

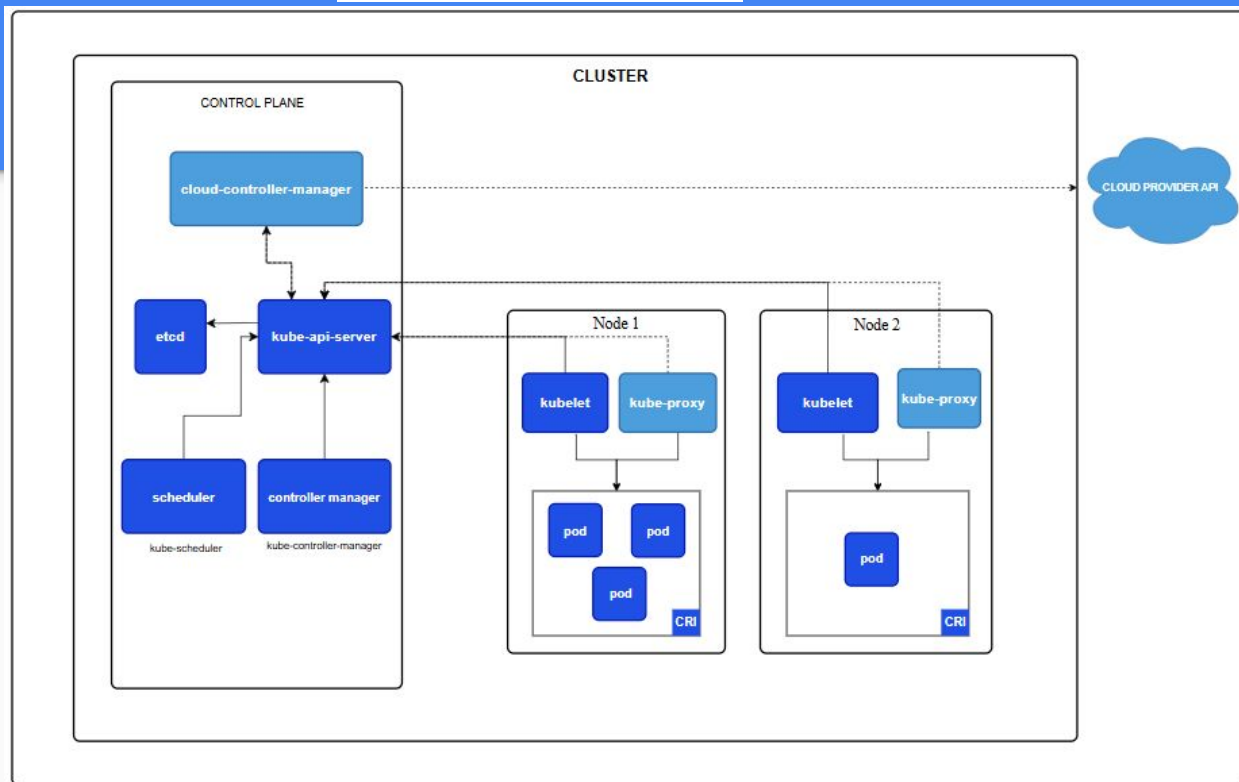
Amazon EKS: Simplified Kubernetes Management



kubernetes

Kubernetes

- Open source orchestration tool
- Orchestrates and manages containers
- Self healing
- Scaling
- Load balancing
- Rolling updates



EKS

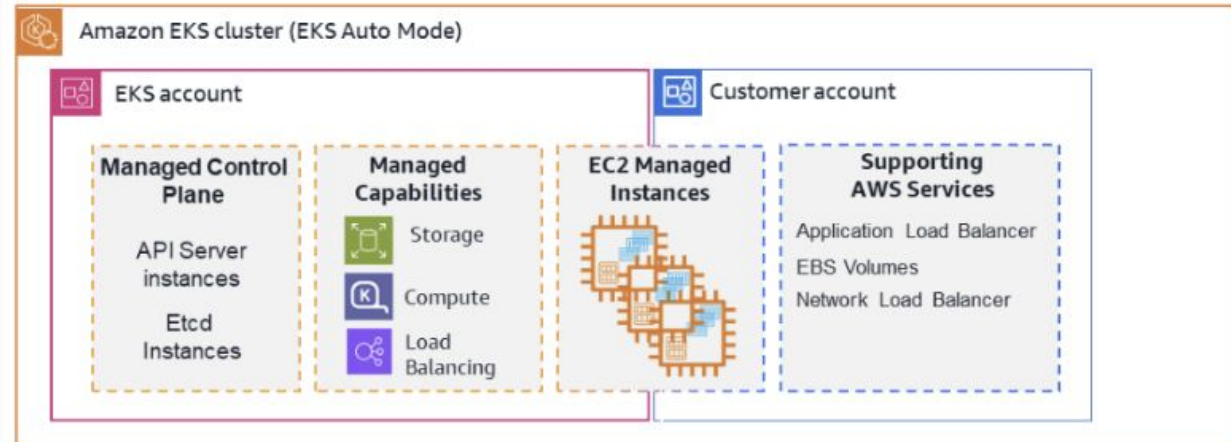
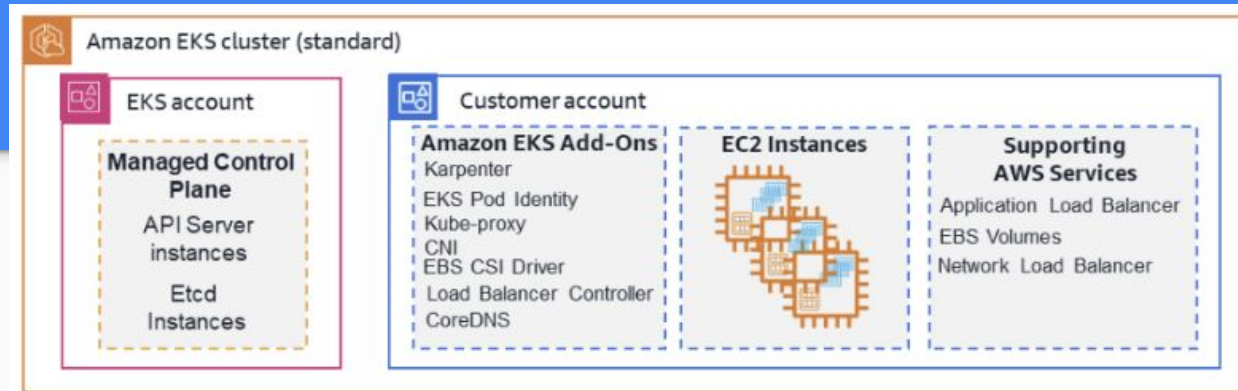
- Without setting up Control Plane
- Save time and effort
- More expensive than Kubernetes



