

Hands-on Lab: Web Application using Flask



Estimated time needed: **30** minutes

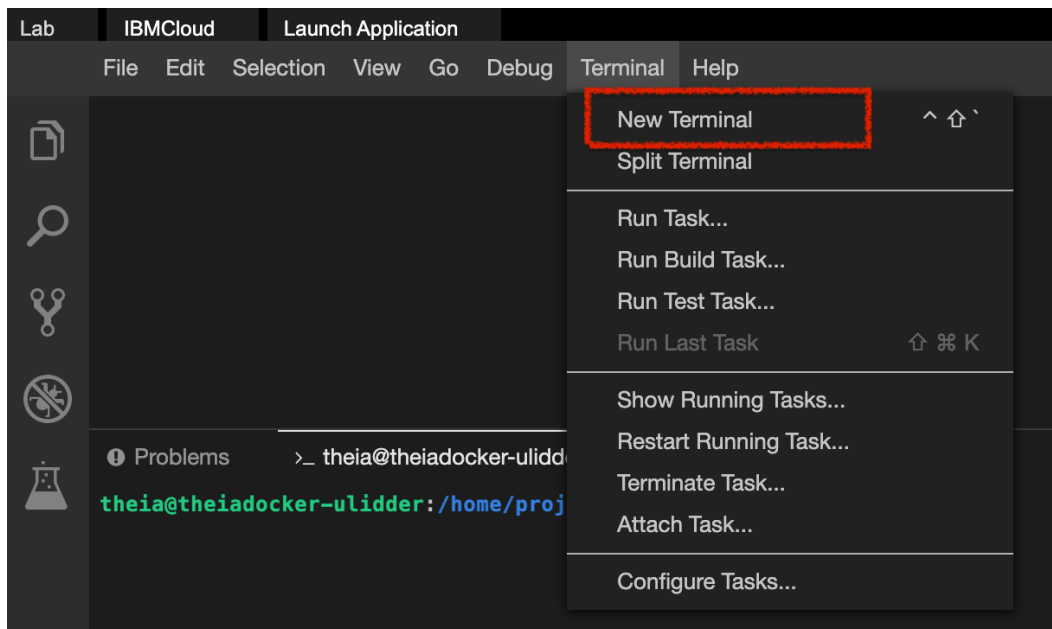
Objectives

After completing this lab you will be able to:

- Install flask
- Create API end points in python
- Create and run a web application using Flask

SetUp

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



2. Install the flask package that you will need to complete the exercises.

- ```
1. 1
1. python3 -m pip install flask
```

Copied!

3. In the terminal, change to your project folder.

- ```
1. 1
1. cd /home/project
```

Copied!

4. Clone the git repository.

- ```
1. 1
1. git clone https://github.com/ibm-developer-skills-network/rlkee-flaskproject.git
```

Copied!

5. Change to the directory named **rlkee-flaskproject/practice\_labs**.

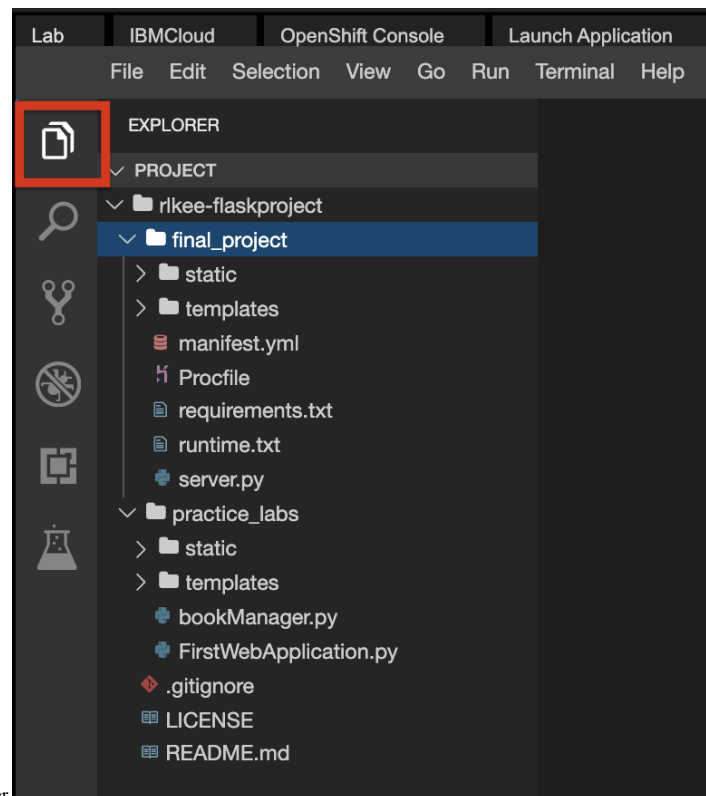
- ```
1. 1
1. cd rlkee-flaskproject/practice_labs
```

