Objective

Host notejam application in Kubernetes/Cloud.

Customer/Business Requirements

- Scalability - Business hours & During events & conferences
- Storage to be retained for 3 years
- Application needs to tolerate data centre failures
- Service to be transferred to different regions in case of emergency.
- Separate environments for dev & production
- Availability
- Logs & metrics to be stored

Solution Overview

Suggested cloud provider – AWS

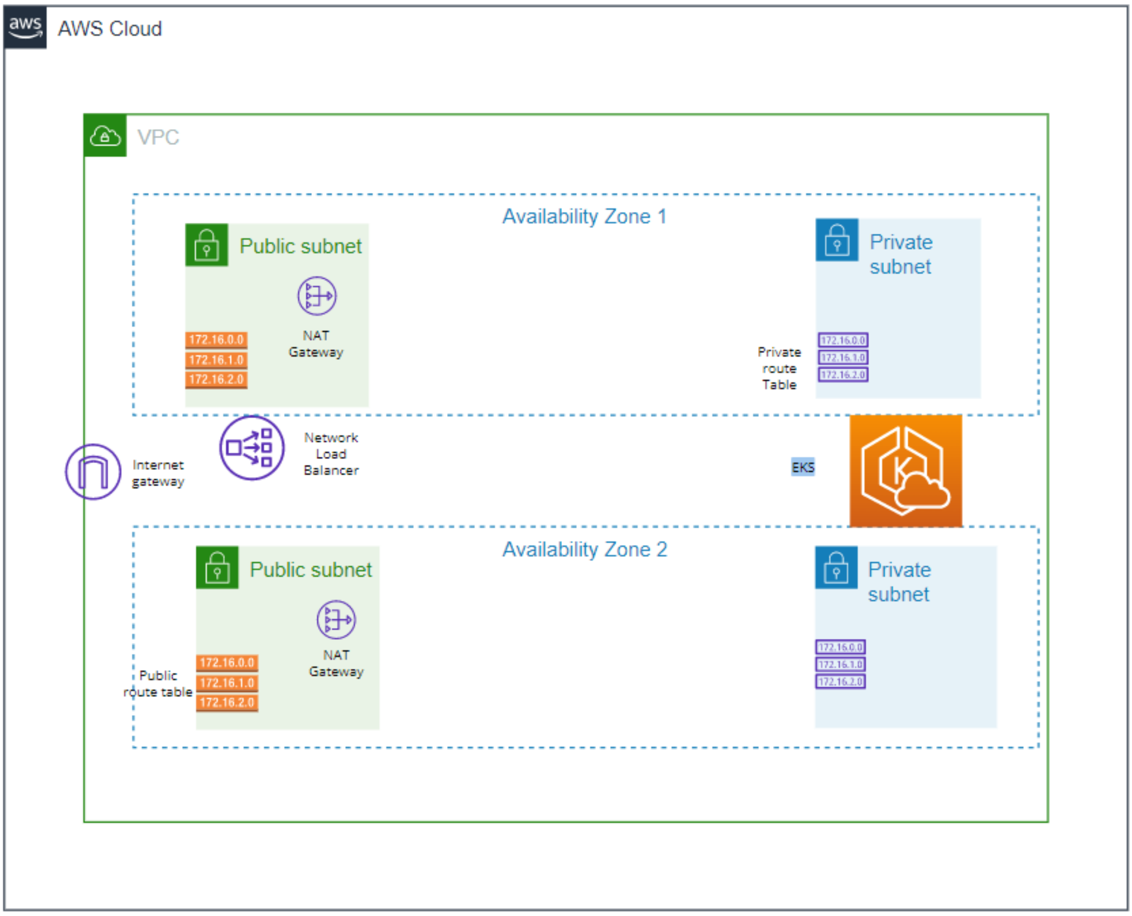
AWS have concept of regions and Availability zones.

AWS region, which is a physical location (Geography) around the world where they group data centers.

Each group of logical data centers - **Availability zone** (AZ)

Each AWS region consists of multiple, isolated, and physically separate AZs within a geographic area.

With AWS, application can span across two Data centers (Availability zone) in same region.



In case of any Data localisation requirements, AWS have regions across EU, Americas, Asia, Africa & Australia.

AWS have managed Kubernetes **EKS** as one of its services.

Most of the administration responsibilities lies with AWS like availability of master nodes in case of DC failure. We can configure for worker nodes by spanning nodes across two Availability zones to tolerate DC failures.

Proposed Kubernetes cluster will span multiple AZs.

Cloud Formation and Terraform are suggested tools and widely used with AWS.

Infrastructure as Code Tool: Terraform preferred

With use of IaC, one can create infrastructure in different region in case of any emergency.

However, this needs to be discussed deeply with team about required features and availability of data etc during this emergency operations.

Steps to create EKS cluster

1. Create VPC

It is private isolated network within a public cloud.

Create subnets within above VPC

Public and private subnets to be created

2. Attach Internet gateway to VPC.

Internet gateway allows communication between VPC and Internet

3. Create Subnets – private and public in 2 Availability zones

Private subnet where EKS nodes are placed don't have direct exposure to internet.

In Public subnet, NAT gateways are placed, communication happens between outside VPC to private subnet.

4. Create NAT gateway and attach it to public subnet.

Assign Elastic Ips to NAT gateway to enable communication with Internet/outside of VPC.

5. Add routing table to both private and public subnet.

Route table of private subnet will have NAT gateway as gateway.

Route table of public subnet will have internet gateway as gateway.

6. Create IAM role & create EKS cluster

IAM role is necessary to access AWS resources, defined role exists and provided by AWS itself

7. Create IAM role & create EKS node group

Node group is to span nodes across Availability zones.

Storage solution

Two types of storage used generally in Kubernetes – Object and block storage.

For the requirement of storing messages for three years, we can use S3 as object storage.

S3-Standard IA (Infrequent access), this will fit for the requirement. But this needs to be discussed before finalising the environment.

Separate environments for Dev &PROD

Either we can opt for separate cluster or create separate namespace for different stages.

To enable multiple users to access EKS cluster,

Appropriate IAM users & roles to be created.

Config map configuration to be updated.

Logging & metrics collections

Ideally ELK with Filebeat/Metric beat can be used for this requirement.

Setup fluentd as daemonset and send logs to CloudWatch.

Autoscaling

Two types of autoscaling can be achieved, node group scaling at Kubernetes worker node levels

POD autoscaling – Vertical & Horizontal scaling. In this git exercise, I have used horizontal auto scaling.

Emergency/DR

Terraform code can be used to create infrastructure in different regions within minutes.

However, availability of data from block storage to be discussed in detail.

Git repo

https://github.com/sabavivek/Kubernetes

This repository has scripts to create AWS EKS cluster.

Additionally, this repo is having EKS deployment files for notejam app.

Notejam app conversion to k8s resources are not done entirely by me,

Referred git: https://github.com/soumya2223/Notejam-Express/tree/master/Kube_File

Not covered in Terraform scripts

Storage configurations – Block and Object

User management

Logging & metrics collection

Advantages

1. DC redundancy achieved by spanning application across two data centers
2. Within few minutes, application can be created in new region in case of emergency.
3. AWS S3 comes with very fair price to meet storage requirements.
4. Different stages can be easily achieved and managed with Kubernetes namespaces.
5. Autoscaling features with AWS as well as Kubernetes in case of varying work loads.
6. AWS in-built solution for logging and metrics collections without much efforts.