**AWS EKS**

The terraform scripts on this folder creates the following

1. An Elastic Kubernetes Cluster using terraform aws\_eks module
2. A VPC in the eu-west-1 region with 3 private(security) and 3 public subnets( for public facing apps)
3. 3 Autoscaling groups for the Kubernetes worker nodes (2 On Demand and 1 Spot On)
4. A cluster autoscaler was deployed on the cluster by attaching relevant policy to the several nodegroups role, attaching relevant tags to the nodegroups, and by deploying the necessary Kubernetes objects to the eks cluster. Documentaion on this can be found on <https://docs.aws.amazon.com/eks/latest/userguide/cluster-autoscaler.html>.
5. A bastian host used to connect to the worker nodes on port 22and also to communicate on to the API server only on port 443

**Horizontal Pod AutoScaler**

To create a horizontal pod auto scaler a metricsets pod need to be deployed on the cluster, this was achieved on this cluster by running the following commands

git clone <https://github.com/kubernetes-sigs/metrics-server.git>

cd metrics-server

kubectl apply -f deploy/1.8+/

**INFRA**

1. A deployment manifest was created for this deployment, the manifest file can be found on this repo with the name deployment.yaml
2. The service definition for this deployment can be found on this deployment.yaml
3. To implement autoscaling this was done by running this command “ kubectl autoscale deploy dubizzle –min=4 –-max=10 –cpu-percent=50”
4. The Kubernetes objects for this deployment was helm packaged to the helm folder of this repo

**Development**

1. A flask python library was used to create an api with the specification given
2. The app was dockerized by creating a Dockerfile(found in this repo with name Dockerfile) and running “docker build . -t tocy/dubizzle-sre-tochukwu.nwoko:1.1”
3. This was then pushed to my docker hub account by running “docker login” and “docker push tocy/dubizzle-sre-tochukwu.nwoko:1.1””

NB This command only runs if u have docker installed find installation steps on <https://docs.docker.com/install/linux/docker-ce/centos/>

Find the link to the docker repo

<https://hub.docker.com/repository/docker/tocy/dubizzle-sre-tochukwu.nwoko>

**Logging and Alerting**

To set up logging, cloudwatch container insights was deployed on this cluser

Find documentation from <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Container-Insights-prerequisites.html>

**NB:** This is not yet working in this cluster as I suspect it’s a permission issue

To set up alerting on this cluster terraform aws\_cloud\_watch\_metric\_alarm module was utilized find more on <https://www.terraform.io/docs/providers/aws/r/cloudwatch_metric_alarm.html>

The terraform scripts achieve this can be found on the file “alerts.tf” on this repo

**NB:** To run this terraform script please find the terraform installation process for respective platforms on <https://learn.hashicorp.com/terraform/getting-started/install.html>

Install git also following this

[https://www.linode.com/docs/development/version-control/how-to-install-git-on-linux-mac-and- windows/](https://www.linode.com/docs/development/version-control/how-to-install-git-on-linux-mac-and-%20%20%20windows/)

clone this repo by running

git clone <repo.git>

cd /<repo>

terraform init

terraform plan

terraform apply