Copied!

6. List the contents of this directory to see the artifacts for this lab. You must have a few exercise files that you will be running in the steps to follow.

- ```
1. 1
1. ls
```

Copied!



7. You can also view the files cloned in the explorer.

## Exercise 1: First WebApplication

In this exercise we are just building a very primitive web application with `flask`, which serves one root endpoint `/` and returns **Hello World!** string when the endpoint is accessed. The root endpoint is when you just type the IP address and the port. For example, `http://127.0.0.1:5000` is a root endpoint.

1. In the files explorer view `FirstWebApplication.py`.

A screenshot of the OpenShift Console showing the code for `FirstWebApplication.py`. The Explorer panel on the left shows the project structure, with `FirstWebApplication.py` selected. The main editor area displays the following Python code:

```
1 from flask import Flask
2
3 app = Flask("My First Flask Application")
4
5
6 @app.route("/")
7 def hello():
8 return "Hello World!"
9
10
11 if __name__ == "__main__":
12 app.run(debug=True)
13 # When no port is specified, starts at default port 5000
```

2. In the terminal run the following command.

1. 1

1. `python3 FirstWebApplication.py`

Copied!

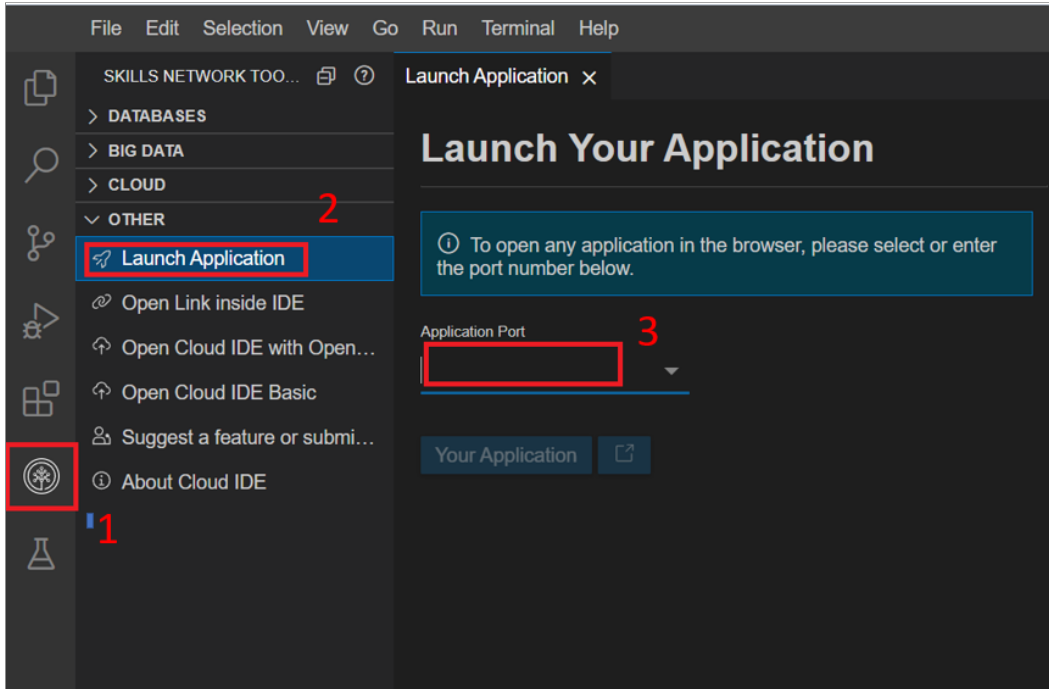
3. This will start the server. You should see the following on your screen, indicating the server is running and is waiting for requests. Note down the port number. The default port number is 5000.

```

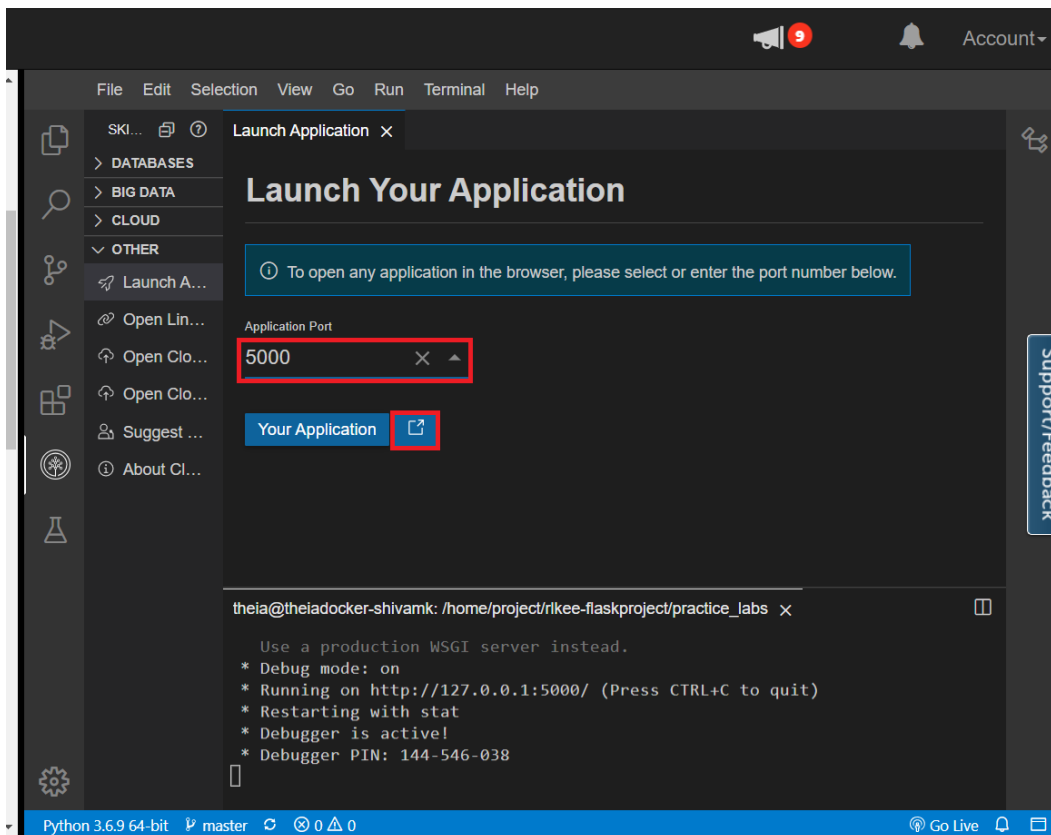
* Serving Flask app "My First Flask Application" (lazy loading)
* Environment: production
 WARNING: This is a development server. Do not use it in a production deployment.
 Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 178-152-102

