Creating a Kubernetes cluster with EKS

Prerequisites

- 1. AWS Account
- 2. AWS CLI to use Kubectl utility
- 3. Instance (To manage cluster by using Kubectl)

Steps Involved:

Step 1: Create IAM role for EKS Cluster

Use cases for other AWS services:

EKS

EKS

Allows EKS to manage clusters on your behalf.

EKS - Cluster

Allows access to other AWS service resources that are required to operate clusters managed by EKS.

IAM > Roles > eks-cluster-role

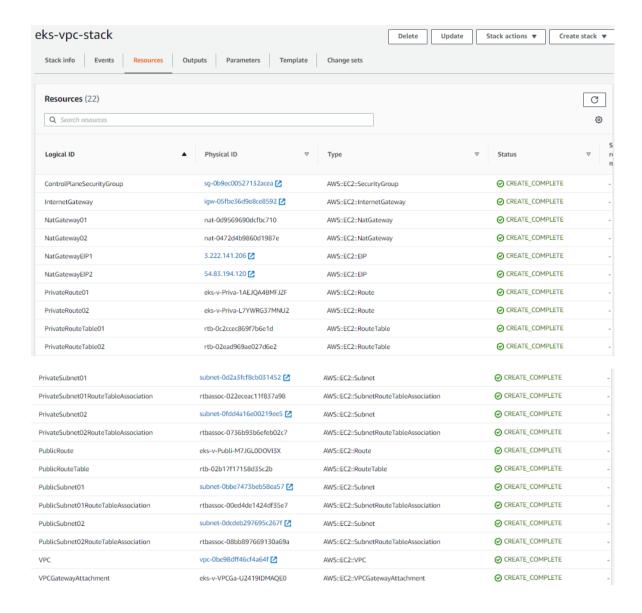
eks-cluster-role

Allows access to other AWS service resources that are required to operate clusters managed by EKS.

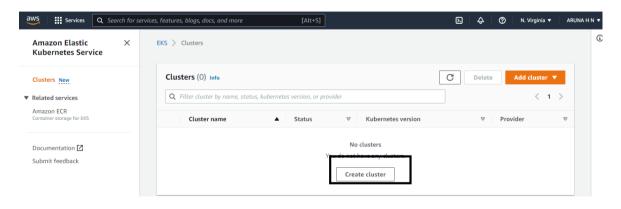
Summary

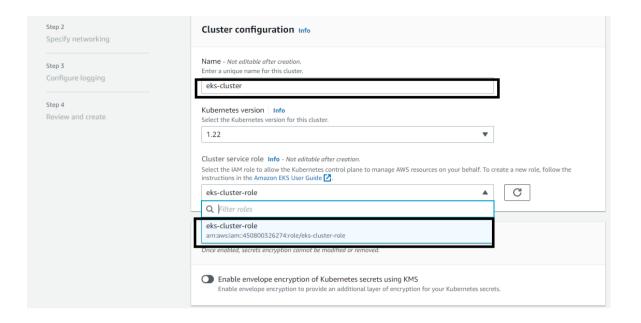
Step 2: Create Dedicated VPC for the EKS Cluster

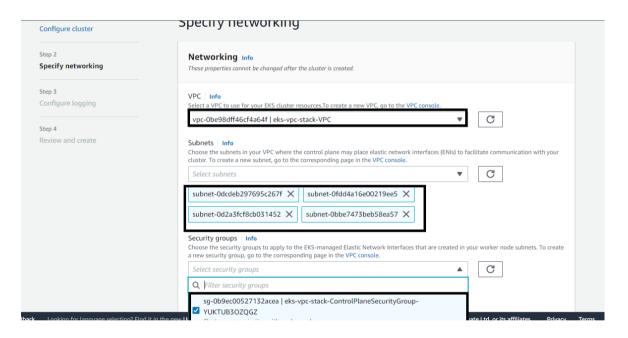
Create Public and Private Subnet using CloudFormation: https://amazon-eks.s3.us-west-2.amazonaws.com/cloudformation/2020-07-23/amazon-eks-vpc-private-subnets.yaml

