Part 1. AWS Create a Master Node

Install kubectl (if not already installed)

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

or
https://kubernetes.io/docs/tasks/tools/

Install AWS CLI

https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

Configure a Policy

Open the IAM console at https://console.aws.amazon.com/iam/

1. Choose Access management -> Policies
2. Click the "Create policy" button
3. Visual editor -> Service -> Choose a service
4. Filter: EKS, Select EKS
5. Click on a checkbox: "All EKS actions (eks:*)
6. Expand "Resources"
7. Click in a **cluster** a checkbox "Any in this account"
8. Click "Next: Tags"
9. Click "Next: Review"
10. Name: **AccessKubernetesApi**
11. Click the "Create policy" button

Configure User and Access and Secret Key

Open the IAM console at https://console.aws.amazon.com/iam/

1. Choose Access management -> Users
2. Click the "Add users" button
3. User name: **rhombus-user**
4. Select AWS credential type: **Access key - Programmatic access**
5. Click "Next Permissions"
6. Pick the "Attach existing policies directly" option
7. Select the checkbox "**AccessKubernetesApi**" name
8. Click "Next Tags"
9. Click "Next Review"
10. Click on the "Create user" button
11. Copy **Access key** and **Secret access key**
12. Click Close button

(Assign to the existing user)

1. Choose Access management -> Users
2. Choose the name of the user

3. Click on the "Security credentials tab"
4. In the **Access keys** section, choose to **Create access key**.

Create a Role

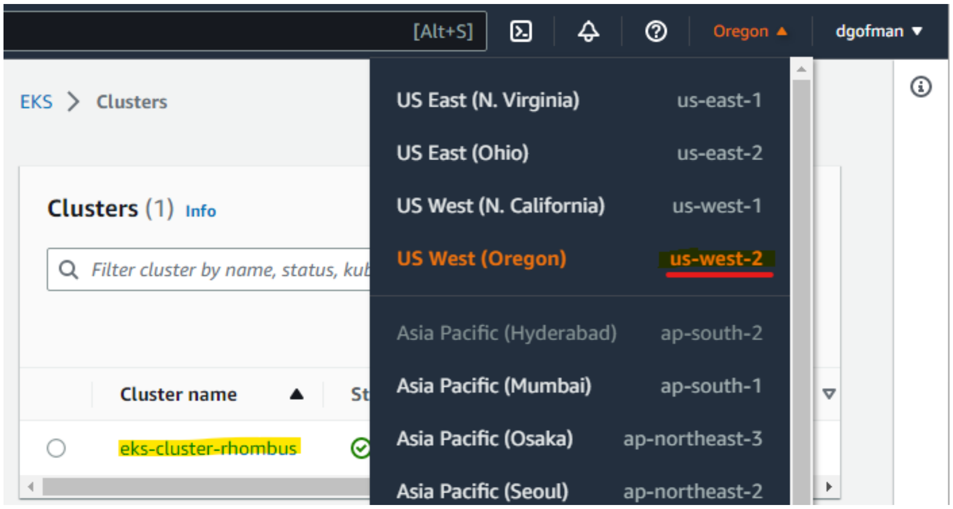
Open the IAM console at https://console.aws.amazon.com/iam/

1. Select "Access management" -> Roles
2. Click the "Create role" button
3.
   a. Trusted entity type: AWS service
   b. Use cases for other AWS services: EKS
   c. EKS – Cluster
4. Click Next button
5. Click Next
6. Role name: **eksClusterRole**
7. Click the "Create role" button

Create EKS Cluster

Open the Elastic Kubernetes Service (EKS) home: https://console.aws.amazon.com/eks/home

1. Select "Add cluster" -> "Create"
2. Name: **eks-cluster-rhombus**
3. Kubernetes version: 1.24
4. Cluster service role: **eksClusterRole**
5. Click Next (Configure cluster)
6. Click Next (Specify networking)
7. Click Next (Configure logging)
8. Click Next (Select add-ons)
9. Click Next (Configure selected add-ons settings)
10. Click Create button (Review and create)
11. Save your region name


[Alt+S]
Oregon ▲
dgofman ▼
EKS > Clusters
Clusters (1) Info
Filter cluster by name, status, kub
Cluster name ▲
St
eks-cluster-rhombus
US East (N. Virginia) us-east-1
US East (Ohio) us-east-2
US West (N. California) us-west-1
US West (Oregon) us-west-2
Asia Pacific (Hyderabad) ap-south-2
Asia Pacific (Mumbai) ap-south-1
Asia Pacific (Osaka) ap-northeast-3
Asia Pacific (Seoul) ap-northeast-2

