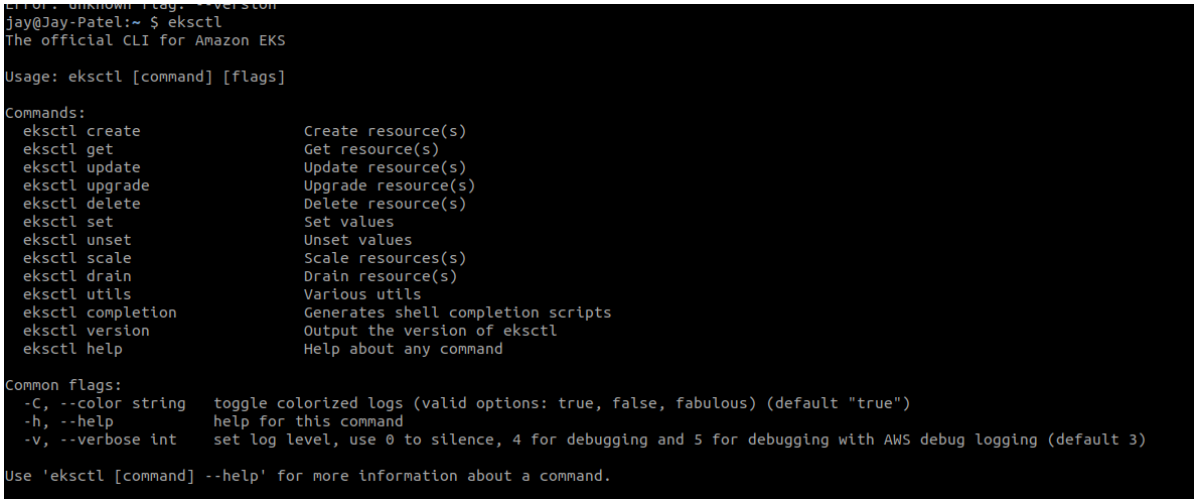
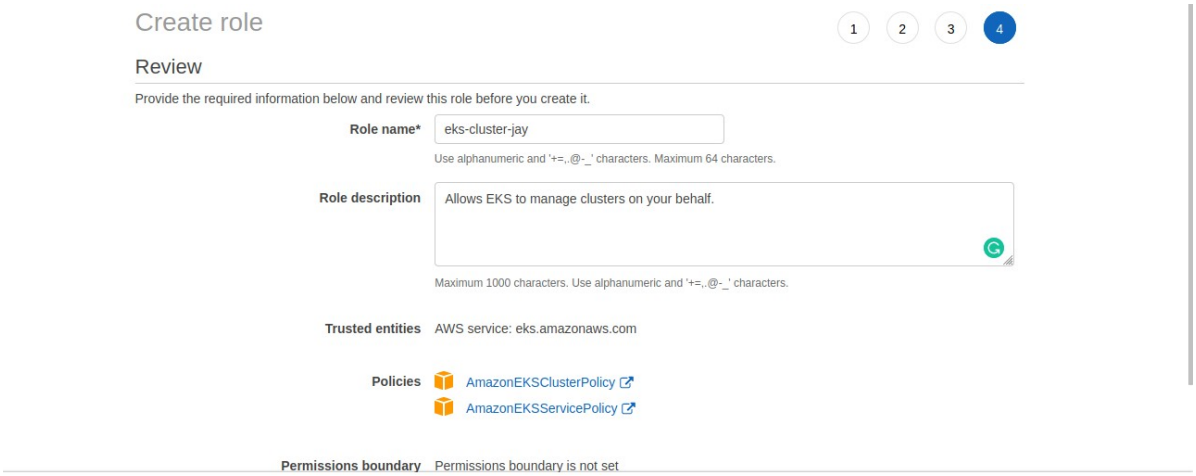
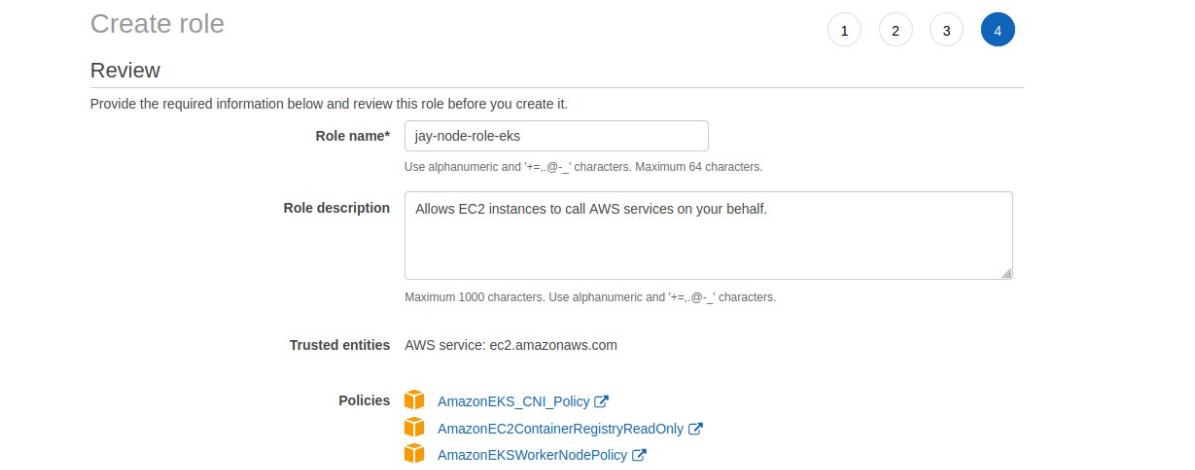
EKS-1

