

EKS-1

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

First we have to create a yaml file for the cluster

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: t34ak-cluster
  region: us-east-1
iam:
  serviceRoleARN: "arn:aws:iam::187632318301:role/eksServiceRole"
vpc:
  securityGroup: "sg-0997df8eb9d968770"
  id: "vpc-0b061c711cd6ec803"
  cidr: "192.168.0.0/16"
  subnets:
    public:
      us-east-1a:
        id: "subnet-0a5a6b106347d1b70"
        cidr: "192.168.64.0/18"
      us-east-1b:
        id: "subnet-033003c92989d26d9"
        cidr: "192.168.128.0/18"
      us-east-1c:
        id: "subnet-070c80956ba0dde0a"
        cidr: "192.168.192.0/18"
nodeGroups:
  - name: t34ak-cluster
    iam:
      instanceProfileARN: "arn:aws:iam::187632318301:instance-profile/EKSNodeInstanceRole"
    availabilityZones: ["us-east-1a", "us-east-1b", "us-east-1c"]
    ssh:
      allow: true
      publicKeyName: t34ak

"cluster.yml" 43L, 1424C 1,1 Top
```

```
nodeGroups:
  - name: t34ak-cluster
    iam:
      instanceProfileARN: "arn:aws:iam::187632318301:instance-profile/EKSNodeInstanceRole"
    availabilityZones: ["us-east-1a", "us-east-1b", "us-east-1c"]
    ssh:
      allow: true
      publicKeyName: t34ak
    securityGroups:
      withShared: true
      withLocal: false
      attachIDs: [sg-0997df8eb9d968770]
    instanceType: t3.medium
    desiredCapacity: 2
    minSize: 1
    maxSize: 5
    volumeSize: 8
```

43,0-1 Bot

Now run eksctl create cluster command to create the cluster and node group in it

```
fahad@fahad ~/$ eksctl create cluster -f cluster.yml
[i] eksctl version 0.13.0
[i] using region us-east-1
[i] using existing VPC (vpc-0b061c711cd6ec803) and subnets (private:[ ] public:[subnet-0a5a6b106347d1b70 subnet-033003c9
2989d26d9 subnet-070c80956ba0dde0a])
[!] custom VPC/subnets will be used; if resulting cluster doesn't function as expected, make sure to review the configura
tion of VPC/subnets
[i] nodegroup "t34ak-cluster" will use "ami-087a82f6b78a07557" [AmazonLinux2/1.14]
[i] using EC2 key pair "t34ak"
[i] using Kubernetes version 1.14
[i] creating EKS cluster "t34ak-cluster" in "us-east-1" region with un-managed nodes
[i] 1 nodegroup (t34ak-cluster) 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=t34ak-cluster'
[i] CloudWatch logging will not be enabled for cluster "t34ak-cluster" in "us-east-1"
[i] you can enable it with 'eksctl utils update-cluster-logging --region=us-east-1 --cluster=t34ak-cluster'
[i] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "t34ak-clus
ter" in "us-east-1"
[i] 2 sequential tasks: { create cluster control plane "t34ak-cluster", create nodegroup "t34ak-cluster" }
[i] building cluster stack "eksctl-t34ak-cluster-cluster"
[i] deploying stack "eksctl-t34ak-cluster-cluster"
[i] building nodegroup stack "eksctl-t34ak-cluster-nodegroup-t34ak-cluster"
[i] deploying stack "eksctl-t34ak-cluster-nodegroup-t34ak-cluster"
[✓] all EKS cluster resources for "t34ak-cluster" have been created
[!] unable to write kubeconfig , please retry with 'eksctl utils write-kubeconfig -n t34ak-cluster': unable to read ext
isting kubeconfig file "/home/fahad/.kube/config": Error loading config file "/home/fahad/.kube/config": open /home/fahad
/.kube/config: permission denied
[i] adding identity "arn:aws:iam::187632318301:role/EKSNodeInstanceRole" to auth ConfigMap
[i] nodegroup "t34ak-cluster" has 0 node(s)
[i] waiting for at least 1 node(s) to become ready in "t34ak-cluster"
[i] nodegroup "t34ak-cluster" has 2 node(s)
[i] node "ip-192-168-148-213.ec2.internal" is ready
```

We can see list all nodes by kubectl get nodes command

