Deploying your first docker image on Code Engine

Skills Network

In this lab, you will learn how to deploy your first Docker image on Code Engine. IBM Cloud Code Engine has been made available to you through this lab environment.

Estimated Time: 20 mins

Learning Objectives:

Aftet completing this lab you will be able to:

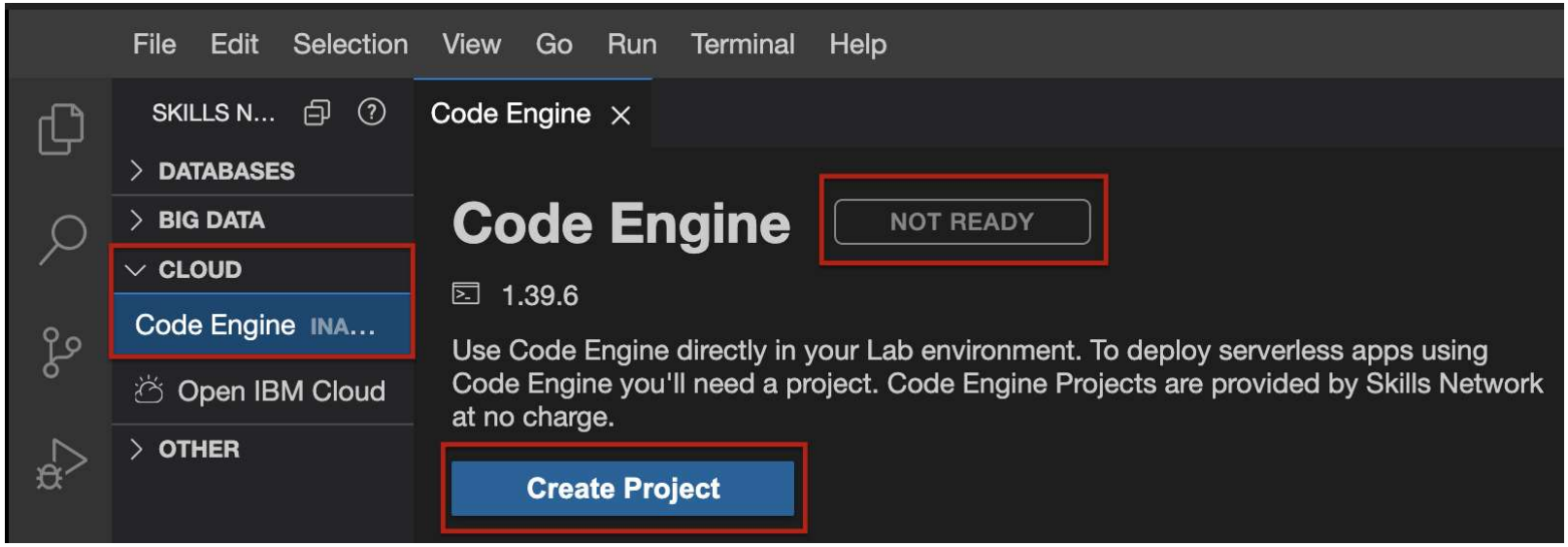
1. Start Code Engine service to create applications
2. Use the code engine service to deploy an application from a docker image and create a remote access URL for the application.
3. List the applications you have deployed.