Create eks cluster using eksctl

1. During creation, Specify
    - Cluster name
    - Kubernetes version
    - Control plane role
    - Subnets for Control Plane
    - Control Plane security Group
    - Add tag: owner, purpose on Control Plane
    - Node Group Name
    - Node Instance Role
    - Subnets for Node Group
    - Node Instance SSH key pair
    - Node Instance Security Group
    - Node Instance Instance Type
    - Node Instance Disk
    - Add tag: owner, purpose on Node Group
    - Node Group Size: min, max







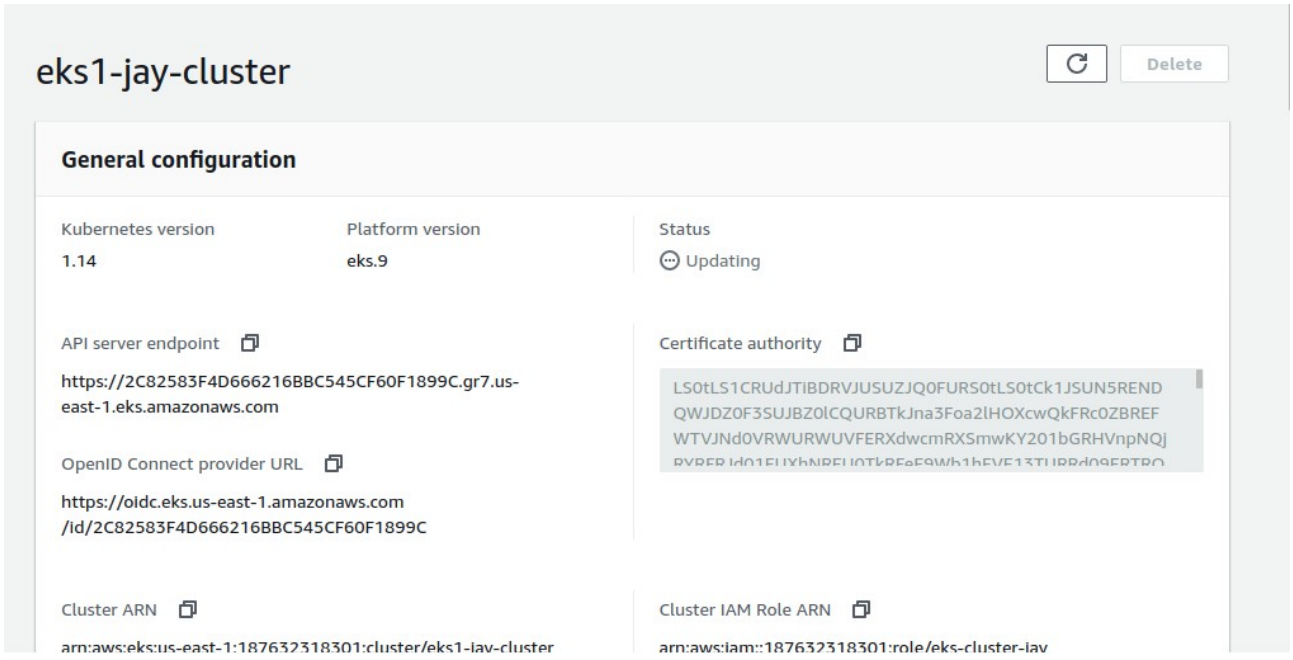

```yaml
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: eks1-jay-cluster
  region: us-east-1
vpc:
  id: "vpc-006b77e885e346f82"
  cidr: "10.0.0.0/16"
  subnets:
    public:
      us-east-1a:
        id: "subnet-087cddd9f9e9fb1b7"
        cidr: "10.0.1.0/24"
      us-east-1b:
        id: "subnet-052a7fb445b1ed477"
        cidr: "10.0.2.0/24"
      us-east-1c:
        id: "subnet-08b47134470eee395"
        cidr: "10.0.3.0/26"

iam:
 serviceRoleARN: "arn:aws:iam::187632318301:role/eks-cluster-jay"
```

```
iam:
  serviceRoleARN: "arn:aws:iam::187632318301:role/eks-cluster-jay"



nodeGroups:
  - name: "jay-ng-1-worker"
    instanceType: t2.micro
    minSize: 1
    desiredCapacity: 2
    maxSize: 3
    availabilityZones: ["us-east-1c", "us-east-1b","us-east-1a"]
    volumeSize: 20
    iam:
     instanceProfileARN: "arn:aws:iam::187632318301:role/jay-node-role-eks"
    ssh:
     publicKeyName: Jay-ALB
     allow: true
    securityGroups:
      withShared: true
      withLocal: true
      attachIDs: ['sg-07847821be8581a15']
    tags:
      'owner': 'jay-eks1-node'
      'purpose': 'jay-eks1-exercise'
```

ne 28, Column 21

```
jay@Jay-Patel:Videos $ eksctl create cluster -f jay-eks.yml
[ℹ]  eksctl version 0.16.0
[ℹ]  using region us-east-1
[✔]  using existing VPC (vpc-006b77e885e346f82) and subnets (private:[] public:[subnet-087cddd9f9e9fb1b7 subnet-052a7fb445b1ed477 subnet-08b47
4470eee395])
[!]  custom VPC/subnets will be used; if resulting cluster doesn't function as expected, make sure to review the configuration of VPC/subnets
[ℹ]  nodegroup "jay-ng-1-worker" will use "ami-0f15d55736fd476da" [AmazonLinux2/1.14]
[ℹ]  using EC2 key pair "Jay-ALB"
[ℹ]  using Kubernetes version 1.14
