Final Project - Product Price Comparison Application

Estimated Time Needed: 1.5 hours

- In this final project you will be deploying multiple microservices to create an integrated application. These components consist of two backend microservices, one developed in Python and the other in Node.js, complemented by a front-end microservice.

Objectives:

After completing this lab, you will be able to:

- Deploy Python and Node.js backend microservices on IBM Cloud Code Engine.
- Deploy the frontend microservices on IBM Cloud Code Engine.

Deploying the Backend Microservices

1. Open the Code Engine CLI.

2. Deploy the microservice for Product Details, which provides API endpoints to retrieve product information.

build-source - `https://github.com/ibm-developer-skills-network/dealer_evaluation_backend.git`

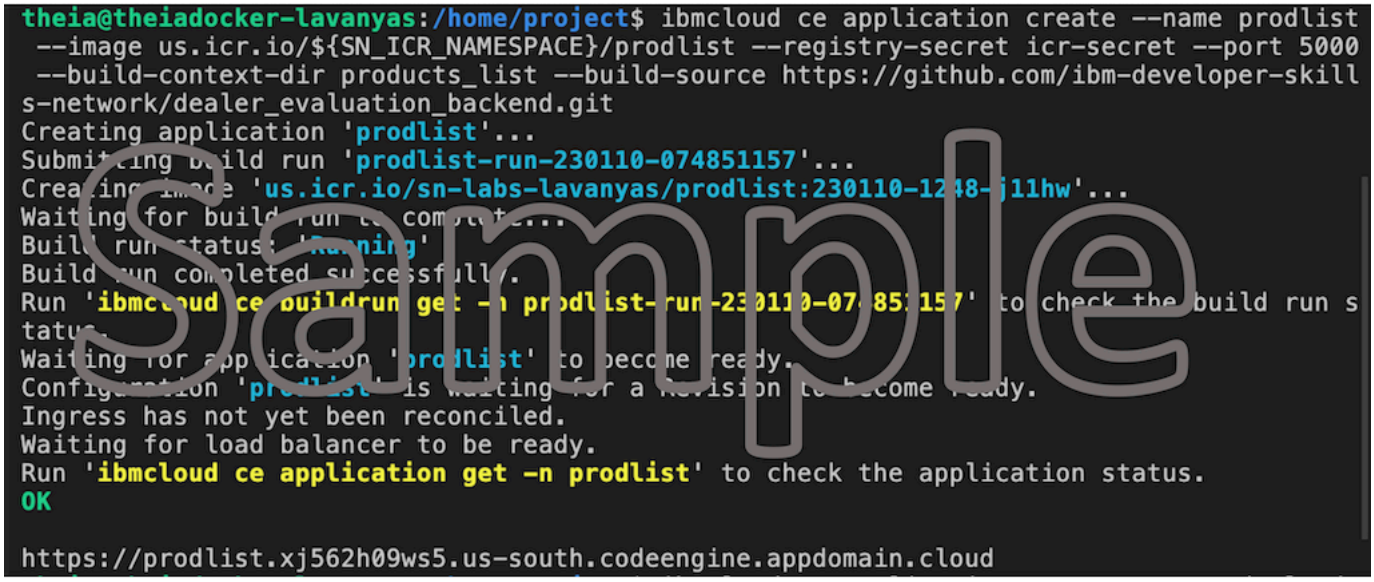
build-context-dir - `products_list`

port - `5000`

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

Copy the deployment URL and save it in a notepad or other text editors.

Take a screenshot of the successful deployment and save it as `product_details_deploy.png`.



Please note that if you encounter the error `FAILED Wait failed for application 'prodlist'`, you can rename the application to `prodlist1` and re-execute the command.

