**AWS EKS**

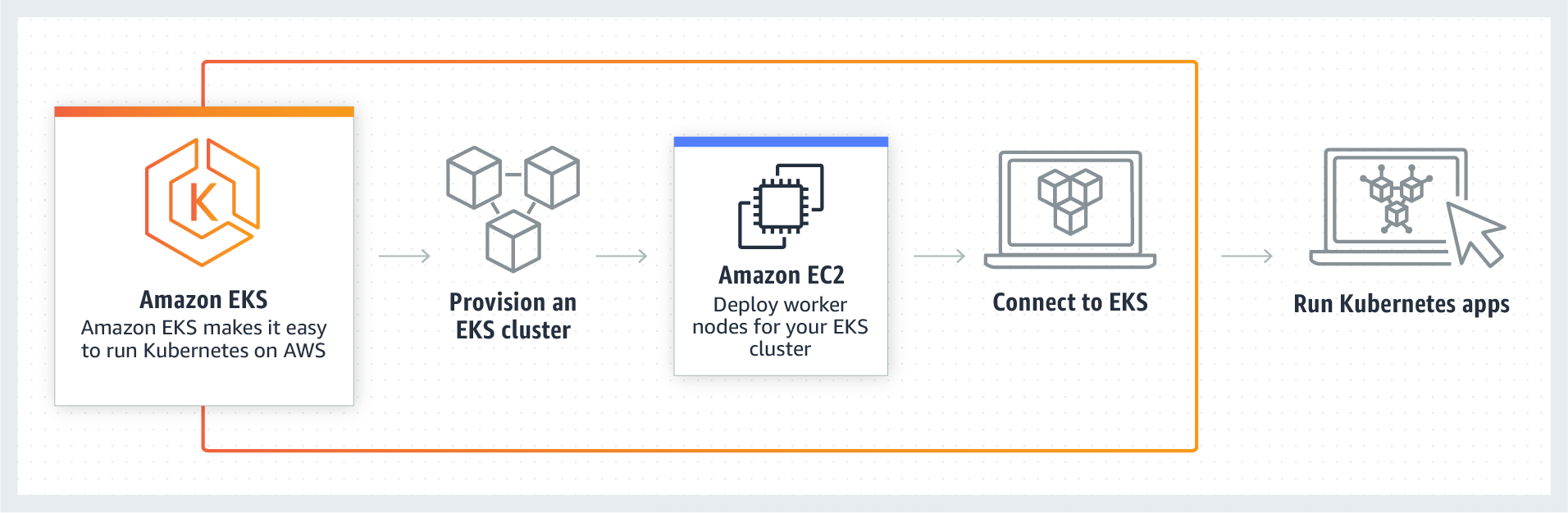
Amazon Elastic Kubernetes Service (Amazon EKS) makes it easy to deploy, manage, and scale containerized applications using [Kubernetes on AWS](https://aws.amazon.com/kubernetes/).

Amazon EKS runs the Kubernetes management infrastructure for you across multiple AWS availability zones to eliminate a single point of failure. Amazon EKS is certified Kubernetes conformant so you can use existing tooling and plugins from partners and the Kubernetes community. Applications running on any standard Kubernetes environment are fully compatible and can be easily migrated to Amazon EKS.