[ℹ]  creating EKS cluster "eks1-jay-cluster" in "us-east-1" region with un-managed nodes
[ℹ]  1 nodegroup (jay-ng-1-worker) was included (based on the include/exclude rules)
[ℹ]  will create a CloudFormation stack for cluster itself and 1 nodegroup stack(s)
[ℹ]  will create a CloudFormation stack for cluster itself and 0 managed nodegroup stack(s)
[ℹ]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=eks1-jay-clus
r'
[ℹ]  CloudWatch logging will not be enabled for cluster "eks1-jay-cluster" in "us-east-1"
[ℹ]  you can enable it with 'eksctl utils update-cluster-logging --region=us-east-1 --cluster=eks1-jay-cluster'
[ℹ]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "eks1-jay-cluster" in "us-east-1"
[ℹ]  2 sequential tasks: { create cluster control plane "eks1-jay-cluster", create nodegroup "jay-ng-1-worker" }
[ℹ]  building cluster stack "eksctl-eks1-jay-cluster-cluster"
[ℹ]  deploying stack "eksctl-eks1-jay-cluster-cluster"

[ℹ]  building nodegroup stack "eksctl-eks1-jay-cluster-nodegroup-jay-ng-1-worker"
[ℹ]  deploying stack "eksctl-eks1-jay-cluster-nodegroup-jay-ng-1-worker"
[✔]  all EKS cluster resources for "eks1-jay-cluster" have been created
[✔]  saved kubeconfig as "/home/jay/.kube/config"
[ℹ]  adding identity "arn:aws:iam::187632318301:role/jay-node-role-eks" to auth ConfigMap
[ℹ]  nodegroup "jay-ng-1-worker" has 0 node(s)
[ℹ]  waiting for at least 1 node(s) to become ready in "jay-ng-1-worker"
[ℹ]  nodegroup "jay-ng-1-worker" has 2 node(s)
[ℹ]  node "ip-10-0-1-211.ec2.internal" is not ready
[ℹ]  node "ip-10-0-2-104.ec2.internal" is ready
[ℹ]  kubectl command should work with "/home/jay/.kube/config", try 'kubectl get nodes'
[✔]  EKS cluster "eks1-jay-cluster" in "us-east-1" region is ready
jay@Jay-Patel:Videos $
jay@Jay-Patel:Videos $
```

```
jay@Jay-Patel:Videos $ eksctl get cluster
NAME                    REGION
diksha-eks              us-east-1
eks1-jay-cluster        us-east-1
jay@Jay-Patel:Videos $
```




```
jay@Jay-Patel:~ $ kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   172.20.0.1   <none>        443/TCP   13m
jay@Jay-Patel:~ $ kubectl get node
NAME                           STATUS     ROLES    AGE   VERSION
ip-10-0-1-132.ec2.internal   NotReady   <none>   1s    v1.14.9-eks-1f0ca9
jay@Jay-Patel:~ $
```