3. Deploy the microservice for Dealer Pricing, which provides API endpoints to retrieve dealer pricing information.

Note: Please use the below parameters for the deploy command

build-source - `https://github.com/ibm-developer-skills-network/dealer_evaluation_backend.git`

build-context-dir - `dealer_details`

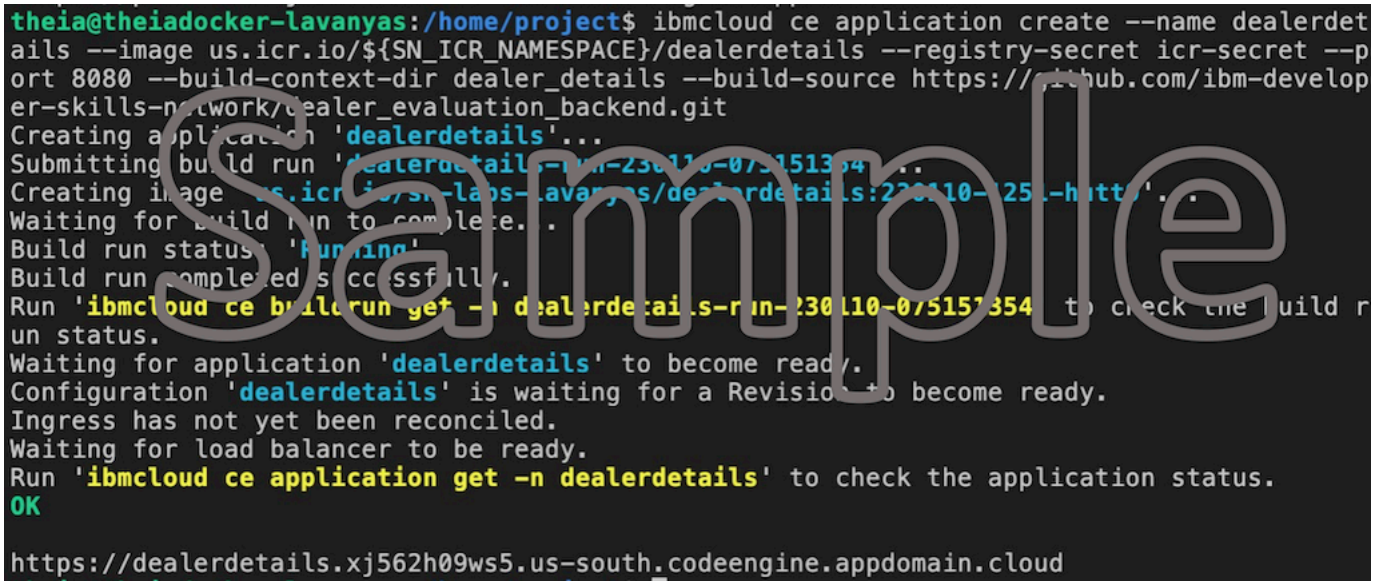
port - `8080`

name - `dealerdetails`

image - `us.icr.io/${SN_ICR_NAMESPACE}/dealerdetails`

Copy the deployment URL and save it in a notepad or other text editors.

Take a screenshot of the successful deployment and save it as `dealer_details_deploy.png`.



Please note that if you encounter the error `FAILED Wait failed for application 'dealerdetails'`, you can rename the application to `dealerdetails1` and re-execute the command.

