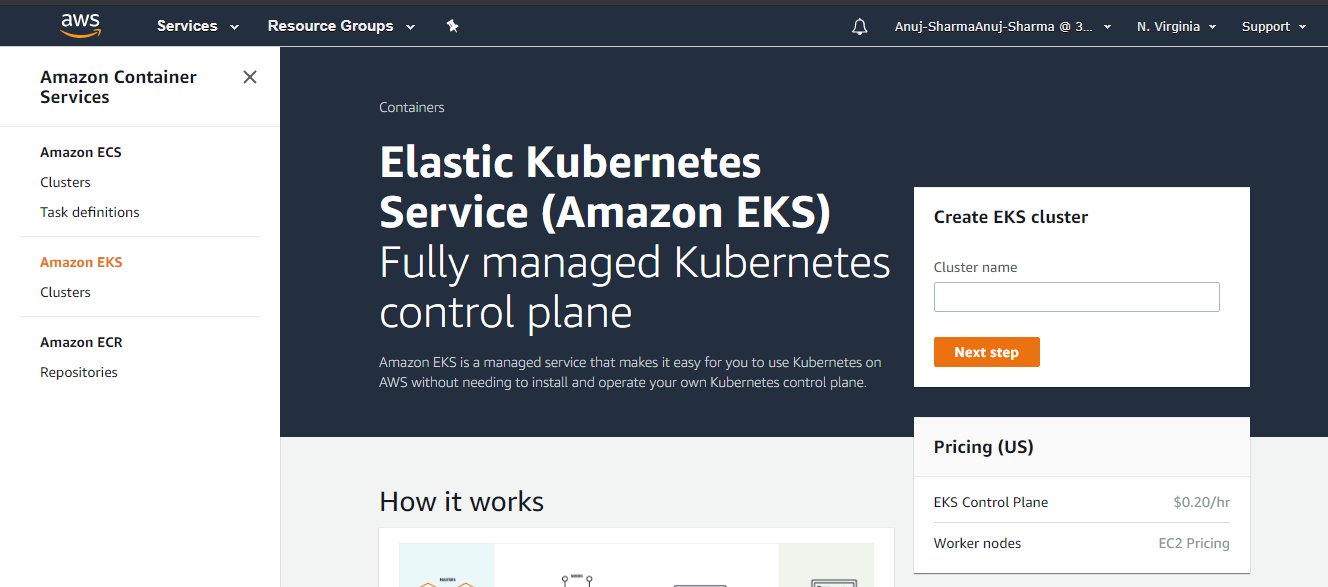
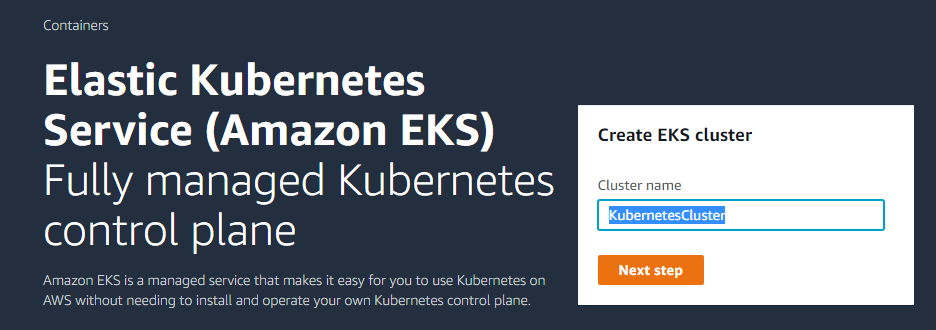
**Step 1:** Creating an AWS EKS cluster

* Connect to AWS console and navigate to EKS service to create an EKS cluster.

