

## İlk docker imajınızı Code Engine üzerinde dağıtma



Bu laboratuvarıda, ilk Docker imajınızı Code Engine üzerinde nasıl dağıtacağınızı öğreneceksiniz. IBM Cloud Code Engine bu laboratuvar ortamı aracılığıyla size sunulmuştur.

**Tahmini Süre: 20 dakika**

### Öğrenme Hedefleri:

Bu laboratuvarı tamamladıktan sonra şunları yapabileceksiniz:

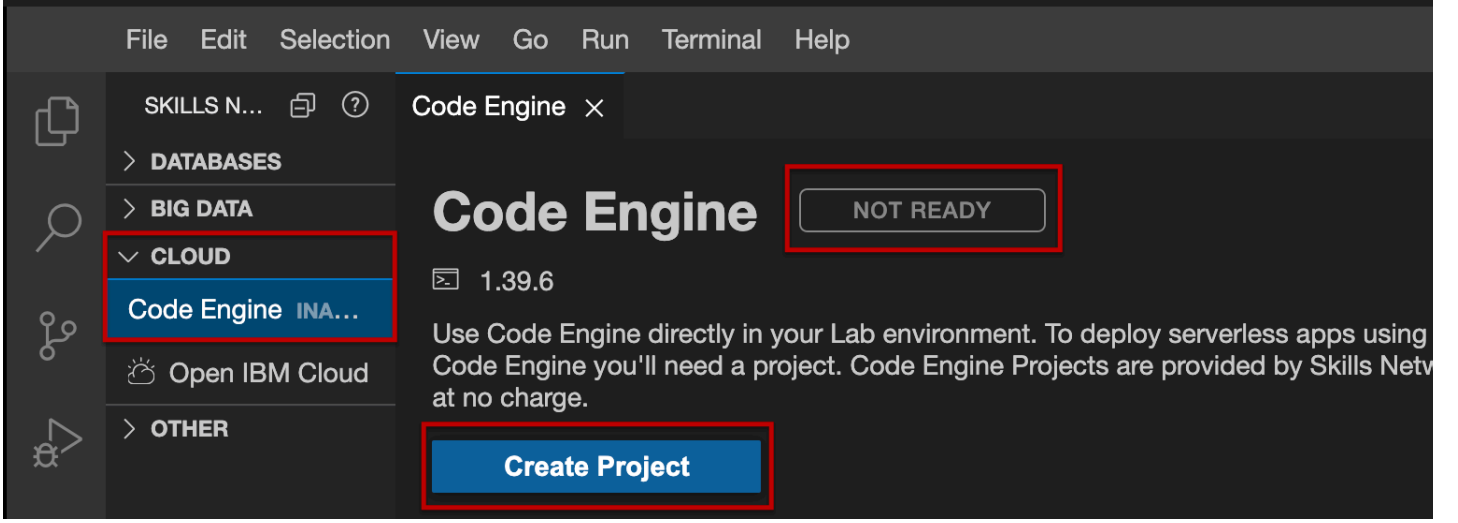
1. Uygulama oluşturmak için Code Engine hizmetini başlatın.
2. Bir docker imajından uygulama dağıtmak ve uygulama için uzaktan erişim URL'si oluşturmak için code engine hizmetini kullanın.
3. Dağıttığınız uygulamaları listeleyin.

