#Amazon EKS

1. Amazon EKS :- AWS EKS is a managed kubernete service by AWS that simplifies running kubernetes cluster in the cloud and on-premises

EKS Handles :-

- Kubernets control plane management
- Cluster scaling
- Upgrades and patching
- Security integration

Note :- Use when you need scalable, secure and production-grade kubernetes without managing it yourself.

What is kubernetes :-

An open-source platform for automating the deployment, scaling, and management of containerized applications.

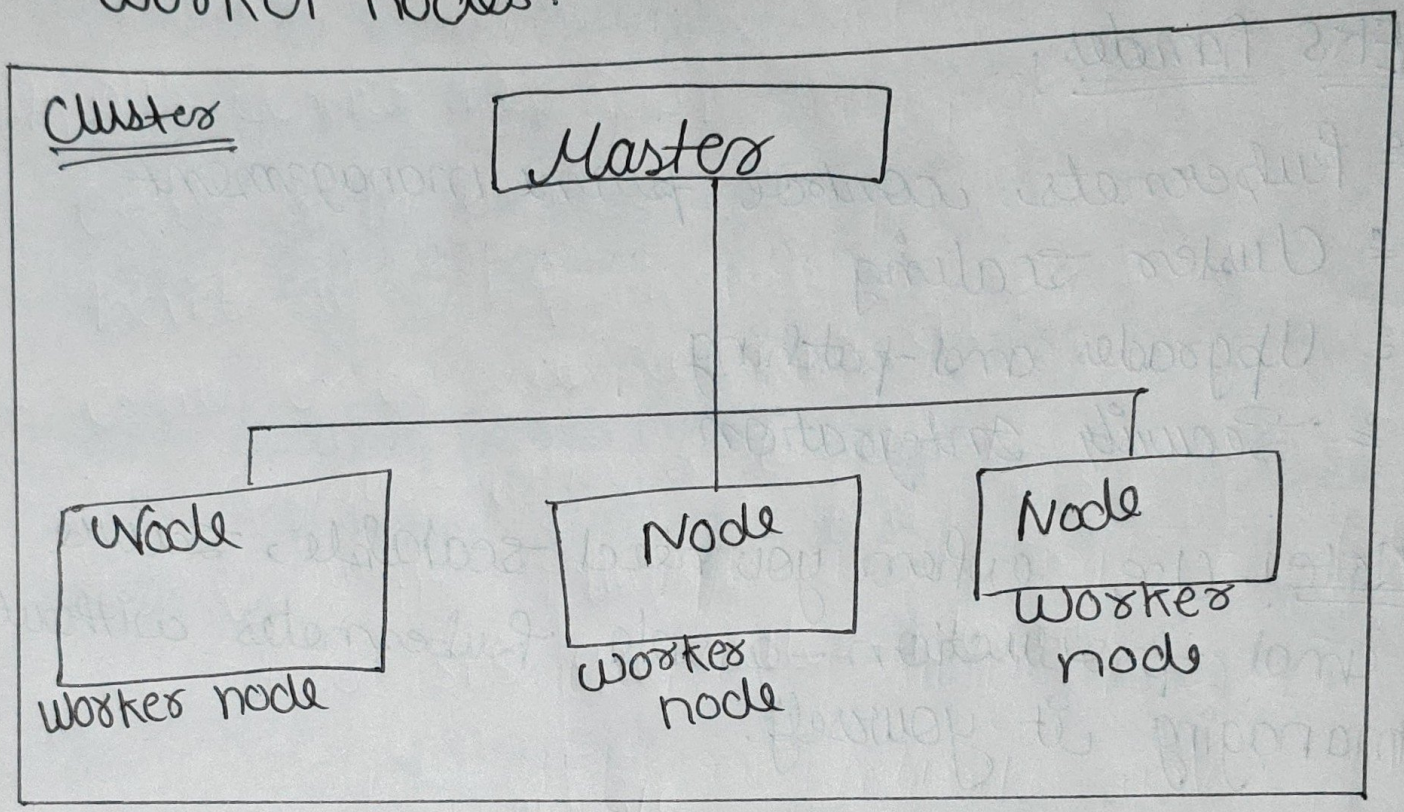
→ Automating Deployment
→ Management of Containerized apps
→ Scaling

Kubernetes

When you deploy Kubernetes, you get a cluster.

Two important parts are:

- Master (control plane) &
- Worker nodes.



Core Components of EKS

(i) Control Plane: Managed by AWS. Includes API Server, etcd, scheduler, and controller manager.

(ii) Worker Nodes: EC2 instances or AWS Fargate to run your containers / pods.