**Benefits**

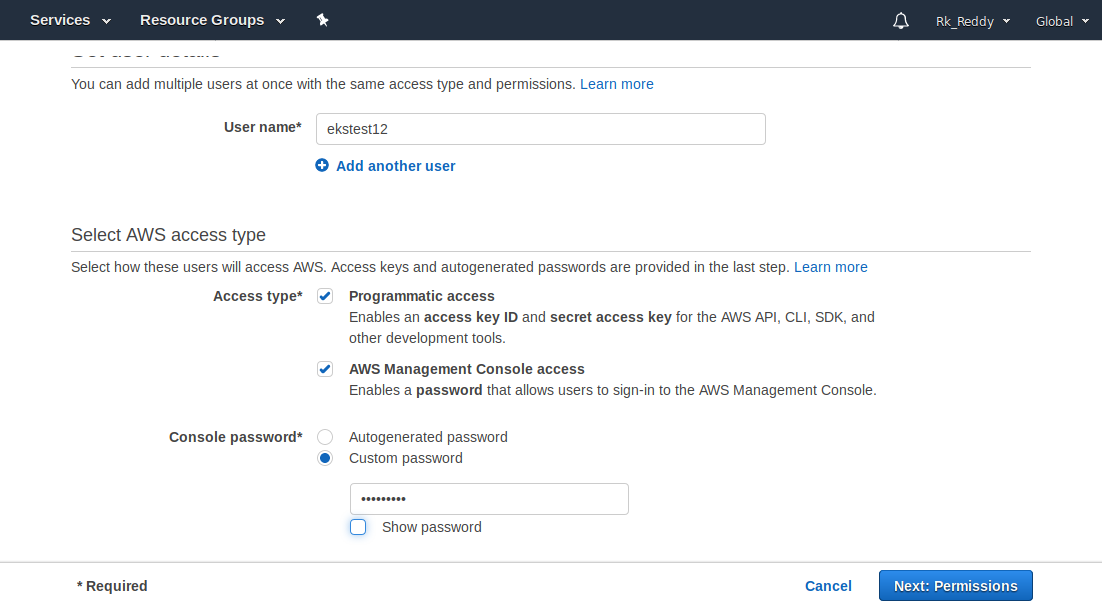
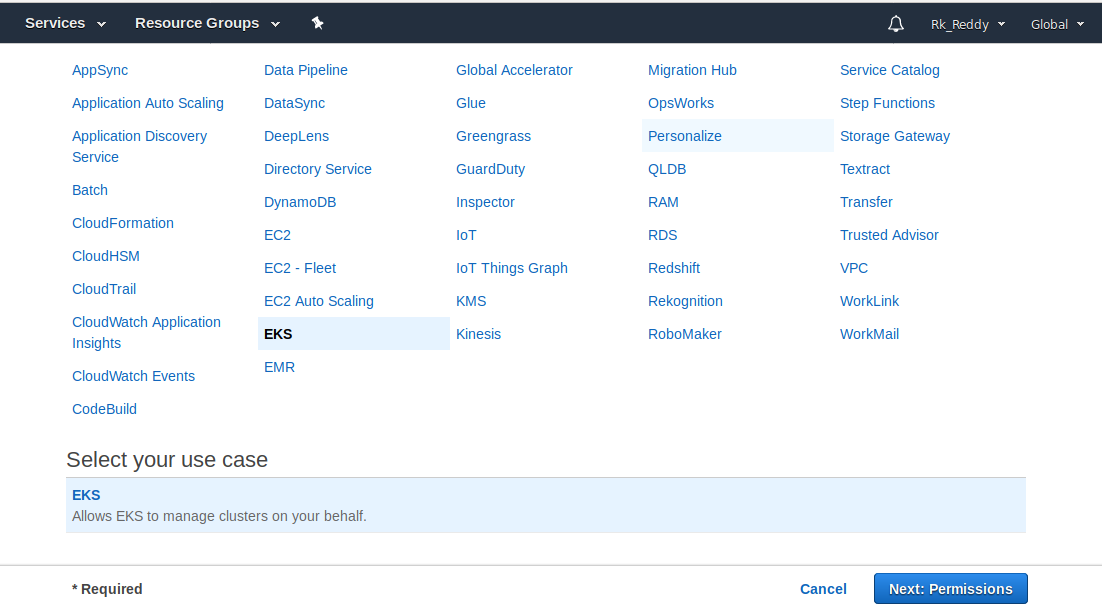
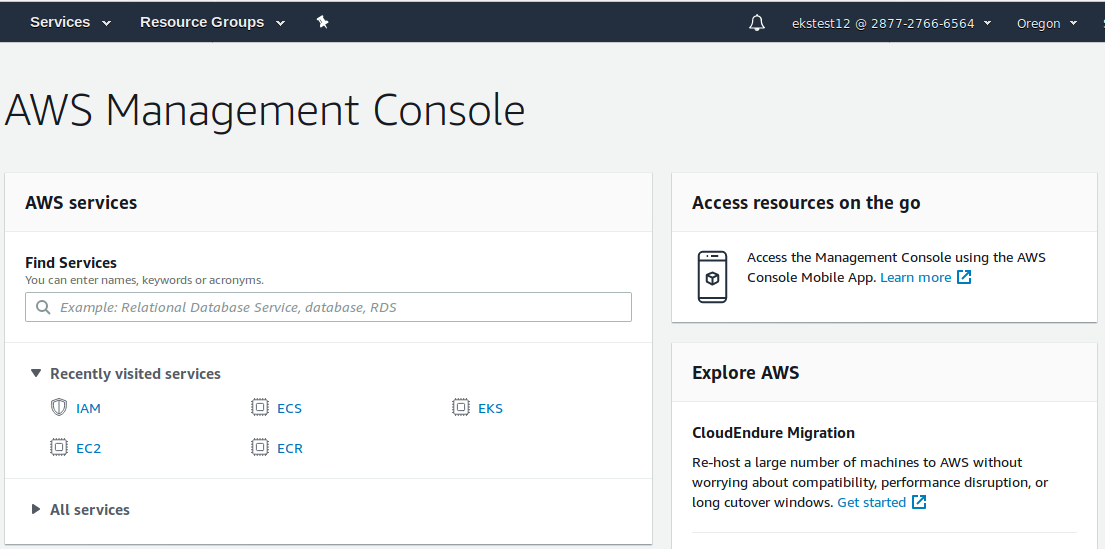
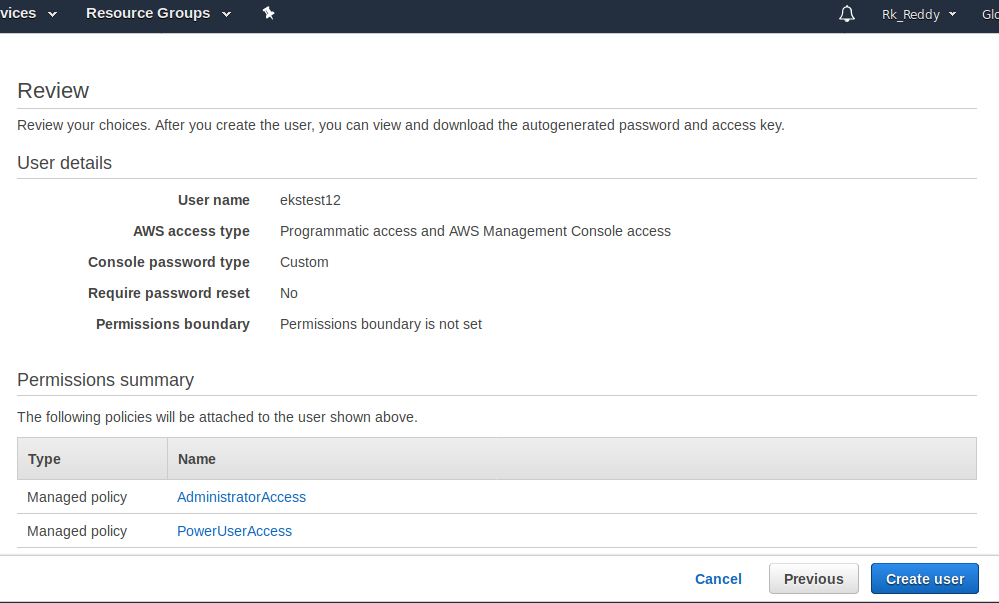
* No control plane to manage
* Conformant and Compatible
* Secure by default
* Optimized for cost
* Built with the community.

