Implementing CI/CD

olsen software

Overview

- Jenkins is a build automation tool, widely used in Citi
  - You can create a Jenkins "build" to automate s/w builds
  - Helps you do Continuous Integration / Continuous Delivery

Contents

- Creating a Jenkins Freestyle project
- Using Jenkins with OpenShift

The Environment

- We've created a separate Linux VM for each student, with Jenkins and OpenShift installed
  - Start your Linux VM via http://student.conygre.com

https://student.conygre.com

Student Machine Manager

Find a Machine Connection Instructions

Using this application you can turn your classroom machine on and off and also

Machine names are like linuxops-dallas1.conygre.com

Step 1: Find your machine

Region:	Machine name	Domain
n-virginia	singaporedevops1	.conygre.com
Where is your virtual machine? Your instructor should have told you. If you are unsure, select **Don't Know**.	Enter the first part of the name of the machine you have been using for the training. So if your machine is **training1.conygre.com**, enter **training1**	The top level domain for your machine.

Find my Machine

Step 2: Turn on Your Machine

We have found your machine! If you plan to use it now, you can turn it on. Use the refresh button once you have turned it on to obtain the connection instructions.

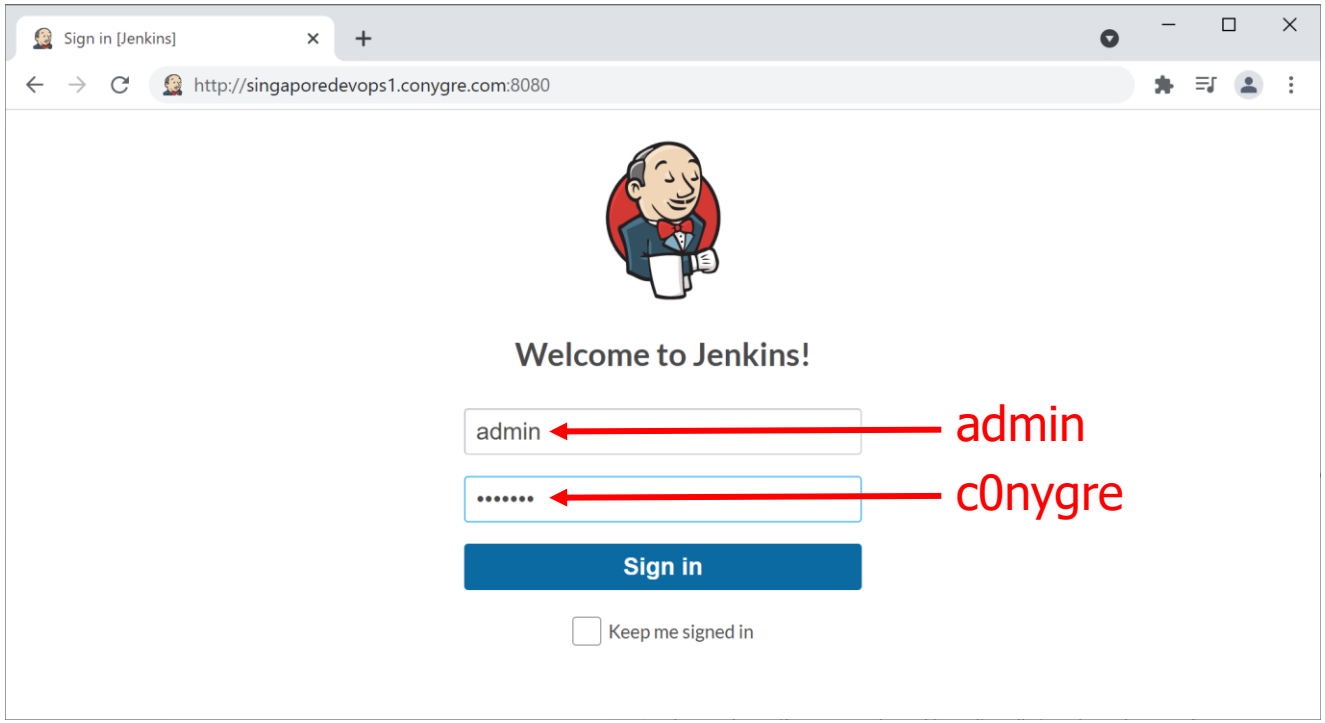
Machine Name/IP	Status	Actions	Available hours (UTC)	Refresh
singaporedevops1.conygre.com 100.25.185.16	running	Stop	From -1 to 16	Refresh Status

Creating a Jenkins Freestyle Project

- Displaying the Jenkins Dashboard
- Creating a Freestyle Project
- Running the build
- Viewing the build log

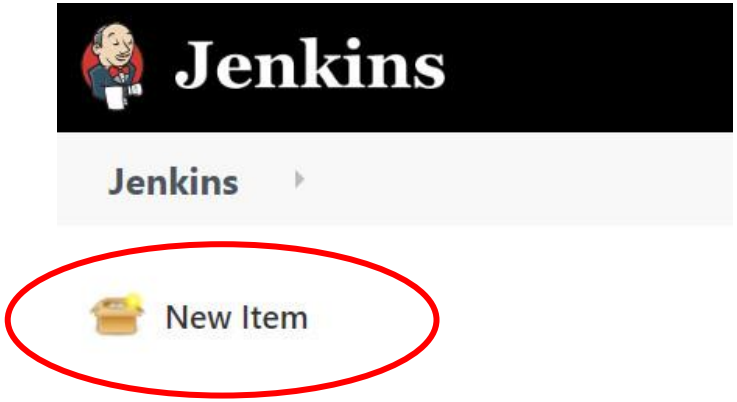
Displaying the Jenkins Dashboard

- We're going to see how to create a Freestyle project in Jenkins on the Linux VM
  - The first step is to open the Jenkins Dashboard on the Linux VM
  - To do this, browse to http://linuxops-dallasX.conygre.com:8080