## Docker imajını dağıtma

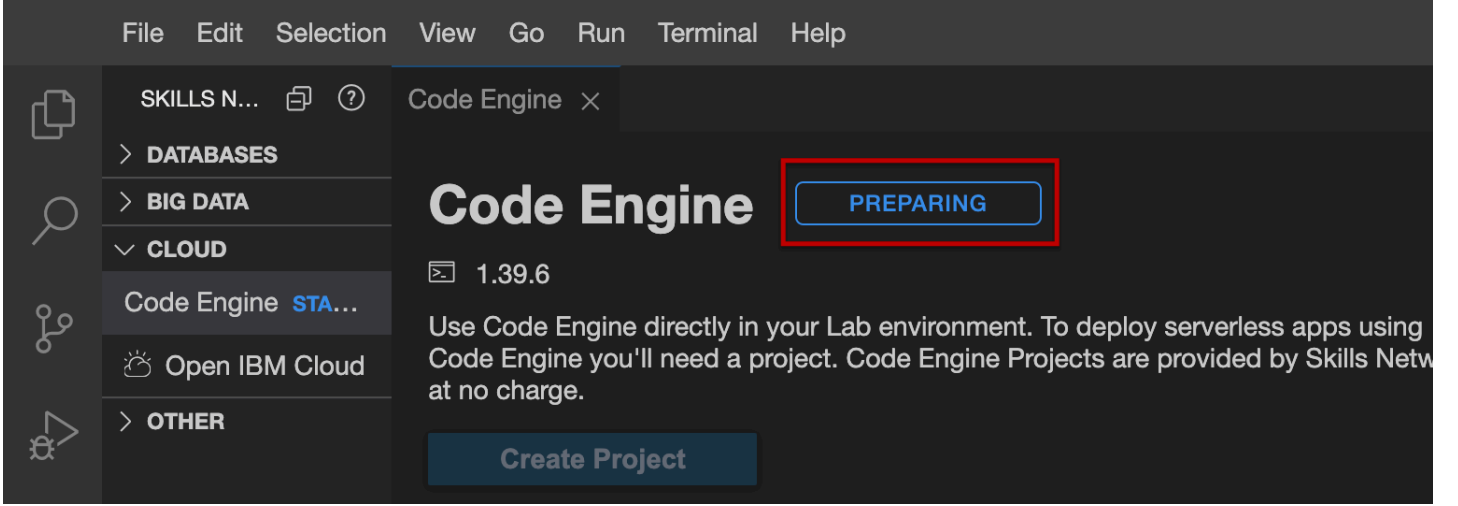
1. Code engine CLI terminaline gidin. Eğer bir tane yoksa, aşağıdaki bağlantıya tıklayarak bir tane ayarlayın.

▼ Bir tane nasıl ayarlayacağınızı görmek için buraya tıklayın

1. Laboratuvar ortamınızdaki menüde, Cloud açılır menüsüne tıklayın ve Code Engine seçeneğini seçin. Code engine ayar paneli açılacaktır. Create Project butonuna tıklayın.



2. Code engine ortamının hazırlanması biraz zaman alır. Ayar panelinde ilerleme durumu gösterilecektir.



3. Code engine ayarı tamamlandığında, aktif olduğunu göreceksiniz. Aşağıdaki terminalde önceden yapılandırılmış CLI'yi başlatmak için Code Engine CLI butonuna tıklayın.

File Edit Selection View Go Run Terminal Help

SKILLS NETWORK T... Code Engine X

> DATABASES

> BIG DATA

✓ CLOUD

Code Engine **ACTIVE**

Open IBM Cloud

> OTHER

# Code Engine

1.39.6

Use Code Engine directly in your Lab environment. To deploy serverless applications, use the Code Engine CLI. Code Engine Projects are provided by Skills Network at no charge.

Delete Project

Summary Project Information Details

Your Skills Network Code Engine Project is now ready to use. You can now deploy serverless applications.

For important information about your project view the Project Information section. If you want to use Code Engine as an IBM Cloud Service, please check out the Details section.

In order to interact with Code Engine please click the following button:

Code Engine CLI

4. Önceden yapılandırılmış CLI'nin başlatıldığını ve ana dizinin mevcut dizine ayarlandığını gözlemleyeceksiniz. Ön yapılandırmanın bir parçası olarak proje ayarlandı ve Kubeconfig yapılandırıldı. Terminalde gösterilen detaylar.

```
ibmcloud ce project current
theia@theiadocker-lavanyas:/home/project$ ibmcloud ce project current
Getting the current project context...
OK

Name:      Code Engine - sn-labs-lavanyas
ID:        ee5183a9-4516-4bd1-8f4e-4a8615cafd81
Subdomain: v9oc2xsjxaz
Domain:    us-south.codeengine.appdomain.cloud
Region:    us-south

Kubernetes Config:
Context:    v9oc2xsjxaz
Environment Variable: export KUBECONFIG="/home/theia/.bluemix/plugins/code-engine/Code Engine
sn-labs-lavanyas-ee5183a9-4516-4bd1-8f4e-4a8615cafd81.yaml"
theia@theiadocker-lavanyas:/home/project$
```

Artık Hello World uygulamasını dağıtmak için CLI'yi kullanacaksınız.

