CMPSC 112
Introduction to Computer Science II
Spring 2014

Laboratory Assignment One: Version Control with Git and Bitbucket

Introduction

Practicing software developers normally use a version control system to manage most of the artifacts produced during the phases of the software development life cycle. In this course, we will always use the Git distributed version control system to manage the files associated with our laboratory assignments. In this laboratory assignment, you will learn how to use the Bitbucket service for managing Git repositories and the `git` command-line tool in the Ubuntu Linux operating system. After connecting to the course's Git repository and creating your own repository, you will compile and run two Java programs, write about their output, and commit your final report to a repository.

Configuring Git and Bitbucket

During this laboratory assignment and subsequent assignments, we will securely communicate with the Bitbucket.org servers that will host all of our projects. In this laboratory assignment, we will perform all of the steps to configure the accounts on the departmental servers and the Bitbucket service. Throughout the assignment, you should refer to the following Web site for additional information: `https://confluence.atlassian.com/display/BITBUCKET/Bitbucket+101`. As you will be required to turn in a report describing each step that you finish in this assignment, please be sure to keep a record of all of the steps that you complete and the challenges that you face. You are also responsible for working with a team to ensure that each member of the team is able to successfully complete each of the steps outlined in this assignment.

1. If you do not already have a Bitbucket account, please go to the Bitbucket Web site and create one — make sure that you use your `allegheny.edu` email address so that you can create an unlimited number of free Bitbucket repositories. Then, upload your ssh key to Bitbucket.

2. If you have never done so before, you must use the `ssh-keygen` program to create secure-shell keys that you can use to support your communication with the Bitbucket servers. Follow the prompts to create your keys and save them in the default directory (press "Enter" after you are prompted: "`Enter file in which to save the key ...   :`", then press "Enter" twice if you do not wish to create a passphrase at this time or type your selected passphrase if you do). Type `man ssh-keygen` and talk with the course instructor to learn more about this program. What files does `ssh-keygen` produce? Where does this program store these files by default? You do not need to complete this step if you have already run `ssh-keygen`.

   Once you have created your ssh keys, you should raise your hand to invite the course instructor to help you with the next steps, thus best ensuring that Bitbucket is configured correctly. First, you must log into Bitbucket and look in the right corner for an account avatar with a down arrow. Click on this blue link and then select the "Manage account" option. Now, scroll down until you found the "SSH keys" option and upload your ssh key to Bitbucket.

You can copy your SSH key by going to the terminal and typing "`cat ~/.ssh/id_rsa.pub`" command. If you cannot complete any of these steps, then please see the instructor. Students who have already created and configured their Bitbucket accounts will not need to complete this step unless you did not use your `allegheny.edu` email account's name for Bitbucket.

3. Now, you need to test to see if you can authenticate with the Bitbucket servers. First, show the course instructor that you have correctly configured your Bitbucket account. Now, ask the instructor to share the course's Git repository with you. Open a terminal window on your workstation and change into the directory where you will store your files for this course. For instance, you might make a `cs112S2015/` directory that will contain the Git repository that I will always use to share files with you. Once you have done so, please type the following command: "`git clone git@bitbucket.org:gkapfham/cs112s2015-share.git`". If everything worked correctly, you should be able to download all of the files that you will need to use for this laboratory assignment. Please resolve any problems that you encountered by first reviewing the Bitbucket documentation and then discussing the matter with your classmates. If you are still not able to run "`git clone`", then please see the instructor.

4. Using your terminal window, you should browse the files that are in this Git repository. In particular, please look in the `labs/lab1/src/` directory and use Vim to study the two Java programs that you find. Remember, the "`cd`" command allows you to change into a directory.

5. If you have not done so already, you should reconfigure your Vim text editor so that it uses the "spf13-vim" configuration so that you have many of the advanced editing features on which we will rely this semester. Those students who have not installed spf13-vim can follow the instructions at `http://vim.spf13.com/` to better configure Vim. If you have already installed this Vim configuration, then please make sure that you know how to use the plugins and features that it provides; see the course instructor if you are stuck on this step.

Creating a New Repository

Now that you have learned how to clone an existing Git repository, you should make a new repository in the `cs112S2015/` directory that you previously created. First, make a new directory called `cs112S2015-<your user name>`. Then, change into this directory and type the command "`git init .`". At this point, you should go into the `cs112s2015-share` repository and use the "`cp -r`" command to copy the entire `labs/` directory from the `cs112s2015-share` repository to `cs112S2015-<your user name>`. Once the files are in your own Git repository, please use the "`git add`" and "`git commit`" commands to add them correctly. If you do not know how to use "`git add`" and "`git commit`" in the terminal window, please learn more about them by searching on the Internet, talking about them with your classmates, and discussing them with the course instructor.

Next, you should use the Bitbucket Web site to create a repository that has the same name as the local directory and local repository. You must follow Bitbucket's instructions to push the code and tags in your local repository to the remote one. When you are finished with this step, you should see in your Web browser that the Bitbucket servers are correctly storing the two Java programs. Once the Git repository contains the correct files, you should share your Bitbucket repository with the course instructor — my Bitbucket user name is "gkapfham".

You can learn more about Git by consulting Web sites like `http://gitimmersion.com/`. At minimum, you should ensure that you understand how to use the following commands: "`git init`", "`git status`", "`git add`", "`git commit`", "`git push`", and "`git pull`". If you want to interact with your Git repository through their Vim text editor—instead of through the terminal window—then you are encourage to learn how to use the Fugitive plugin that is part of spf13-vim.

Compiling, Running, and Understanding Java Programs

Once you have mastered the use of Git and version control, you should return to the `labs/lab1/src/` directory that contains the two Java programs. Now, use the Java compiler to compile the `Hooray.java` program. That is, you should type "`javac Hooray.java`" in the terminal window. Next, you can run this program by typing "`java Hooray`" in the terminal window. What output does this program produce? Why does it create this output? How do you stop this program?

After compiling, using, and studying the `Hooray` program, you should complete the same steps for the `Weeee.java` program. Go ahead and compile and run this program. What output does it produce? Why does it create this output? How is the output similar to and different from that which was created by the `Hooray` program? Once you have finished studying and understanding these two programs, add comments to the code to explain what they do and how they work. Finally, please make sure that the commented version of each program is correctly committed to your Git version control repository hosted by Bitbucket and you have responded to all of the below prompts.

Summary of the Required Deliverables

This assignment invites you to submit one printed and electronic version of these deliverables:

1. A description of the steps that a user must take to configure Git and Bitbucket
2. A description of the inputs, outputs, and behavior of the six aforementioned Git commands
3. A commented version of the `Hooray.java` and `Weeee.java` programs
4. A one-page report that clearly responds to the following four prompts:
   (a) The steps that you took to compile and run both of these programs
   (b) The output that each of these programs produce
   (c) An explanation for why these programs create the output that they do
   (d) A discussion of the similarities and differences between these programs and their output

Before you turn in this assignment, you also must ensure that the course instructor has read access to your Bitbucket repository that is named according to the convention `cs112S2015-<your user name>`. Please note that each student in the class is responsible for completing and submitting their own version of this assignment. While it is acceptable for members of this class to have high-level conversations, you should not share source code or full command lines with your classmates. That is, it is necessary to distinguish carefully between the student who discusses the principles underlying a problem with others and the student who produces assignments that are identical to, or merely variations on, someone else's work. Source code or output that is largely similar to other submissions or to online material will be judged as evidence of violating the Honor Code. Please see the course instructor if you have questions about the policies for this laboratory assignment.