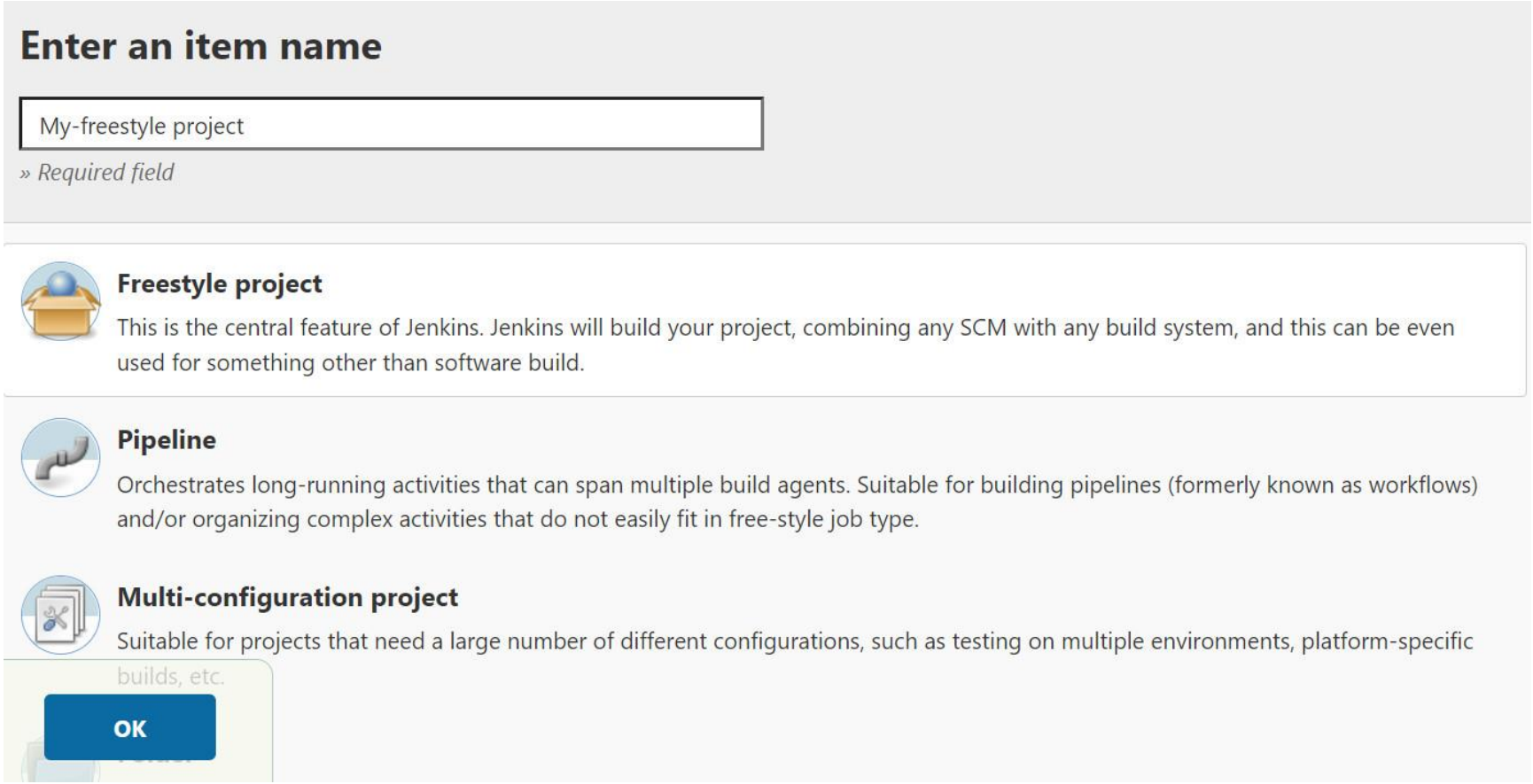
Creating a Freestyle project

- In the Jenkins Dashboard click **New Item**



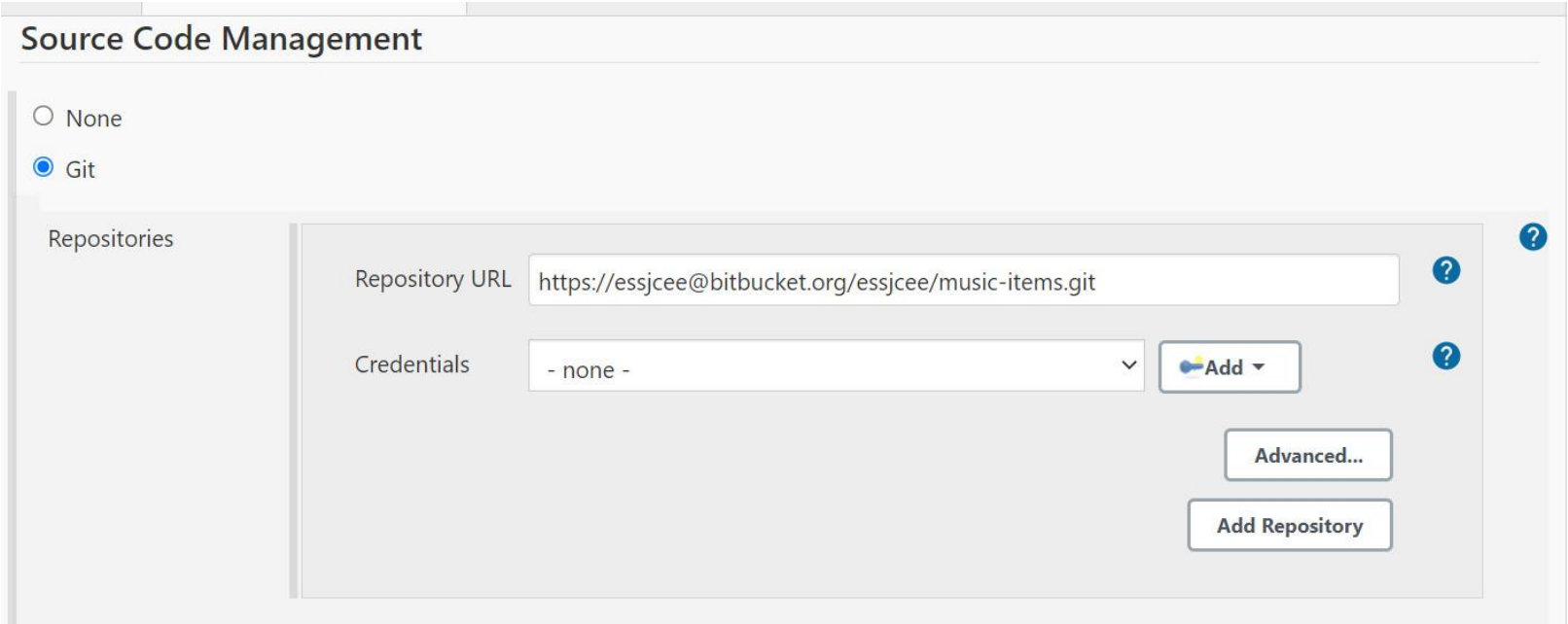
Creating a Freestyle Project

- On the next page give the project a name
- Select **Freestyle project** and click **OK**



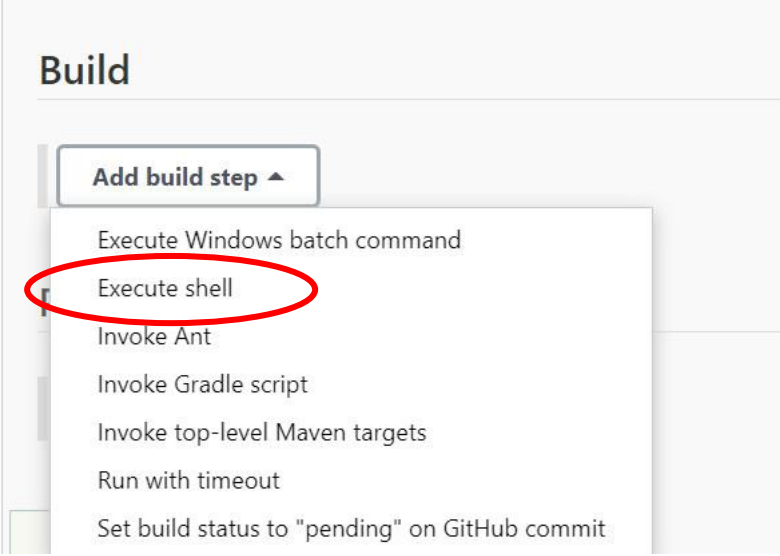
Creating a Freestyle Project

- On the next page under **Source Code Management** select **Git** and enter this repository url:
  - https://essjcee@bitbucket.org/essjcee/music-items.git
  - You don't need any credentials
  - After doing this scroll down to the **Build** section



Creating a Freestyle Project

- In the **Build** section select **Execute shell** from the **Add builds step** dropdown



Creating a Freestyle Project

- Paste the provided script into the **Execute shell Command** text area and click **Save**

Build

Execute shell

Command

```
docker stop musicapp || echo 'App not running'
docker container rm musicapp || echo 'No container to remove'
docker image rm music/app:latest || echo 'No image to remove'
chmod a+x mvnw
./mvnw package
docker build -t music/app .
docker run -d -p 8081:8080 --name musicapp music/app
```