Deploy the Frontend Microservice

1. Open new terminal, go to `/home/project` directory.

   ```
   cd /home/project
   ```

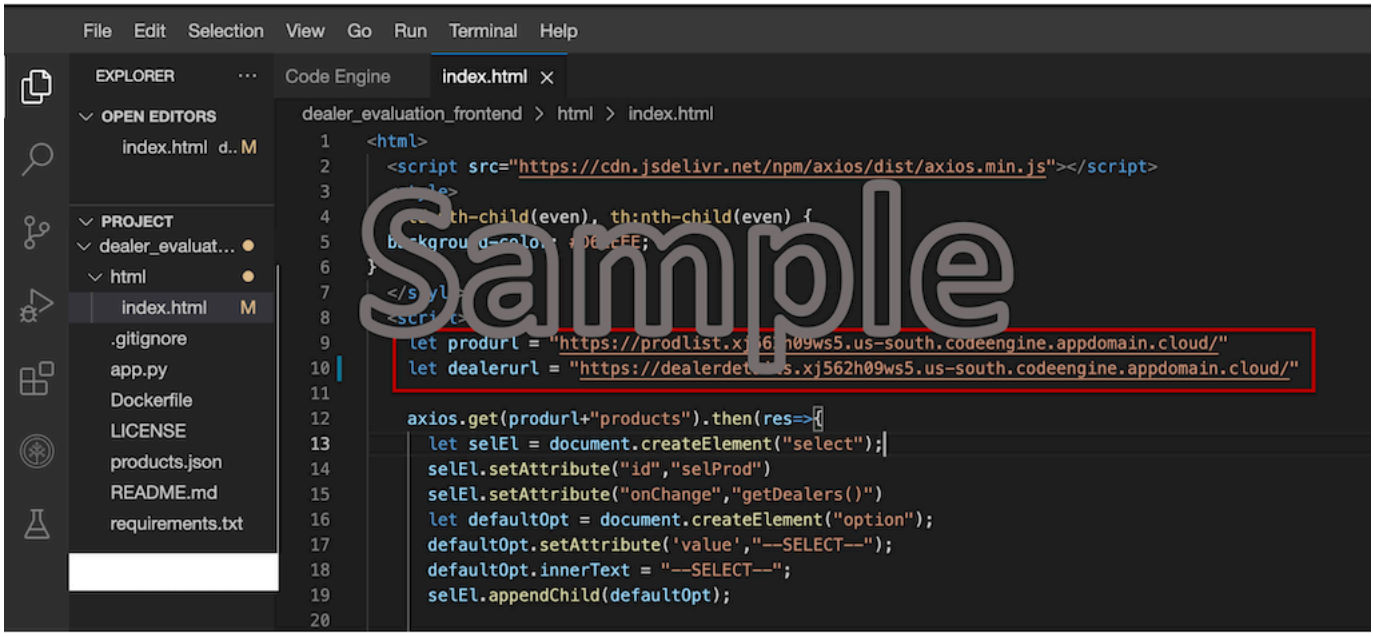
2. Clone the repository https://github.com/ibm-developer-skills-network/dealer_evaluation_frontend.git in your /home/project directory.

   Take a screenshot of the successful git cloning and save it as `git_clone.png`.

3. Change to the `dealer_evaluation_frontend` directory.

4. Update the index.html file with the deployment URLs obtained from the microservice deployments. (`http://localhost:5000/` and `http://localhost:8080/`), copy the deployment URLs you copied in the appropriate location. Make sure you end the URLs with a `/`.

   Take a screenshot of the changes and save it as `index_urlchanges.png`.



5. Deploy the Dealer Evaluation frontend microservice by pointing the build-source to the current directory.

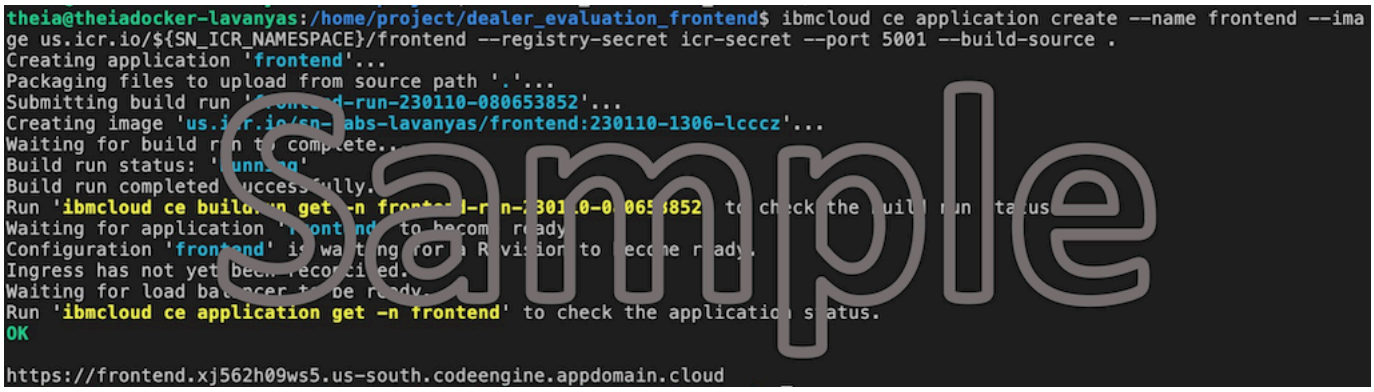
build-source - `.`

port - `5001`

name - `frontend`

image - `us.icr.io/${SN_ICR_NAMESPACE}/frontend`

Take a screenshot of the successful deployment and name it `frontend_deploy.png`.



Please note that if you encounter the error `FAILED Wait failed for application 'frontend'`, you can rename the application to `frontend1` and re-execute the command.

