

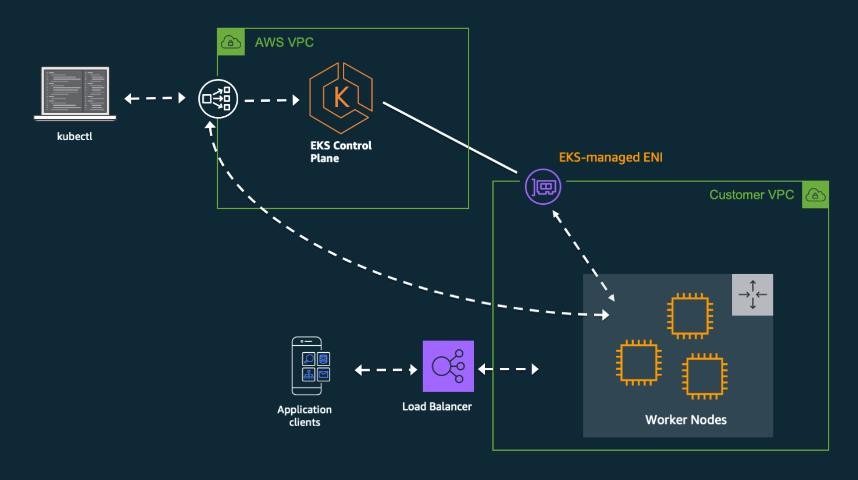
# What is Amazon EKS?

* Amazon Elastic Kubernetes Service (EKS) is a managed Kubernetes service provided by AWS. It makes it easy to deploy, manage, and scale containerized applications using Kubernetes.

# Benefits of Amazon EKS

* Scalability: EKS allows you to scale your applications effortlessly as your requirements change.
* High Availability: It offers a highly available and reliable environment for your applications.
* Easy Management: AWS manages the control plane, so you can focus on your applications.

# Architectural Diagram:



**Setting up Amazon EKS:**

## Create an AWS Account

* + If you haven't already, sign up for an AWS free tier account.

## Install AWS CLI

* + Install and configure the AWS Command Line Interface (CLI) on your local.

**COMMANDS:**

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

sudo apt install unzip unzip awscliv2.zip

sudo ./aws/install -i /usr/local/aws-cli -b /usr/local/bin --update

## Create an IAM Role for EKS

* + Create an IAM role with the necessary permissions for EKS.

**STEPS:**

## Create an IAM User:

* + Go to the AWS IAM console.
  + Create a new IAM user named "eks-admin."
  + Attach the "AdministratorAccess" policy to this user.

## Create Security Credentials:

* + After creating the user, generate an Access Key and Secret Access Key for this user.

1. **Launch AWS instance and get access to the instance**
2. **Configure AWS CLI:**
3. **Configure the AWS CLI with the Access Key and Secret Access Key from step 2:**
4. **Kubernetes tools setup:**

aws configure

* + **Install kubectl:**

Install kubeclt:

curl -LO "https://dl.k8s.io/release/**$(**curl -L -s [https://dl.k8s.io/release/stable.txt](https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl\")**[)](https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl\")**[/bin/linux/amd64/kubectl"](https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl\")

curl -LO "https://dl.k8s.io/release/**$(**curl -L -s [https://dl.k8s.io/release/stable.txt](https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256\")**[)](https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256\")**[/bin/linux/amd64/kubectl.sha256"](https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256\")

echo "**$(**cat kubectl.sha256**)** kubectl" | sha256sum --check

sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

kubectl version --client

## Install eksctl:

Install eksctl:

# for ARM systems, set ARCH to: `arm64`, `armv6` or `armv7`

ARCH=amd64

PLATFORM=$(uname -s)\_$ARCH

curl -sLO "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl\_$PLATFORM.tar.gz"

# (Optional) Verify checksum

curl -sL "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl\_checksums.txt" | grep $PLATFORM | sha256sum --check

tar -xzf eksctl\_$PLATFORM.tar.gz -C /tmp && rm eksctl\_$PLATFORM.tar.gz

sudo mv /tmp/eksctl /usr/local/bin

1. **EKS Cluster Setup:**



Use eksctl to create the EKS cluster

**NOTE:** Make sure to replace <cluster-name> and <region> with your desired values

eksctl create cluster --name <cluster-name> --region <region> --node-type t2.micro --nodes-min 2 --nodes-max 2



Update your kubeconfig to connect to the newly created EKS cluster:

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

1. **Verify Nodes:**

kubectl get nodes

# Basic to Advanced EKS Commands

## Get Cluster Information

aws eks describe-cluster --name <cluster-name> --region <region>

## List Worker Nodes

kubectl get nodes

## Deploy an application

kubectl apply -f <yaml-file>

## Scale a Deployment

kubectl scale deployment <deployment-name> --replicas=<number>

## View Pods in a Namespace

kubectl get pods -n <namespace>

## Check Cluster Events

kubectl get events

## Create a Persistent Volume

kubectl apply -f <pv-definition.yaml>

## Apply a Rolling Update

kubectl set image deployment/<deployment-name> <container-name>=<new- image>

## Enable Autoscaling

kubectl autoscale deployment <deployment-name> --min=3 --max=5