GSP321

Task - 1 : Create development VPC manually

1. gcloud compute networks create griffin-dev-vpc --subnet-mode custom

2. gcloud compute networks subnets create griffin-dev-wp --network=griffin-dev-vpc --region us-east1 --range=192.168.16.0/20

3. gcloud compute networks subnets create griffin-dev-mgmt --network=griffin-dev-vpc --region us-east1 --range=192.168.32.0/20

Task - 2 : Create production VPC manually

1. gsutil cp -r gs://cloud-training/gsp321/dm .

2. cd dm

   sed -i s/SET_REGION/us-east1/g prod-network.yaml

3. gcloud deployment-manager deployments create prod-network \
   --config=prod-network.yaml

   cd ..

Task - 3 : Create bastion host

1. gcloud compute instances create bastion
   --network-interface=network=griffin-dev-vpc,subnet=griffin-dev-mgmt
   --network-interface=network=griffin-prod-vpc,subnet=griffin-prod-mgmt --tags=ssh
   --zone=us-east1-b

   gcloud compute firewall-rules create fw-ssh-dev --source-ranges=0.0.0.0/0 --target-tags ssh
   --allow=tcp:22 --network=griffin-dev-vpc

   gcloud compute firewall-rules create fw-ssh-prod --source-ranges=0.0.0.0/0 --target-tags ssh
   --allow=tcp:22 --network=griffin-prod-vpc

Task - 4 : Create and configure Cloud SQL Instance

1. gcloud sql instances create griffin-dev-db --root-password password --region=us-east1

2. gcloud sql connect griffin-dev-db
3. password

4. CREATE DATABASE wordpress;
   GRANT ALL PRIVILEGES ON wordpress.* TO "wp_user"@"%" IDENTIFIED BY
   "stormwind_rules";
   FLUSH PRIVILEGES;

5. exit

Task - 5 : Create Kubernetes cluster

1. gcloud container clusters create griffin-dev \
   --network griffin-dev-vpc \
   --subnetwork griffin-dev-wp \
   --machine-type n1-standard-4 \
   --num-nodes 2 \
   --zone us-east1-b

2. gcloud container clusters get-credentials griffin-dev --zone us-east1-b

3. cd ~/

4. gsutil cp -r gs://cloud-training/gsp321/wp-k8s .

Task - 6 : Prepare the Kubernetes cluster

- **Open Editor -> wp-k8s -> wp-env.yaml Change username and password to:**

  username : wp_user
  password : stormwind_rules

- **SAVE**
  1. cd wp-k8s

  2. kubectl create -f wp-env.yaml

  3. gcloud iam service-accounts keys create key.json \

     --iam-account=cloud-sql-proxy@$GOOGLE_CLOUD_PROJECT.iam.gserviceaccount.com

  4. kubectl create secret generic cloudsql-instance-credentials \

     --from-file key.json

Task - 7 : Create a WordPress deployment

- **In editor: wp-deployment.yaml -> replace YOUR_SQL_INSTANCE with griffin-dev-db.**
- **Save.**

```
kubectl create -f wp-deployment.yaml
kubectl create -f wp-service.yaml
```

Task - 8 : Enable monitoring

- **Navigation Menu -> Kubernetes Engine -> Services and Ingress -> Copy Endpoint's address.**
- **Navigation Menu -> Monitoring -> Uptime Checks -> + CREATE UPTIME CHECK**

Title : Wordpress Uptime

- **Next -> Target**

Hostname : {Endpoint's address} (without http...)
Path : /

- **Next -> Next -> Create**
-

Task - 9 : Provide access for an additional engineer

- **Navigation Menu -> IAM & Admin -> IAM -> ADD**

New Member : {Username 2 from Lab instruction page}
Role : Project -> Editor

Save.