RoundTower Technologies.

Continuous Integration and Continuous Delivery Fundamentals

Lab 1

Tennis Smith

1.0

Table of Contents

Prerequisites

Students will need a computer with:

- SSH
- Web Browser (Chrome, Safari, Firefox)

Credentials, Links, etc.

Systems & Credentials:

- roundtower Training Portal (http://goo.gl/SRGcuX)
  - User: *TRAINER PROVIDED*
  - Password: *TRAINER PROVIDED*
- Developer Desktop (http://gitlab.localdomain)
  - User: *student*
  - Password: *roundtower*
- GitLab (http://gitlab.localdomain)
  - User: *student*
  - Password: *roundtowerstudent*
- Jenkins (http://jenkins.localdomain:8080)
  - User: *student*
  - Password: *roundtower*
- GitLab (http://gitlab.localdomain)
  - User: *student*
  - Password: *roundtowerstudent*

Lab 1

Step 1: Log Into the Desktop

- Using a web browser, connect to the roundtower Training Portal, using the credentials provided by your trainer:

  ```
  http://goo.gl/SRGcuX
  ```

- You will see the lab environment listed as:

  ```
  CICD-Training-v11
  ```

- Under the "Actions" heading, select ***View***.
- In this view you will see the systems in your lab environment. Find the VM named ***Developer Desktop*** and open the virtual console by clicking the ***Console*** button.

> Pop-up Blockers are known to interfere with the virtual console. If you aren’t able to open the virtual console, disable any pop-up blockers and try again.

- Log into the Developer Desktop using the credentials specified above.

Step 2: Log Into GitLab

- On the developer desktop, open Firefox. (A link is provided in the dock on the left of the screen.)
- Connect to GitLab, using the credentials listed above:

```
http://gitlab.localdomain/
```

- You are now connected to GitLab. This system is your source code management and collaboration tool.
- On your dashboard, select the project ***Administrator/Lab1***.
- In the left sidebar, select ***Issues***.
- Click on the issue ***Story: New Content for Site***.

Step 3: Review the User Story

- In this lab, you will assume the role of a developer. The user story ***Story: New Content for Site*** has been assigned to you. Take a moment to review the the content of the user story.
- Before you alter the site content, you must set up your Continuous Integration toolchain.

Step 4: Set Up Continuous Integration Toolchain

- On the Developer Desktop, open a new tab in Firefox
- Connect to Jenkins, using the credentials listed above:

```
http://jenkins.localdomain:8080
```

- Click the ***New Item*** link in the left side menu.
- Name the new pipeline ***Lab1***.
- Select the ***Pipeline*** option to define the job type.
- Select ***OK***.
- On the next page, select the option ***GitHub project***, and enter the following in the ***Project url*** field:

```
http://gitlab.localdomain/root/Lab1
```

- Ensure that ***Student GitLab*** is selected for ***GitLab connection***. If it isn't, please select it from the dropdown list.
- Select the option ***Build when a change is pushed to GitLab***. Accept the default selections.
- Under the ***Pipeline*** heading, select ***Pipeline script from SCM***. For ***SCM***, select ***Git***. For ***Repository URL***, enter the following:

```
http://gitlab.localdomain/root/Lab1.git
```

- Select ***Save***. Your continuous integration pipeline is now configured.
- Verify that everything is working properly by selecting ***Build Now***.

Step 5: Get the Source Code for the Feature

- Now that you Continuous Integration toolchain is working properly, you can begin the development process.
- First, you must clone your project's source code repository.
- On the Developer Desktop, open the Terminal. (A link is provided in the dock on the left of the screen.)
- To clone the source code repository, type the following command:

```
git clone http://gitlab.localdomain/root/Lab1.git
```

- Change your working directory to your source code repository

```
cd Lab1
```

Step 6: Develop the Tests

- Before you alter the source code, you must first develop unit tests. For this lab, you will alter existing tests to verify that your new content is present.
- First, change your working directory to the "tests" directory

```
cd tests
```

- In this directory are two tests: ***index-test*** and ***layout-test***. First, you will modify ***index-test***:

```
nano index-test.txt
```

- In this file, you define the content that must be present in the site. Replace these two lines with the requested content from Scenario 1 & 2 of the User Story.
- *Remember, the text you specify here will be looked for **exactly** the new site, so aware of spaces and capitalization.*

> You can save a file by pressing "ctrl+O". You can close a file by pressing "ctrl+X".

- Now, alter the second test with the requested content from Scenario 3 & 4 of the User Story.

```
nano layout-test.txt
```

- Now that the tests are complete, you can alter the site content.

Step 7: Alter the Site Content

- Now you can alter the site content. To begin, change your working directory:

```
cd ..
cd src/site/views
```

- Now you alter the index with the requested content from Scenario 1 & 2 from the user story.

```
nano index.jade
```

- Once that is complete, alter the layout with the requested content from Scenario 3 & 4.

```
nano layout.jade
```

Step 8: Commit Changes and Push to Central Repository

Now that your site has been modified to according to the user story, you can commit your changes and push them to GitLab. * First, change the working directory.

```
cd ~/Lab1
```

- Next, add all altered files to the change set.

```
git add .
```

- Next, commit the changes.

```
git commit -m "Issue #1 - Tests Altered, Feature Complete"
```

- Last, push the change to GitLab.

```
git push
```

- This push will trigger the continuous integration system (Jenkins) to build and test the change. You can observe it’s progress by switching back to Firefox.
- By hovering over each phase of the pipeline, you can view the logs for that phase.
- If any phase of your pipeline fails, utilize the logs to identify the error. Please ask your trainer for assistance if required.