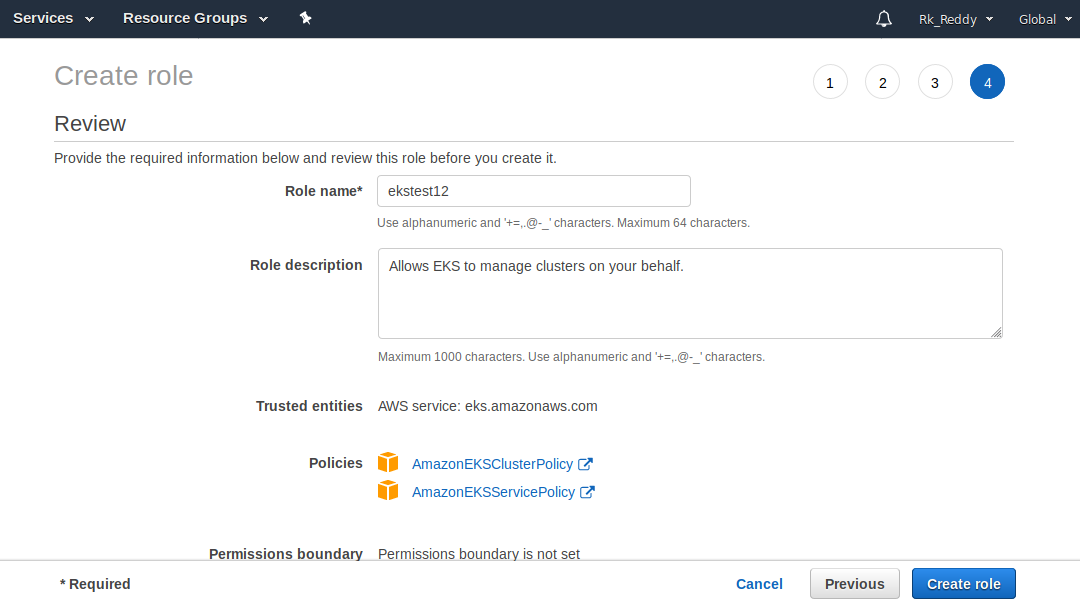
**HOW IT WORKS**

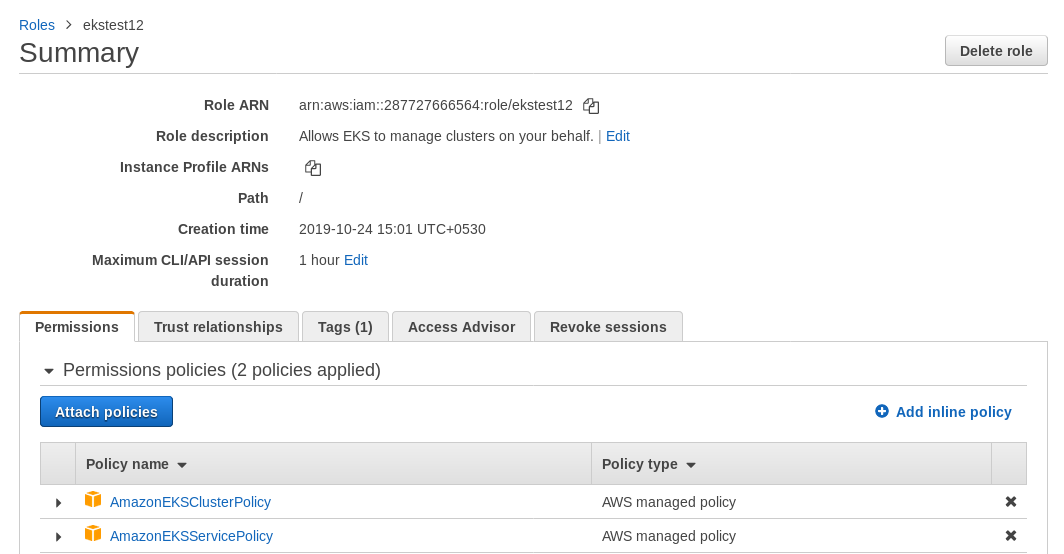


**1. Create a role in IAM and give EKS permissions**

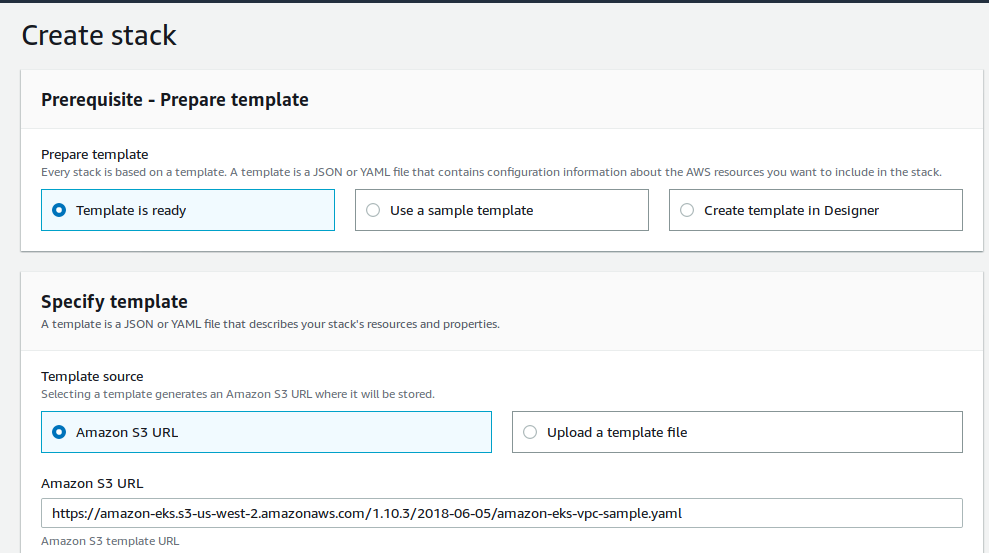
**First create an IAM user. Here, I created “ekstest12” and I given EKS permissions.**