```

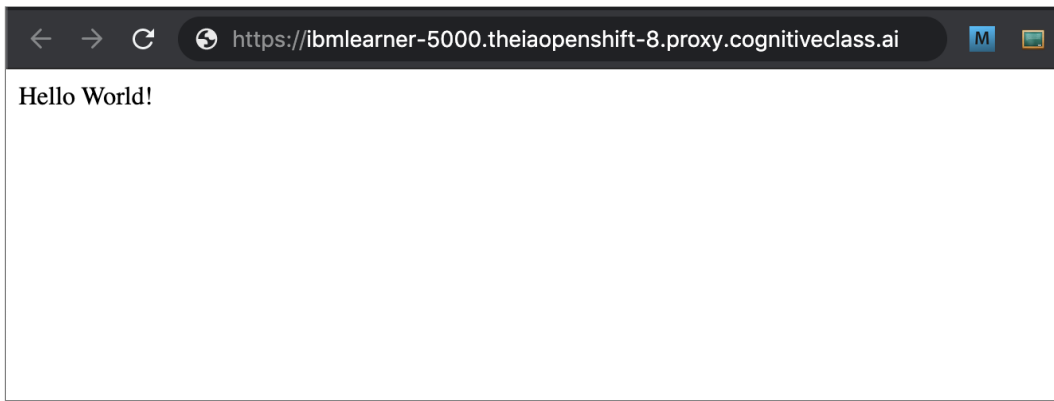
4. Click on the Skills Network button on the left, it will open the Skills Network Toolbox. Then click the Other then Launch Application. From there you should be able to enter the port.



5. Enter the port number at which the server is running and click Launch button.



6. A new browser tab opens up.



7. Close the browser tab.

8. Press Ctrl+C in the terminal to stop the server.

## Exercise 2: WebApplication with Template Rendering and static resources

In this exercise, we will go further than just a primitive web application. We will create a **Book Manager** web application.

This application will render HTML page as template, through the root endpoint /. We will also have other endpoints which will be used internally by the HTML page through Javascript.

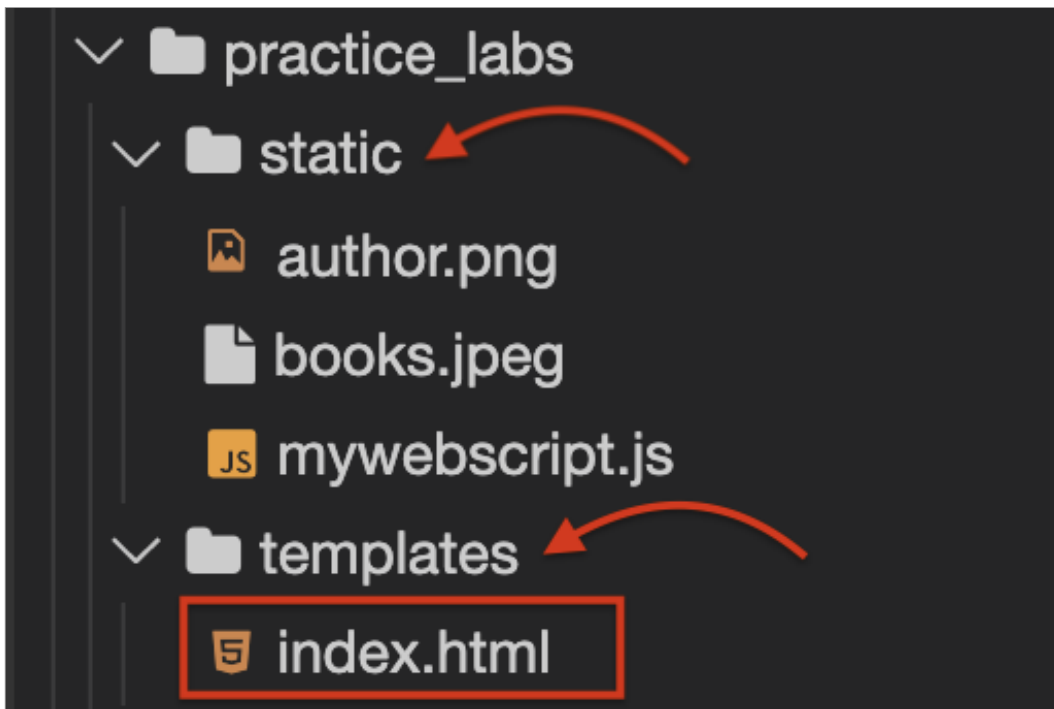
1. In the terminal, check if you are in the practice\_labs directory. If not change to that directory with the following command.