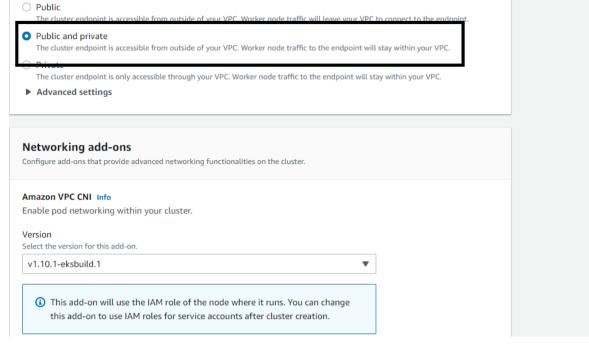


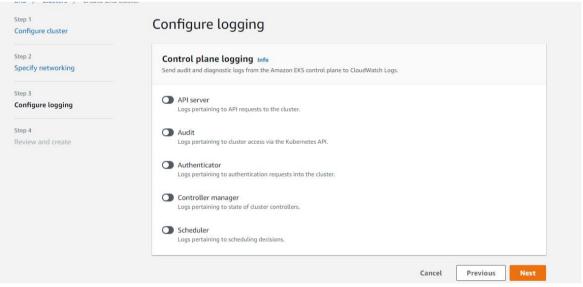
Step 3: Create EKS Cluster

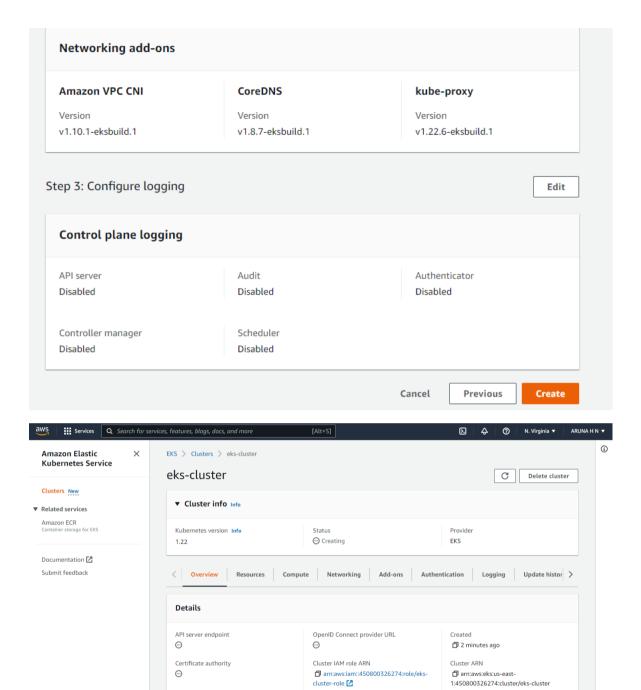












1:450800326274:cluster/eks-cluster

Now Master Node is Created.

Create and AWS CLI and kubectl

step 4: Install AWS CLI on Linux ->

aws --version

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86 64-2.0.30.zip" -o "awscliv2.zip"

unzip awscliv2.zip

sudo ./aws/install

pip3 install --user --upgrade awscli

Step 5: Configure AWS CLI

secret Key:

Access Key

Step6: Installing Kubectl

https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html

a. For Version 1.22 -->

curl -o kubectl https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.6/2022-03-09/bin/linux/amd64/kubectl

b. Verify the downloaded binary with the SHA-256 sum for your binary.

curl -o kubectl.sha256 https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.6/2022-03-09/bin/linux/amd64/kubectl.sha256

```
[root@ip-172-31-82-181 ec2-user]# curl -o kubectl https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.6/2022-03-09/bin/linux/amd64/kubectl
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 44.7M 100 44.7M 0 0 5210k 0 0:00:08 0:00:08 --:--:- 5685k
[root@ip-172-31-82-181 ec2-user]# curl -o kubectl.sha256 https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.6/2022-03-09/bin/linux/amd64/kubectl.sha256
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 73 100 73 0 0 260 0 --:--:- --:--:- 260
```

Check the SHA-256 sum for your downloaded binary.

openssl sha1 -sha256 kubectl

```
[root@ip-172-31-82-181 ec2-user]# openssl sha1 -sha256 kubectl
SHA256(kubectl)= 860c3d37a5979491895767e7332404d28dc0d7797c7673c33df30ca80e215a07
```

c. Apply execute permissions to the binary.

chmod +x ./kubectl

d. Copy the binary to a folder in your PATH. If you have already installed a version of kubectl, then we recommend creating a \$HOME/bin/kubectl and ensuring that \$HOME/bin comes first in your \$PATH.

mkdir -p \$HOME/bin && cp ./kubectl \$HOME/bin/kubectl && export PATH=\$PATH:\$HOME/bin

e. Add the \$HOME/bin path to your shell initialization file so that it is configured when you open a shell.

echo 'export PATH=\$PATH:\$HOME/bin' >> ~/.bashrc

f. After you install kubectl, you can verify its version with the following command:

kubectl version --short --client

```
[root@ip-172-31-82-181 ec2-user]# kubectl version --short --client
Client Version: v1.22.6-eks-7d68063
```

Step 7: Configure kubectl

aws eks describe-cluster --name eks-cluster

fetch all configurations from master node to kubernetes kubectl on local m/c

aws eks update-kubeconfig --name eks-cluster

```
[root@ip-172-31-82-181 ec2-user]# aws eks update-kubeconfig --name eks-cluster
Added new context arn:aws:eks:us-east-1:450800326274:cluster/eks-cluster to /root/.kube/config
```

configure kubectl configuration to master node

kubectl get svc

```
[root@ip-172-31-82-181 ec2-user]# kubectl get svc

VAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 8m24s
```