

EKS { A Cloud Guru }

133 → New

Jali

(1) Module #1 { EKS Fundamentals }

↳ Microservices :

= "Many modern application made up of bunch of independent microservices running in containers" =

i.e

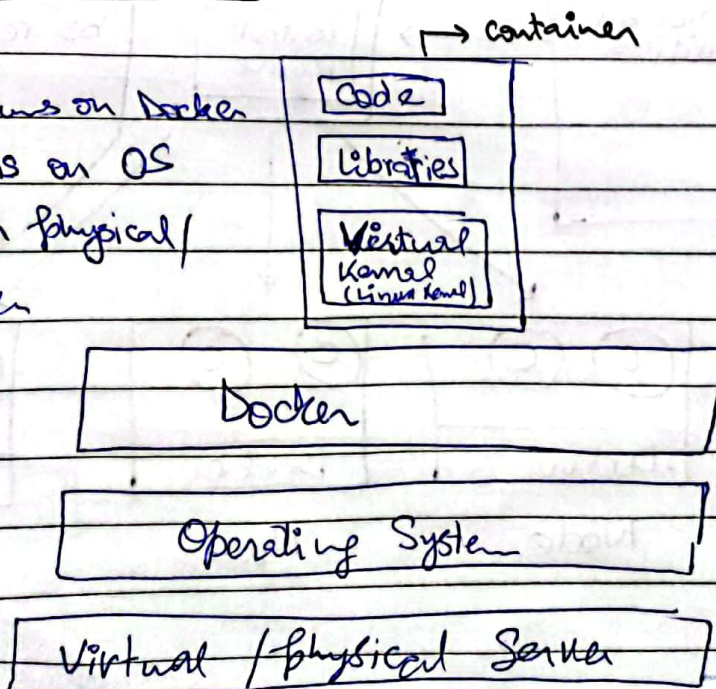
Ecommerce website

- → Shopping Cart
- → Recommendation Engine
- → Product Review
- → Marketing preferences

All of these services can be a separate microservice running in different containers. So you have separate modules component instead of a big monolithic website.

↳ Container Architecture

- Container runs on Docker
- Docker runs on OS
- OS runs on physical / Virtual Server



② Kubernetes: Container Orchestration tool. ②
↳ Deploy, manage & maintain containers using single central control plane.

• Highly Available: Restart/replace failed containers.

• Deployment & Updates:

Deploy hundred or 1000's of containers with single command

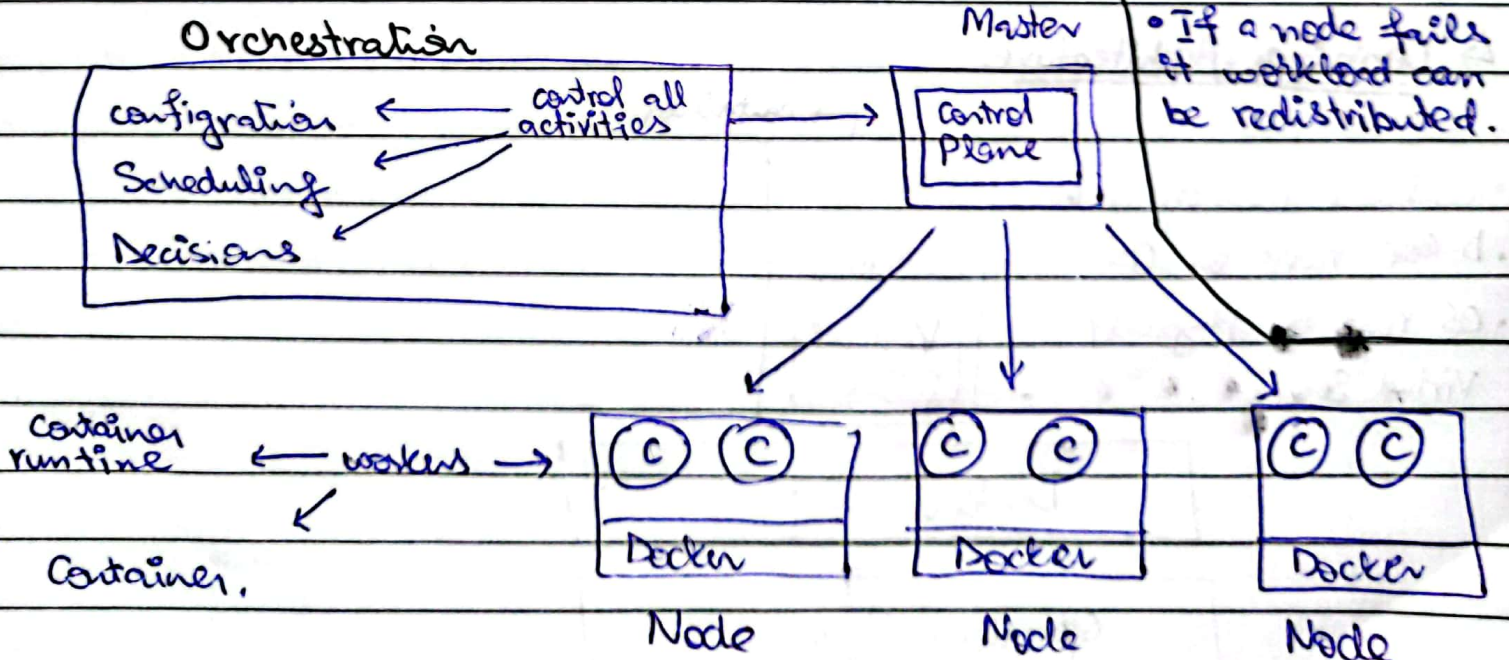
• Scalability:

