Part D: Deploy an Application to Cloud Foundry

Estimated time needed: 30 minutes

Lab Overview

In this hands-on lab, you will deploy a "Hello world" Node.js appliation to IBM Cloud using Cloud Foundry and the CLI (Command Line Interface).

Objectives

After completing this lab, you will be able to:

- 1. Launch the IBM Cloud Shell
- 2. Explain how to use the IBM Cloud CLI to deploy applications to Cloud Foundry

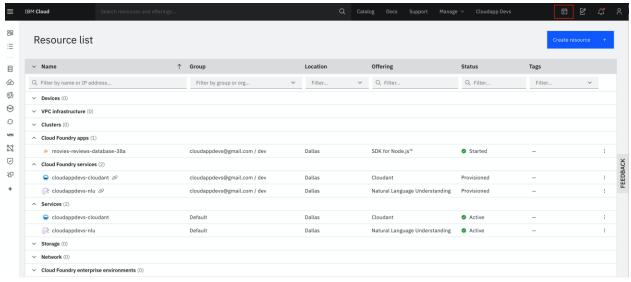
Lab Instructions

This lab is broken up into the following tasks:

- Part D: Deploy an Application to Cloud Foundry
 - Lab Overview
 - o **Objectives**
 - Lab Instructions
 - 1. Launch the IBM Cloud Shell
 - Fork the repository and clone in the shell
 - 3. Change application name in the manifest file
 - 5. Deploy the application
 - 5. Access the web app
 - Author(s)
 - Changelog

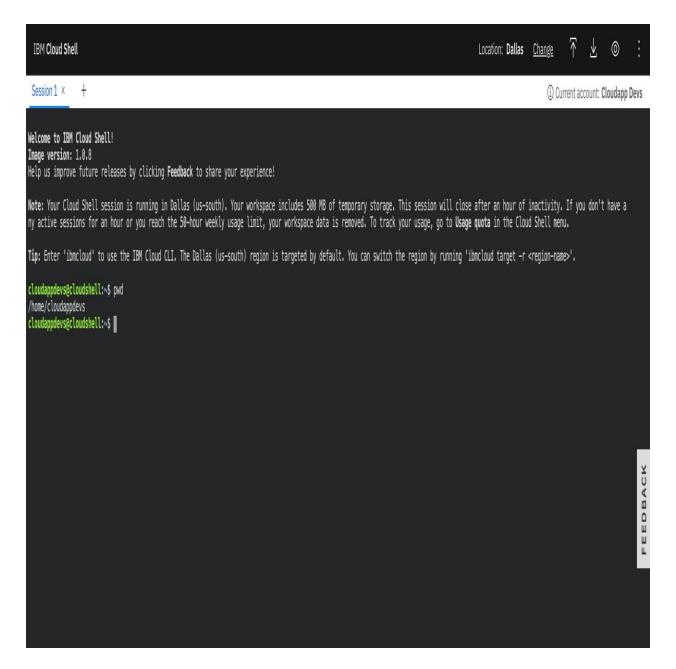
1. Launch the IBM Cloud Shell

For this next step we'll be using the *IBM Cloud Shell*, which is available by clicking on the terminal icon on the top right.



