Assessment 2

CRC Application

Prepared By: Yash Gupta
Prepared For: Notes Application

Index

1. Introduction:

CRC application is a backend application that helps to check the records of the courts. This is a python and django frame work application. APIs on this application will be called from the frontend

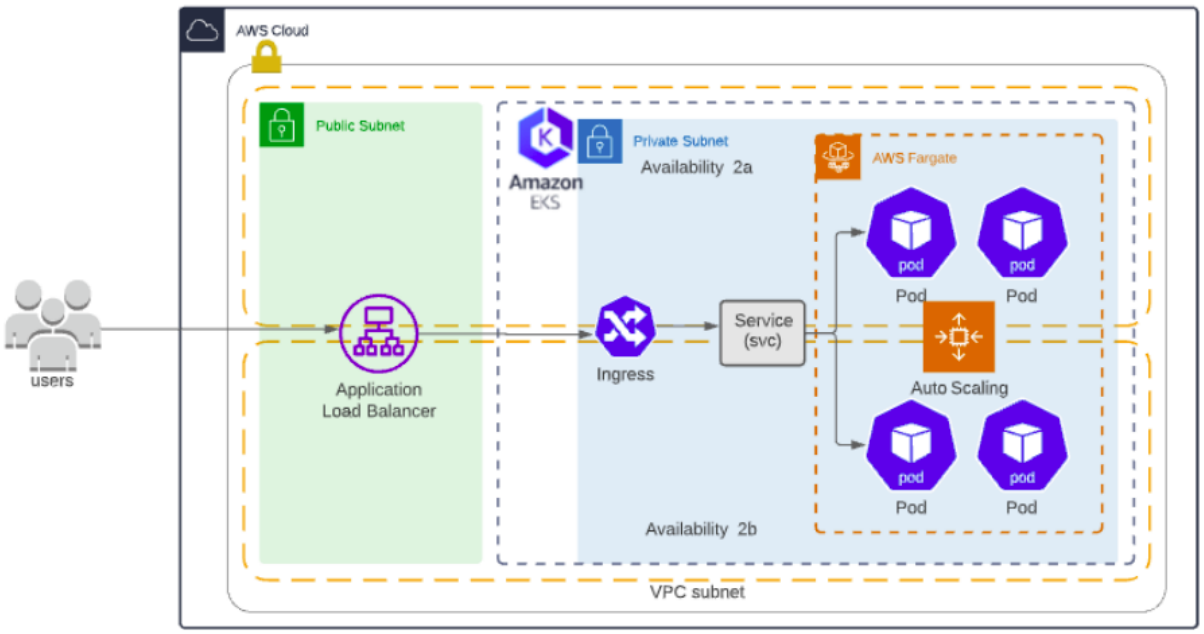
2. Objective:

Since the application is to be hosted on AWS Cloud, the objective of this document is to outline the solution architectures that can be proposed for Notes Application.

3. AWS Solution Architecture:

3.1. Architecture.

3.1.1. Architecture Diagram



3.1.2. Cost Estimation

S.No	Service	Configuration	Cost($)/Month
1	Elastic Cloud Compute(bastion)	t2.micro (gp2 8GB)	9.11
2	Elastic Cloud Compute(nodes)	t2.small(gp2 8GB)	17.43
3	EKS cluster		73.00
TOTAL			**99.54**

Note: Cost will be Increased accordingly in case of any instances launched by the autoscaling group.

3.1.3. AWS Services

a. Elastic Cloud Compute(EC2)

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. It is a scalable, user-configurable compute service that supports multiple methods for encrypting data at rest.

b. Route 53

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service.

c. Elastic Load Balancer(ELB)

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones.

d. EKS Cluster

The Amazon EKS control plane consists of control plane nodes that run the Kubernetes software, such as etcd and the Kubernetes API server. The control plane runs in an account managed by AWS, and the Kubernetes API is exposed via the Amazon EKS endpoint associated with your cluster