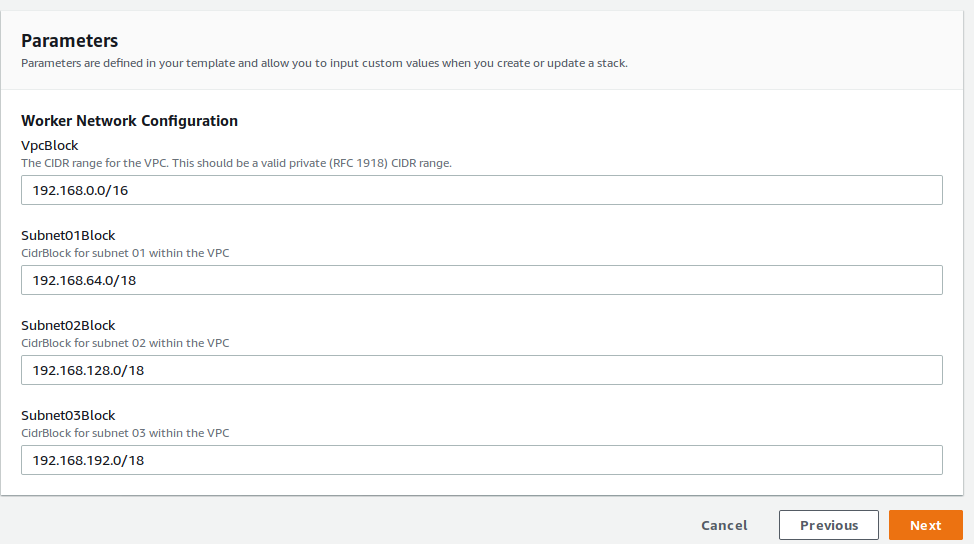




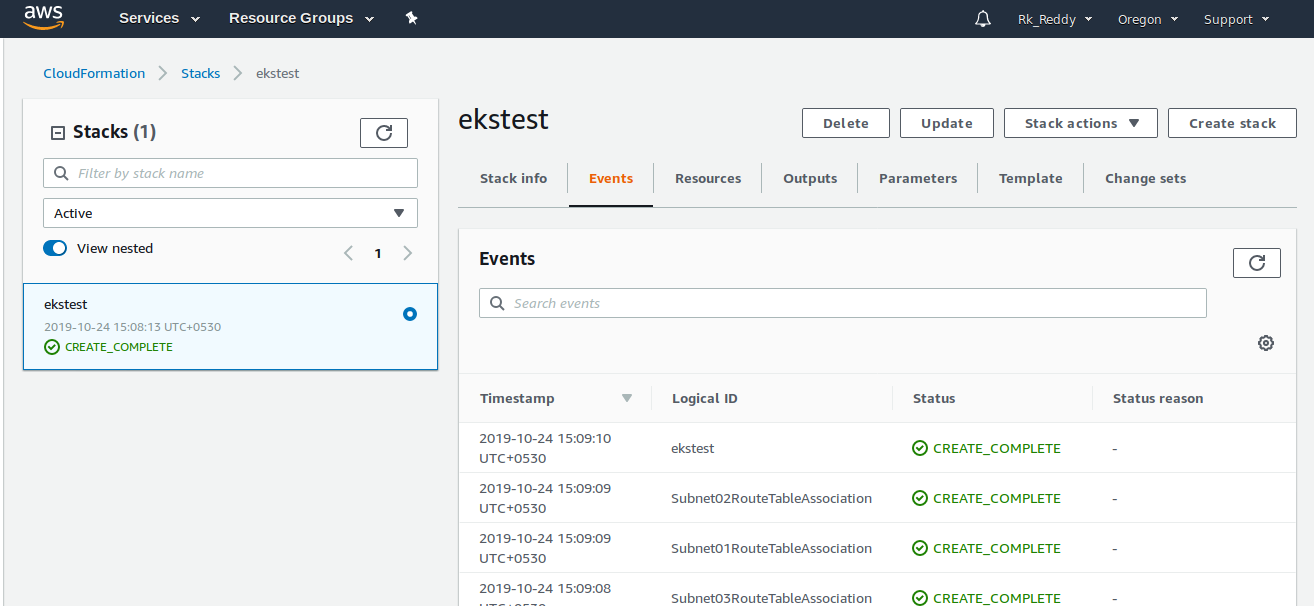


**2.Open service “Cloud Formation” and create a stack and fill the parameters which allows us to create a stack**

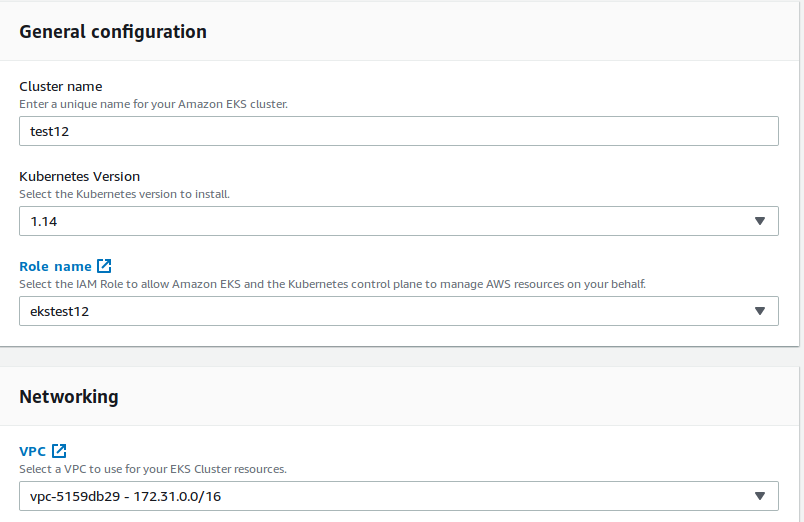


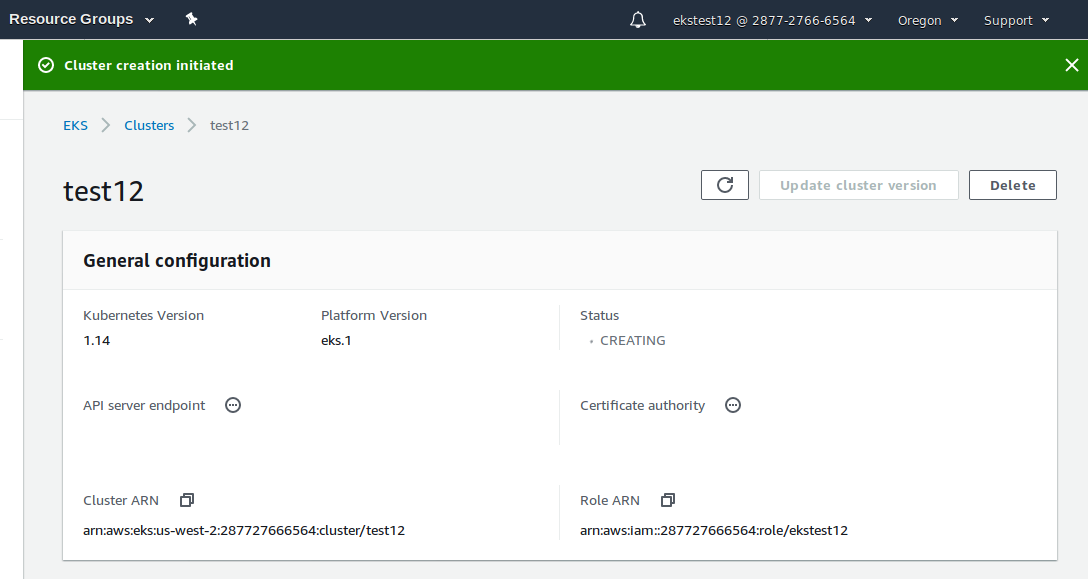


**Here, I created ekstest stack**

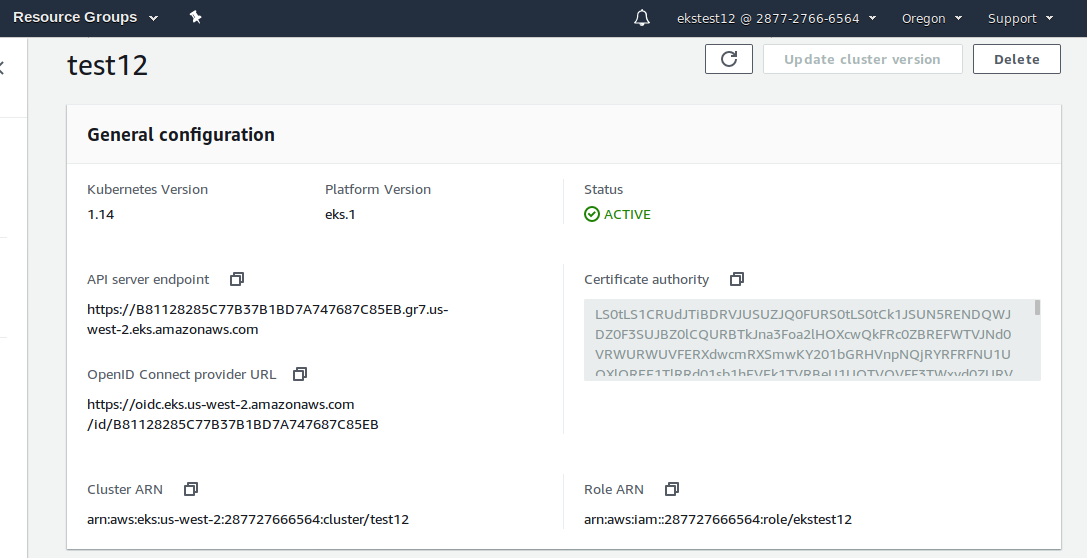