1. 1

1. `cd /home/project/r1kee-flaskproject/practice_labs`

Copied!

2. Analyze the code in bookManager.py in the explorer. See the end points that are being served. Explore the templates and the static folder. Observe, how the code in index.html uses the templates and the static files.



3. In the terminal run the following command, to start the application server.

1. 1

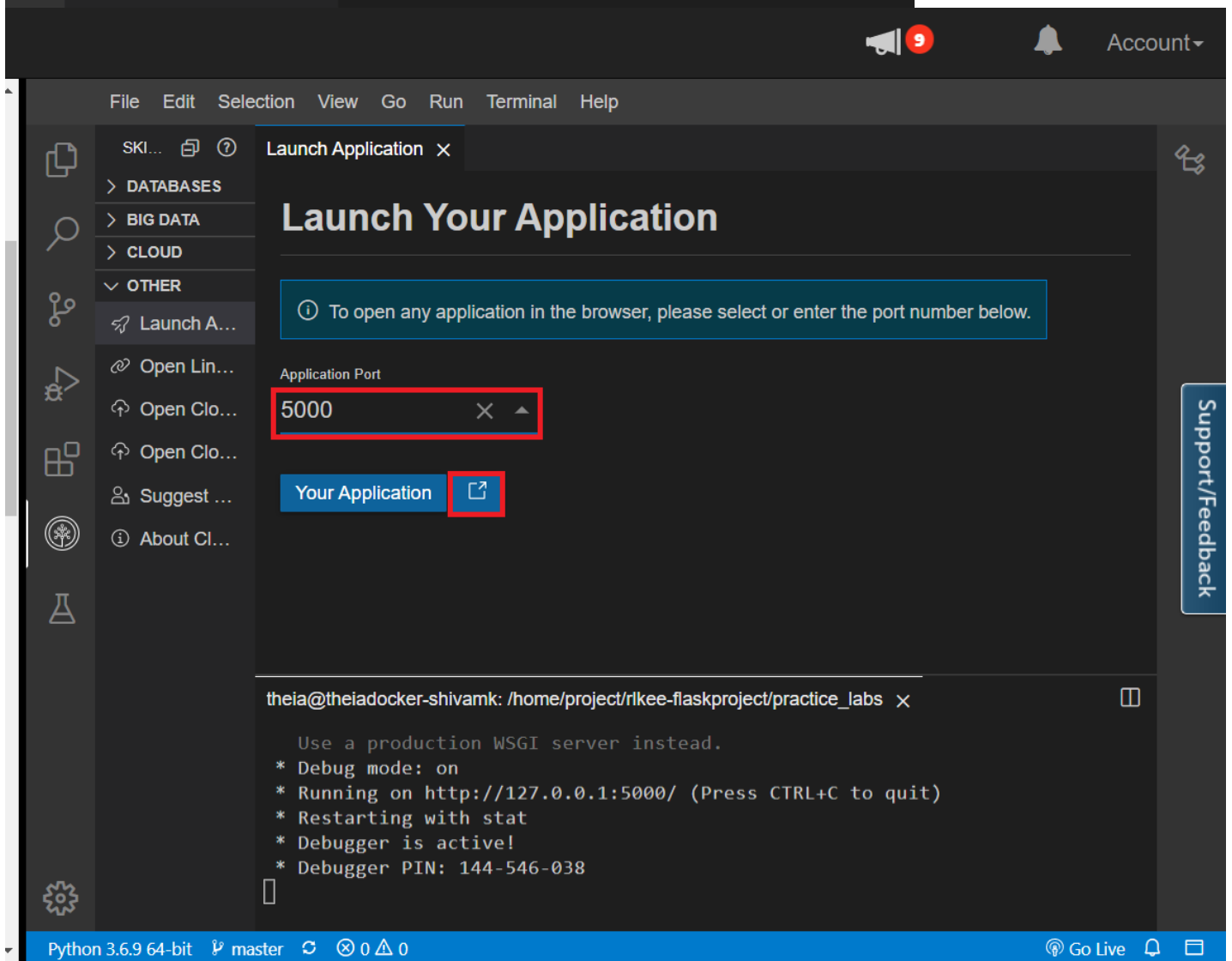
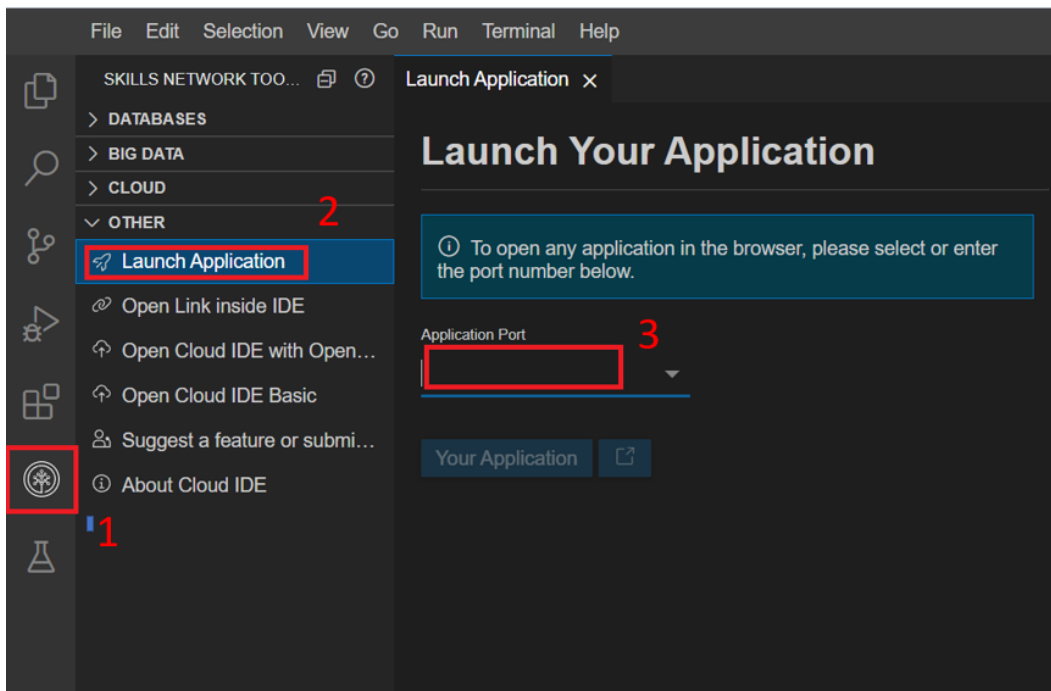
1. `python3 bookManager.py`