{ width=1024 height=1024 }

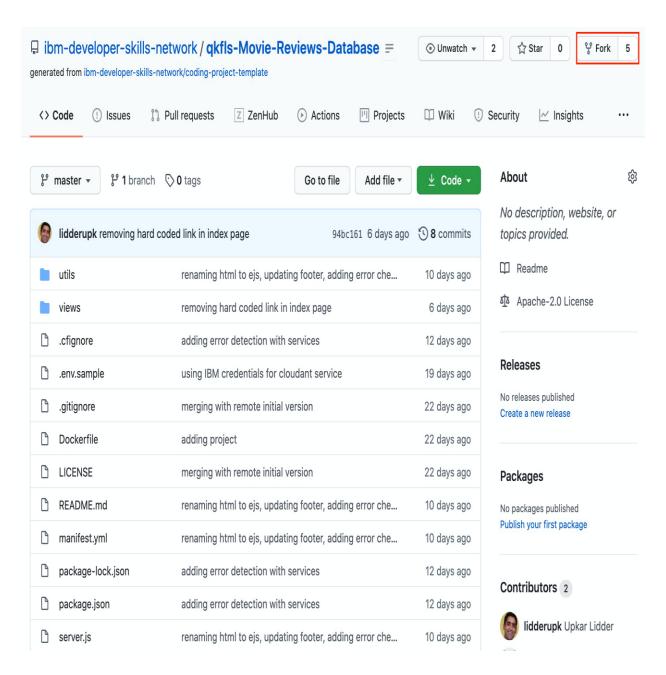
You should see a terminal



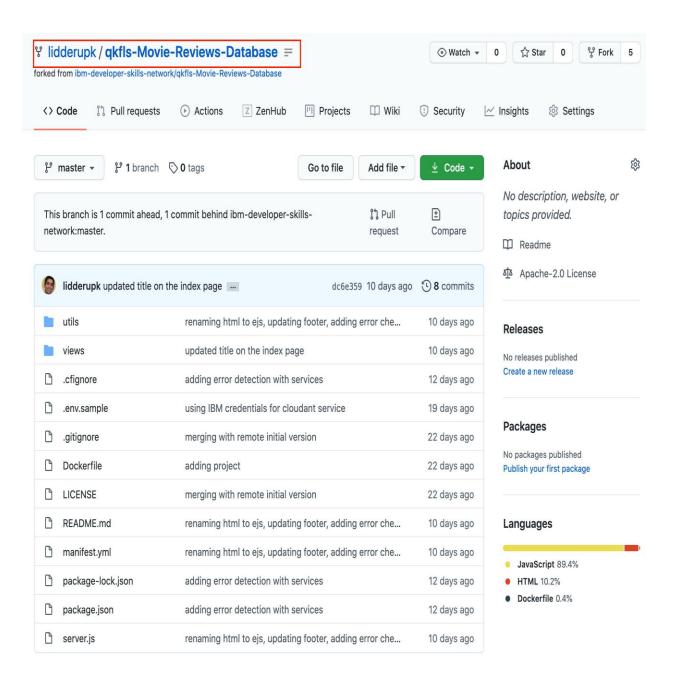
Fork the repository and clone in the shell

Open the Movies Reviews Database Repository available at

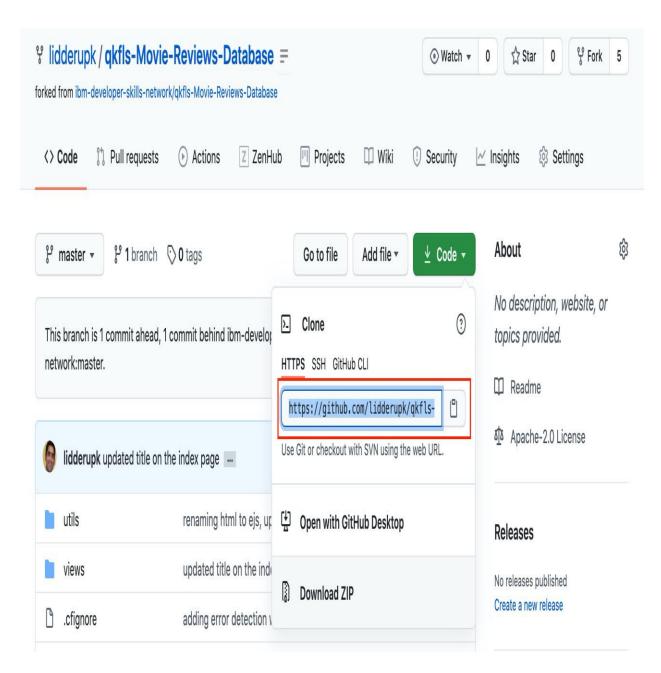
https://github.com/ibm-developer-skills-network/qkfls-Movie-Reviews-Database



Once the repository is forked successfully, you will be taken to your copy under your github account.



Next, clone the forked code repository that has the source code we want to deploy. Copy the following command into the terminal. You can get the https URL of the repository as shown here:



In my case, the command looks as follows:

git clone https://github.com/lidderupk/qkfls-Movie-Reviews-Database.git

Welcome to IBM Cloud Shell!

Image version: 1.0.9

Help us improve future releases by clicking Feedback to share your experience!

Note: Your Cloud Shell session is running in Dallas (us-south). Your workspace includes 500 MB of temporary storage. This session will c lose after an hour of inactivity. If you don't have any active sessions for an hour or you reach the 50-hour weekly usage limit, your wo rkspace data is removed. To track your usage, go to Usage quota in the Cloud Shell menu.

Tip: Enter 'ibmcloud' to use the IBM Cloud CLI. The Dallas (us-south) region is targeted by default. You can switch the region by runnin g 'ibmcloud target -r <region-name>'.

ulidder@cloudshell:~\$ git clone https://github.com/lidderupk/qkfls-Movie-Reviews-Database.git
Cloning into 'qkfls-Movie-Reviews-Database.git

remote: Enumerating objects: 73, done.
remote: Counting objects: 100% (73/73), done.
remote: Compressing objects: 100% (47/47), done.

remote: Total 73 (delta 23), reused 62 (delta 16), pack-reused 0

Unpacking objects: 100% (73/73), done.

ulidder@cloudshell:~\$

Go into the directory of the cloned repository

cd qkfls-Movie-Reviews-Database

3. Change application name in the manifest file

Let's modify the source code to produce your changes. Use the Nano text editor to open the manifest.yml file.

nano manifest.yml



Figure 1-13 Nano text editor

1. Using the cursor keys to navigate, change the application name from **movies-reviews-database** to the name of your application **movies-reviews-database- xxx**.



Figure 1-14 Editing text

2. Press CTRL+X to save the file, press Y to save the modified buffer.



Figure 1-15 Saving the file

3. Verify that the **File Name to Write** is **manifest.yml**, and then press Enter.



Figure 1-15 verify the file

You can confirm that the manifest file has the right application name by writing it to the terminal using the cat command.