See the list of available environment variables

Add build step

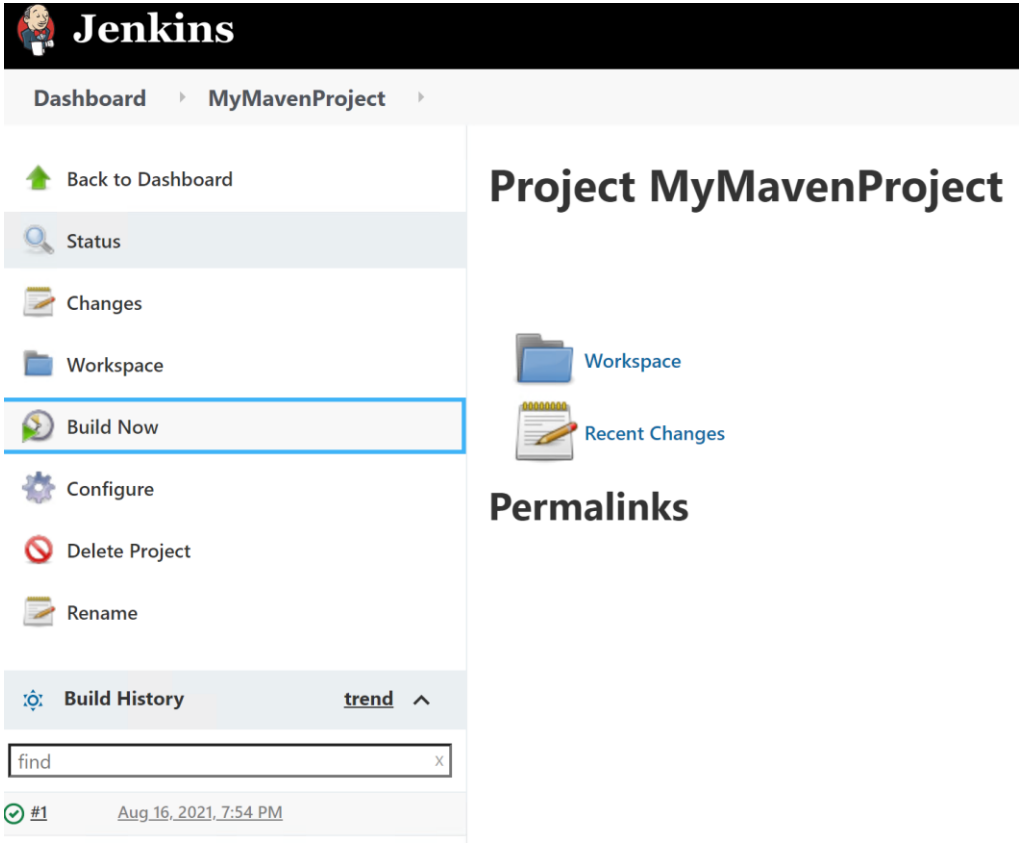
Post-build Actions

Add post-build action

Save Apply

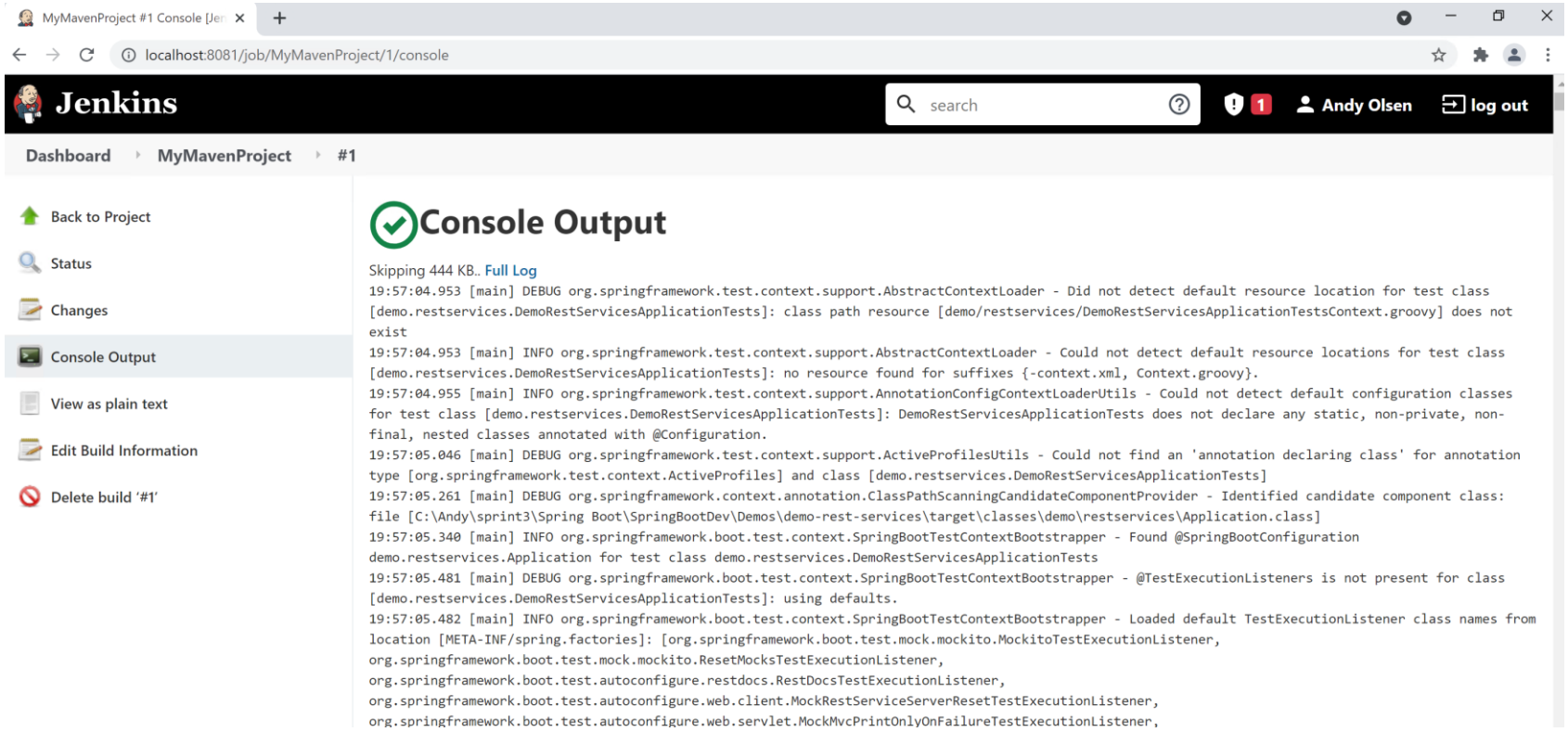
Running the Build

- In the Project page click **Build Now**
- The build indicator will go blue or red, indicating whether the build succeeded or failed