Automatically scale up and down based on load balancing service discovery.

Dealing with failure

- Control plane monitor state, performance & health.
- Failed containers are replaced.

Architecture of Kubernetes



③ EKS 101 (Elastic Container Service for Kubernetes) Date:

① Features

i) Certified Kubernetes Conformant (compatible with all k8s plugins, No code modifications required)

ii) Provision & Scaling Cluster:

Across multiple AZ's for

high availability

iii) Self-Healing: Automatically detects, replace unhealthy worker nodes. Patching for the control plane

② Kubernetes Terminology

deployment → How to create instances of your container. control plane will schedule, monitor & restart instances.

image → The docker container image that your deployment will use.

Service → Expose the service externally (using an elastic load balancer)

Store your container image in a repo

Elastic Container Registry
Artifactory
Docker Hub

Pod → Grp of one or more container.

Pod Node → Pod runs on node. Shared IP space.

(III) Ectl:

Simple tool to create, delete and get information about our cluster.

(IV) Kubectl

- The K8s command line utility.
- Use to control Kubernetes cluster.

(4) Advantages of EKS

- It is deeply integrated with many of the features of AWS.

(5) LAB Overview

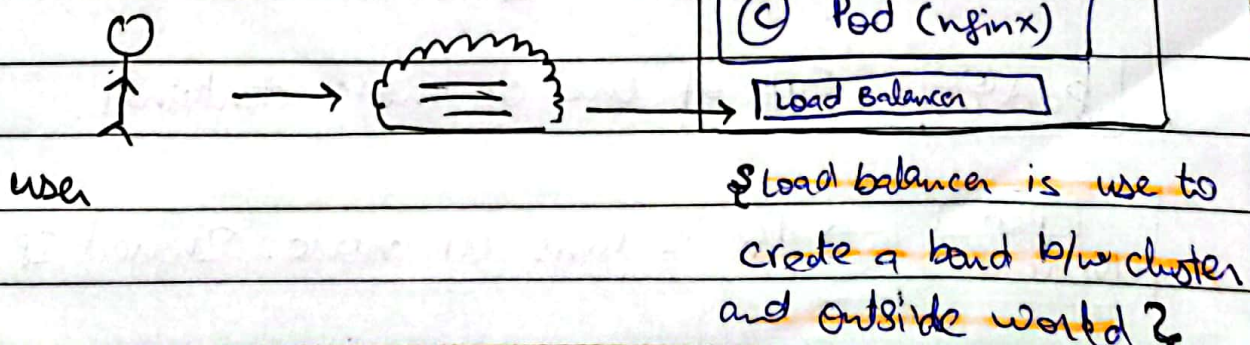
config CLI with
Kubernetes admin user.

i) create IAM (K8 admin)

ii) EC2/Local/VM

- Install AWS CLI (V2)
- Install eksctl
- Install kubectl

will create EKS cluster
control K8s



\$load balancer is use to
create a band b/w cluster
and outside world 2

⑤ Prerequisites for EKS

i) Aws console

(ii) EC2 / VM & Acts as admin workstation

Project

i) Iam (Create User)

- Create user → Programmatic Access
- Attach policy → (Administration Access)
- Create user

(ii) Create EC2 Instance

- Goto Ec2 → (Amazon Linux AMI) → Networking → Auto assign IP (enable)
- Update version of AWS CLI → 2
 - * Download AWS CLI V2 (curl) → unzip
 - * Install AWS CLI V2 (using aws documentation)
 - * Configure AWS
 - * Install kubectl (Via Aws doc) → `chmod +x ./kubectl` → `mkdir -p $HOME/bin`
 - * Install eksctl (Via Aws doc)

(iii) Create Cluster (using eks)

```
# eksctl create cluster --name dev-cluster --version 1.16
```

```
--region us-east-1 --node-group-name worker --node-type t2.micro
--nodes 2 --nodes-min 1 --nodes-max 3 --managed.
```

// Enable us to connect to cluster.

```
# aws eks update-kubeconfig --name dev-cluster --region us-east-1
```

(iv) Deployment On Eks.

clone/create deployment and svc file

// first create svc

```
# kubectl apply -f ./svc.yaml.
```

```
# kubectl apply -f ./deployment.yaml.
```