5. Deploy the application

To deploy the application with Cloud Foundry we first we need to target a Cloud Foundry API endpoint. To do this, run the following interactive command.

ibmcloud target --cf

You should see output like the example below:

Getting app info...

Updating app with these attributes...

name: movies-reviews-database-38a

path: /home/cloudappdevs/qkfls-Movie-Reviews-Database

command: npm start

disk quota: 1G
health check type: port

instances: 1- memory: 64M

+ memory: 128M

stack: cflinuxfs3

services:

cloudappdevs-cloudant
cloudappdevs-nlu

routes:

movies-reviews-database-38a.us-south.cf.appdomain.cloud

Updating app movies-reviews-database-38a...

Mapping routes...

Comparing local files to remote cache...

Packaging files to upload...

. . .

Waiting **for** app to start...

name: movies-reviews-database-38a

requested state: started

routes: movies-reviews-database-38a.us-south.cf.appdomain.cloud

last uploaded: Mon 26 Oct 19:31:17 UTC 2020

stack: cflinuxfs3
buildpacks: sdk-for-nodejs

type: web
instances: 1/1
memory usage: 128M
start command: npm start

state since cpu memory disk details

#0 running 2020-10-26T19:31:41Z 0.0% 40K of 128M 8K of 1G

```
Location: Dallas
  IBM Cloud Shell
                                                                                                         Change
                                                                                                                                       0
 Session 1 X
                                                                                                               (i) Current account: Cloudapp Devs
cloudappdevs@cloudshell:~/qkfls-Movie-Reviews-Database$ ibmcloud app push
Invoking 'cf push'...
Pushing from manifest to org cloudappdevs@gmail.com / space dev as cloudappdevs@gmail.com...
Using manifest file /home/cloudappdevs/qkfls-Movie-Reviews-Database/manifest.yml
Getting app info...
Updating app with these attributes...
                           movies-reviews-database-38a
  name:
  path:
                           /home/cloudappdevs/qkfls-Movie-Reviews-Database
  command:
                           npm start
  disk quota:
                           1Ġ
  health check type:
                           port
  instances:
                           cflinuxfs3
  stack:
  services:
     cloudappdevs-cloudant
     cloudappdevs-nlu
                                                                                                                                                  FEEDBACK
     movies-reviews-database-38a.us-south.cf.appdomain.cloud
Updating app movies-reviews-database-38a...
Mapping routes...
Comparing local files to remote cache...
Packaging files to upload...
Uploading files...
31.21 KiB / 31.21 KiB [===
                                                                                                                             =] 100.00% 1s
Waiting for API to complete processing files...
Stopping app...
Staging app and tracing logs...
   Downloading liberty-for-java_v3_50-20201019-1521...
   Downloading dotnet-core...
   Downloading sdk-for-nodejs...
   Downloading swift_buildpack...
Downloaded dotnet-core
   Downloading noop-buildpack...
   Downloaded swift_buildpack
   Downloading xpages_buildpack...
Downloaded sdk-for-nodejs
   Downloading swift_buildpack_v2_0_18-20190303-1915...
Downloaded liberty-for-java_v3_50-20201019-1521
   Downloading swift_buildpack_v2_0_20-20190401-2122...
   Downloaded noop-buildpack
   Downloading staticfile_buildpack...
```

5. Access the web app

Navigate to the URL that is show in the routes value, in the example above it is:

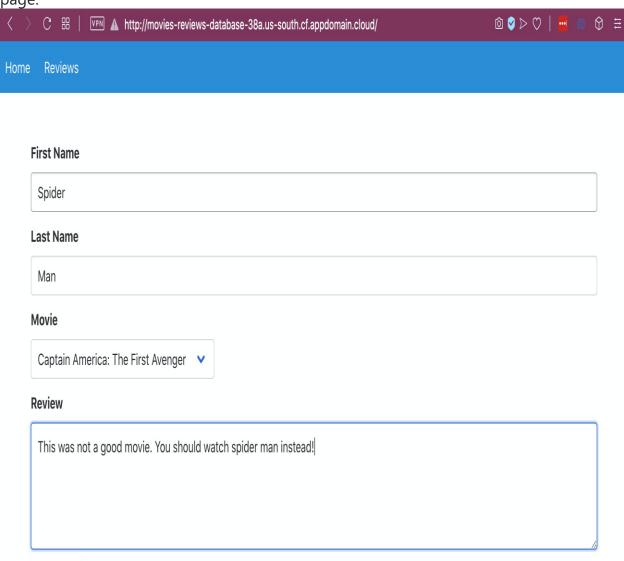
movies-reviews-database-38a.us-south.cf.appdomain.cloud

If all goes well - you should see a website with two menu items. **Home** should have a form that lets the user enter a movie review. **Reviews** should show the reviews that users have already entered. If the services were not created or connected to the application, you will see some error messages on both pages. If everything was connected properly, the pages will look as follows:

Home

page:

Submit



Reviews



Current Reviews



Congratulations! We hope you've deployed an application to Cloud Foundry on IBM Cloud using the command line interface.