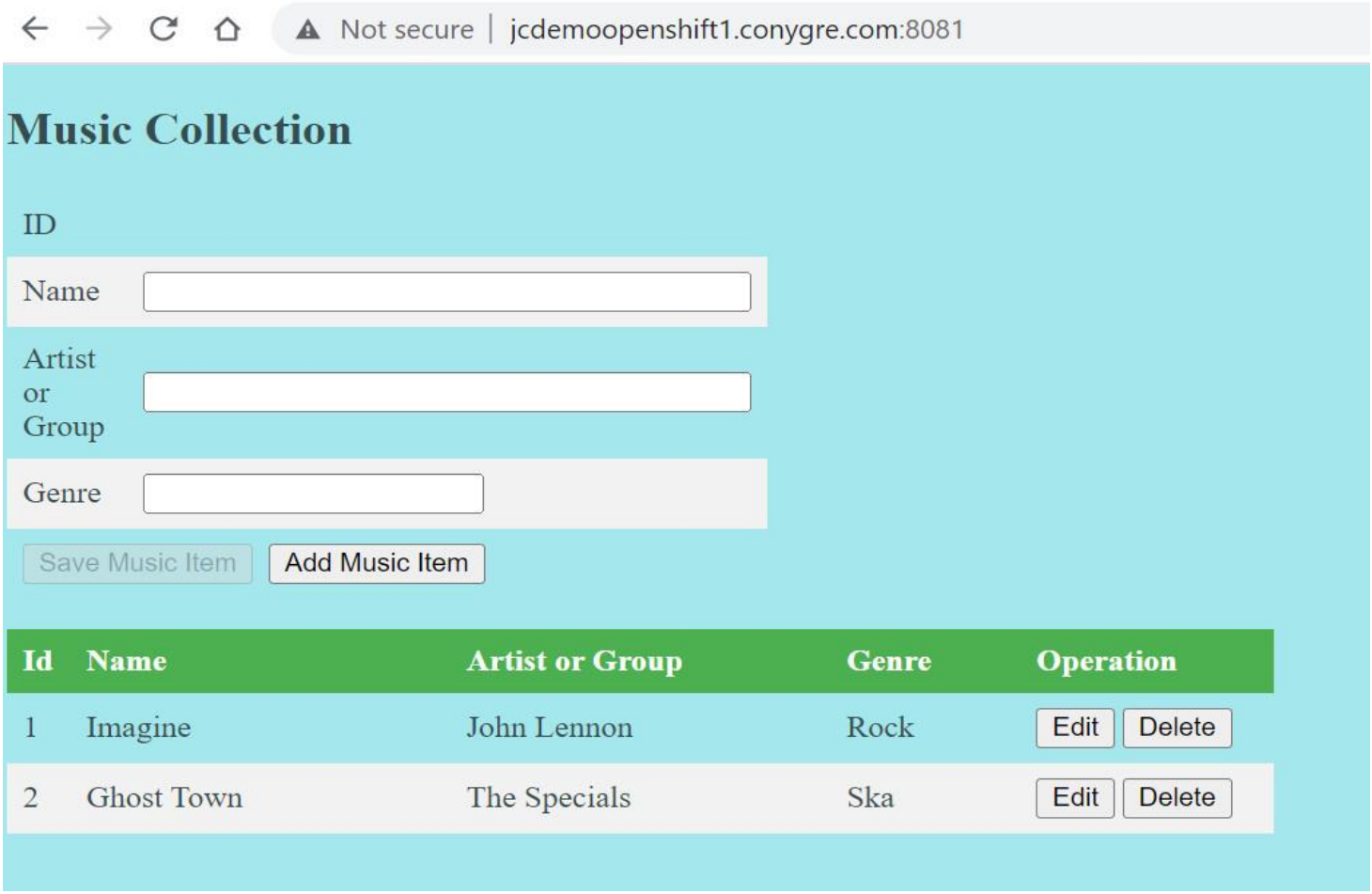
Viewing the Build Log

- To view the build log to see what happened:
  - Click the #1 link in the Build History section
  - In the next screen, click Console Output



View the deployed containerized Spring Boot Application

- Browse to port 8081 on your VM to see the deployed Spring Boot Application



Using Jenkins with OpenShift

- Overview
- The environment
- Connecting to the Linux VM
- Displaying the Jenkins Dashboard
- Creating a pipeline build
- Running a build
- Verifying the deployed app in OpenShift
- Pinging the app via the exposed Route

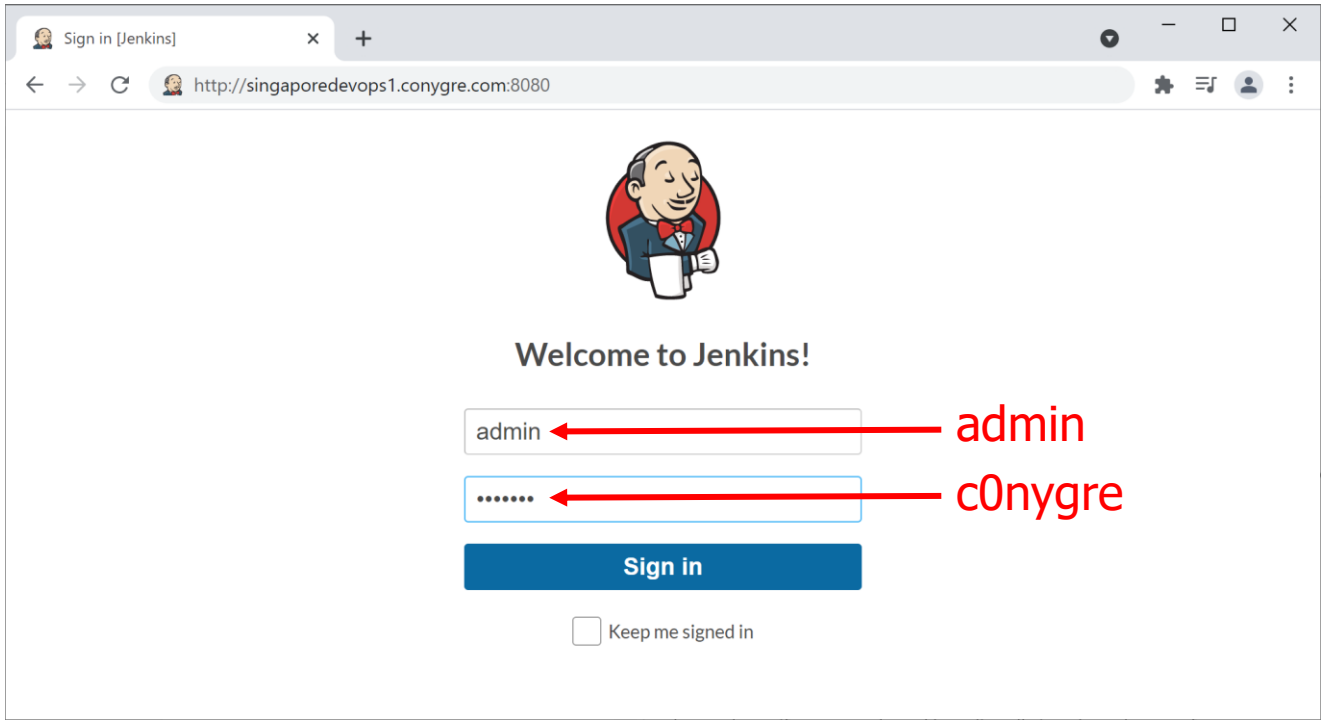
Overview

- In this section we'll show how to use Jenkins and OpenShift together

- We'll show how to write a Jenkinsfile (Jenkins build script) that defines a pipeline as follows:
  1. Build a Docker image for our app
  2. Deploy and run the containerized app in OpenShift