```
root@fahad:cluster # kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
ip-192-168-148-213.ec2.internal    Ready    <none>    15m   v1.14.8-eks-b8860f
ip-192-168-235-201.ec2.internal    Ready    <none>    15m   v1.14.8-eks-b8860f
root@fahad:cluster #
```

Now we can see that cluster is created

The screenshot shows the AWS Management Console interface for the 't34ak-cluster'. The breadcrumb navigation is 'EKS > Clusters > t34ak-cluster'. The cluster name 't34ak-cluster' is displayed at the top with 'Refresh' and 'Delete' buttons. A notification bar indicates 'A new Kubernetes version is available for this cluster' with a 'Learn more' link and an 'Update now' button. The 'General configuration' section contains the following details:

Property	Value
Kubernetes version	1.14
Platform version	eks.9
Status	Active
API server endpoint	https://00D6CBB04950F453FE78A1C2D43E1742.gr7.us-east-1.eks.amazonaws.com
OpenID Connect provider URL	https://oidc.eks.us-east-1.amazonaws.com/id/00D6CBB04950F453FE78A1C2D43E1742
Certificate authority	LS0tLS1CRUdJTiBDRVJUSUZJQ0FUR50tLS0tCk1JSUN5RENDQWJDZ0F3SUJBZ0lCQU... (truncated)
Cluster ARN	arn:aws:eks:us-east-1:187632318301:cluster/t34ak-cluster
Cluster IAM Role ARN	arn:aws:iam::187632318301:role/eksServiceRole

And the instances are also created and running

The screenshot shows the AWS Management Console interface for the instances of the 't34ak-cluster'. The breadcrumb navigation is 'Launch Instance > Connect > Actions'. The search bar shows 'search: t34ak' and 'Add filter'. The table below lists the instances:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
t34ak-cluster...	i-015f758f98257acf	t3.medium	us-east-1b	terminated	None	None	-	-	-
t34ak-cluster...	i-0f1c08fe1bd090542	t3.medium	us-east-1c	terminated	None	None	-	-	-

Below the table, the details for the instance 'i-015f758f98257acf (t34ak-cluster-t34ak-cluster-Node)' are shown. The 'Description' tab is active, displaying the following information:

Property	Value
Instance ID	i-015f758f98257acf
Instance state	terminated
Instance type	t3.medium
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more
Private DNS	-
Private IPs	-
Secondary private IPs	-
VPC ID	-
Subnet ID	-
Network interfaces	-
Source/dest. check	False
Public DNS (IPv4)	-
IPv4 Public IP	-
IPv6 IPs	-
Elastic IPs	-
Availability zone	us-east-1b
Security groups	-
Scheduled events	-
AMI ID	amazon-eks-node-1.14-v20191213 (ami-087a82f6b78a07557)
Platform	-
IAM role	-
Key pair name	t34ak

2. Authentication Management

- Add new 2 IAM user into the cluster
- Add arn of the users whom we want to give access to the cluster


```
sudo bash
File Edit View Search Terminal Tabs Help
sudo bash x v1 cluster.yml
# reopened with the relevant failures.
#
apiVersion: v1
data:
  mapRoles: |
    - groups:
      - system:bootstrappers
      - system:nodes
      rolearn: arn:aws:iam::187632318301:role/EKSNodeInstanceRole
      username: system:node:{{EC2PrivateDNSName}}
  mapUsers: |
    - userarn: arn:aws:iam::187632318301:user/kaushlendra.singh@tothenew.com
      username: kaushlendra
      groups:
        - system: master
    - userarn: arn:aws:iam::187632318301:user/vaibhav.gupta2@tothenew.com
      username: vaibhav
      groups:
        - system: master
kind: ConfigMap
metadata:
  creationTimestamp: "2020-03-15T15:53:57Z"
  name: aws-auth
  namespace: kube-system
  resourceVersion: "71828"
  selfLink: /api/v1/namespaces/kube-system/configmaps/aws-auth
  uid: 2e25f19e-66d5-11ea-a143-123356d6b705
~
~
~
```

Now we can see the nodes using kubectl get nodes on that user's

```
root@vaibhav:/home/vaibhav/.aws# kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
ip-192-168-114-217.ec2.internal     Ready    <none>    46m   v1.14.9-eks-1f0ca9
ip-192-168-181-203.ec2.internal     Ready    <none>    46m   v1.14.9-eks-1f0ca9
root@vaibhav:/home/vaibhav/.aws#
```

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

Login to a instance and download aws, kubectl and aws-iam-authenticator

