Kubernetes Kubeconfig & Connecting to Private Endpoint EKS Cluster

# 1. What is kubeconfig in Kubernetes?

A kubeconfig file is a YAML configuration file used by Kubernetes command-line tools like kubectl to connect to clusters. It contains all the details needed for authentication and communication.

Default location:

~/.kube/config

Key components:

- clusters → Defines the Kubernetes cluster API server endpoint and CA certificate.

- users → Authentication details (AWS IAM, tokens, certificates, etc.).

- contexts → A combination of cluster + user + namespace.

- current-context → The default context used by kubectl commands.

🔍 Example kubeconfig snippet:

apiVersion: v1  
kind: Config  
clusters:  
- cluster:  
 server: https://ABCD1234.gr7.us-east-1.eks.amazonaws.com  
 certificate-authority-data: <base64-encoded-ca>  
 name: my-eks-cluster  
users:  
- name: aws-user  
 user:  
 exec:  
 apiVersion: client.authentication.k8s.io/v1beta1  
 command: aws  
 args:  
 - eks  
 - get-token  
 - --cluster-name  
 - my-eks-cluster  
contexts:  
- context:  
 cluster: my-eks-cluster  
 user: aws-user  
 namespace: default  
 name: my-context  
current-context: my-context

# 2. Connecting to a Private Endpoint EKS Cluster

When creating an Amazon EKS cluster, the Kubernetes API server endpoint can be Public, Private, or Both.

- Public → Accessible over the internet (with IAM authentication and security groups).

- Private → Accessible only within the VPC.

- Both → Combination of the above.

## Private Endpoint Requirements

To connect to a private-only endpoint, your client machine must have network access to the VPC hosting the EKS cluster.

✅ Options to achieve this:

1. VPN Connection → Use AWS Client VPN, OpenVPN, or a third-party VPN.

2. AWS Direct Connect → For on-premise environments, establish a Direct Connect link to the VPC.

3. Bastion Host (Jump Box) → Deploy an EC2 instance in the VPC, SSH into it, and run kubectl commands from there.

4. AWS PrivateLink / VPC Peering → Connect from another VPC using PrivateLink or VPC peering.

## Steps to Configure kubeconfig

Run the following command from a machine that has network access:

aws eks update-kubeconfig --region <region> --name <cluster\_name>

Test the connection:

kubectl get svc

## 📊 Architecture Diagram

