

**TO  
THE  
NEW**™



**EKS**

Trainee Name : Chhavi Sharma

Newers ID : 4023

College : UPES

1. Create eks cluster using eksctl

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 Instance Type
- Node Instance Disk
- Add tag: owner, purpose on Node Group
- Node Group Size: min, max

Ans.

Step 1: Install eksctl

```
chhavi@chhavi:~$ curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
chhavi@chhavi:~$ sudo mv /tmp/eksctl /usr/local/bin
chhavi@chhavi:~$ eksctl version
0.16.0
```

Step 2: Write a yaml file for cluster configuration.

```

apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: chhavieks
  region: us-east-1

vpc:
  id: "vpc-093a4253d4c9ab207"
  cidr: "10.0.0.0/16"
  subnets:
    public:
      us-east-1a:
        id: "subnet-0af58143cd499547c"
        cidr: "10.0.0.0/24"
      us-east-1b:
        id: "subnet-0342b9c54db410a4e"
        cidr: "10.0.2.0/24"
      us-east-1c:
        id: "subnet-007036a437b689c11"
        cidr: "10.0.4.0/24"

iam:
  serviceRoleARN: "arn:aws:iam::187632318301:role/eks-service-role"

managedNodeGroups:
  - name: mynodegroup

managedNodeGroups:
  - name: mynodegroup
    instanceType: t3.medium
    desiredCapacity: 2
    minSize: 1
    maxSize: 3
    availabilityZones: ["us-east-1a", "us-east-1b", "us-east-1c"]
    volumeSize: 15
    tags:
      owner: chhavi
      purpose: ekscluster
    ssh:
      publicKeyName: 'ansiblekeypair'
      allow: true

```

Step 3: Run the command below to create the cluster

```

chhavi@chhavi:~/eks$ eksctl create cluster -f eksconfig.yml
[i] eksctl version 0.16.0
[i] using region us-east-1
[✓] using existing VPC (vpc-093a4253d4c9ab207) and subnets (private:[] public:[subnet-007036a437b689c11 subnet-0af58143cd499547c subnet-0342b9c54db410a4e])
[!] custom VPC/subnets will be used; if resulting cluster doesn't function as expected, make sure to review the configuration of VPC/subnets
[i] using EC2 key pair "ansiblekeypair"
[i] using Kubernetes version 1.14
[i] creating EKS cluster "chhavi eks" in "us-east-1" region with managed nodes
[i] 1 nodegroup (mynodegroup) was included (based on the include/exclude rules)
[i] will create a CloudFormation stack for cluster itself and 0 nodegroup stack(s)
[i] will create a CloudFormation stack for cluster itself and 1 managed nodegroup stack(s)
[i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=chhavi eks'
[i] CloudWatch logging will not be enabled for cluster "chhavi eks" in "us-east-1"
[i] you can enable it with 'eksctl utils update-cluster-logging --region=us-east-1 --cluster=chhavi eks'
[i] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "chhavi eks" in "us-east-1"
[i] 2 sequential tasks: { create cluster control plane "chhavi eks", create managed nodegroup "mynodegroup" }
[i] building cluster stack "eksctl-chhavi eks-cluster"
[i] deploying stack "eksctl-chhavi eks-cluster"
[i] building managed nodegroup stack "eksctl-chhavi eks-nodegroup-mynodegroup"
[i] deploying stack "eksctl-chhavi eks-nodegroup-mynodegroup"
[✓] all EKS cluster resources for "chhavi eks" have been created
[✓] saved kubeconfig as "/home/chhavi/.kube/config"

[i] building cluster stack "eksctl-chhavi eks-cluster"
[i] deploying stack "eksctl-chhavi eks-cluster"
[i] building managed nodegroup stack "eksctl-chhavi eks-nodegroup-mynodegroup"
[i] deploying stack "eksctl-chhavi eks-nodegroup-mynodegroup"
[✓] all EKS cluster resources for "chhavi eks" have been created
[✓] saved kubeconfig as "/home/chhavi/.kube/config"
[i] nodegroup "mynodegroup" has 2 node(s)
[i] node "ip-10-0-0-45.ec2.internal" is ready
[i] node "ip-10-0-4-66.ec2.internal" is ready
[i] waiting for at least 1 node(s) to become ready in "mynodegroup"
[i] nodegroup "mynodegroup" has 2 node(s)
[i] node "ip-10-0-0-45.ec2.internal" is ready
[i] node "ip-10-0-4-66.ec2.internal" is ready
[i] kubectl command should work with "/home/chhavi/.kube/config", try 'kubectl get nodes'
[✓] EKS cluster "chhavi eks" in "us-east-1" region is ready