6. Click the link to load the homepage. Please note the page takes time to load the first time you access it.

7. Click the products drop down to see if the products have been populated.

Take a screenshot of the home page showing the products list and name it `homepage.png`.

8. Choose a specific dealer for the product and verify the price is displayed.

   Take a screenshot of the entire page showing the product chosen, and dealers that supply the listed product returned by the microservice and name it `product_dealer.png`.

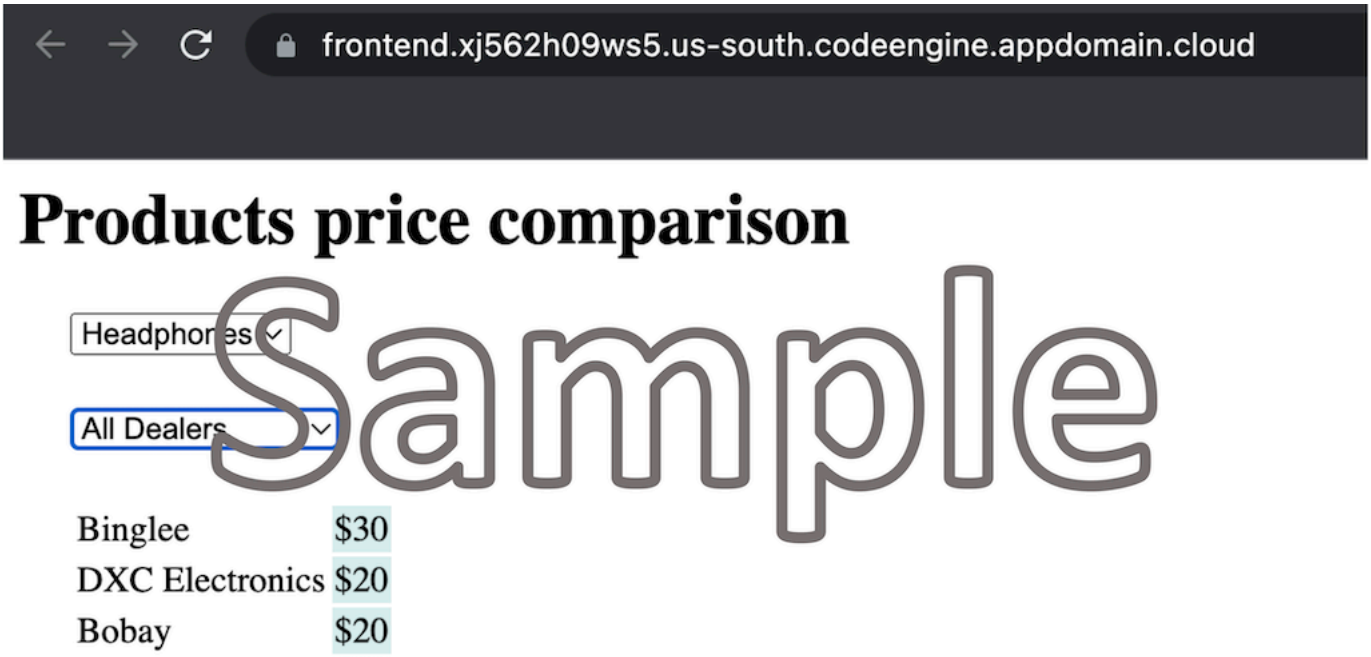
9. After the dealers dropdown populates, choose a particular dealer for the product and see if the price charged by that dealer is displayed.

   Allow 10 to 20 secs to load the page.

   Take a screenshot of the entire page showing the product chosen, dealer chosen, and the price returned by the microservice and name it `product_dealer_price.png`.

10. Choose `All Dealers` option for a product (make sure you choose a product which has more than one dealer). Pricing of all dealers offering the product should be shown on the screen.

    Take a screenshot of the entire page showing the product chosen, `All Dealers` option chosen, and the prices charged by all dealers returned by the microservice and name it `product_all_dealers_prices.png`.



Congratulations! You have completed the Final project!

Summary:

In this lab, you've successfully deployed multiple microservices to build an integrated application. This includes two backend microservices, one developed in Python and the other in Node.js, along with a front-end microservice.

Author(s)

Lavanaya T S