

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>

eks-vpc-stack

DeleteUpdateStack actions▼Create stack▼

Stack infoEventsResourcesOutputsParametersTemplateChange sets

Resources (22)

Q Search resources

Logical ID	Physical ID	Type	Status	
ControlPlaneSecurityGroup	sg-0b9ec00527132acea	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
InternetGateway	igw-05fbc36d9e8ce8592	AWS::EC2::InternetGateway	CREATE_COMPLETE	-
NatGateway01	nat-0d9569690dcfbc710	AWS::EC2::NatGateway	CREATE_COMPLETE	-
NatGateway02	nat-0472d4b9860d1987e	AWS::EC2::NatGateway	CREATE_COMPLETE	-
NatGatewayEIP1	3.222.141.206	AWS::EC2::EIP	CREATE_COMPLETE	-
NatGatewayEIP2	54.83.194.120	AWS::EC2::EIP	CREATE_COMPLETE	-
PrivateRoute01	eks-v-Priva-1AEJQA4BMFJZF	AWS::EC2::Route	CREATE_COMPLETE	-
PrivateRoute02	eks-v-Priva-L7YWRG37MNU2	AWS::EC2::Route	CREATE_COMPLETE	-
PrivateRouteTable01	rtb-0c2ccec869f7b6e1d	AWS::EC2::RouteTable	CREATE_COMPLETE	-
PrivateRouteTable02	rtb-02ead969ae027d6e2	AWS::EC2::RouteTable	CREATE_COMPLETE	-
PrivateSubnet01	subnet-0d2a3fcf8cb031452	AWS::EC2::Subnet	CREATE_COMPLETE	-
PrivateSubnet01RouteTableAssociation	rtbassoc-022ceac11f837a98	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
PrivateSubnet02	subnet-0fdd4a16e00219ee5	AWS::EC2::Subnet	CREATE_COMPLETE	-
PrivateSubnet02RouteTableAssociation	rtbassoc-0736b93b6efeb02c7	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
PublicRoute	eks-v-Publi-M7JGL0DOVI3X	AWS::EC2::Route	CREATE_COMPLETE	-
PublicRouteTable	rtb-02b17f17158d35c2b	AWS::EC2::RouteTable	CREATE_COMPLETE	-
PublicSubnet01	subnet-0bbe7473beb58ea57	AWS::EC2::Subnet	CREATE_COMPLETE	-
PublicSubnet01RouteTableAssociation	rtbassoc-00ed4de1424df35e7	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
PublicSubnet02	subnet-0dcdeb297695c267f	AWS::EC2::Subnet	CREATE_COMPLETE	-
PublicSubnet02RouteTableAssociation	rtbassoc-08bb897669130a69a	AWS::EC2::SubnetRouteTableAssociation	CREATE_COMPLETE	-
VPC	vpc-0be98dff46cf4a64f	AWS::EC2::VPC	CREATE_COMPLETE	-
VPCGatewayAttachment	eks-v-VPCGa-U2419IDMAQE0	AWS::EC2::VPCGatewayAttachment	CREATE_COMPLETE	-

Step 3: Create EKS Cluster

Amazon Elastic Kubernetes Service

Clusters New

Related services

Amazon ECR
Container storage for EKS

Documentation

Submit feedback

EKS > Clusters

Clusters (0) Info

Filter cluster by name, status, kubernetes version, or provider

Cluster name

Status

Kubernetes version

Provider

No clusters

Create cluster

Step 2

Specify networking

Step 3

Configure logging

Step 4

Review and create

Cluster configuration

Info

Name - Not editable after creation.

Enter a unique name for this cluster.

eks-cluster

Kubernetes version

Info

Select the Kubernetes version for this cluster.

1.22

Cluster service role

Info - Not editable after creation.

Select the IAM role to allow the Kubernetes control plane to manage AWS resources on your behalf. To create a new role, follow the instructions in the [Amazon EKS User Guide](#).

eks-cluster-role

Q Filter roles

eks-cluster-role

arn:aws:iam::450800326274:role/eks-cluster-role

Once enabled, secrets encryption cannot be modified or removed.

Enable envelope encryption of Kubernetes secrets using KMS

Enable envelope encryption to provide an additional layer of encryption for your Kubernetes secrets.

Configure cluster

Specify networking

Step 2

Specify networking

Step 3

Configure logging

Step 4

Review and create

Networking

Info

These properties cannot be changed after the cluster is created.

VPC

Info

Select a VPC to use for your EKS cluster resources. To create a new VPC, go to the [VPC console](#).

vpc-0be98dff46cf4a64f | eks-vpc-stack-VPC

Subnets

Info

Choose the subnets in your VPC where the control plane may place elastic network interfaces (ENIs) to facilitate communication with your cluster. To create a new subnet, go to the corresponding page in the [VPC console](#).

Select subnets

subnet-0dcdeb297695c267f X

subnet-0fdd4a16e00219ee5 X

subnet-0d2a3cf8cb031452 X

subnet-0bbe7473beb58ea57 X

Security groups

Info

Choose the security groups to apply to the EKS-managed Elastic Network Interfaces that are created in your worker node subnets. To create a new security group, go to the corresponding page in the [VPC console](#).

Select security groups

Q Filter security groups

☒ sg-0b9ec00527132acea | eks-vpc-stack-ControlPlaneSecurityGroup-YUKTUB3OZQGZ

☐ Public

The cluster endpoint is accessible from outside of your VPC. Worker node traffic will leave your VPC to connect to the endpoint.

☒ Public and private

The cluster endpoint is accessible from outside of your VPC. Worker node traffic to the endpoint will stay within your VPC.

☐ Private

The cluster endpoint is only accessible through your VPC. Worker node traffic to the endpoint will stay within your VPC.

► **Advanced settings**

Networking add-ons

Configure add-ons that provide advanced networking functionalities on the cluster.


Amazon VPC CNI [Info](#)

Enable pod networking within your cluster.

Version

Select the version for this add-on.

v1.10.1-eksbuild.1

 This add-on will use the IAM role of the node where it runs. You can change this add-on to use IAM roles for service accounts after cluster creation.

Step 1

Configure cluster

Step 2

Specify networking

Step 3

Configure logging

Step 4

Review and create

Configure logging

Control plane logging [Info](#)

Send audit and diagnostic logs from the Amazon EKS control plane to CloudWatch Logs.



API server

Logs pertaining to API requests to the cluster.



Audit

Logs pertaining to cluster access via the Kubernetes API.



Authenticator

Logs pertaining to authentication requests into the cluster.



Controller manager

Logs pertaining to state of cluster controllers.



Scheduler

Logs pertaining to scheduling decisions.

Cancel

Previous

Next


```
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     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100 73    100 73      0     0  260      0  0:00:00  0:00:00 --:--:-- 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

```
[root@ip-172-31-82-181 ec2-user]# aws eks describe-cluster --name eks-cluster
{
  "cluster": {
    "status": "ACTIVE",
    "endpoint": "https://2BABEA56E63C22F36B9CB3F72C6724C1.sk1.us-east-1.eks.amazonaws.com",
    "logging": {
      "clusterLogging": [
        {
          "enabled": false,
          "types": [
            "api",
            "audit",
            "authenticator",
            "controllerManager",
            "scheduler"
          ]
        }
      ]
    }
  },
  "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
[root@ip-172-31-82-181 ec2-user]#
```

configure kubectl configuration to master node

kubectl get svc

```
[root@ip-172-31-82-181 ec2-user]# kubectl get svc
NAME                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes          ClusterIP     10.100.0.1   <none>        443/TCP    8m24s
[root@ip-172-31-82-181 ec2-user]#
```