```

Step 4: Check if the cluster and the nodes are created successfully.

chhavi eks

Delete

A new Kubernetes version is available for this cluster. [Learn more](#)

Update now

General configuration

Kubernetes version

1.14

Platform version

eks.9

Status

Active

API server endpoint

https://49D220D28277EDE27B99C57595290447.gr7.us-east-1.eks.amazonaws.com

Certificate authority

LS0tLS1CRUdJTIBDRVJUSUZJQ0FUR50tLS0tCk1JSUN5REN  
DQWJDZ0F3SUJBZ0lCQURBTkJna3Foa2lHOXcwQkFRc0ZBR  
EFWTVJNd0VRWURWUWFERXdwcmRXSmwKY201bGRHVnp

OpenID Connect provider URL

https://oidc.eks.us-east-1.amazonaws.com/id/49D220D28277EDE27B99C57595290447

Cluster ARN

Cluster IAM Role ARN

```

chhavi@chhavi:~/eks$ eksctl get cluster
NAME          REGION
chhavi eks    us-east-1
chhavi@chhavi:~/eks$ kubectl get nodes
NAME                                     STATUS    ROLES    AGE     VERSION
ip-10-0-0-45.ec2.internal               Ready    <none>    8m36s   v1.14.9-eks-1f0ca9
ip-10-0-4-66.ec2.internal               Ready    <none>    8m48s   v1.14.9-eks-1f0ca9
chhavi@chhavi:~/eks$

```

2. Authentication Management
  - a. Add new 2 IAM user into the cluster

Ans.

```

chhavi@chhavi:~/eks$ kubectl edit -n kube-system configmap/aws-auth
configmap/aws-auth edited

```



```

apiVersion: v1
data:
  mapRoles: |
    - groups:
      - system:bootstrappers
      - system:nodes
      rolearn: arn:aws:iam::187632318301:role/eksctl-chhavi-eks-nodegroup-mynode-NodeInstanceRole-QARWH7TMTW
    username: system:node:{{EC2PrivateDNSName}}

  mapUsers: |
    - userarn: arn:aws:iam::187632318301:user/gargi.sharma@tothenew.com
      username: gargi
      groups:
        - system: masters
    - userarn: arn:aws:iam::187632318301:user/ekanshu.dargan@tothenew.com
      username: ekanshu
      groups:
        - system: masters

kind: ConfigMap
metadata:
  creationTimestamp: "2020-04-06T07:21:44Z"
  name: aws-auth
  namespace: kube-system
-- INSERT --

```

20,24 57%

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

Step 1: Create a policy for service eks

The screenshot shows the AWS IAM console's 'Visual editor' for a policy named 'EKS (8 actions)'. The policy is configured with the following settings:

- Service:** EKS
- Actions:**
  - List:** ListFargateProfiles, ListNodegroups, ListTagsForResource, ListUpdates
  - Read:** DescribeCluster, DescribeFargateProfile, DescribeNodegroup, DescribeUpdate
- Resources:**
  - arn:aws:eks:us-east-1:187632318301:cluster/chhavi-eks
  - arn:aws:eks:\*:\*:fargateprofile/\*/\*/\*
  - arn:aws:eks:\*:\*:nodegroup/\*/\*/\*

## Create policy

1

2

### Review policy

Name\*

EKS

Use alphanumeric and '+-=, @- \_' characters. Maximum 128 characters.

Description

|

Maximum 1000 characters. Use alphanumeric and '+-=, @- \_' characters.

Summary

Filter

Service

Access level

Resource

Request condition

Allow (1 of 226 services) [Show remaining 225](#)

EKS

Full: List, Read

Multiple

None

## Step 2: Create a new role and attach the above policy.

## Create role

1

2

3

4

### Review

Provide the required information below and review this role before you create it.