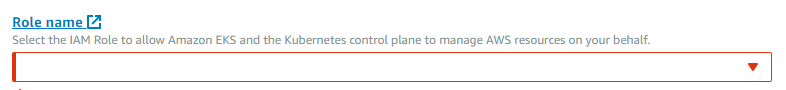




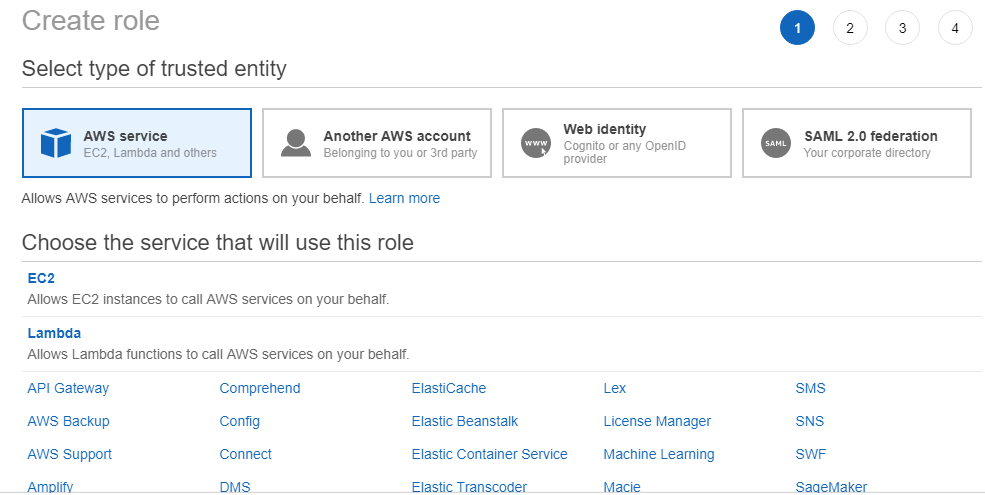
* Provide a cluster name and click on **Next Step**.



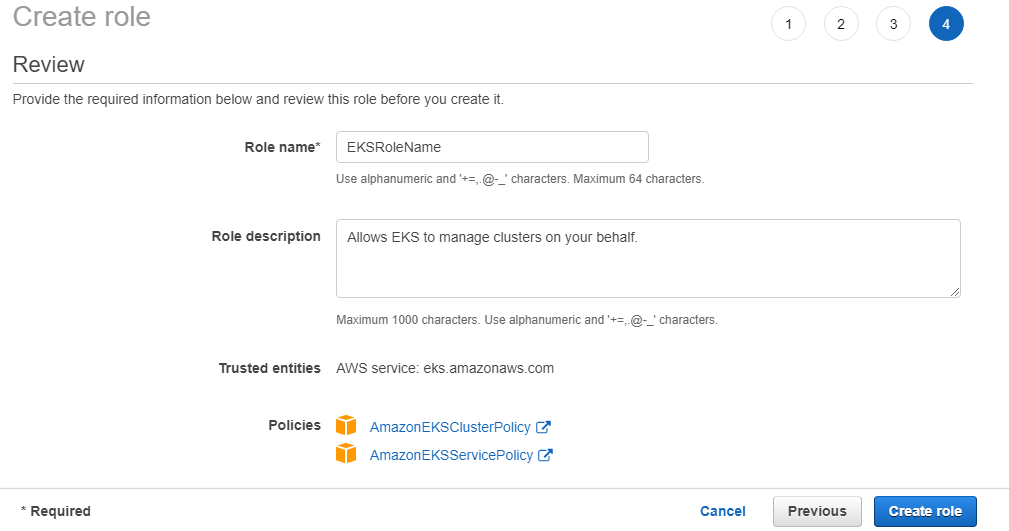
* Configure **Role name** used by EKS rest.



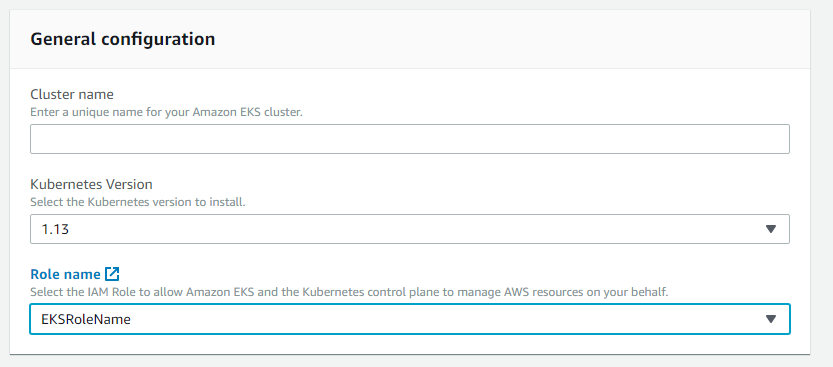
* Click on **Create Role** and provide the policy details. Select **EKS** from the service list.