Configure AWS CLI

1. Open a Terminal
2. Type: aws configure **or** aws configure –profile **default**
3. Enter an **Access key**
4. Enter a **Secret Access Key**
5. Default region name: **us-west-2**
6. Default output format: **json**

Create SSH key

1. Open a Terminal
2. cd ~/.ssh
3. ssh-keygen -t rsa -C "ec2-key"
4. aws ec2 import-key-pair --key-name "ec2-key" --public-key-material fileb://~/.ssh/id_rsa.pub

Configure access to AWS cluster via kubectl

1. Open a Terminal
2. aws eks list-clusters
3. aws eks describe-cluster --query cluster.status --region **us-west-2** --name **eks-cluster-rhombus**

4. aws eks update-kubeconfig --region <region> --name <cluster-name>
   Example:
   aws eks update-kubeconfig --region **us-west-2** --name **eks-cluster-rhombus**
   or
   aws sts get-caller-identity --profile **default** (copy "Arn": xxxxx)
   aws eks update-kubeconfig --profile **default** --region <region> --name <cluster-name> --role-arn <arn>

5. kubectl config view –minify
6. kubectl config get-contexts
7. kubectl config get-users

(Optional) Create cluster from command line

1. Open a Terminal
2. Install https://docs.aws.amazon.com/eks/latest/userguide/eksctl.html
3. eksctl create --cluster <cluster-name> --region <region> --nodegroup-name <node-ame> --node-type t3.micro --nodes 1 --nodes-min 1 --nodes-max 3 --ssh-access
   Example:
   eksctl create cluster --name **rhombus-master** --region **us-west-2** --nodegroup-name **master-node1** --node-type t3.micro --nodes 1 --nodes-min 1 --nodes-max 3 --ssh-access

Add worker node(s)

Create a new permission

Open the IAM console at https://console.aws.amazon.com/iam/

1. Select "Access management" -> Roles
2. Click the "Create role" button
3.
    a. Trusted entity type: AWS service
    b. Use case: EC2
    c. Click the Next button
    d. Filter by: EKS
    e. Pick "AmazonEKSWorkerNodePolicy" and "AmazonEKS_CNI_Policy" items
    f. Click the "Clear filters" button
    g. Filter by: EC2C
    e. Checkbox "AmazonEC2ContainerRegistryReadOnly" option
4. Click Next button
5. Role name: **eksWorkerRole**
6. Click the "Create role" button

Create a worker node(s)

Open the Elastic Kubernetes Service (EKS) home: https://console.aws.amazon.com/eks/home

1. Be sure you selected a primary node region
2. Click on the created cluster
3. Select a "Compute" tab
4. Click the "Add node group" button
5. Name: worker-group1
6. Node IAM role: **eksWorkerRole**
7. Click Next button
8. AMI type: Amazon Linux 2 (AL2_x68_64)
9. Instance types: t3.medium
10. Disk size: 5 Gib
11. Enable: Configure SSH access to nodes
12. Select from the dropdown "ec2-key."
13. Security groups, select all checkboxes.
14. Click the Next button
15. Click the Create button

Cleanup

Open the Elastic Kubernetes Service (EKS) home: https://console.aws.amazon.com/eks/home

1. Be sure you selected a primary node region
2. Click on the created cluster
3. Select the "Compute" tab
4. Select the radio button "worker-group1"
5. Click Delete button
6. Type: worker-group1
7. Click Delete button

   Repeat the same step for "master-node1"
8. Click the "Delete cluster" button
9. Type a cluster name
10. Click Delete

Open the IAM console at https://console.aws.amazon.com/iam/

11. Choose Access management -> Policies
12. Select a radio button **AccessKubernetesApi**
13. Click on the "Actions->Delete" menu button
14. Enter confirmation text
15. Click Delete
16. Choose Access management -> Roles
17. Select eksClusterRole and eksWorkerRole checkboxes
18. Click the Delete button
19. Enter confirmation text
20. Click Delete
21. Choose Access management -> Users
22. Select a checkbox **rhombus-user**
23. Click the Delete button
24. Enter confirmation text
25. Click Delete