

Computer Science 2212b - Winter 2014
Introduction to Software Engineering

Lab 5 - Continuous Integration with Jenkins

Part I

Introduction to Continuous Integration

Estimated Time: 90-120 minutes

1 Introduction

In this lab, you will set up a Jenkins Continuous Integration (CI) server on an Amazon EC2 instance (virtual machine). Working through a five-part video series, you will setup an EC2 instance, install Jenkins on it, and configure Jenkins to automatically build and test your team's code, every time a change is pushed to your GitHub repository.

Note: It is **very** important that you carefully follow the directions given in the videos. Nearly all problems that students have with labs are due to not following instructions properly. There are a lot of instructions given in the videos, and therefore ensuring that you are entering the correct commands in the correct order is essential.

2 Prerequisites

In order to complete this lab, you will need to email your instructor and obtain:

- A username and password for the EC2 console
- An SMTP server hostname, along with a username and password for the server

You are not to share this information with anyone else, nor should it be checked into a GitHub repository. It is also not to be used for any purpose other than this lab, and for the purpose of running a CI server for your team project.

3 Rules of Engagement

Before we continue, it's important to set some ground rules for working with Amazon EC2 in this course. We wanted to give you direct access to the Amazon EC2 console to give you practical experience with EC2 that may just come in handy for you out in the workforce (many companies are using EC2!). The console allows you to work with your own instance and to stop it, reboot, and terminate it (delete it) as needed. However, it will allow you to do the same with anyone else's instance in the class. Hence, we remind you of the following excerpt from the department's **Rules of Ethical Conduct**:

The following are considered to be scholastic offences:

- Unauthorized inspection, alteration, deletion, publication, copying or tampering with files.
- To engage in any action which denies or unreasonably restricts the use of computer facilities to their authorized users.

Hence, tampering with another student's or team's instance, security group, or key pair in any way will not be tolerated and will be treated as an academic offense. This includes, but is not limited to, stopping or rebooting an instance, or modifying or deleting an instance, key pair, or security group.

In short, you can work with your own instance, security group, and key pair as needed, but please do not touch those of other students or teams.

Additionally, you may have the need to delete your instance and start over (for example, if you mess up your instance and can't get it work properly). This is quite fine, and actually a real benefit of using virtual machines. If you mess up your instance, you can simply terminate it, start a new one, and continue working. However, the rules we will follow in this course are as follows:

- You may only ever have 1 instance running at a given time. If you decide to terminate your instance and start a new one, you must wait for the old instance to reach the **terminated** state before you start a new one.
- You may only ever start an instance of type `t1.micro` (you may use `m1.small` **only if** you are the individual selected by your team to perform Lab 5). Amazon EC2 provides a variety of different instance types, each with their own specifications for RAM size and CPU speed, and each with their own hourly price. Amazon has generously allocated a specific budget to us for use in this course, and to avoid exceeding this allocation, we will be using the cheapest instances available. Nevertheless, even these micro instances are more powerful than we actually require for the purposes of CS 2212, so they will do just fine.
- Your instance is intended for your CS 2212 assignments and project only. You are not permitted to use it for other purposes.

Note that your actions in the EC2 console are logged and we will be monitoring the logs frequently to detect any unauthorized actions.

Alright, enough legalese. Let's get back to Jenkins on EC2.

4 A Word about the Videos

You should watch the videos in this lab in full screen and 720p resolution. Note that, on YouTube, sometimes you may have to rewind the video several times after selecting 720p before the high quality picture starts to display. You will know you are in 720p if the text in the video is clear and not blurry.

5 Introduction to Continuous Integration

If you were in class for the CI lecture, you can skip this part. Otherwise, you should watch the following video:

<http://youtu.be/1JSOGJQAhtE>

6 Setting Up an Amazon EC2 Instance

Work through the following video: <http://youtu.be/zojMg2c6k3Q>

- **If you are the designated individual completing lab 5 for your team**
 - Use an instance type of `m1.small`
 - Name your instance, security group, and key pair `teamN`, where `N` is your team number
- **If you are simply completing lab 5 for interest's sake**
 - Use an instance type of `t1.micro`
 - Name your instance, security group, and key pair after your UWO username

7 Configuring Jenkins

Work through the following video: <http://youtu.be/0ZS2BL5R3Ow>

8 Configuring a Project

1. Change to your individual Git repository, create a `lab5` directory, and change to it:

```
$ cd ~/courses/cs2212/labs
$ mkdir lab5
$ cd lab5
```

2. Create the directory structure to host our project:

```
$ mkdir -p src/{main,test}/{java,resources}
```

3. Create the directory structure to house your Java package and its tests:

```
$ mkdir -p src/main/java/ca/uwo/csd/cs2212/USERNAME
$ mkdir -p src/test/java/ca/uwo/csd/cs2212/USERNAME
```

Again, `USERNAME` should be your UWO username in **lowercase**.

4. Obtain `pom.xml`, `BankAccount.java`, and `TestBankAccount.java` from the following Gist:

<https://gist.github.com/jsuwo/9038610>

- `pom.xml` should go in your `lab5` directory and should be customized with your UWO username.
- `BankAccount.java` should go in `src/main/java/ca/.../USERNAME`. The package line in the file should be customized with your UWO username.
- `TestBankAccount.java` should go in `src/test/java/ca/.../USERNAME`. The package line in the file should be customized with your UWO username.
- Do **NOT** add the contents of `pom2.xml` to your `pom.xml` yet.

5. Commit and push your code to GitHub.

6. Work through the following video: <http://youtu.be/SRAQzs41ct4>

- The first project you create should use your own individual GitHub repository.
- You should later set up another Jenkins project for your team repository – it will be part of assignment 3, so you might as well do it now while the knowledge is fresh in your mind.

9 Advanced Jenkins Configuration

Work through the following video: <http://youtu.be/IZ99VwrF6t4>

10 Submitting Your Lab

- Create a Jenkins user `automarker` and give it a password.
- Create a file `password.txt` inside your `lab5` directory:
 - On line 1, paste the address of your Jenkins server including the `http://`, e.g. `http://myserver.com`.
 - On line 2, enter the password you chose for the `automarker` user.
 - There should be nothing else on lines 1 or 2, and no other content in the file.
- Commit your code and push to GitHub.
- To submit your lab, create the tag `lab5` and push it to GitHub. For a reminder on this process, see Lab 1.