

Instruction for Replication

In this report, we started at creating docker image.

Project ID	socialbook-455415
Repository Name	django-repo
Image Name	django-app
Zone	us-central1

When replicating, please replace these parameters with yours.

YAML files:

Namespace: namespace.yaml

Development: postgres-dev.yaml, django-dev.yaml

Production: django-prod.yaml, secrets.yaml

1. Build Docker Image

```
docker build -t social_book_app .
```

```
docker tag social_book_app us-central1-docker.pkg.dev/socialbook-455415/django-repo/django-app:v1
```

2. GCP Authorization

```
gcloud auth configure-docker us-central1-docker.pkg.dev
```

3. Artifact Registry

```
gcloud artifacts repositories create django-repo --repository-format=docker --location=us-central1 --description="Docker repository for Django applications"
```

```
docker push us-central1-docker.pkg.dev/socialbook-455415/django-repo/django-app:v1
```

4. Create GKE Cluster

```
gcloud container clusters create django-cluster --num-nodes 3 --zone us-east1
```

```
gcloud container clusters get-credentials django-cluster --zone us-east1
```

5. Namespace Isolation

```
kubectl apply -f namespaces.yaml
```

6. Deploy to Development Environment

```
kubectl apply -f postgres-dev.yaml
```

```
kubectl apply -f django-dev.yaml
```

7. Check the Status (Optional)

```
kubectl get pods -n development
```

```
kubectl get services -n development
```

```
C:\Users\buwen\AppData\Local\Google\Cloud SDK>kubectl get pods -n development
NAME          READY   STATUS    RESTARTS   AGE
postgres-pod  1/1     Running   0           5m23s
web-pod       1/1     Running   0           11s

C:\Users\buwen\AppData\Local\Google\Cloud SDK>kubectl get services -n development
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
django-service  LoadBalancer  34.118.233.3     <pending>        80:30446/TCP     28s
postgres       ClusterIP      34.118.237.245   <none>           5432/TCP         4m25s
```

8. Prepare for Production APIs

Create a **Cloud SQL** instance in GCP, modifying the secret values in secrets.yaml with yours.

Create a new user (e.g., yushi) in Cloud SQL and grant all privileges to it:

```
GRANT ALL PRIVILEGES ON DATABASE postgres TO yushi;
```

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO yushi;
```

```
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO yushi;
```

Store the base64 code of username and password in secrets.yaml.

Enable **Cloud Storage** and run:

```
gsutil mb -p socialbook-455415 -l us-central1 gs://socialbook-media-bucket/
```

Enable **IAM & Admin** and create a service account:

```
gcloud iam service-accounts create default --display-name="Django GCS Service Account"
```

```
gcloud projects add-iam-policy-binding socialbook-455415 --
```

```
member=serviceAccount:default@socialbook-455415.iam.gserviceaccount.com --
```

```
role=roles/cloudsql.client
```

Create secret key:

```
gcloud iam service-accounts keys create key.json --iam-account= default@socialbook-455415.iam.gserviceaccount.com
```

Store the base64 code in secrets.yaml (parameter key).

Generate your secret key (Base64 string of length 50) and store it in secrets.yaml (parameter secret-key)

9. Deploy to Production Environment

```
kubectl apply -f secrets.yaml
```

```
kubectl apply -f django-prod.yaml
```

10. Check the Status (Optional)

```
kubectl get pods -n production
```

```
kubectl get services -n production
```

```
C:\Users\buwen\AppData\Local\Google\Cloud SDK>kubectl get pods -n production
NAME                                READY   STATUS    RESTARTS   AGE
django-deployment-5dc4f667b4-9dcg4  2/2     Running   1 (3s ago)  9s

C:\Users\buwen\AppData\Local\Google\Cloud SDK>kubectl get services -n production
NAME            TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
django-service  LoadBalancer  34.118.227.197  35.227.10.198    80:30904/TCP     3h26m
```

(It restarts because I created a new instance for test

and it did not have enough quota to run replications)

Now you shall be able to visit the website through external-IP and try all the features.