Documentation:

1. Git clone repo (backstage)
2. Installed NPM and node
3. Verified node and npm versions for installing yarn
4. Yarn installed
5. Yarn tsc
6. Yarn build

Now, created a dockerfile and .dockerignore files

docker image build . -f packages/backend/Dockerfile --tag backstage-image:latest

**Now, we will use this Image in our terraform scripts to launch a pod of backstage.**

**GKE cluster creation using terraform:**

1. Created a account on google cloud and get the access keys.
2. Wrote down terraform scripts for launching a 3 node GKE cluster in Europe-west1 region.

Files: main.tf, variables.tf and output.tf

Used provider “Google” and module gke\_auth and gke\_network for connecting through terraform.

Terraform init, terraform validate, terraform plan, terraform apply. And GKE cluster created in **project id: nkg-backstage**

**Deploying backstage using terraform on GKE cluster:**

1. Used provider “kubernetes” for deploying Kubernetes resources on GKE through terraform.
2. For connecting with GKE cluster kube, need to add the config\_file path withing Kubernetes block.
3. Now, created persistent volume, persistent volume claim terraform file and persistent disk to be used in persistent volume.
4. Created a persistent disk for postgresql database and used gce\_persistent\_disk resource for this.
5. Created a namespace.tf file for creating a namespace named as backstage
6. Created separate secret files for db and app deployments and credentials added there and passed this secret\_ref in env\_from section of postgres volume.
7. For backstage deployment, image reference is used with previously created image.
8. Created secret file , deployment with replica 1 and it will use postgre db pod for connection.
9. Created two service file for eachpostgresql and backstage.

After adding all these .tf file

Terraform plan and then terraform apply.

Finally, after few challenges, pods got successfully running.

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a computer

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Challenges faced and there solution:**

1. As it was to be deployed on GCP and GKE cluster, it took time for creating an account on GCP and using GKE\_Auth module in terraform to be successfully connected through GCP cloud resources.
2. Launched a GKE cluster from scratch using terraform with subnet ranges and module GKE\_network.
3. Getting issues in postgreql pod to be running as it was using persistent volume and persistent volume claim, there was must to use gce\_persistent\_disk to be created . after this, PVC came in Bound state from previously pending state.
4. Now, the biggest challenge was with , making this pod successful, but it was not coming up . needed to add the volume mounts and then needed to create sub\_path for volume with new directory created.
5. Finally, Postgresql DB pod came up.
6. My docker desktop, on windows got blocked due to new policies by docker inc. and that’s why I was not able to build image locally.
7. There was sufficient information was available on backstage site, but deploying on GKE was the different thing totally , rather than deploying on any in-houce Kubernetes.

**Total time to be taken: more than 4 hours** as it required to create a GKE cluster from scratch using terraform and then all Kubernetes resources for backstage app. Solving errors of persisten disk in terraform was the biggest challenge faced.

References:

1. <https://backstage.io/docs/deployment/>
2. <https://kubernetes.io/docs/concepts/storage/volumes/#gcepersistentdisk>
3. <https://registry.terraform.io/providers/hashicorp/google/latest/docs>
4. <https://registry.terraform.io/providers/hashicorp/google/latest/docs/guides/using_gke_with_terraform#interacting-with-kubernetes>
5. <https://nodejs.org/en/download/>
6. <https://cloud.google.com/sdk/docs/install>
7. <https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs>
8. <https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs/resources/persistent_volume_v1#mount_options>