Introduction to Cloud Computing Final Project - Guess the Capital

Estimated time needed: 30 minutes

In this final project, you will be deploying "Guess the Capital" on the cloud. It is a web application that asks you to guess the capital of a country from 4 choices.

You will use the source code and the steps provided to practice hands-on how an application can be developed and deployed on the cloud.

Objectives:

1. Clone the source code
2. Build Docker image
3. Deploy on Docker
4. Tag and Push image to IBM Cloud
5. Deploy on IBM Code Engine

Background

Docker

Containers are isolated environments that package applications and their dependencies. Each container runs as an isolated process on the host operating system.

Docker is an open-source platform that enables developers to automate the deployment and management of applications inside lightweight, isolated containers.

IBM Cloud

IBM Cloud is a cloud computing platform and suite of cloud-based services offered by IBM. It provides a range of infrastructure, platform, and software services to support the development, deployment, and management of various types of applications and workloads in the cloud.

IBM Code Engine

IBM Cloud Code Engine is a serverless compute platform provided by IBM Cloud. It allows developers to deploy and run containerized applications without the need to manage the underlying infrastructure. Abstracting away the complexities of server provisioning, scaling, and maintenance, enabling developers to focus on writing code and building applications.

Working with files in Cloud IDE

If you are new to Cloud IDE, this section will show you how to create and edit files, which are part of your project, in Cloud IDE.

To view your files and directories inside Cloud IDE, click on this files icon to reveal it.

If you have cloned (using `git clone` command) boilerplate/starting code, then it will look like below:

Otherwise a blank project looks like this:

Create a new file

You can right-click and select the New File option to create a file in your project.

You can also choose `File -> New File` to do the same.

It will then prompt you to enter name of this new file. In the example below, we are creating `sample.html`.

Clicking on the file name `sample.html` in the directory structure will open the file on the right pane. You can create all different types of files; for example `FILE_NAME.js` for JavaScript file.

In the example, we just pasted some basic html code and then saved the file.

And saving it by:

- Going in the menu.
- Press 梠 + `S` on Mac or `CTRL + S` on Windows.
- Or it can Autosave it for you too.

Verify the environment and command line tools

1. Open a terminal window by using the menu in the editor: `Terminal > New Terminal`.

   Note:If the terminal is already opened, please skip this step.

2. Verify that `docker` CLI is installed.

```
docker --version
```

You should see the following output, although the version may be different:

3. Verify that `ibmcloud` CLI is installed.

   ```
   ibmcloud version
   ```

You should see the following output, although the version may be different:

Start Code Engine

1. On the menu in your lab environment, Click on SN logo icon and then click the `Cloud` dropdown menu and select `Code Engine`. The code engine setup panel appears. Click `Create Project` to begin.

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

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

4. You will observe that the pre-configured CLI startup and the home directory are 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 are shown on the terminal as follows.

Set-up : Create application

1. Open a terminal window by using the menu in the editor: **Terminal > New Terminal**.

2. If you are not currently in the project folder, copy and paste the following code to change to your project folder.

   ```
   cd /home/project
   ```

3. Run the following command to clone the Git repository that contains the starter code needed for this project if the Git repository doesn't already exist.

   ```
   [ ! -d 'fyidw-guess-the-capital' ] && git clone https://github.com/ibm-developer-skills-network/fyidw-guess-the-capital.git
   ```

4. Change to the directory **fyidw-guess-the-capital** to start working on the lab.

   ```
   cd fyidw-guess-the-capital
   ```

5. List the contents of this directory to see the artifacts for this lab.

   ```
   ls
   ```

6. Run the following command on the terminal to host your web page.

   ```
   python3 -m http.server
   ```

7. To test your application in your browser, run the application first. To view your application, click the Skills Network icon on the left panel (refer to number 1). This action will open the **SKILLS NETWORK TOOLBOX**. Next, click **Launch Application** (refer to number 2). Enter port number **8000** in **Application Port** (refer to number 3) and click . You can also click on the button given below to launch your application.

Launch Application

8. It will look like this:

9. In your terminal, press `CTRL + C` to stop your web server.

Task 1: Containerise the application

Let's start modernising our application. The first step towards it is to containerise it using Docker.

Create Dockerfile

Your tasks:

1. Paste the following content in

Open **Dockerfile** in IDE

Use the below as Dockerfile content.

```
# Use the official Nginx base image from Docker Hub
FROM nginx
# Copy the favicon to the default Nginx HTML directory
COPY favicon.ico /usr/share/nginx/html/favicon.ico
# Copy the main HTML file to serve as the landing page
COPY index.html /usr/share/nginx/html/index.html
# Copy the JavaScript file to enable client-side functionality
COPY script.js /usr/share/nginx/html/script.js
# Copy the CSS file to define the visual presentation and styling
COPY style.css /usr/share/nginx/html/style.css
# Copy the JSON data file to serve structured content to the frontend
COPY data.json /usr/share/nginx/html/data.json
```

And it should look like below:

2. Build an image from a Dockerfile

   ```
   docker build -t guess-the-capital .
   ```

Giving you the output similar to:

3. List built images

```
docker images
```

4. Run the image

```
docker run -it -d -p 8080:80 guess-the-capital
```

5. Verify in browser

Launch Application

Task 2: Deploy on IBM Cloud

Let's start with launching Code Engine CLI.

Create Code Engine Project in IDE

```
cd /home/project/fyidw-guess-the-capital
docker build . -t us.icr.io/${SN_ICR_NAMESPACE}/guess-the-capital
```

Push the image to IBM Cloud

```
docker push us.icr.io/${SN_ICR_NAMESPACE}/guess-the-capital
```

Deploy the image on IBM CE

```
ibmcloud ce application create --name guess-the-capital --image us.icr.io/${SN_ICR_NAMESPACE}/guess-the-capital --registry-secret icr-secret --port 80
```

Take Cloud URL from the output; which looks something like: `https://guess-the-capital.somerandomalphanumeric.us-south.codeengine.appdomain.cloud` and open in your browser.

Optionally check the status

```
ibmcloud ce application get --name guess-the-capital
```

Congratulations

You have completed this final lab that showed you how to deploy and host a standard JavaScript application in Docker and on IBM Cloud.

Author(s)

Muhammad Yahya