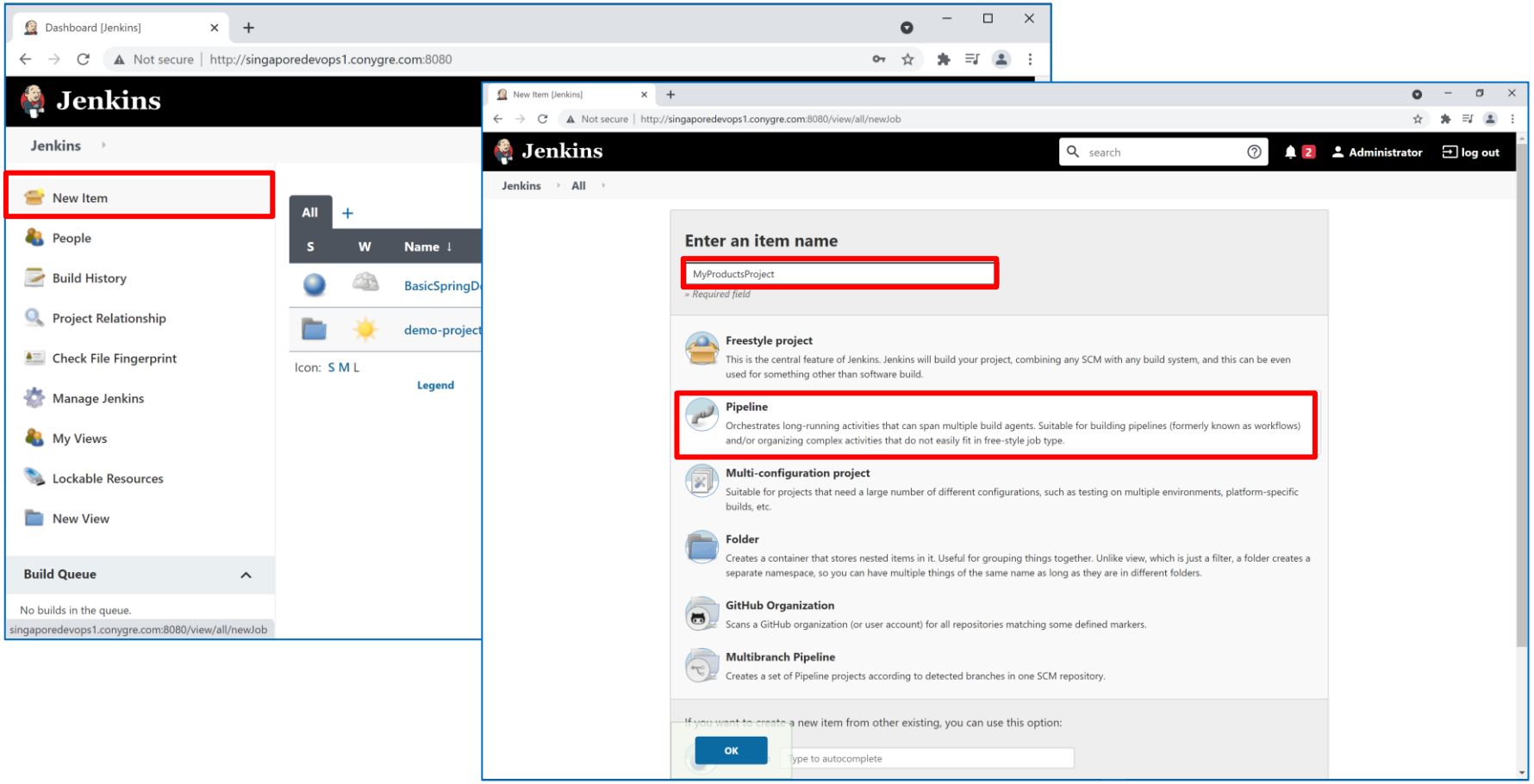
Displaying the Jenkins Dashboard

- We're going to see how to create a "build pipeline" in Jenkins on the Linux VM
  - The first step is to open the Jenkins Dashboard on the Linux VM
  - To do this, browse to http://linuxops-dallasX.conygre.com:8080



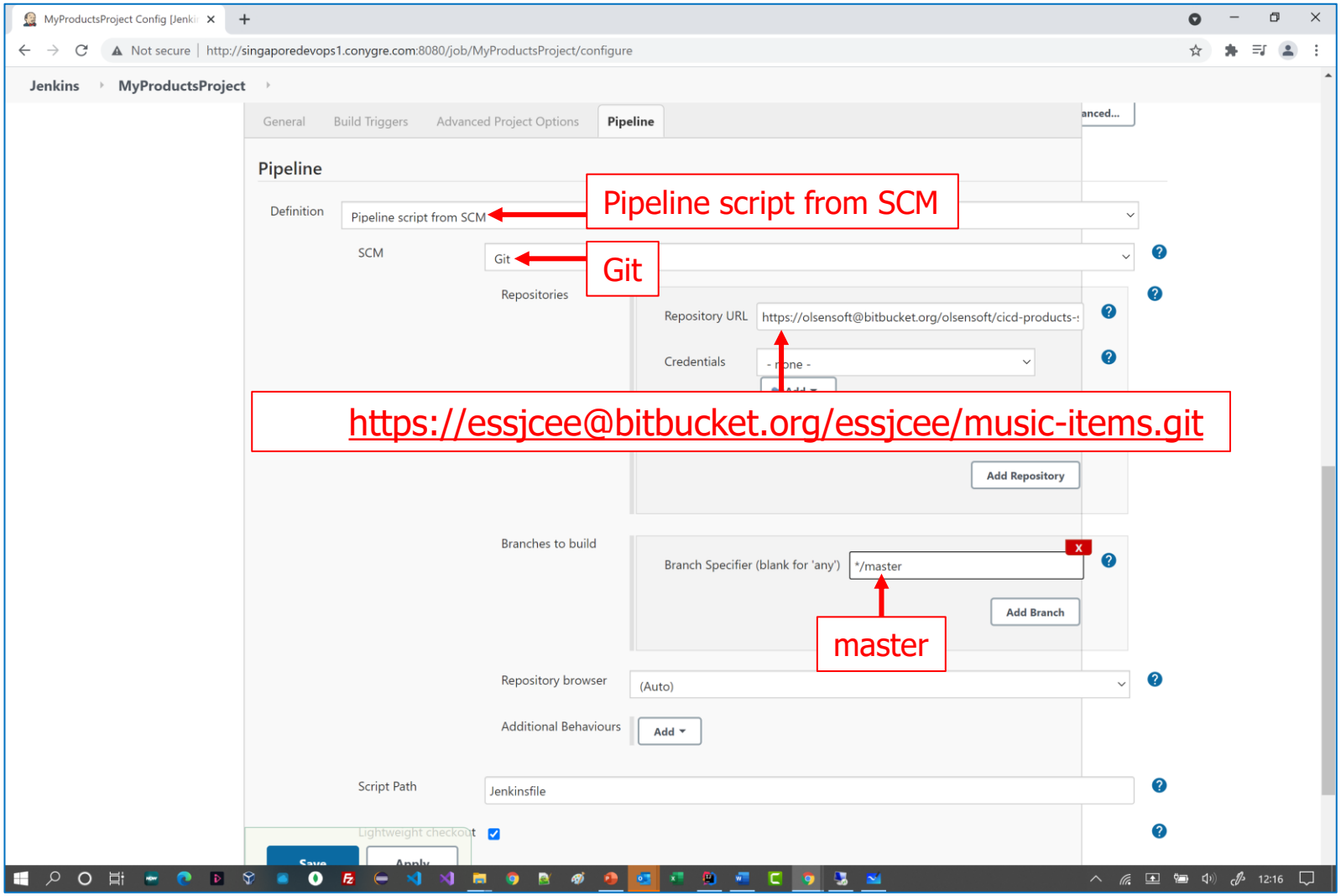
Creating a Pipeline Build (1 of 3)

- Create a new "pipeline build" as follows:
  - In the Jenkins Dashboard, click New Item
  - Enter a project name, click Pipeline, then click OK



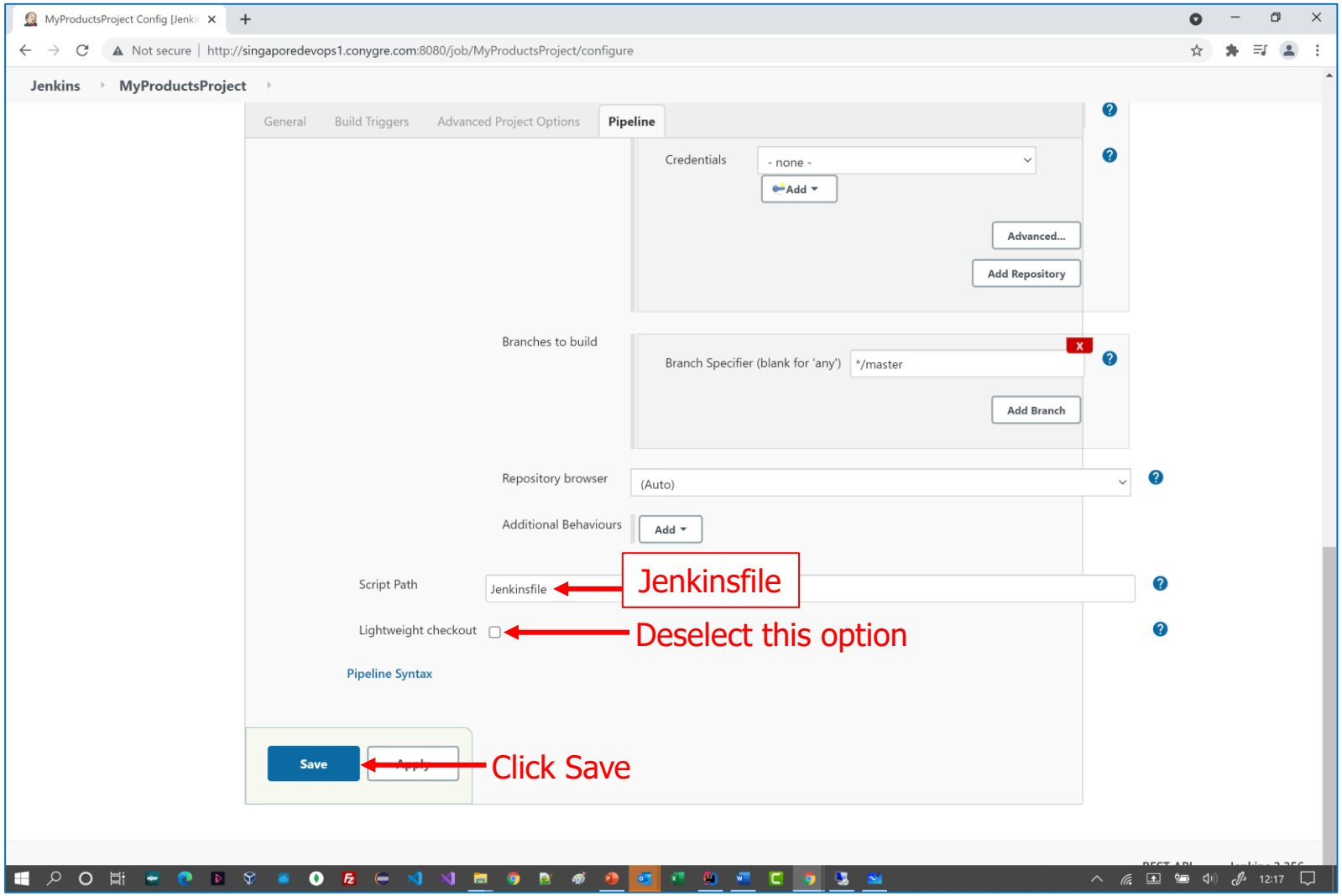
Creating a Pipeline Build (2 of 3)