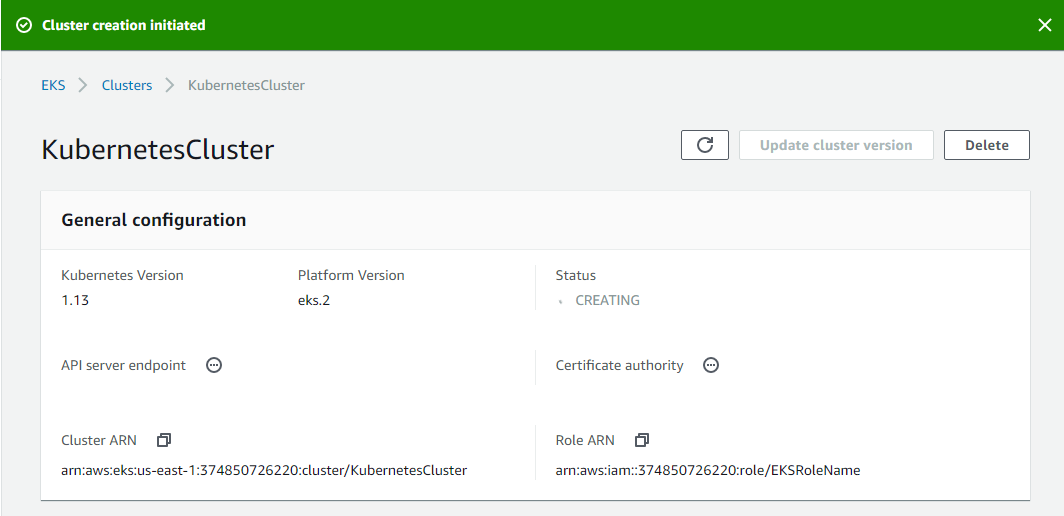


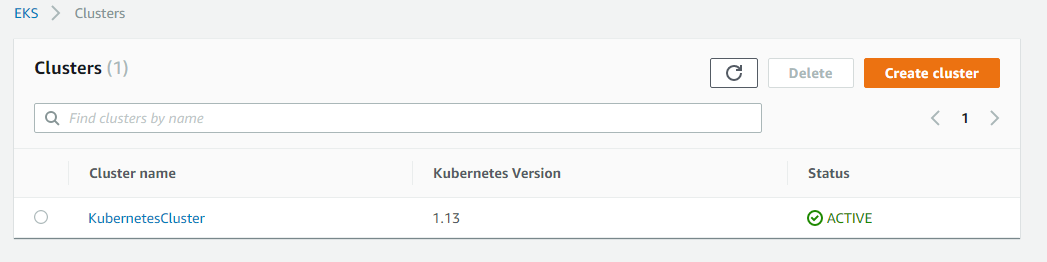


* Select the newly created role name from the list while creating the EKS cluster.



**Please Note:** Once configurations are saved and the EKS cluster is created, it may take some time to bring the cluster online. Configure **kubectl** command line only when EKS will be completely online.





**Step 2:** Setting up kubectl command line with EKS cluster

**chmod +x kubectl**

**./kubectl**



* Configure **kubectl** in **PATH** variable to call **kubectl** command globally. Follow the set of commands mentioned 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**. 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>



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

**apt install python-pip**

**pip install awscli**

**aws --version**

* Create **Access keys** in AWS IAM Console.



**Please Note:** Create Access keys and keep them saved in a document.



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



* Update the local **kubeconfig** to configure locally installed **kubectl** with EKS configurations.

**awseks --region us-east-1 update-kubeconfig --name KubernetesCluster**

**kubectl get svc**