Amazon EKS

EKS Architecture

- EKS Standard
- EKS Auto



EKS Vs Elastic Beanstalk

	EKS	Elastic Beanstalk
Ease of use	More complex, and requires Kubernetes knowledge.	Easy to get started for developers who do not want to manage infrastructure. Minimal setup.
Use case	Scalable systems, microservices, containers.	Simple web apps, REST APIs
Scaling	Highly customizable and powerful.	Automatic but basic.
Control	Full control over infrastructure and how apps are deployed and run.	Limited control over the underlying infrastructure (AWS handles almost everything).
Technology	Runs containerized apps with Kubernetes	Works with EC2/ELB, no containers required.

Our experimentation with EKS - Node View

[Amazon Elastic Kubernetes Service](#) > [Clusters](#) > [interesting-bluegrass-dinosaur](#) > [research-nodegroup](#)

Amazon Elastic Kubernetes Service

Clusters

Settings

Instance settings

Amazon EKS Anywhere

Enterprise Subscriptions

Related services

Amazon ECR

Amazon S3 Batch

Documentation

research-nodegroup

Refresh

Edit

Delete

Node group configuration

Kubernetes version

1.32

AMI type

AL2023_x86_64_STANDARD

Status

Active

AMI release version

1.32.3-20250501

Instance types

t3.medium

Disk size

20 GiB

<

Details

Nodes

Health issues 0

Kubernetes labels

Update config

Kubernetes taints

Update history >

Nodes (2)

Filter Nodes by property or value

< 1 >

Node name	Instance type	Compute	Managed by	Created	Status
ip-172-31-2-180.ec2.internal	t3.medium	Node group	research-nodegroup	2 hours ago	Ready
ip-172-31-25-158.ec2.internal	t3.medium	Node group	research-nodegroup	2 hours ago	Ready

Our experimentation with EKS - Pods

The screenshot displays the Amazon EKS console interface. The breadcrumb navigation at the top reads: Amazon Elastic Kubernetes Service > Clusters > interesting-bluegrass-dinosaur > ccbda-web-app. The left-hand navigation pane includes sections for Amazon Elastic Kubernetes Service, Clusters, Settings (with a link to Console settings), Amazon EKS Anywhere (with a link to Enterprise Subscriptions), and Related services (with links to Amazon ECR and AWS Batch). A Documentation link is also present at the bottom of the sidebar.

The main content area is titled 'ccbda-web-app' and features two tabs: 'Structured view' (selected) and 'Raw view'. The 'Details' section is divided into three columns:

- Status:** 2 Desired | 2 Available | 2 Ready | 0 Pending
- Created:** an hour ago
- Namespace:** ccbda

Below the status column, the **Selector** is shown as 'app=ccbda-web-app'. The **Last transition time** is 'an hour ago'. The **Images** section lists the container image: '675835039352.dkr.ecr.us-east-1.amazonaws.com/django-webapp-docker-repo:v1.0.1'.

The **Strategy type** is 'RollingUpdate'.

The **Pods (2)** section includes an 'Info' link and a table listing the pods. The table has columns for Name, Status, Created, and IP. Both pods are in a 'Running' state and were created 'an hour ago'.

Name	Status	Created	IP
ccbda-web-app-769c7c7b4-qfzhp	Running	an hour ago	172.31.19.146
ccbda-web-app-769c7c7b4-z52dp	Running	an hour ago	172.31.13.13

Our main Sources

<https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html>

<https://kubernetes.io/docs/concepts/architecture/>

<https://kubernetes.io/docs/tutorials/kubernetes-basics/>