**3. Now, Open AWS management console using IAM user which you created. Open EKS and Create a cluster using the VPC, Subnets and Security Groups.**





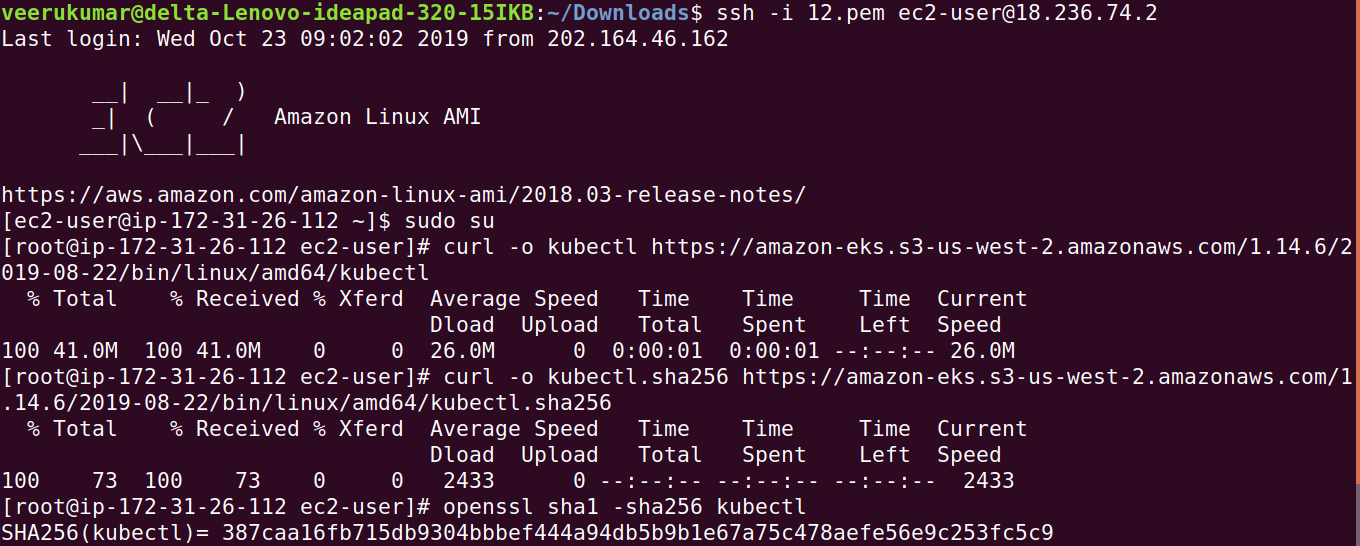
**Here, I created “test12” cluster.**

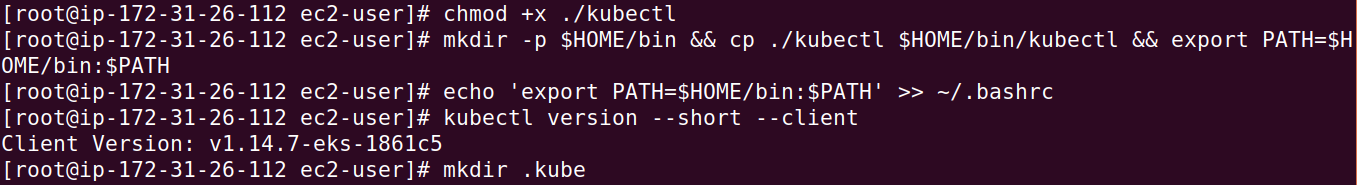


**4. Installing and configuring kubectl in our terminal.**

**Here, Kubectl used for communication with the cluster API server.**

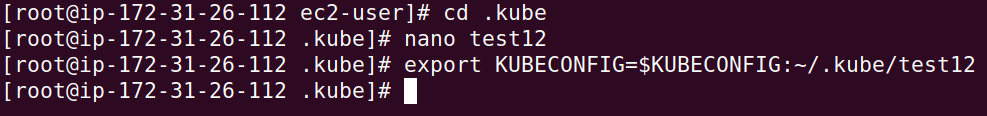
**First, we need to launch an EC2 server.**

