

GKE Cluster Oluřturma ve Flask Python Uygulaması Deploy Rehberi

GKE Cluster Oluřturmak İin Gerekli Yetkiler

1. **Kubernetes Engine Admin:** `Kubernetes Engine` cluster'larını ynetme yetkisini saęlar.
2. **Compute Admin:** Compute Engine kaynaklarını ynetme yetkisini saęlar. GKE node'ları bu kaynaklar zerinde alıřır.
3. **Storage Admin:** GKE'nin disk ve veri depolama iřlemleri iin gerekli izinleri saęlar.
4. **Service Account User:** Servis hesaplarını kullanma yetkisini saęlar.
5. **Viewer:** Projedeki kaynakları grntleme yetkisini saęlar (genellikle opsiyonel).

Bu yetkiler, genellikle `roles` olarak atanır. rneęin:

- `roles/container.admin`
- `roles/compute.admin`
- `roles/storage.admin`
- `roles/iam.serviceAccountUser`

GKE Cluster Oluřturma Adımları

1. Google Cloud SDK'yı Kurma ve Konfigre Etme

İlk olarak, Google Cloud SDK'nın (gcloud) ykl ve yapılandırılmıř olması gerekir. Eęer ykl deęilse, <https://cloud.google.com/sdk/docs/install> takip ederek yklebilirsiniz.

2. Google Cloud Hesabına Giriř Yapma

Google Cloud hesabınıza giriř yapın. Bu iřlem, `gcloud` komutlarının sizin kimlięinizle Google Cloud zerinde iřlemler yapabilmesini saęlar:

```
gcloud auth login
```

Komut alıřtırıldığında, web tarayıcınızda Google hesabınızla giriř yapmanız gereken bir sayfa aılacaktır. Giriř yaptıktan sonra, terminalde bařarılı bir řekilde giriř yaptığınız mesajını grmelisiniz.

3. GCP Projesi ve Zonu Seęme

```
gcloud config set project <PROJECT_ID>
```

```
gcloud config set compute/zone <ZONE>
```



Project info



Project name

My First Project

Project number

639998474696

Project ID

rosy-elem-43794018-b9

Burada **<PROJECT_ID>** GCP projenizin kimliği, **<ZONE>** ise cluster'ınızı oluşturmak istediğiniz bölge.

Örneğin:

```
gcloud config set project rosy-elem-43794018-b9
gcloud config set compute/zone us-central1-a
```

```
(base) ~ ➦ gcloud config set project rosy-elem-43794018-b9
Updated property [core/project].
(base) ~ ➦ gcloud config set compute/zone us-central1-a
Updated property [compute/zone].
(base) ~ ➦ gcloud services enable container.googleapis.com
Operation "operations/acf.p2-639908474696-53764b95-c366-47b2-a766-5ac79b9aed5e" finished successfully.
```

4. GKE API'ını Etkinleştirme

GKE API'nın etkinleştirilmiş olması gerekir:

```
gcloud services enable container.googleapis.com
```



Kubernetes Engine API

[Google Enterprise API](#)

Builds and manages container-based applications, powered by the open source Kubernetes technology.

ENABLE

TRY THIS API [↗](#)

5. GKE Cluster Oluşturma

```
gcloud container clusters create <CLUSTER_NAME> \
--num-nodes=2 \
--machine-type=n1-standard-2 \
--zone=<ZONE>
```

Burada <CLUSTER_NAME> cluster'ınızın adı, --num-nodes node sayısı, --machine-type node tipini belirtir. --zone parametresi, cluster'ın yer alacağı bölgeyi belirtir.

