

# Elastic Kubernetes Service

By

Sai Sashreek D

Date: - 05/02/2023

# Elastic Kubernetes Service (EKS)

AWS Kubernetes Service is also called as AWS EKS

EKS is a managed service that lets you run Kubernetes on AWS.

It is built for scaling with Kubernetes.

## What is Kubernetes?

Kubernetes is an open-source software.

It helps you deploy and manage containerized applications.

Kubernetes has a large community.

AWS continuously keeps the AWS EKS service updated to the latest Kubernetes features.

## How AWS EKS Works?

AWS EKS is used to run and scale Kubernetes applications in the cloud and on-premises.

Deploy application in 3 different ways:

- Cloud Deployment
- Deployment on your infrastructure
- Deployment with your tools

## What is Amazon EKS?

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed service that you can use to run Kubernetes on AWS without needing to install, operate, and maintain your own Kubernetes control plane or nodes.

Kubernetes is an open source for automating the deployment, scaling, and management of containerized applications.

## About Amazon EKS

Runs and scales the Kubernetes control plane across multiple AWS Availability Zones to ensure high availability.

Automatically scales control plane instances based on load, detects and replaces unhealthy control plane instances, and it provides automated version updates and patching for them.

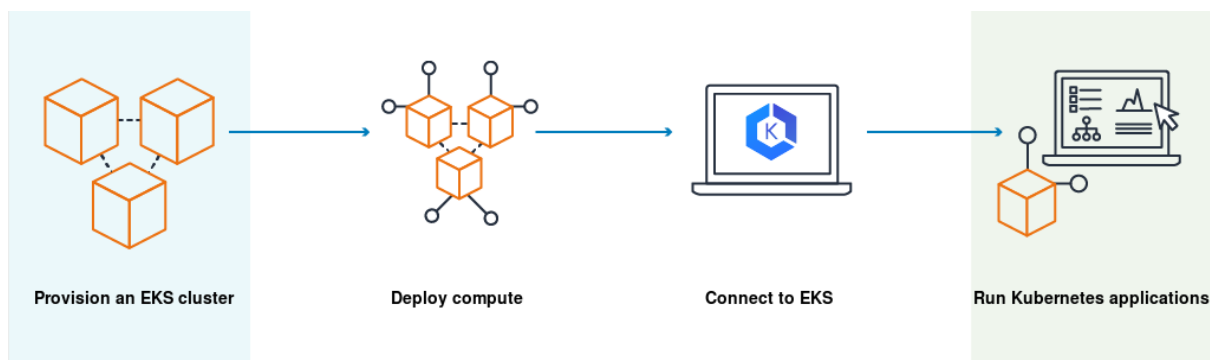
## Amazon EKS Control Plane Architecture

It uses Amazon VPC network policies to restrict traffic between control plane components to within a single cluster.

Control Plane components for a cluster can't view or receive communication from other clusters or other AWS accounts, except as authorized with Kubernetes RBAC policies.

This secure and highly available configuration makes Amazon EKS reliable and recommended for production workloads.

## How does AWS EKS work?



Getting started with AWS EKS is easy:

Create an Amazon EKS cluster in the AWS Management Console or with the AWS CLI or one of the AWS SDKs.

Launch managed or self-managed Amazon EC2 nodes, or deploy your workloads to AWS Fargate.

When your cluster is ready, you can configure your favorite Kubernetes tools, such as kubectl, to communicate with your cluster.

Deploy and manage workloads on your Amazon EKS cluster the same way that you would do with any other Kubernetes environment.

## Amazon Elastic Kubernetes Components (Amazon EKS)

There are 4 components in Amazon Elastic Kubernetes Service.

- Clusters
- Nodes
- Workloads
- EKS Connector

### **Cluster: -**

An EKS cluster is made up of two main components:

- EKS Control Plane
- EKS Nodes

The API server endpoint is public to the internet by default, but you can enable private access to keep communication between nodes and the API server within VPC.

EKS supports two autoscaling products:

Cluster Autoscaler – uses AWS Autoscaling groups.

Karpenter - works directly with the Amazon EC2 fleet.

By default, cluster control plane logs aren't sent to CloudWatch Logs.

To send logs for your cluster, you must enable each log type individually.

EKS cluster uses IAM / OIDC for authentication and Kubernetes RBAC for authorization.

### **EKS Control Plane: -**

It is made up of nodes that run the Kubernetes software (API server & etcd)

Each cluster is single-tenant and unique, and runs on its own set of EC2 instances.

Cluster control plane is provisioned across multiple AZs and frontend by an ELB Network Load Balancer.

Use AWS KMS to encrypt data stored by etcd nodes and associated EBS volumes.

### **EKS Nodes: -**

A cluster consists of one or more EC2 nodes on which pods are scheduled.

Connects to the cluster's control plane via the API server endpoint.

## **Nodes: -**

Nodes must be in same VPC as the subnets you chose when creating a cluster.

From the perspective of the Kubernetes API, nodes represent the compute resources provisioned for your cluster.

Taints and tolerations prevent pods from being scheduled on the wrong nodes.

### **Self-managed nodes**

A cluster can have several node groups.

A node group is a collection of one or more EC2 instances deployed in an Amazon EC2 Auto Scaling group.

In a node group, instances must have the following characteristics:

- Same instance type
- Running the same AMI
- Uses the same EKS node IAM role.

Node groups with different instance types and host operating systems can exist in a cluster.

There are two methods for updating self-managed node groups in a cluster to use a new AMI:

- Migrating to a new node group
- Updating an existing self-managed node group

### **Manage node groups**

Automates the provisioning and life-cycle management of nodes in EKS cluster.

Every managed node is provisioned as part of Amazon EC2 auto Scaling group.

When nodes are launched as part of a managed node group, they are automatically tagged for auto-discovery by Kubernetes Cluster Autoscaler.

Use node group to apply Kubernetes labels to nodes.

Multiple managed node groups can exist in a single cluster.

When you create a managed node group, you have the option of selecting On-Demand or Spot instances.

To ensure that your applications remain available, node updates and terminations drain nodes automatically.

## AWS Fargate

You must first define a Fargate profile before scheduling pods on Fargate in your cluster.

If a pod matches more than one Fargate profile, Amazon EKS picks one at random.

Fargate profiles are immutable and contains the following components:

- Pod execution role
- Subnets
- Selectors
- Namespace
- Labels

Fargate runs only one pod per node.

Pod storage is ephemeral, and data is encrypted with AWS Fargate managed keys.

To encrypt ephemeral pod storage, you can use AWS Fargate managed keys.

## **Workloads: -**

Workloads are deployed in containers and define the applications that run on a Kubernetes cluster.

A pod can contain one or more containers.

Vertical Pod Autoscaler adjusts your pods 'CPU and memory reservations.

Horizontal Pod Autoscaler adjusts the number of pods in a deployment, replication controller, or replica set based on CPU utilization.

## **EKS Connector: -**

Enables you to register and connect any Kubernetes cluster to AWS.

You can view the status, configuration, and workloads of the cluster in the Amazon EKS console after it has been connected.