Using Code Engine

Estimated time needed: 30 minutes

In the last video, you learned about IBM Code Engine, a service on IBM Cloud that let's you deploy microservices at scale and without the need to manage the infrastructure at the backend. Let's learn about some of the commands available in Code Engine that will assist you to complete the final project in this course.

Objectives

After completing this reading, you will be able to:

1. Explain how to interact with IBM Code Engine using the IBM Cloud CLI
2. Describe projects in Code Engine
3. Describe the differences between applications and jobs in Code Engine
4. Explain the steps required to deploy your code to the Code Engine

Code Engine CLI

All the exercises in the next lab and the final capstone lab will use the Code Engine CLI. Therefore, it is important to learn how the CLI works. The Code Engine CLI can be installed as a plugin with the IBM Cloud CLI.

1. Ensure you have IBM Cloud CLI installed. You can check this with the `ibmcloud version` command. You should see an output similar to the following:

```
1. 1
2. 2
1. $ ibmcloud version
2. ibmcloud version 2.13.0+05fd406-2022-11-30T21:03:31+00:00
```
Copied!

2. The Code Engine plugin can be invoked by using the `ce` command.

```
1. 1
2. 2
3. 3
1. ibmcloud ce
2. FAILED
3. 'ce' is not a registered command. Check your list of installed plug-ins. See 'ibmcloud help'.
```
Copied!

   As you can see `ce` is not recognized as a command. We can fix this by installing the `code-engine` plugin in the IBM Cloud CLI.
3. Install the code-engine plugin with the following command:

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
1. ibmcloud plugin install code-engine
2. Looking up 'code-engine' from repository 'IBM Cloud'...
3. Plug-in 'code-engine 1.41.1' found in repository 'IBM Cloud'
4. Installing binary...
5. OK
6. Plug-in 'code-engine 1.41.1' was successfully installed
```
Copied!

4. You should now be able to use the code engine CLI:

```
1. 1
2. 2
1. $ ibmcloud ce version
2. version:  v1.41.1
```
Copied!

The lab environment in this course comes with the IBM Cloud CLI and the Code Engine plugin preinstalled.

Projects in IBM Code Engine

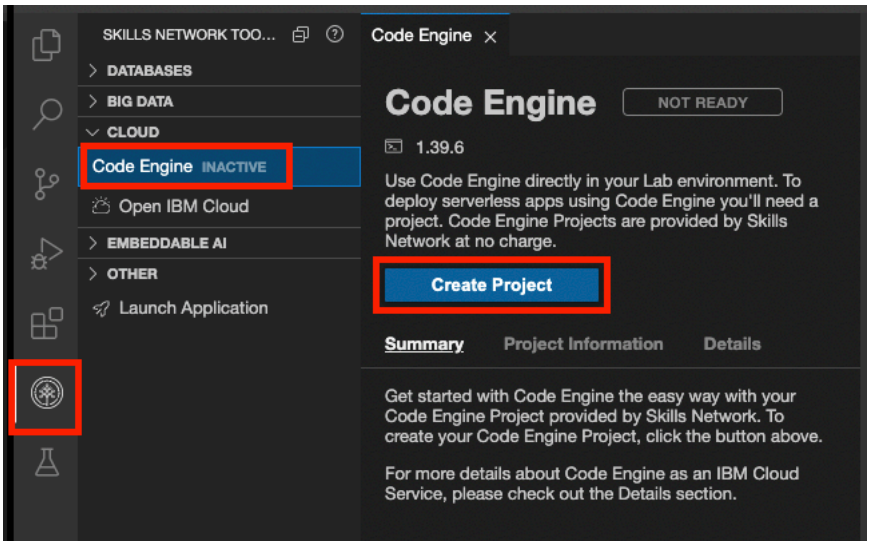
You will first create a project in the Code Engine. A project groups together different entities like applications, jobs, builds, environment variables, secrets, and others.

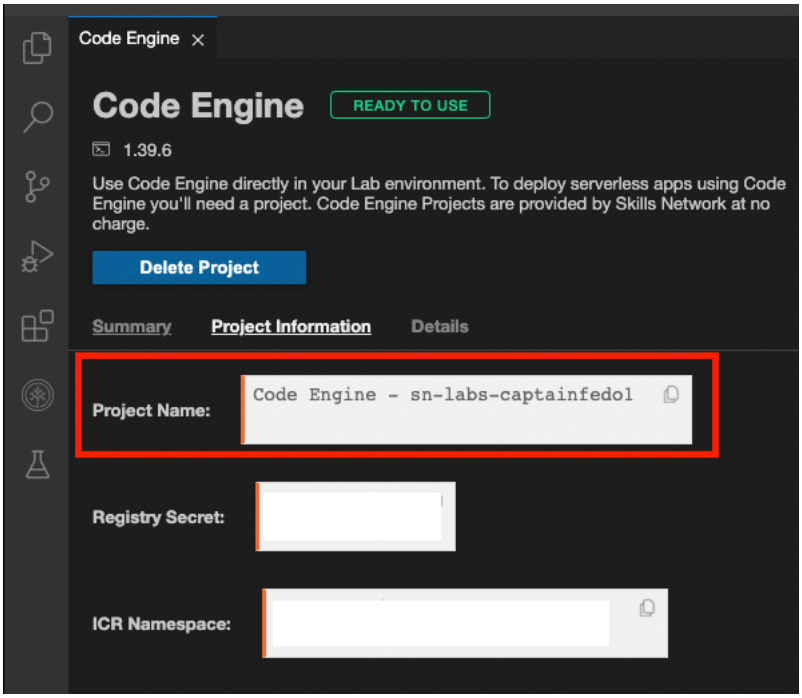
1. To create a project in the Code Engine using the CLI, you will execute the following command:

