

Develop your Google Cloud Network

```
export REGION=[your_lab_region]
export ZONE=[your_lab_zone]
export ADDITIONAL_ENGINEER_EMAIL=[your_lab_username2]
```

Create development VPC

```
gcloud compute networks create griffin-dev-vpc --subnet-mode=custom
```

```
gcloud compute networks subnets create griffin-dev-wp \
  --network=griffin-dev-vpc \
  --range=192.168.16.0/20 \
  --region=$REGION
```

```
gcloud compute networks subnets create griffin-dev-mgmt \
  --network=griffin-dev-vpc \
  --range=192.168.32.0/20 \
  --region=$REGION
```

Create production VPC

```
gcloud compute networks create griffin-prod-vpc --subnet-mode=custom
```

```
gcloud compute networks subnets create griffin-prod-wp \
  --network=griffin-prod-vpc \
  --range=192.168.48.0/20 \
  --region=$REGION
```

```
gcloud compute networks subnets create griffin-prod-mgmt \
  --network=griffin-prod-vpc \
  --range=192.168.64.0/20 \
  --region=$REGION
```

Create a bastion host

```
gcloud compute instances create griffin-bastion \
--machine-type=e2-medium \
--zone=$ZONE \
--tags=bastion \
--network-interface=subnet=griffin-dev-mgmt \
--network-interface=subnet=griffin-prod-mgmt \
--metadata=startup-script='#!/bin/bash
sudo apt-get update
sudo apt-get install -yq git http
'\
--scopes=cloud-platform \
--image-family=debian-10 \
--image-project=debian-cloud
```

Firewall rules allowing TCP traffic on port 22:

```
gcloud compute firewall-rules create griffin-dev-allow-ssh \
--network=griffin-dev-vpc \
--allow=tcp:22 \
--source-ranges=0.0.0.0/0 \
--target-tags=bastion \
--description="Allow SSH access to bastion host"
```

```
gcloud compute firewall-rules create griffin-prod-allow-ssh \
--network=griffin-prod-vpc \
--allow=tcp:22 \
--source-ranges=0.0.0.0/0 \
--target-tags=bastion \
--description="Allow SSH access to bastion host in production"
```

Create and Configure Cloud SQL Instance

```
gcloud sql instances create griffin-dev-db \  
  --database-version=MYSQL_5_7 \  
  --tier=db-n1-standard-1 \  
  --region=$REGION
```

```
gcloud sql databases create wordpress --instance=griffin-dev-db
```

```
gcloud sql users create wp_user --host=% --instance=griffin-dev-db --password=password123
```

```
gcloud sql connect griffin-dev-db --user=root << EOF  
CREATE DATABASE wordpress;  
CREATE USER 'wp_user'@'%' IDENTIFIED BY 'stormwind_rules';  
GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'%';  
FLUSH PRIVILEGES;  
EOF
```

Create Kubernetes cluster

```
gcloud container clusters create griffin-dev \  
  --zone=$ZONE \  
  --num-nodes=2 \  
  --machine-type=e2-standard-4 \  
  --network=griffin-dev-vpc \  
  --subnetwork=griffin-dev-wp
```

Prepare the Kubernetes cluster

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

```
cd wp-k8s
```

```
ls
```

```
nano wp-env.yaml
```

Update the values of `username` to `wp_user` and `password` to `stormwind_rules`

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

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

```
kubectl create secret generic cloudsql-instance-credentials \
```

```
--from-file key.json
```

Create WordPress deployment

```
gcloud sql instances describe griffin-dev-db --format='value(connectionName)'
```

```
nano wp-deployment.yaml
```

replace the placeholder `YOUR_SQL_INSTANCE` with the instance connection name

```
cat wp-deployment.yaml
```

```
kubectl apply -f wp-env.yaml
```

```
kubectl apply -f wp-deployment.yaml
```

```
kubectl apply -f wp-service.yaml
```

Enable Monitoring

```
kubectl get services
```

```
export WORDPRESS_SITE_URL=[EXTERNAL_IP]
```

he EXTERNAL-IP column will contain the IP address you use as the WordPress site URL

```
gcloud monitoring uptime create griffin-dev-wp-uptime-check \
  --display-name="Griffin Dev WP Uptime Check" \
  --resource-labels=host=$WORDPRESS_EXTERNAL_IP
```

Provide access for an additional engineer

```
gcloud projects add-iam-policy-binding $GOOGLE_CLOUD_PROJECT \
  --member="user:$ADDITIONAL_ENGINEER_EMAIL" \
  --role="roles/editor"
```