- Scroll down to Pipeline, then enter Destination info:



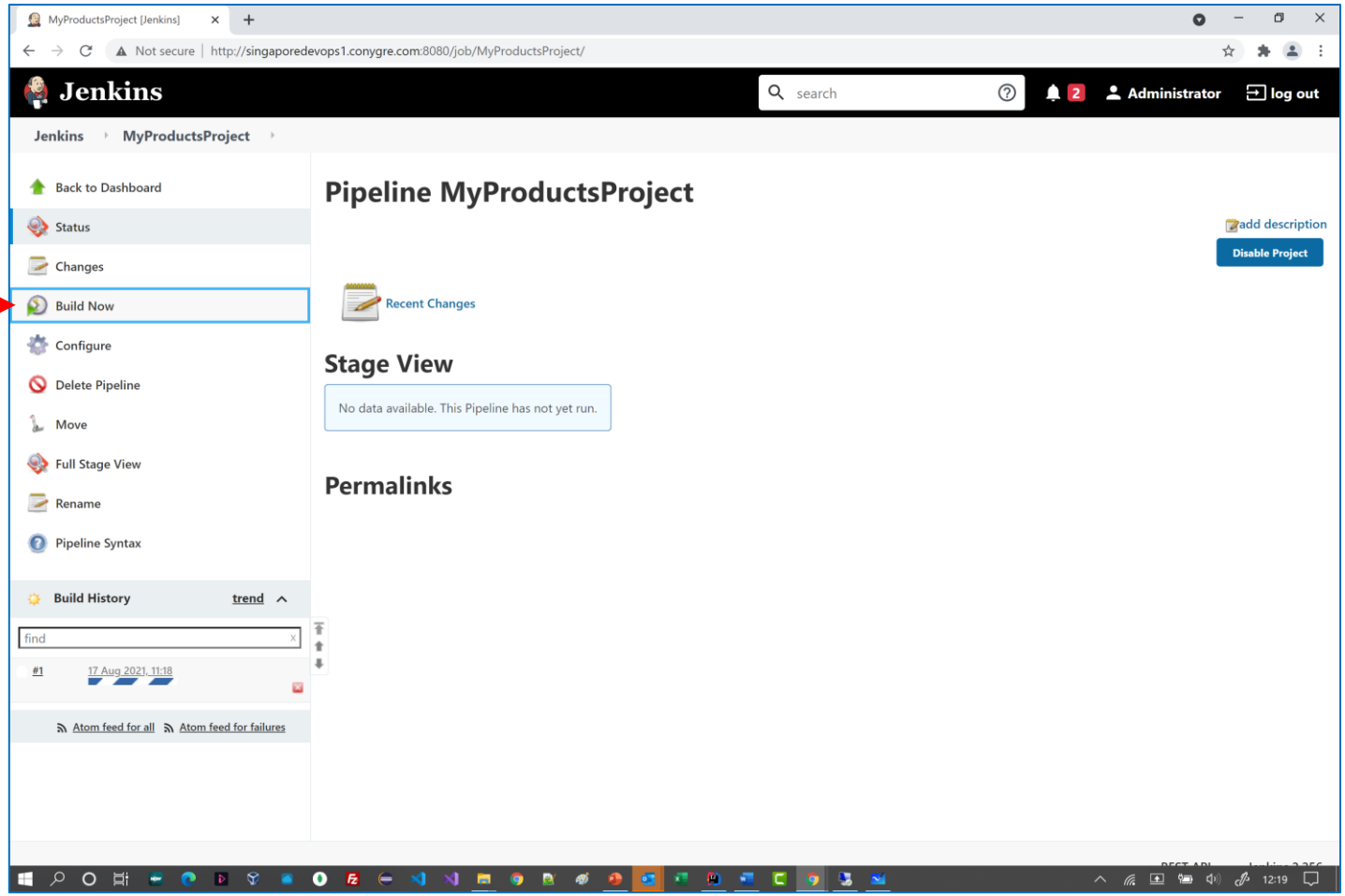
Creating a Pipeline Build (3 of 3)

- Also enter the following additional details:



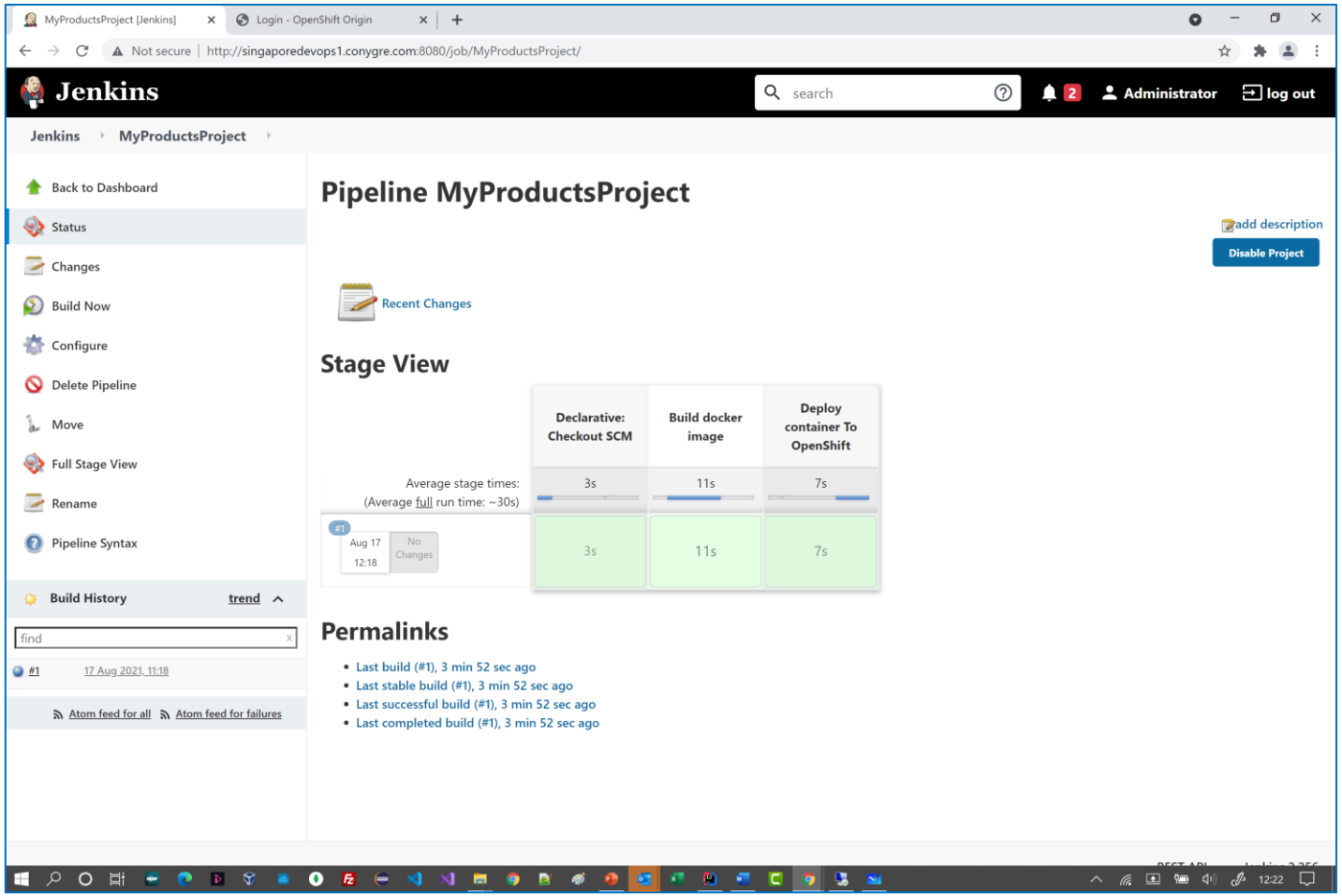
Running a Build (1 of 2)