Role name\*

EKS

Use alphanumeric and '+-=, @- \_' characters. Maximum 64 characters.

Role description

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+-=, @- \_' characters.

Trusted entities

AWS service: ec2.amazonaws.com

Policies

EKS

Permissions boundary

Permissions boundary is not set

The new role will receive the following tags

Roles > EKS

## Summary

**Role ARN** arn:aws:iam::187632318301:role/EKS [🔗](#)

**Role description** Allows EC2 instances to call AWS services on your behalf. | [Edit](#)

**Instance Profile ARNs** arn:aws:iam::187632318301:instance-profile/EKS [🔗](#)

**Path** /

**Creation time** 2020-04-06 13:42 UTC+0530

**Last activity** Not accessed in the tracking period

**Maximum CLI/API session duration** 1 hour [Edit](#)

**Permissions** | Trust relationships | Tags (2) | Access Advisor | Revoke sessions

▼ Permissions policies (1 policy applied)

[Attach policies](#) [+ Add inline policy](#)

Policy name ▼	Policy type ▼
▼ EKS	Managed policy <a href="#">✕</a>

Step 3: Now launch an instance and attach the role created to it.

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

### Step 3: Configure Instance Details

**Number of instances** ⓘ  [Launch into Auto Scaling Group ⓘ](#)

**Purchasing option** ⓘ ☐ Request Spot instances

**Network** ⓘ vpc-d38d68b7 | default (default) [↕](#) [Create new VPC](#)

**Subnet** ⓘ subnet-06680a5b651f104dc | default | us-east-1c [↕](#) [Create new subnet](#)  
65512 IP Addresses available

**Auto-assign Public IP** ⓘ Use subnet setting (Enable) [↕](#)

**Placement group** ⓘ ☐ Add instance to placement group

**Capacity Reservation** ⓘ Open [↕](#) [Create new Capacity Reservation](#)

**IAM role** ⓘ EKS [↕](#) [Create new IAM role](#)

Step 4: SSH into the instance and check if you can describe the cluster created.



```
ubuntu@ip-172-31-142-249:~$ aws eks describe-cluster --name chhavi eks --region us-east-1
{
  "cluster": {
    "name": "chhavi eks",
    "arn": "arn:aws:eks:us-east-1:187632318301:cluster/chhavi eks",
    "createdAt": "2020-04-06T07:10:06.993000+00:00",
    "version": "1.14",
    "endpoint": "https://49D220D28277EDE27B99C57595290447.gr7.us-east-1.eks.amazonaws.com",
    "roleArn": "arn:aws:iam::187632318301:role/eks-service-role",
    "resourcesVpcConfig": {
      "subnetIds": [
        "subnet-0af58143cd499547c",
        "subnet-0342b9c54db410a4e",
        "subnet-007036a437b689c11"
      ],
      "securityGroupIds": [
        "sg-0f27d9451079f4e47"
      ],
      "clusterSecurityGroupId": "sg-023b1f5698945412a",
      "vpcId": "vpc-093a4253d4c9ab207",
      "endpointPublicAccess": true,
      "endpointPrivateAccess": false,
      "publicAccessCidrs": [
        "0.0.0.0/0"
      ]
    }
  },
}
```

### 3. Eksctl command to terminate the stack

Ans.

Execute the command below to delete the cluster .

```
chhavi@chhavi:~/eks$ eksctl delete cluster -f eksconfig.yml
[i] eksctl version 0.16.0
[i] using region us-east-1
[i] deleting EKS cluster "chhavi eks"
[i] deleted 0 Fargate profile(s)
[✓] kubeconfig has been updated
[i] 2 sequential tasks: { delete nodegroup "mynodegroup", delete cluster control plane "chhavi eks" [async]
}
[i] will delete stack "eksctl-chhavi eks-nodegroup-mynodegroup"
[i] waiting for stack "eksctl-chhavi eks-nodegroup-mynodegroup" to get deleted
[i] will delete stack "eksctl-chhavi eks-cluster"
[✓] all cluster resources were deleted
chhavi@chhavi:~/eks$
```