```
ubuntu@ip-192-168-96-84:~$ aws --version
aws-cli/2.0.3 Python/3.7.3 Linux/4.15.0-1057-aws botocore/2.0.0dev7
ubuntu@ip-192-168-96-84:~$ kubectl get nodes
kubectl: command not found
ubuntu@ip-192-168-96-84:~$ curl -o kubectl https://amazon-eks.s3-us-west-2.amazonaws.com/1.14.6/2019-08-22/bin/linux/amd64/kubectl
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 41.0M 100 41.0M 0 0 15.3M 0 0:00:02 0:00:02 --:--:-- 15.3M
ubuntu@ip-192-168-96-84:~$ chmod +x ./kubectl
ubuntu@ip-192-168-96-84:~$ mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin
ubuntu@ip-192-168-96-84:~$ echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
ubuntu@ip-192-168-96-84:~$ kubectl version --short --client
Client Version: v1.14.7-eks-1861c5
ubuntu@ip-192-168-96-84:~$ kubectl get nodes
The connection to the server localhost:8080 was refused - did you specify the right host or port?
ubuntu@ip-192-168-96-84:~$ curl -o aws-iam-authenticator https://amazon-eks.s3-us-west-2.amazonaws.com/1.14.6/2019-08-22/bin/linux/amd64/
aws-iam-authenticator
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 17.7M 100 17.7M 0 0 7691k 0 0:00:02 0:00:02 --:--:-- 7691k
ubuntu@ip-192-168-96-84:~$ chmod +x ./aws-iam-authenticator
ubuntu@ip-192-168-96-84:~$ mkdir -p $HOME/bin && cp ./aws-iam-authenticator $HOME/bin/aws-iam-authenticator && export PATH=$PATH:$HOME/bin
ubuntu@ip-192-168-96-84:~$ echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
ubuntu@ip-192-168-96-84:~$ aws-iam-authenticator help
A tool to authenticate to Kubernetes using AWS IAM credentials
```

Attach a role whose ARN is present in the aws auth file for cluster

Instances > Attach/Replace IAM Role

Attach/Replace IAM Role

Select an IAM role to attach to your instance. If you don't have any IAM roles, choose Create new IAM role to create a role in the IAM console. If an IAM role is already attached to your instance, the IAM role you choose will replace the existing role.

Instance ID I-0c880c56d8ab48e6b ⓘ

IAM role* EKSNodeInstanceRole ▼

⌂ Create new IAM role ⓘ

* Required

Cancel Apply

Then we will be able to see all the nodes

```
Use "aws-iam-authenticator [command] --help" for more information about a command.
ubuntu@ip-192-168-96-84:~$ kubectl get nodes
The connection to the server localhost:8080 was refused - did you specify the right host or port?
ubuntu@ip-192-168-96-84:~$ aws eks --region us-east-1 update-kubeconfig --name group5
Added new context arn:aws:eks:us-east-1:187632318301:cluster/group5 to /home/ubuntu/.kube/config
ubuntu@ip-192-168-96-84:~$ kubectl get nodes
error: You must be logged in to the server (Unauthorized)
ubuntu@ip-192-168-96-84:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
ip-192-168-114-217.ec2.internal     Ready    <none>   35m   v1.14.9-eks-1f0ca9
ip-192-168-181-203.ec2.internal     Ready    <none>   35m   v1.14.9-eks-1f0ca9
ubuntu@ip-192-168-96-84:~$
```

3. Eksctl command to terminate the stack

```
rishabh@rishabh:Downloads$ eksctl delete cluster --region=us-east-1 --name=group5
[i] eksctl version 0.14.0
[i] using region us-east-1
[i] deleting EKS cluster "group5"
[i] deleted 0 Fargate profile(s)
[✓] kubeconfig has been updated
[i] cleaning up LoadBalancer services
[i] 2 sequential tasks: { delete nodegroup "g5-node1", delete cluster control plane "group5" [async] }
[i] will delete stack "eksctl-group5-nodegroup-g5-node1"
[i] waiting for stack "eksctl-group5-nodegroup-g5-node1" to get deleted
[i] will delete stack "eksctl-group5-cluster"
[✓] all cluster resources were deleted
```