Copied!

4. You will see the server running on port 5000, the default port.

```
theia@theiaopenshift-lavanyas:/home/project/rlkee-flaskproject/practice_labs$ python bookManager.py
* Serving Flask app "Book Manager" (lazy loading)
* Environment: production
 WARNING: This is a development server. Do not use it in a production deployment.
 Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 178-152-102
```

5. Click on **launch application** and enter the port number 5000 just as you did in the previous exercise.



6. Enter the port number 5000 and click **OK** to launch the browser tab.

← → ↻ 🌐 https://ibmlearner-5000.theiaopenshift-8.proxy.cognitiveclass.ai Update ⋮

📖 Book Name

👤 Author

Add Book

7. Enter *Book* and *Author* and click on **Add Book** and see the list of books added.

Congratulations! You have completed this lab section. You should now be familiar with creating Web Application with Flask.

## Authors

Lavanya

## Change Log

| Date (YYYY-MM-DD) Version Changed By |     |         | Change Description                                                   |
|--------------------------------------|-----|---------|----------------------------------------------------------------------|
| 2021-05-14                           | 0.1 | Lavanya | Created initial version of the lab                                   |
| 2022-09-01                           | 0.2 | Ratima  | Updated screenshot and step of Lauch Application                     |
| 2022-10-21                           | 0.3 | Ratima  | Updated steps of Lauch Application and Skill Network Logo screenshot |

Copyright © 2020 IBM Corporation. All rights reserved.