- Back in the Dashboard it shows your pipeline
  - In the menu on the left, click Build Now



Running a Build (2 of 2)

- You'll see a series of green squares
  - Indicating it has completed the all the stages of the Jenkins build

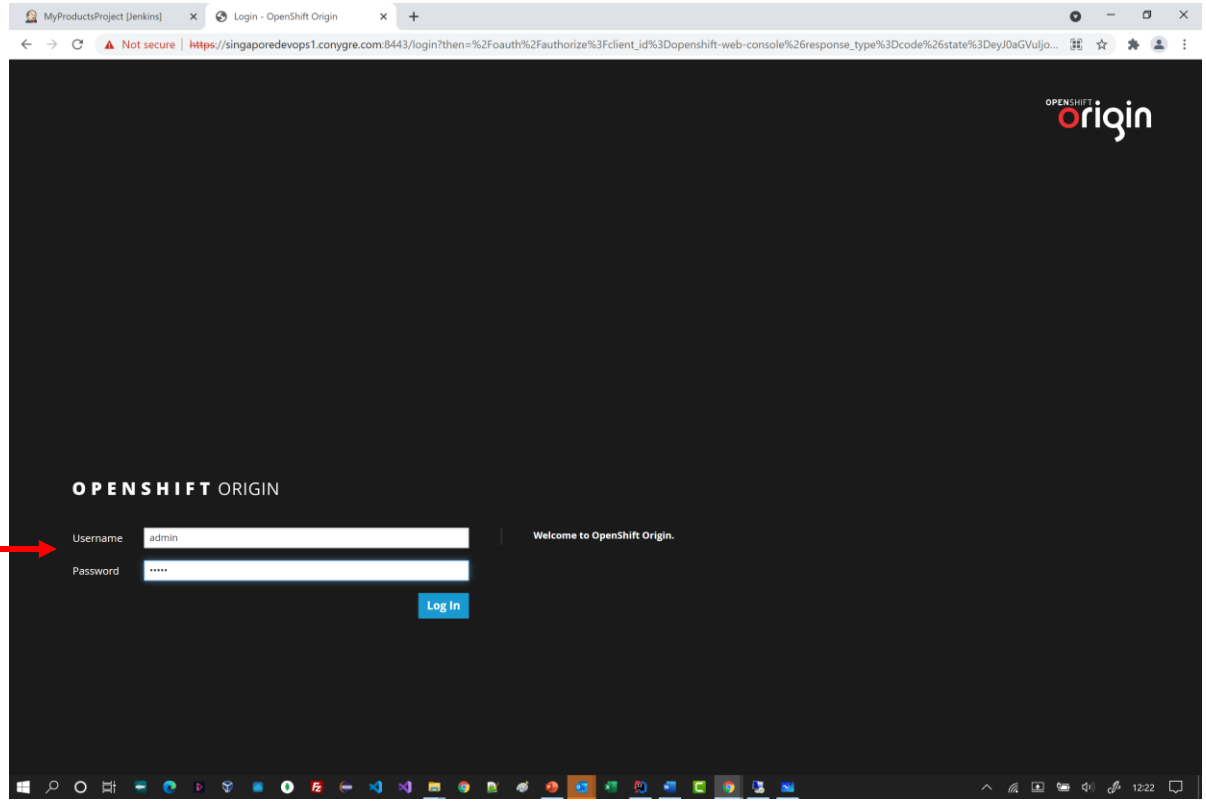


Verifying the Deployed App in OpenShift (1 of 3)

- The Jenkins build has deployed the app to OpenShift

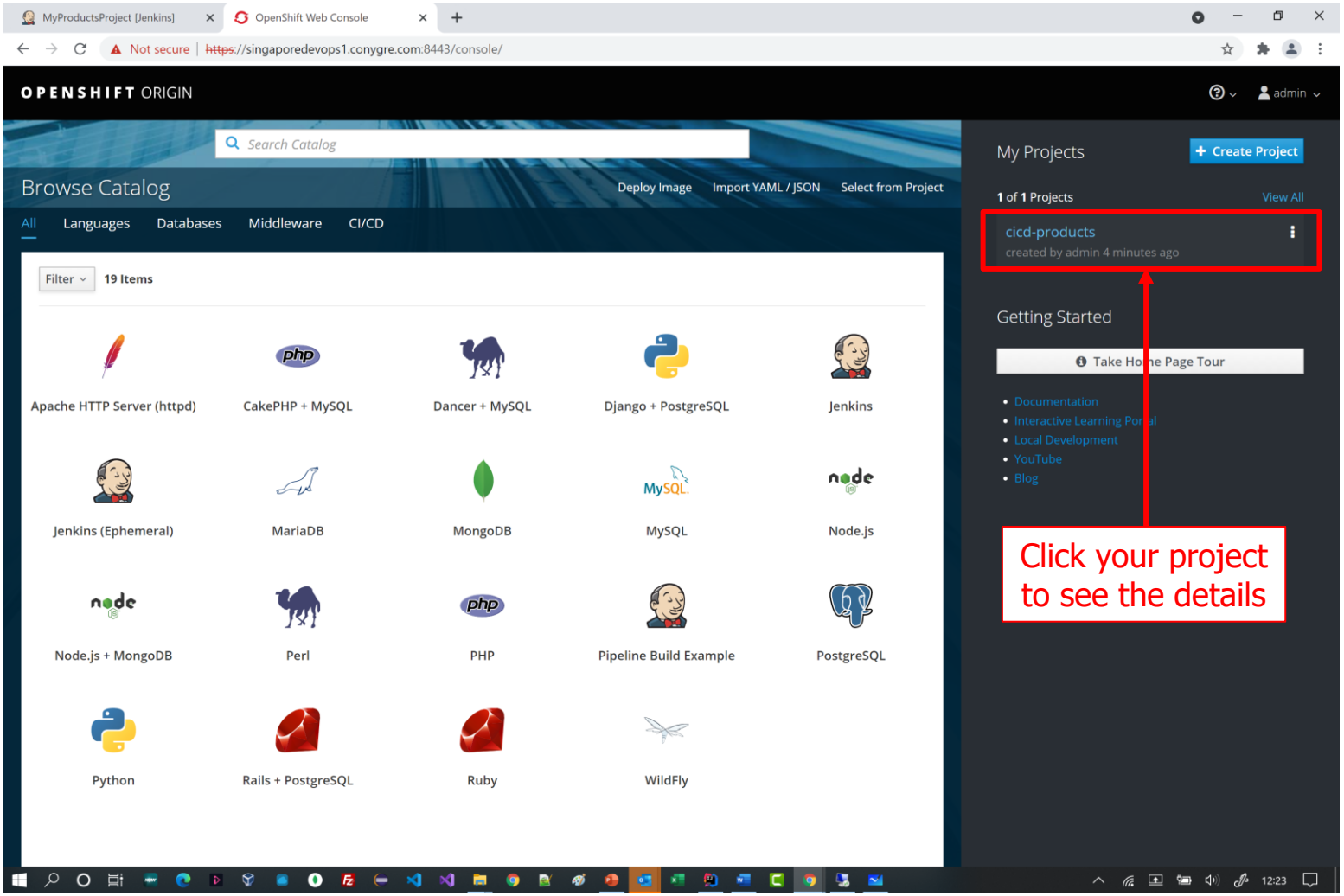
- To verify this, browse to the OpenShift console as follows:
  - https://linuxops-dallasX.conygre.com:8443/