Deploying the docker image

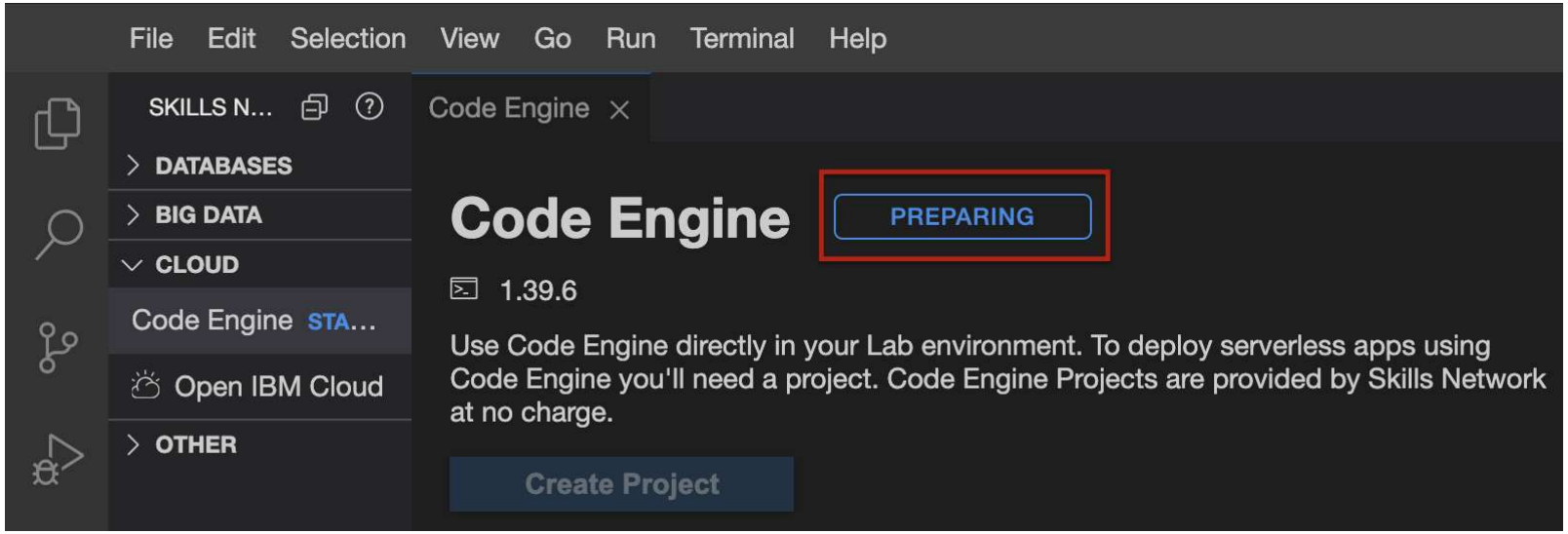
1. Go to the Code engine CLI terminal. If you don't have one, click below to set it up.

▼ Click here to see how to set up one

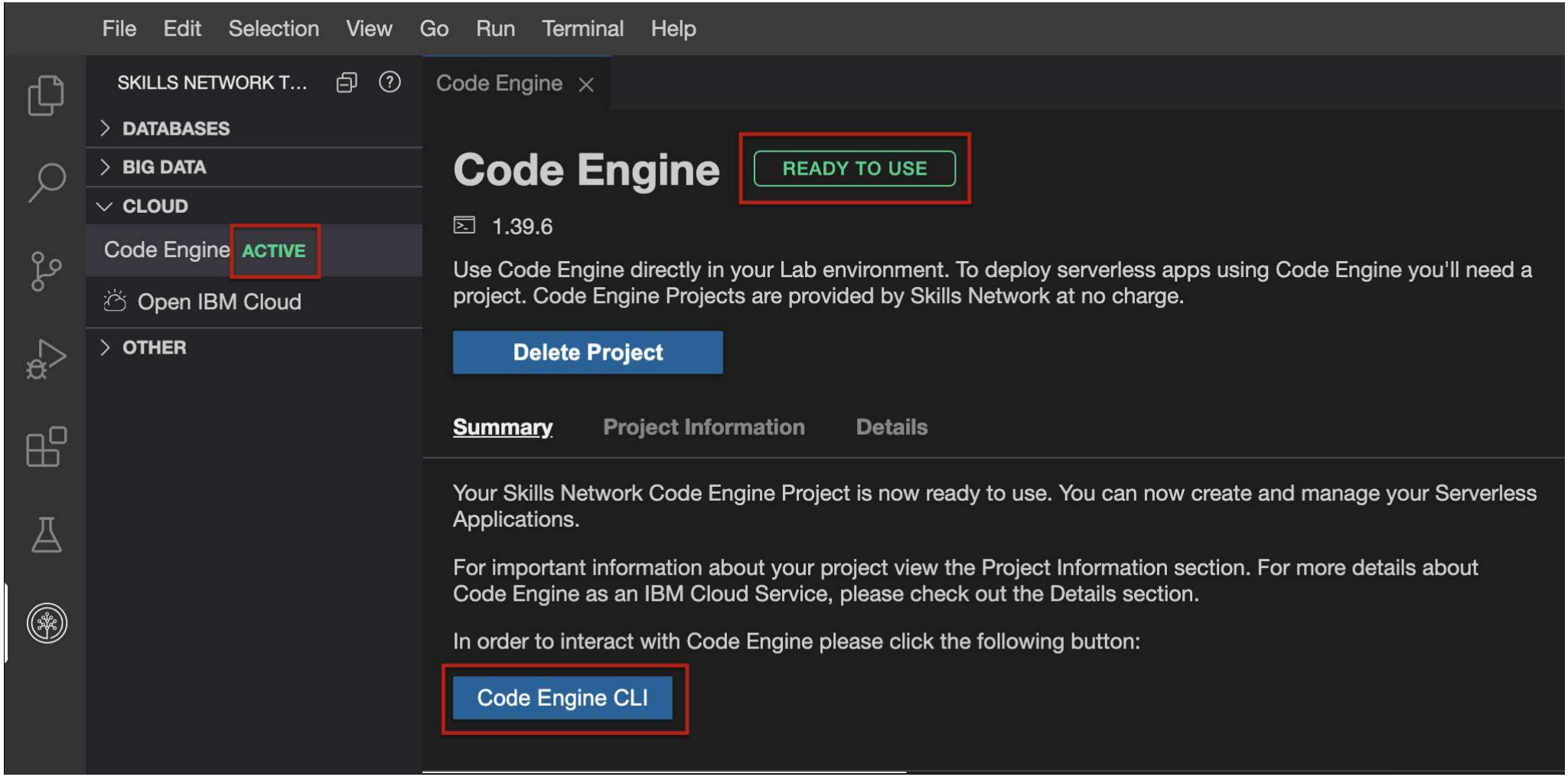
1. On the menu in your lab environment, Click `Cloud` dropdown and choose `Code Engine`. The code engine set up panel comes up. Click `Create Project`.



