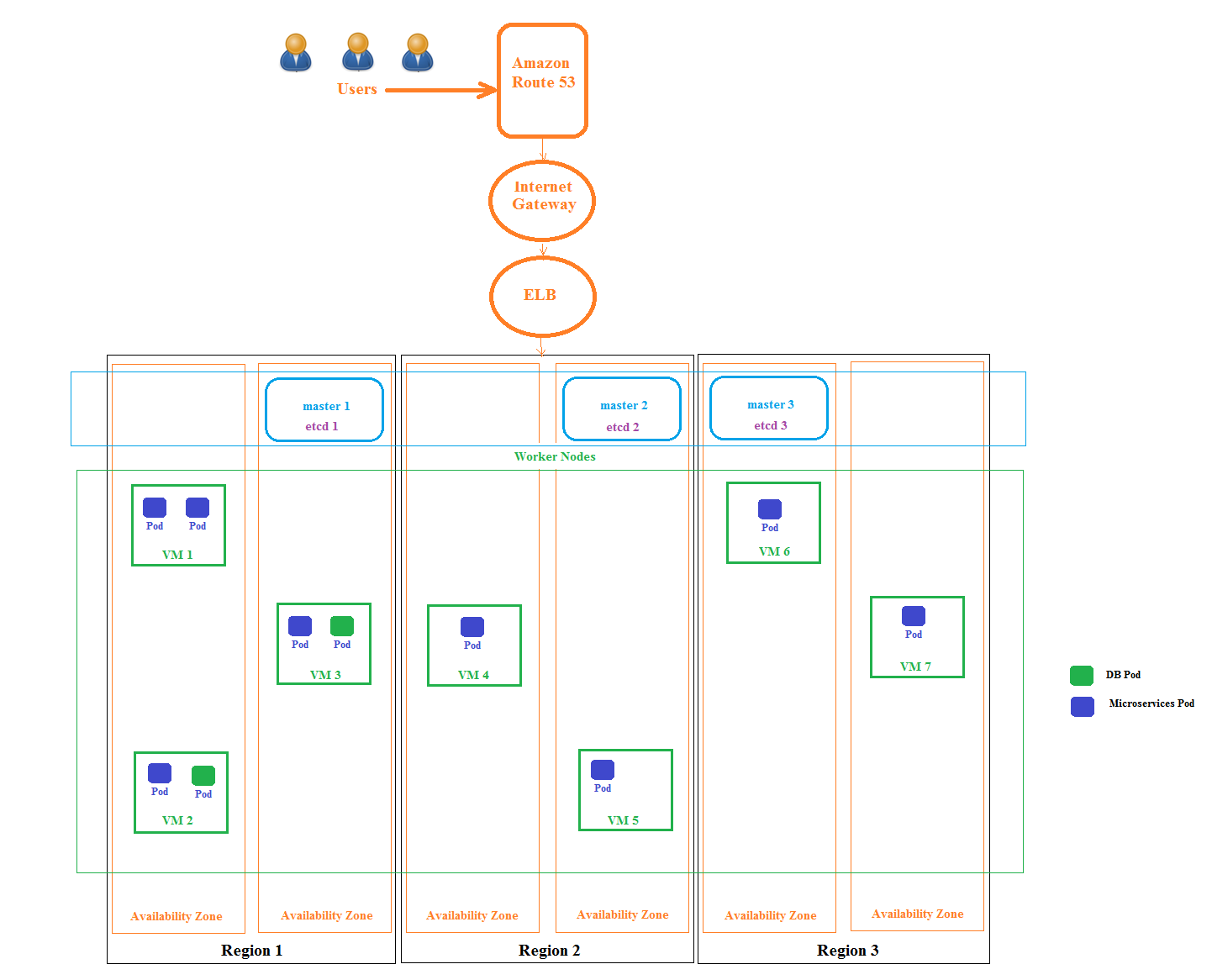
1. Hello-world Microservices application is built using Spring Boot framework
   1. Maven is used as build tool
   2. Junit as unit testing framework
   3. Derby Database is used as a POC, we can consider Mongo DB for production pods.
   4. Dockerfile is created to make docker containers
2. System diagram for AWS deployment



Key design points are listed below

* Route 53 is used as a DNS service
* Elastic Load Balancer (ELB) is used as External Load Balancer for routing TCP and HTTP traffic, SSL termination would happen at this level.
* Kubernetes cluster is created using EC2 compute instances.
  + Three Different Regions and six availability zones are used to ensure high availability and fault tolerance of instances
  + Pods (having Microservices and Db pod) comes with in-built auto heal, auto scale, highly available features from Kubernetes orchestrator.

1. CI/CD :
   1. GiHub / Gitlab can be used as source code repository tool
   2. Jenkins can be used as Continuous Integration Tool
   3. Using Docker hub as image registry, we can consider using Amazon Elastic Container Registry for private repository

Jenkins Configuration: For Rolling Update

1. Clone source code from github repository.

git clone <repo.git>

2. Inject environment var for kubernetes cluster details.

KUBECONFIG=/home/admin/k8s/admin.conf

3. Execute shell

IMAGE\_NAME="danishsh/home-task:latest"  
docker build . -t $IMAGE\_NAME  
docker login -u danishsh -p ${DOCKER\_HUB}  
docker push $IMAGE\_NAME  
  
kubectl apply -f ${WORKSPACE}/home-task-deploy.yaml

1. Content of home-task-deploy.yaml as attachment below

kubectl create –f home-task-deploy.yaml



We need to create Services in kubernetes cluster using below yaml file.

kubectl create –f home-task-web.yaml