2. Mevcut uygulamaların listesini görmek için aşağıdaki komutu çalıştırın.

```
ibmcloud ce app list
```

3. GitHub'dan kodu klonlayacak, bunu dockerize edecek ve kök seviyede bir REST API uç noktası sunan ve Hello World dizesini döndüren web uygulamasını dağıtacaksınız. Kodu klonlamak için aşağıdaki komutu çalıştırın.

```
git clone https://github.com/ibm-developer-skills-network/danum-pythonflaskserver
```

4. Aşağıdaki komutu çalıştırarak klonlanan dizine geçin.

```
cd danum-pythonflaskserver
```

5. Şimdi mevcut dizinde `docker build` komutunu çalıştırın ve resmi etiketleyin. Aşağıdaki komutta uygulamayı `helloworld2` olarak adlandırdığımıza dikkat edin çünkü proje alanında hala önceki `helloworld` örneği bulunabilir.

```
docker build . -t us.icr.io/${SN_ICR_NAMESPACE}/helloworld2
```

6. Şimdi resmi ad alanına iterek çalıştırabilirsiniz.

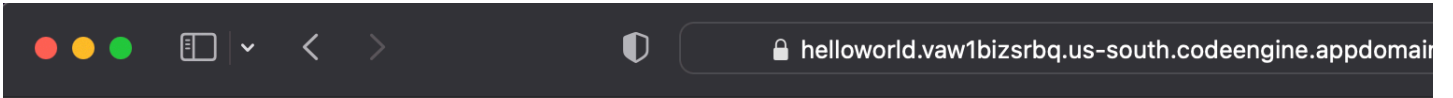
```
docker push us.icr.io/${SN_ICR_NAMESPACE}/helloworld2
```

7. Artık görüntü dağıtımına hazır olduğuna göre, aşağıdaki komutu çalıştırın. Daha önce görüntüyü oluşturup göndermiş olduğumuz için, uygulamayı oluştururken yapı kaynağını belirtmemize gerek olmadığını unutmayın. Komutun uygulamayı oluşturduğunu ve ayrıca gerekli altyapıyı içsel olarak ayarladığını göreceksiniz. Bu birkaç saniye alır ve sonunda URL ile birlikte bir onay verir.

```
ibmcloud ce application create --name helloworld2 --image us.icr.io/${SN_ICR_NAMESPACE}/helloworld2 --registry-secret icr-secret --port 5000
```

```
theia@theiadosker-lavanyas:/home/project/danum-pythonflaskserver$ ibmcloud ce application create --name helloworld2 --image us.icr.io/${SN_ICR_NAMESPACE}/helloworld2 --registry-secret icr-secret --port 5000
Creating application 'helloworld2'...
Configuration 'helloworld2' is waiting for a Revision to become ready.
Ingress has not yet been reconciled.
Waiting for load balancer to be ready.
Run 'ibmcloud ce application get -n helloworld2' to check the application status.
OK
https://helloworld2.vjuxldzxxcg.us-south.codeengine.appdomain.cloud
```

8. ctrl(Windows)/cmd(Mac) tuşlarına basın ve oluşturulan bağlantıya tıklayın. Alternatif olarak, bağlantıyı kopyalayıp bir tarayıcı sayfasına yapıştırsanız ve enter tuşuna basarsanız, `hello world` uygulama sayfası aşağıda verilen şekilde görüntülenir.



# Hello World!

## Uygulama Alıştırması:

1. Dosya menüsüne gidin, `danum-pythonflaskserver/app.py` dosyasını açın ve mesajı “Hello World” yerine “Merhaba yourname!” olarak değiştirin.
2. Dosyayı kaydedin, `docker`’ı yeniden oluşturun ve uygulamayı `ibmcloud ce application update` ile güncelleyin.

▼ Çözüm için buraya tıklayın

```
ibmcloud ce application update --name helloworld2 --image us.icr.io/${SN_ICR_NAMESPACE}/helloworld2 --registry-secret icr-secret --port 5000
```

3. Oluşturulan URL'yi açın ve uygulamanın güncellenip güncellenmediğine bakın.

Tebrikler! Bu laboratuvarı başarıyla tamamladınız ve Code Engine üzerinde ilk uygulamanızı dağıttınız.

## Yazar(lar)

Lavanya T S

© IBM Corporation. Tüm hakları saklıdır.