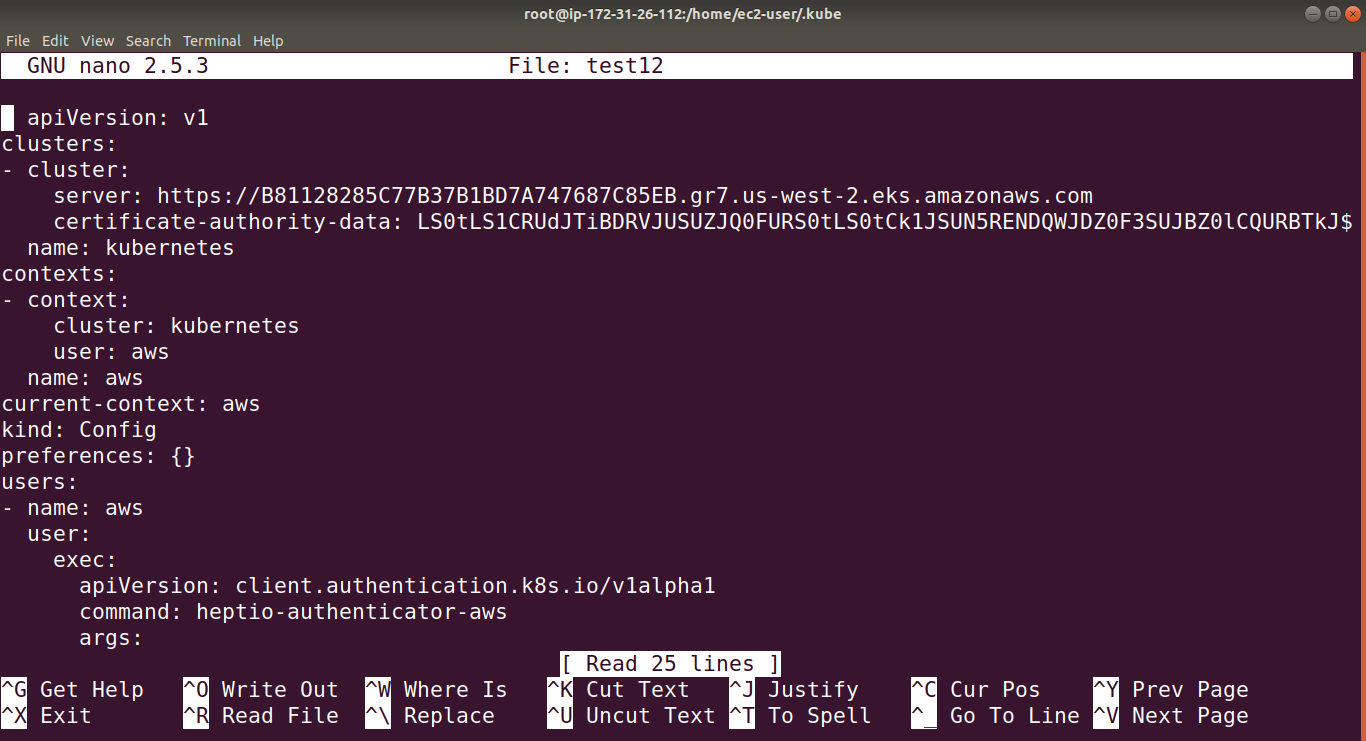




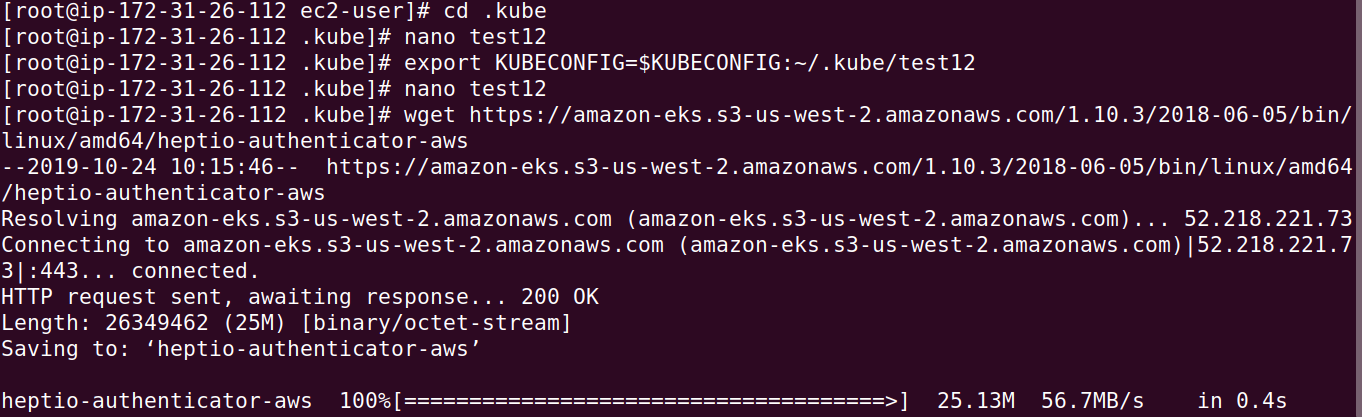
**Create one directory named .kube and create a file inside the directory.**

