**Project Plan**

**Tools, Technologies, Services and Resources to be used for this project.**

GitHub Repository : <https://github.com/viveknambi/my_project.git>

Location where developers push their code after a build validation and testing using automated tools such as Maven, ant etc..

Terraform: Configure and manage infrastructure as code for initial setup.

Create a terraform file locally to build an EC2 instance in AWS with a centOS7 ami-image with following tools installed and configured. Execute this and keep the instance ready.

1. Git
2. Ansible
3. Docker and associated tools
4. Terraform
5. Jenkins

VPC in AWS Cloud

Ec2 Server1

AWS EKS (Elastic Kubernetes service)

DB Storage Volume

CentOS7

Jenkins

Git

Ansible

Docker

Docker-compose

Terraform

Developer Front

Freestyle job - Terraform

WEB

Pipeline Job for deploying app

API

Git-Hub repo

LB1

DB

LB2

User Front

Terraform

Create a terraform code to build the infrastructure where application has to be deployed.

Infrastructure: - Create a VPC with all relevant networking resources and an (EKS)Elastic Kubernetes Service with 3 nodes for the deployment of application. Commit it to local git.

Jenkins: Configure Jenkins to create CI/CD Pipeline

1. Jenkins listens for any update in git-hub repository using SCM poll.
2. Create a freestyle job which uses the later terraform file to create the main infrastructure and commit it to local git. This job will be run only once unless there is modification in infrastructure.
3. Create a pipeline job to use a docker compose file in the git-hub repo which in turn uses Docker file of the respective tier to build the docker images of web, api and db and stores it locally (first ec2 instance).
4. Another linked job in pipeline to use an ansible playbook to push these 3 images to docker-hub, preferably on-premise one, else Private/public.
5. Linked job in pipeline to run an ansible playbook in AWS EKS cluster nodes to deploy the 3-tier application with AWS load balancers. This ensures scalability, high availability and resilience.
6. Once all these codes are written, push the code to git-hub repository. Jenkins service will initiate the job as configured.

Note:- This is the overall envisaged plan. Since many of the AWS resources, essential to implement this are chargeable, implementing this is not feasible. Hence as a temporary solution I have a terraform code to launch a single EC2 instance in AWS. Once the instance is launched, we can deploy application using Jenkins or directly running docker-compose file. Details are mentioned below.

**Project Temporary**

VPC with EC2 instance in AWS

Ec2 Instance – server1 - ubuntu

Terraform – infra\_1.tf

Jenkins

Docker

Docker-compose

Git

Local repo

web

Git-hub

api

db

http://<AWS\_public\_IP>:3000/ to access the web service

Clone the git repo given at top of the document .to local folder

Running the Terraform code file **infra\_1.tf** from local folder will create a VPC in AWS with an EC2 instance having public IP with all Network resources and relevant tools.

Login to Newly created EC2 instance and clone the git repo and run

docker-compose.yml file manually. This will deploy the application.

Or

Login to newly created Ec2 instance and configure Jenkins. It will run the deployment docker compose file as a freestyle job and enabling SCM with git repo will do an auto update.

Go to http://<AWS\_publica\_IP>:3000/ to access the web service.