Örnek:

```
gcloud container clusters create test-gke \
--num-nodes=1 \
--machine-type=e2-medium \
--image-type=COS_CONTAINERD \
--zone=us-central1-a
```

```
(base) ~ ➤ gcloud container clusters create test-gke \
--num-nodes=2 \
--machine-type=n1-standard-2 \
--zone=us-central1-a
WARNING: Starting in January 2021, clusters will use the Regular release channel by default when '--cluster-version', '--release-channel', '--no-enable-autoupgrade', and '--no-enable-autorepair' flags are not specified.
WARNING: Currently VPC-native is the default mode during cluster creation for versions greater than 1.21.0-gke.1500. To create advanced routes based clusters, please pass the '--no-enable-ip-alias' flag
WARNING: Starting with version 1.18, clusters will have shielded GKE nodes by default.
WARNING: Your Pod address range ('--cluster-ipv4-cidr') can accommodate at most 1008 node(s).
WARNING: Starting with version 1.19, newly created clusters and node-pools will have COS_CONTAINERD as the default node image when no image type is specified.
Creating cluster test-gke in us-central1-a...:
```

6. GKE Cluster ile Etkileşime Geçmek İçin Kubeconfig Dosyasını Ayarlama

```
gcloud container clusters get-credentials <CLUSTER_NAME> --zone=<ZONE>
```

Örnek:

```
gcloud container clusters get-credentials test-gke --zone=us-central1-a
```

Bu komut, kubectl komutlarının GKE cluster'ınızla etkileşimde bulunabilmesi için .kub/config dosyanızı günceller.

“Flask Python Uygulamasını Deploy Etme

1. Flask Uygulaması İçin Dockerfile Oluşturma

```
FROM python:3.8-slim
WORKDIR /app
RUN pip install --upgrade pip setuptools
COPY requirements.txt .
RUN pip install -r requirements.txt --verbose
COPY . .
EXPOSE 5000
```

```
CMD ["gunicorn", "--bind", "0.0.0.0:5000", "app:app"]
```

2. Docker Image Build ve Push

Flask uygulamanızın Docker image'ını oluşturun ve Google Container Registry'ye (GCR) push edin.

Build Docker image

```
docker build -t hbayraktar/flask-app:2.2 .
```

alternatif olarak mac kullanıcıları image'ın linuxda çalışabilmesi için multi platform desteği ile derleyebilirsiniz

```
docker build --platform linux/amd64 -t  
hbayraktar/flask-app:2.2 .
```

Push Docker image to dockerhub

```
docker push hbayraktar/flask-app:2.2
```

3. Flask Uygulaması İçin Kubernetes Deployment ve Service Tanımları

flask-deployment.yaml:

```
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: flask-app  
spec:  
  replicas: 2  
  selector:  
    matchLabels:  
      app: flask-app  
  template:  
    metadata:  
      labels:  
        app: flask-app  
    spec:  
      containers:  
        - name: flask-app  
          image: hbayraktar/flask-app:2.2  
          ports:  
            - containerPort: 5000
```

flask-service.yaml:

```
apiVersion: v1
kind: Service
metadata:
  name: flask-app-service
spec:
  selector:
    app: flask-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 5000
  type: LoadBalancer
```

requirements.txt

```
Flask==2.0.3
gunicorn==20.1.0
```

4. Deployment ve Service'i Uygulama

```
kubectl apply -f flask-deployment.yaml
kubectl apply -f flask-service.yaml
```

5. Uygulama URL'sini Kontrol Etme

GKE, **LoadBalancer** türündeki servislere bir IP adresi atayacaktır. IP adresini almak için aşağıdaki komutu kullanın:

```
kubectl get services
```

```
(base) GKE ➔ k get svc
NAME                TYPE           CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
flask-app-service   LoadBalancer  34.118.228.201 34.136.208.239 80:32590/TCP     32m
kubernetes          ClusterIP      34.118.224.1   <none>         443/TCP          47m
```

Bu komut, **EXTERNAL-IP** kolonunda uygulamanızın erişim adresini gösterecektir. Tarayıcınızda bu IP adresini ziyaret ederek Flask uygulamanızı görebilirsiniz.