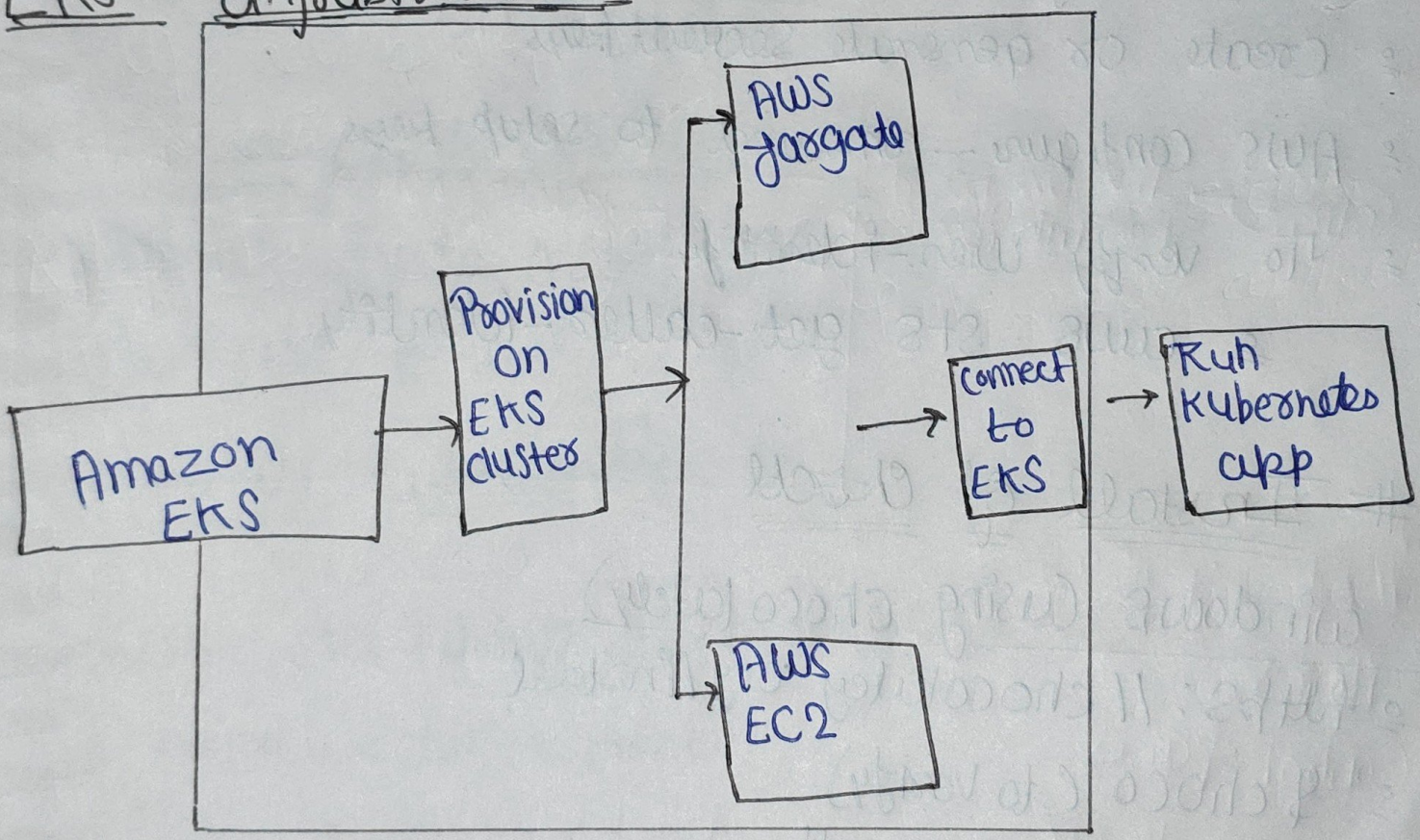
(iii) EKS Cluster: Logical grouping of control planes and nodes.

Node Group :- Group of EC2 worker nodes managed together.

Fargate profiles :- Use Serverless containers AWS provisions infrastructure.

kubeconfig :- Config file for kubectl to access your cluster securely.

Eks infrastructure



Practical

Prerequisites

Before you begin, ensure you have the following prerequisities set up to use Amazon Eks.

Installation of AWS CLI

→ Https://docs.aws.amazon.com/cli/latest/userguide/

- Window - Install using cli
- Mac - Install using pkg
- Verify using : aws --version

AWS Credential Setup

- Create a IAM user
- Provide sufficient permissions.
- Create or generate Secreat keys
- AWS configure - command to setup keys
- To verify user-identify.
  - aws sts get-caller-identity.