```
1. 1
1. ibmcloud ce project create --name myproject
```
Copied!

   This will create a project named `myproject` in your Code Engine account.

   The lab environment automatically creates a project for you when you start code engine using the `Create Project` button.





2. You can use the `ibmcloud ce project list` command to see all the projects in the Code Engine service. You should see a single project in the lab environment:

```
1. 1
2. 2
3. 3
4. 4
5. 5
1. theia@theiadocker-captainfedo1:/home/project$ ibmcloud ce project list
2. Getting projects...
3. OK
4. Name                              ID                                    Status  Enabled  Selected  Tags                        Region    Reso
5. Code Engine - sn-labs-captainfedo1  54413f5c-5b19-4dc8-866e-43527fbee246  active  true     true      captainfedo1,skillsnetwork  us-south  prod
```

Copied!

Applications in Code Engine

An application or app runs your code to serve HTTP requests. An application has a URL for incoming requests. The number of running instances of an application is automatically scaled up or down (to zero) based on the incoming workload.

1. You can use the `ibmcloud ce app create` command to create a new application. An application can be created from your source code, Dockerfile, or even a pre-existing image in an image registry.
2. Let's create an application from code in the GitHub repository `https://github.com/IBM/CodeEngine`. The command is as follows:

```
1. 1
2. 2
3. 3
4. 4
5. 5
1. ibmcloud ce application create --name myapp \
2. --build-source https://github.com/IBM/CodeEngine \
3. --build-context-dir hello \
4. --image us.icr.io/${SN_ICR_NAMESPACE}/hello \
```

```
5. --registry-secret icr-secret
```
Copied!

- `--name`: Name of the new application.
- `--build-source`: The URL of the Git repository that contains your source code; for example, `https://github.com/IBM/CodeEngine`.
- `--build-context-dir`: The directory in the repository that contains the buildpacks file or the Dockerfile. This value is optional.
- `--image`: The name of the image that is used for this application. The format is 'REGISTRY/NAMESPACE/REPOSITORY:TAG' where 'REGISTRY' and 'TAG' are optional.
- `--registry-secret`: The name of the image registry access secret. The image registry access secret is used to authenticate with a private registry when you download the container image.

If the repostory code contains a Dockerfile, that will be used to create the image for the application. If there is no Dockerfile, Code Engine is smart enough to inspect the directory and create an image from the source code directly.

Jobs in Code Engine

A job runs one or more instances of your executable code. Unlike applications, which include an HTTP Server to handle incoming requests, jobs are designed to run one time and exit.

1. As with images, you can create jobs from source code or pre-existing images. Let's create a job with a pre-existing image called `firstjob` in the IBM container registry `icr.io/codeengine`:

```
1. 1
2. 2
3. 3
1. ibmcloud ce job create \
2. --name myjob  \
3. --image icr.io/codeengine/firstjob
```
Copied!

   - `--image`: The name of the image that is used to run the job.
   - `--name`: The name of the new job.
2. You should see an output as follows:

```
1. 1
2. 2
1. Creating job 'myjob'...
2. OK
```
Copied!

3. After you have created a job in Code Engine, you can use the following command to execute the job:

```
1. 1
1. ibmcloud ce jobrun submit --name testfirstjobrun --job firstjob
```
Copied!

   - `--name`: The name of the new job run.
   - `--job`: The job to run. You will run the `firstjob` job you created in the previous step.

Author(s)

CF