**LAB#1-Setting up a Secure EKS (Elastic Kubernetes**

**Service) Cluster**

\*\*Tasks\*\*:

- Configure IAM roles for EKS.

- Set up VPC with security groups.

- Launch and test EKS.

- \*\*Documentation\*\*: Explain IAM roles,

VPCs, and the significance of security groups.

**Prerequisite :**

**Create instance for eks cluster**

**Configure IAM roles for EKS**

To configure IAM roles for EKS, you can use the following steps

<https://docs.aws.amazon.com/eks/latest/userguide/eksctl.html>

1. Create an IAM role for your EKS cluster.
   * Open the IAM console and choose Roles.
   * Click Create role.
   * Under Trusted entity type, select AWS service.
   * Under Use cases for other AWS services, select EKS.
   * Choose EKS - Cluster for your use case.
   * Enter a name for your role and click Next.
   * On the Add permissions tab, click Next.
   * Review the permissions that are included in the default EKS cluster policy and make any necessary changes.
   * Click Create role.
2. Attach the IAM role to your EKS cluster.
   * Open the EKS console and choose Clusters.
   * Choose the name of the cluster that you want to attach the IAM role to.
   * Click Edit.
   * Under Role ARN, enter the ARN of the IAM role that you created in step 1.
   * Click Update.
3. Configure your Kubernetes service accounts to use the IAM role.
   * Create a Kubernetes service account for each application that needs to use the IAM role.
   * Annotate the service account with the ARN of the IAM role.

**Set up VPC with security groups**

To set up a VPC with security groups for EKS, you can use the following steps:

1. Create a VPC.
   * Open the VPC console and choose Create VPC.
   * Enter a name for your VPC and select the CIDR block that you want to use.
   * Click Create VPC.
2. Create a subnet for each Availability Zone where you will launch your EKS cluster.
   * Open the VPC console and choose Subnets.
   * Click Create subnet.
   * Select the VPC that you created in step 1 and select the Availability Zone where you want to launch your EKS cluster.
   * Enter a name for your subnet and select the CIDR block that you want to use.
   * Click Create subnet.
3. Create a security group for your EKS cluster.
   * Open the VPC console and choose Security groups.
   * Click Create security group.
   * Enter a name for your security group.
   * Under Description, enter a description for your security group.
   * Under Inbound, add the following rules:
     + TCP 22 from anywhere for SSH access
     + TCP 443 from anywhere for HTTPS access
   * Under Outbound, add the following rules:
     + All TCP ports to all addresses
   * Click Create security group.
4. Associate the security group to the subnets that you created in step 2.
   * Open the VPC console and choose Security groups.
   * Select the security group that you created in step 3 and click Edit.
   * Under VPC, select the subnets that you created in step 2.
   * Click Save.

**Launch and test EKS:**

To launch and test EKS, you can use the following steps:

1. Launch an EKS cluster.
   * Open the EKS console and choose Create cluster.
   * Enter a name for your cluster and select the region where you want to launch it.
   * Select the VPC and subnets that you created in step 2.
   * Select the security group that you created in step 3.
   * Click Create cluster.
2. Wait for your EKS cluster to launch.
   * Once your EKS cluster has launched, you can connect to it using kubectl.

| kubectl get nodes |
| --- |

This command should list the nodes in your EKS cluster.

1. Test your EKS cluster by deploying a simple application.

| kubectl create deployment my-app --image nginx |
| --- |

This command will deploy a simple Nginx application to your EKS cluster.

| kubectl get pods |
| --- |

This command should list the pods in your EKS cluster. You should see a pod named my-app.

| kubectl exec -it my-app-74946c8495-96jls -- nginx -T |
| --- |

============

**Conclusion of this lab:**

Understanding IAM roles, VPCs, and the role of security groups forms the foundation of secure and controlled infrastructure management within AWS.

* IAM Role Management: Utilizing IAM roles ensures controlled access to AWS services, allowing organizations to implement least privilege principles and manage permissions effectively.
* VPC Configurations: Establishing VPCs allows users to design and manage a secure network environment with precise control over networking aspects, facilitating secure communication between resources.
* Security Group Functionality: Leveraging security groups enhances resource-level security by defining traffic rules, thereby controlling and monitoring inbound and outbound traffic flow to AWS resources.

These fundamental AWS components play a critical role in creating a secure and well-structured cloud infrastructure. Employing best practices, such as role segregation, least privilege, and regularly updated security rules, ensures a robust and secure AWS environment for diverse applications and services.