Kubernetes Kubeconfig & Connecting to Private Endpoint EKS Cluster

## 1. What is kubeconfig?

A kubeconfig file is a configuration file used by kubectl (and other Kubernetes clients) to communicate with a cluster.

### Contents of kubeconfig:

- Clusters: API server endpoint and certificate authority.  
- Users: Authentication details (tokens, IAM roles, client certs).  
- Contexts: Combination of cluster + user + namespace.

**It tells kubectl:**

Where the cluster API server is located.

How to authenticate (user/role, tokens, certificates).

Whichcontext to use (cluster + namespace + user).

### Default Location:

~/.kube/config  
  
You can specify a different file using:  
kubectl --kubeconfig=/path/to/config get pods

**Structure of kubeconfig:**

apiVersion: v1

kind: Config

clusters:

- name: my-cluster

cluster:

server: https://<cluster-endpoint>

certificate-authority-data: <base64-ca-cert>

users:

- name: my-user

user:

token: <authentication-token>

contexts:

- name: my-context

context:

cluster: my-cluster

user: my-user

namespace: default

current-context: my-context

**clusters** → Defines Kubernetes API server endpoints.

**users** → Credentials for authentication.

**contexts** → Links cluster + user + namespace.

**current-context** → Active context kubectl uses by default.

**Why kubeconfig is Important?**

Without kubeconfig, kubectl doesn’t know:

* Which cluster to talk to.
* How to prove your identity.
* Which namespace to use by default.

## 2. Connecting to an EKS Cluster with Private Endpoint

Amazon EKS clusters can expose their API server as:  
- Public endpoint – Accessible over the internet.  
- Private endpoint – Accessible only within the VPC.  
  
If your cluster uses a private endpoint, you must have network access to the VPC where the cluster resides.

## 1. Steps to Connect

### Step 1: Generate kubeconfig for EKS

Use the AWS CLI to generate kubeconfig:  
aws eks update-kubeconfig --name <cluster-name> --region <region>  
This updates ~/.kube/config with the correct cluster info and authentication.

* Downloads the cluster endpoint & certificate.
* Configures IAM authentication.
* Updates your ~/.kube/config.

### Step 2: Ensure Network Access

Since the API server is private, you need access to the VPC where the EKS cluster resides. Options:

1. Bastion Host (EC2 approach)  
 - Launch an EC2 instance inside the VPC.  
 - SSH into it and run kubectl from there.  
  
2. VPN / Direct Connect  
 - Connect your local machine to the AWS VPC using a VPN or Direct Connect.  
 - Once connected, you can run kubectl directly from your laptop.  
  
3. AWS SSM Session Manager  
 - Use Session Manager to run commands inside an EC2 instance in the VPC.  
 - No need for SSH or open inbound ports.

### Step 3: IAM Authentication

EKS uses IAM for authentication.  
- Ensure your IAM user or role is mapped in the aws-auth ConfigMap inside the cluster.  
- Without this, kubectl will fail with Unauthorized errors.

## 4.Verification Steps

Once kubeconfig, networking, and IAM are set up, test:

kubectl get svc

kubectl get nodes

kubectl get pods -A

## 5. Summary

- kubeconfig = file containing connection details for Kubernetes clusters.  
- For private EKS clusters:  
 1. Generate kubeconfig with aws eks update-kubeconfig.  
 2. Ensure network access to the VPC (via Bastion Host, VPN, or SSM).  
 3. Verify IAM permissions are correctly set in the cluster.