2. The code engine environment takes a while to prepare. You will see the progress status being indicated in the set up panel.



3. Once the code engine set up is complete, you can see that it is active. Click on `Code Engine CLI` to begin the pre-configured CLI in the terminal below.



4. You will observe that the pre-configured CLI statrup and the home directory is set to the current directory. As a part of the pre-configuration, the project has been set up and Kubeconfig is set up. The details that are shown on the terminal.

```
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$
```

You will now use the CLI to deploy the Hello World application.

2. Run the following command to see the list of applications that exist.

```
1
ibmcloud ce app list
```

Copied! Executed!

3. You will clone the code from github, dockerize it and deploy the web application which serves one REST API endpoint at the root level and returns the string `Hello World`. Run the following command to clone the code.

```
1
git clone https://github.com/ibm-developer-skills-network/danum-pythonflaskserver
```

Copied! Executed!

4. Change to the cloned directory by running the following command.

```
1
cd danum-pythonflaskserver
```

Copied! Executed!

5. Now run `docker build` in the current directory and tag the image. Note that in the below command we are naming the app `helloworld2` as we may have the earlier instance of `helloworld` still in the project space.

```
1
docker build . -t us.icr.io/${SN_ICR_NAMESPACE}/helloworld2
```

Copied! Executed!

6. Now push the image to the namespace so that you can run it.

```
1
docker push us.icr.io/${SN_ICR_NAMESPACE}/helloworld2
```

Copied! Executed!

7. Now that the image is all set to be deployed, run the following command. Please note that since we already built and pushed the image, we can create the application without mentioning the build source. You will see that the command creates the application and also internally sets up the required infrastructure. It takes a a few seconds and it finally gives a confirmation along with the URL.

```
1
ibmcloud ce application create --name helloworld2 --image us.icr.io/${SN_ICR_NAMESPACE}/helloworld2 --registry-secret icr-secret --port 5000
```

Copied! Executed!

```
theia@theiadocker-lavanyas:/home/project/danum-pythonflaskserver$ ibmcloud ce appli
cation 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. Press ctrl(Windows)/cmd(Mac) and the link that is created. Alternatively copy the link and paste it in a browser page and press enter. The `hello world` application page renders as given below.

Hello World!

Practice Exercise:

1. Go to the file menu, open `pythonflaskserver/app.py` and change the message from “Hello World” to “Hello yourname!”.
2. Save the file, build docker again and update the application using `ibmcloud ce application update`.

▼ Click here for the solution

```
1. 1
```

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

Copied! Executed!

3. Open the URL that is generated and see if the application has got updated.

Congratulations! You have completed this lab successfully and deployed your first application on Code Engine.

Author(s)

Lavanya T S

Changelog

Date	Version	Changed by	Change Description
2022-11-21	0.1	Lavanya	Initial version created
2023-07-18	0.2	Sapthashree	Updated the lab overview