**Question and Answer**

**Question 1**. In a few words, describe how you provision cloud infrastructure from scratch to host this application with High Availability in mind. (Output should not be code, but a description in your own words). Some topics you might want to include:

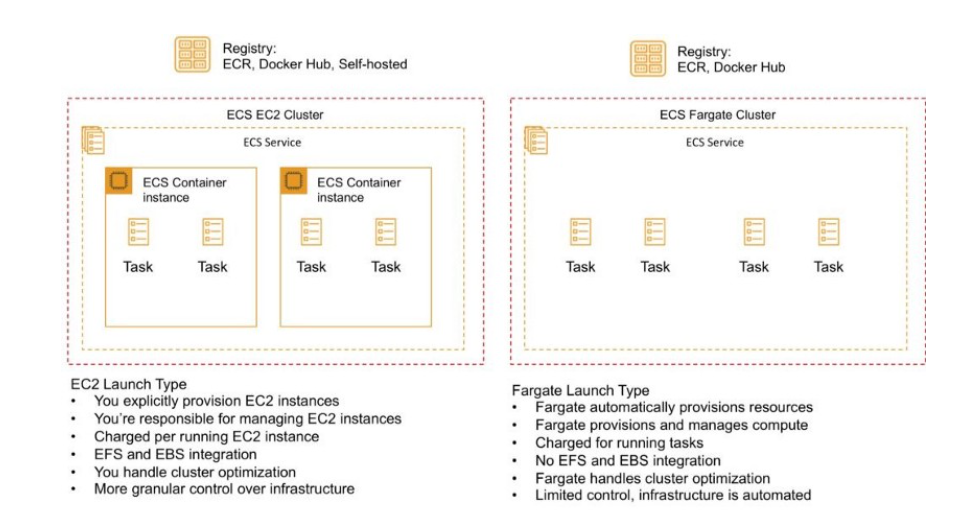
**ANS**

1. Cloud: AWS cloud

2. Infrastructure design and components: VPC, IAM Role and Policy, EKS and NodeGroup

3. Automation and deployment tools: Terraform

We can create aws eks cluster with 3 master and 3 nodes to keep High Availability in mind. Also if we will use Farget Profile instead of Node Group that will be best.



Steps:

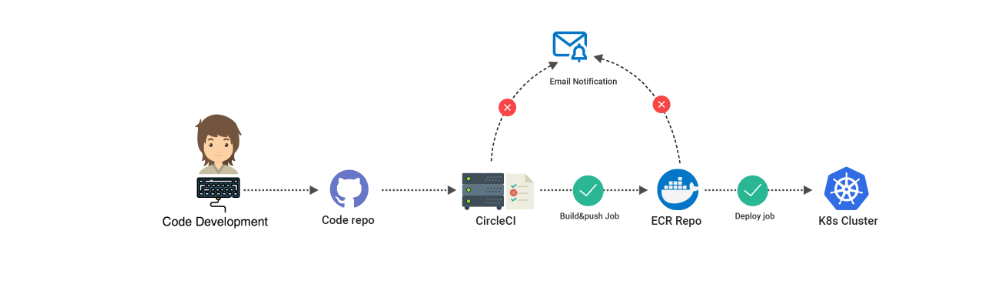
* Create aws VPC
* Create IAM Role and policy for nodegroup
* Create EKS cluster
* Create NodeGroup
* Attach role and policy to Nodegroup
* Attach node Group with EKS cluster

**Question 2.** In a few words, describe how you would create a continuous delivery pipeline to automatically build and deploy this application to dev –> int –> uat –> prod environments with every Git push. (Output should not be code, but a description in your own words). Some topics you might want to include:

**ANS**

We can create the pipeline for a flask application. The pipeline will first build the binary, create a docker image from it, push the image to ECR, then deploy it on the Kubernetes cluster using its helm chart. We can automate this using circle ci

We need to create a folder named .circleci in the root folder of our repository and create a file named config.yml in it. In our config.yml we have defined two jobs one is build&pushImage and deploy.



1. CI/CD: CircleCI and Helm Chart

2. Procedural build steps

* Set the tag for the image, we will concatenate the app version and circle build number with a `-` char in between
* Build the docker image
* Install AWS cli
* Login to ECR
* Tag the image with ECR repo name
* Push the image the ECR repo

3.Procedural deployment steps

* Install AWS cli
* Set the tag for the image, we will concatenate the app version and circle build number with a `-` char in between
* Install and configure kubectl
* Install and configure kubectl aws-iam-authenticator
* Install latest awscli version
* Get the kubeconfig file
* Install and configure helm
* Initialize helm
* Release flask app using helm chart