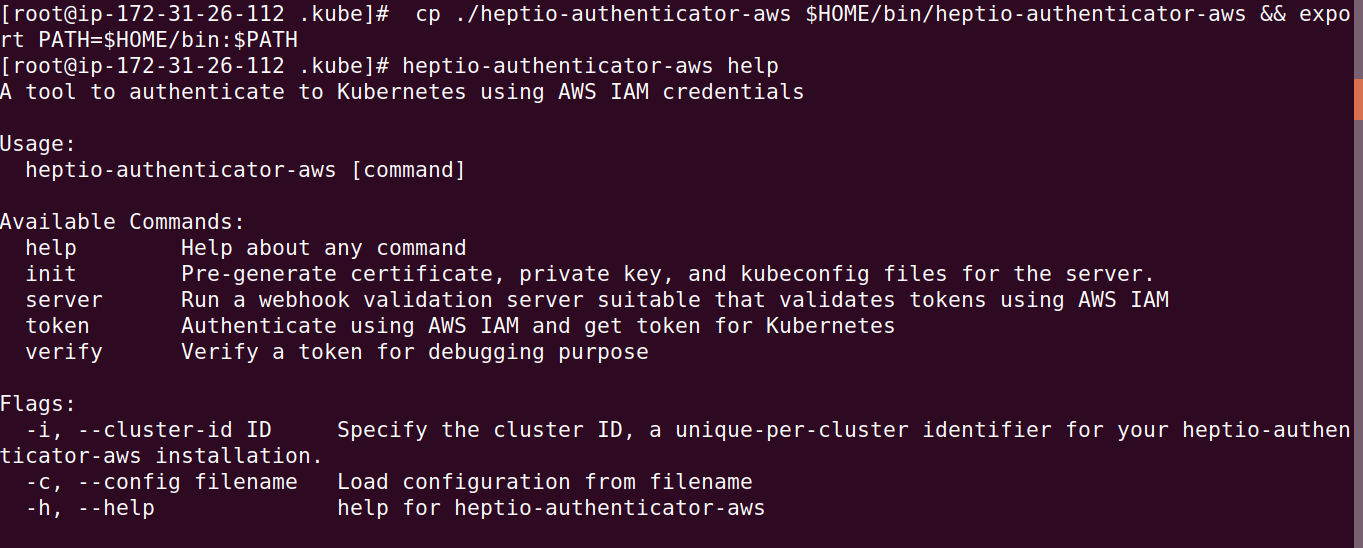
**Here, I named the file as test12.**





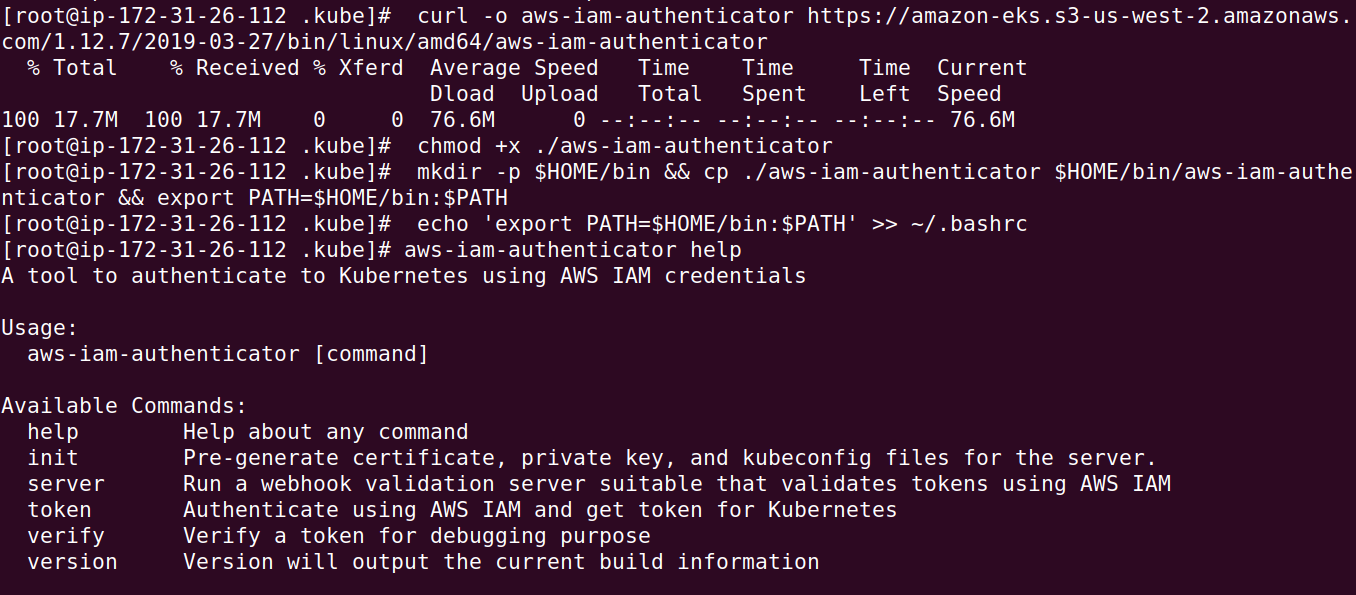
**5. Install hepito-authenticator-AWS for EKS**