admin
admin



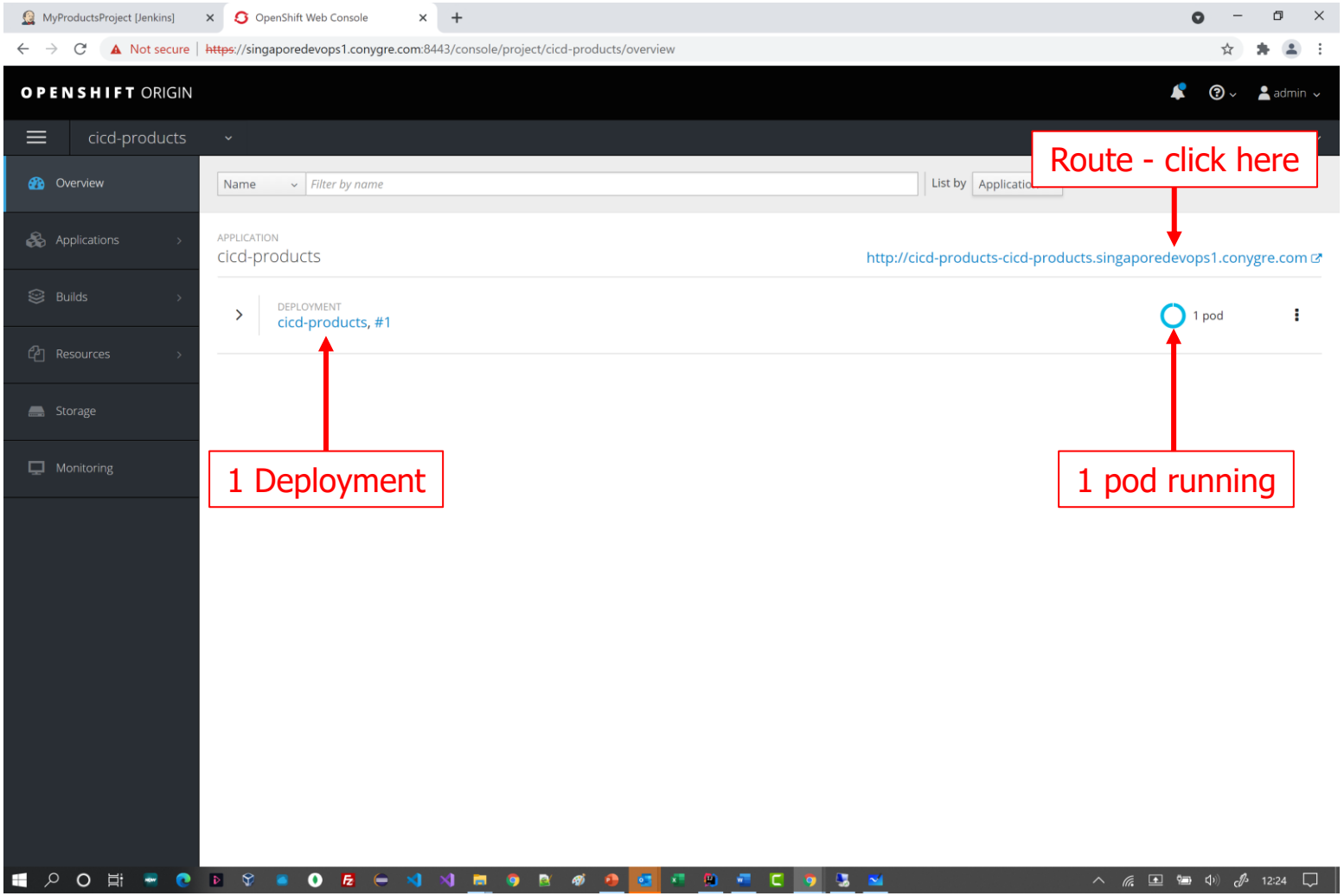
Verifying the Deployed App in OpenShift (2 of 3)

- On the right, you'll see your project in the list of projects



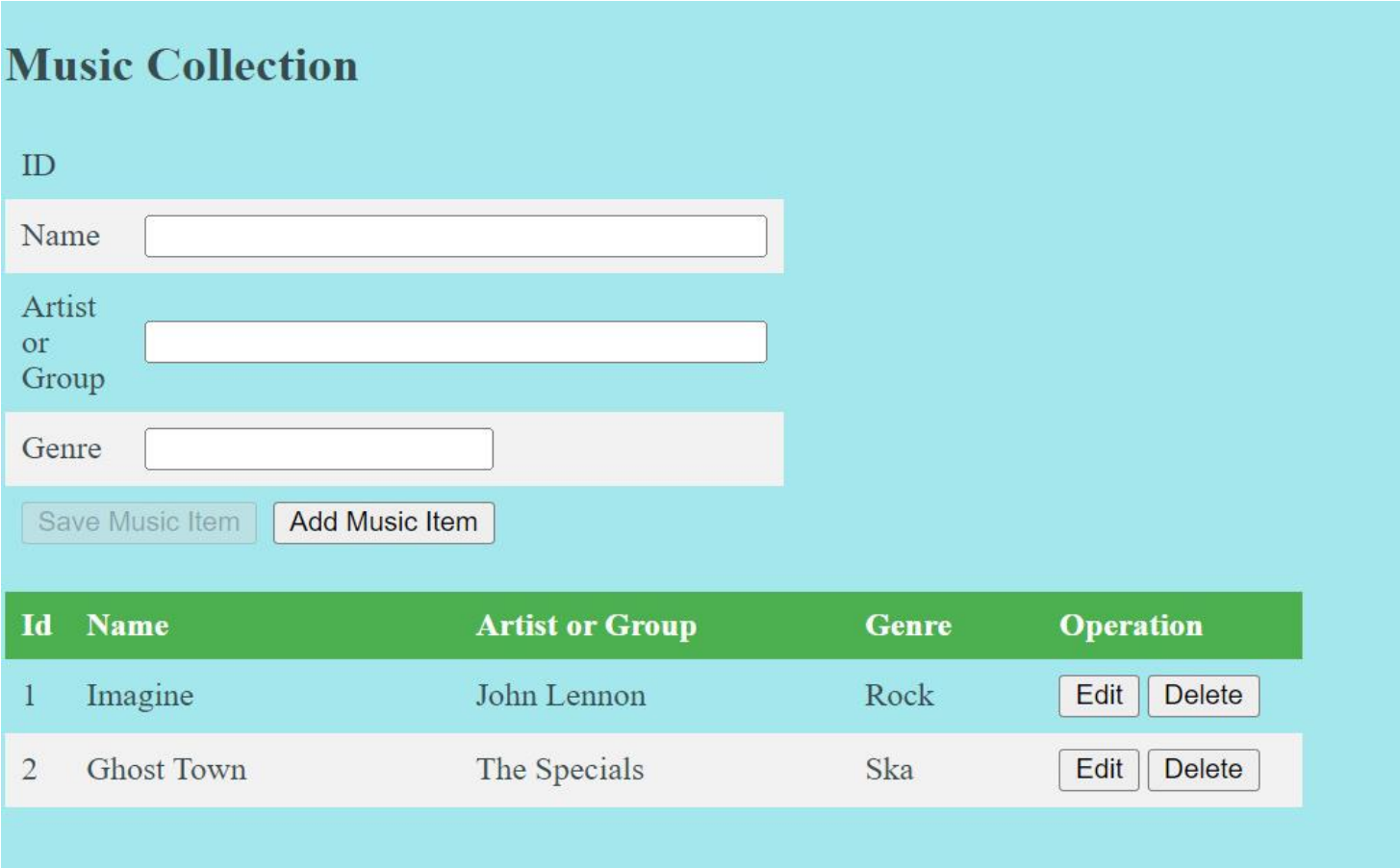
Verifying the Deployed App in OpenShift (3 of 3)

- Your project appears as follows



Pinging the App via the Exposed Route

- When you click the Route that exposes the app, you've just pinged your running application - success!



Any Questions?