Install of eksctl

- windows (using chocolately)
  - https://chocolatey.org/install
  - $ choco (to verify)
  - choco install eksctl
- MAC (using homebrew)
  - https://brew.sh/
  - brew tap weaveworks/tap
  - brew install weaveworks/tap/eksctl

Installation of kubectl

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

- MAC - brew install kubectl
- Windows - choco install kubernetes-cli

EKS setup steps

(1) Create EKS cluster

- Use AWS Console, CLI (eksctl), or Terraform.
- Define VPC, subnets, cluster name, IAM roles.

(2) Launch worker Nodes

- Create node group (EC2) or configure fargate profile.

(3) Update kubeconfig

aws eks update-kubeconfig --region <region> --name <cluster_name>.

(4) Deploy Apps

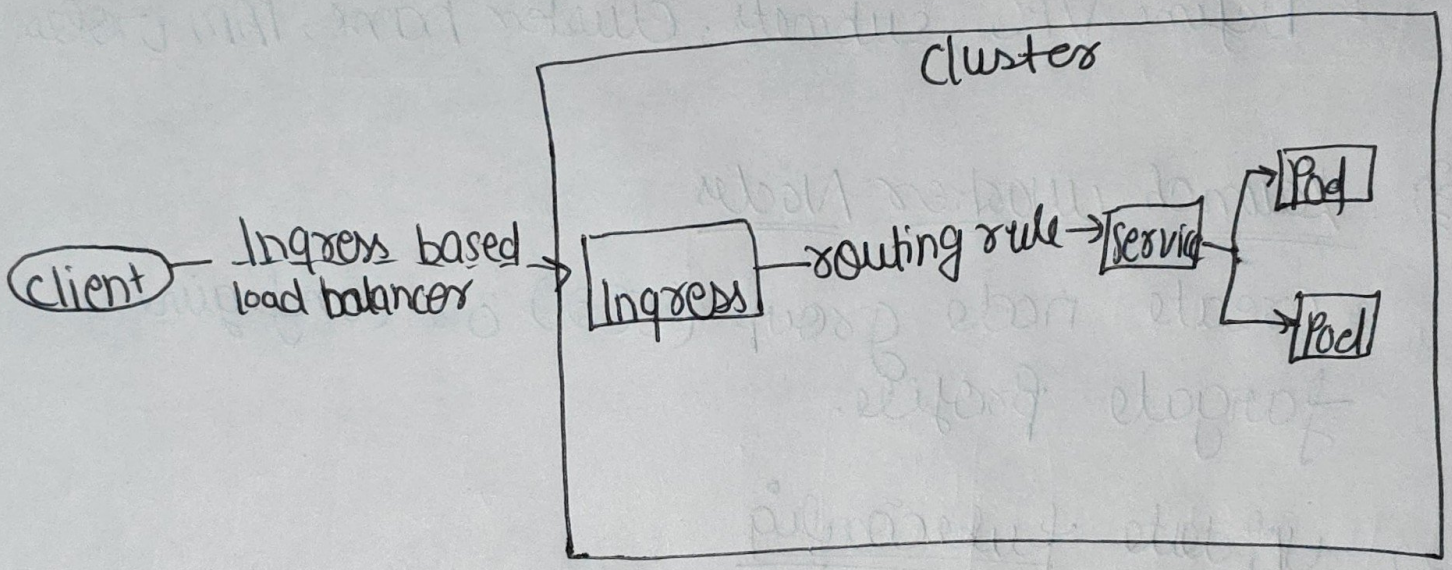
Use standard kubectl commands

kubectl apply -f deployment.yaml.

Ingress :- Ingress exposes Http & Https routes from outside the cluster to services within the cluster.

→ Traffic routing is controlled by rules defined on the Ingress resources.

Here is a simple example where an Ingress sends all its traffic to one Service:



EKS Pricing

items	price
Control Panel	~$74/month per cluster
Worker Nodes (EC2)	Billed seperately based on instance type
Fargates	Charged per VCPU & GB used
Load balancers	Additional cost (ALB/NLB)
Cloudwatch Logs	Cost based on data volume.

Pros & Cons

Pros:

- Fully managed Kubernetes.
- Integrated with AWS ecosystem.
- Secure & Scalable
- Can Use EC2 or fargate.
- Production-grade resilience

Cons :-

- Costly for small apps
- Requires kubernetes experties
- Slower to provision than Lambda / App Runner
- Limited free-tier Usage.

Alternatives to EKS

Use Case	Better Alternative
Simple web app	Aws App Runner
Serverless workloads	Aws Lambda
Basic container hosting	Aws fargate (without EKS)
Cost-efficient VM	Aws Lightsail
High control with K8s	Self managed k8s on EC2