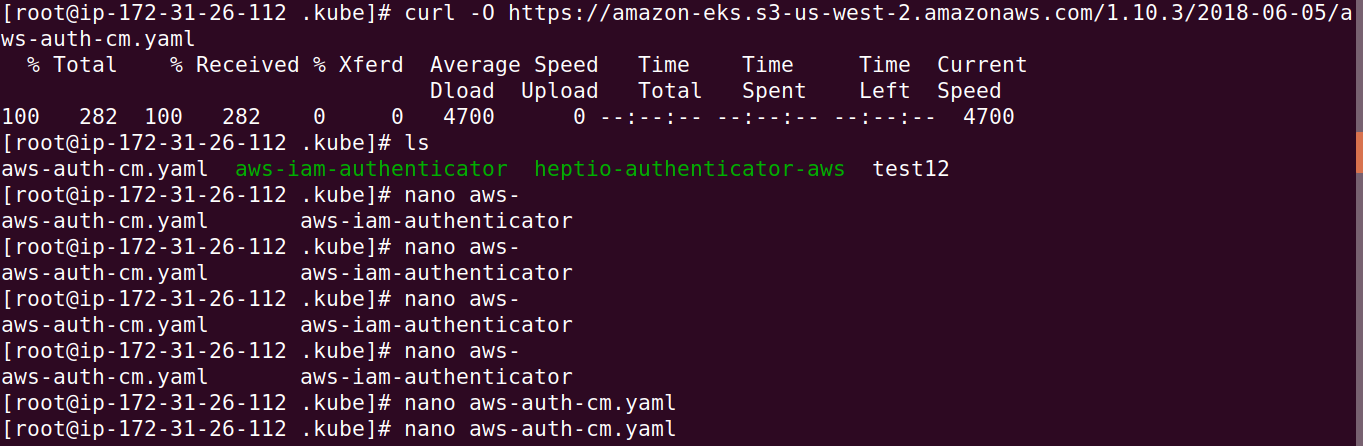




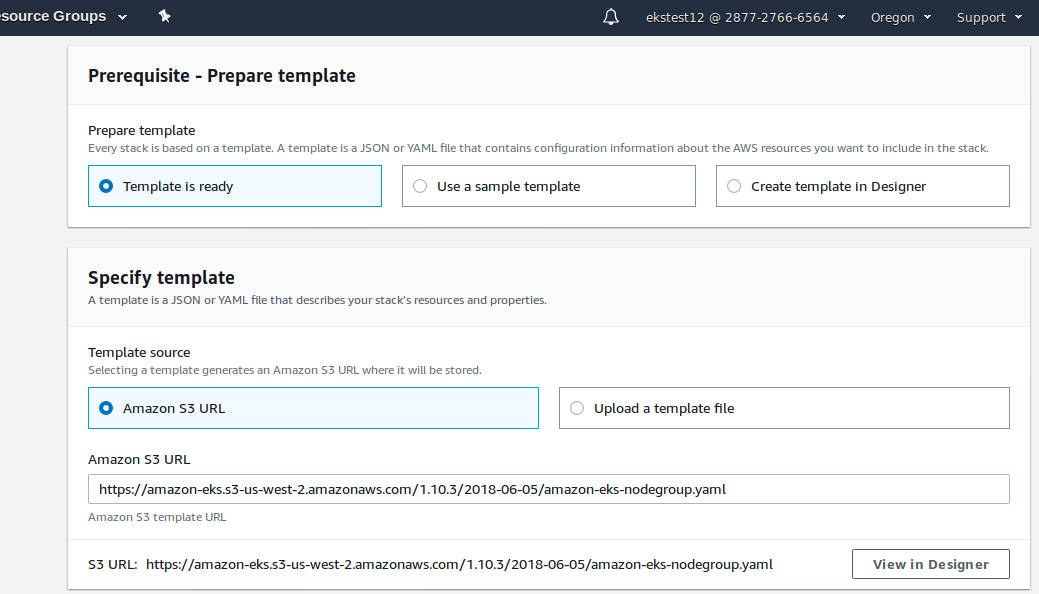
**(or)**

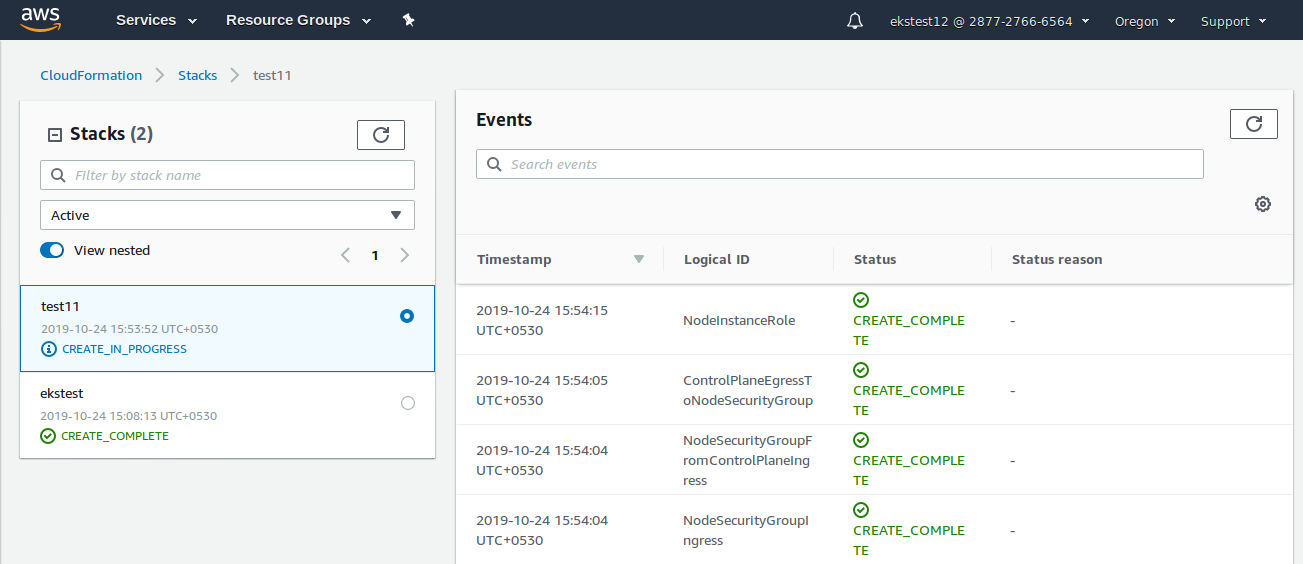
**> Installation of IAM authenticator for EKS**



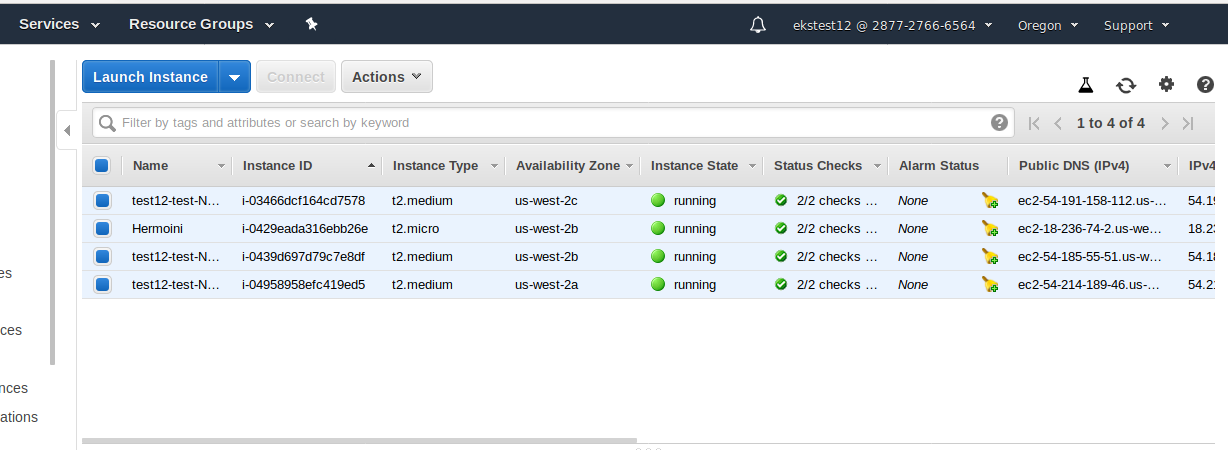


**6.Create a stack for nodes to join a cluster. Here, I created a stack named “test11” under Cloud Formation.**





**Here, We can see that instances are created.**



**7. To enable work nodes to join the cluster.**

**apiVersion: v1**

**kind: ConfigMap**

**metadata:**

**name: aws-auth**

**namespace: kube-system**

**data:**

**mapRoles: |**

**- rolearn: <** **NodeInstanceRole> #value of nodeinstancerole.**

**username: system:node:{{EC2PrivateDNSName}}**

**groups:**

**- system:bootstrappers**

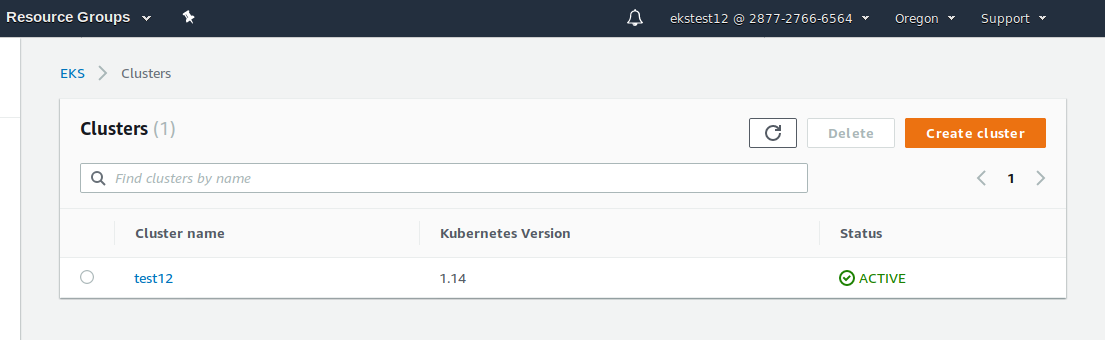
**- system:nodes**

**Configure cluster and EKS**

**Here, We can create cluster in two ways using command.**

**(or)**

**We can create cluster manually in AWS management console. Here, I created Manually.**



**Above cluster “test12” created Manually in AWS Management Console.**

**Commands for creating and deleting the cluster.**

**> eksctl create cluster --name=test12 --nodes=2 --region=us-west-2**

**> eksctl delete cluster --name=test12 --nodes=2 --region=us-west-2**