2. Authentication Management

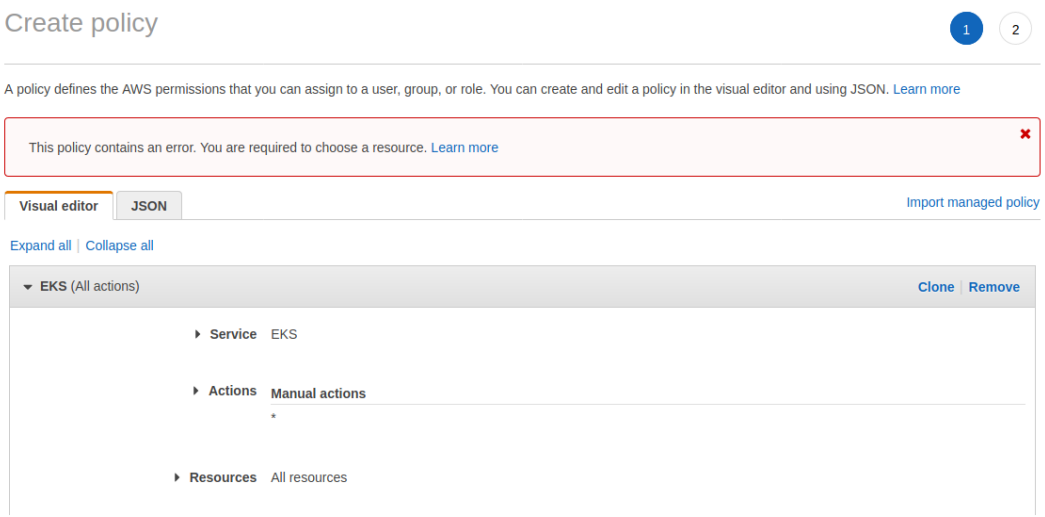
1. Add new 1 IAM user into the cluster

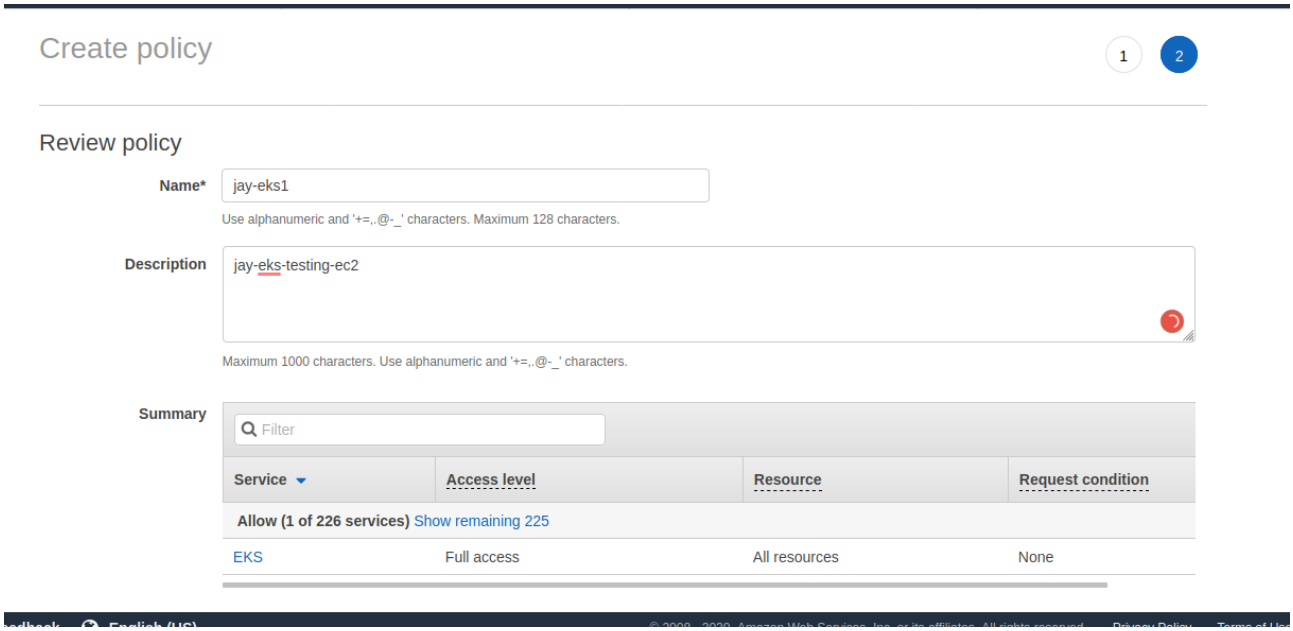
```
jay@Jay-Patel:~ $ kubectl edit -n kube-system configmap/aws-auth
```

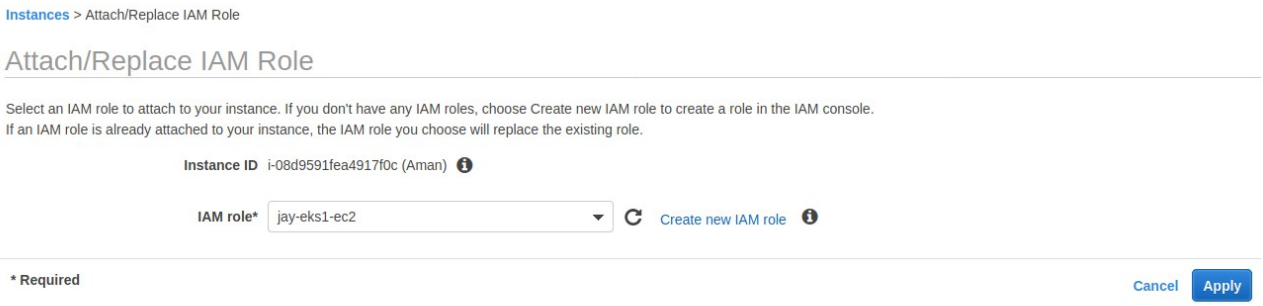
Add this in configmap/aws-auth in kube-system to give access to eks cluster to deploy the pods.

```
mapUsers: |
  - userarn: arn:aws:iam::187632319301:user/jay.patel@tothenew.com
    username: jay
    groups:
      - system:masters
```

2. Enable a EC2 server to access Cluster master API without using access/secret key







aws eks describe-cluster --name eks1-jay-cluster --region us-east-1

3. Eksctl command to terminate the stack

eksctl delete cluster -f jay-eks.yml