

Andreas Landgrebe
Pledge:
Labraotry Assignment # 1
Introducing Data Management Concepts and Tools
Wednesday September 3, 2014

Andreas Landgrebe

Laboratory Assignment # 1 : Introducing Data Management Concepts and Tools

1. **A description of the steps that a user must take to configure Git and Bitbucket.**

In order to properly set up Git and Bitbucket to work correctly, one must undergo this steps. The first step to complete is to read a quick overview of SSH concepts.

The next step to properly set up Git and Bitbucket is to ensure you have an SSH client installed. To ensure this, you must write a command:

```
ssh -v
```

If you have an SSH client installed, then you can skip this step.

If you do not have a SSH client installed, then you will have to install the SSH client. In order to do so, you will have to write the command:

```
ssh-keygen
```

This command will generate ssh keys. One will now save the key to a default location so one will just accept the location to save the key. After so, one will be prompt to create a password and re-enter a password for ones SSH keys. After one has successfully create a password and re-enter the same password, one good idea to do is to list and contents of `/.ssh` to view the key files to make sure that one has successfully installed the ssh keys. In order to do so, one would type the command:

```
ls -a /.ssh
```

If you had saved the ssh keys to the default location.

The next step is to start the SSH-agent and load ones keys. To do this, one must type the command:

```
ps -e — grep [s]sh-agent
```

To see if the agent is running. If it is not running, then one would type the command to start it manually:

```
ssh-agent /bin/bash
```

After one has start to run the agent, then you would load your new identity into the ssh-agent management program by running the following command:

```
ssh-add ~/.ssh/id_rsa
```

Now one want to use the ssh-add command to list the keys that the agent is managing. To do so, one would type the following command:

```
ssh-add -l
```

Since Git does compression for one when sending or retrieving data using SSH, one does not need to enable SSH compression. If one was using Mercurial, then one would need to enable SSH compression. After this step, one needs to install the public key on a Bitbucket account. In order to do so, one must open an internet browser and create a Bitbucket account. If one has already created an account, then one must log into ones account. After so, one must choose avatar and click on "Manage Account". The "Manage Account" setting will be under the application menu. After so, the system will display the "Account settings" page. Now, one must go on the tag "SSH keys". When clicked on, this tag will display all of the existing keys that ones "Bitbucket" account has. If this is ones first time using a Bitbucket account, then there should be no SSH keys configured. To configure this, one must copy the contents of the public key file. To do so, one could "cat" the contents. To do this, write the following command into the terminal.

```
cat ~/.ssh/id_rsa.pub
```

Now back in the browser, enter a label for ones new keys. This label could be anything so one could just call it "Default public key". After so, paste the copied key into the SSH key field.

There is just one more step to set up Bitbucket and Git correctly. The last step is to change ones repository from a HTTPS to the SSH protocol to be able to use the SSH keys one has configured correctly. After so, one would want to create a repository and look for a clone button on the new repository. If one is able to find this command to clone and successfully clone this repository into ones local files, then one has successfully set up Bitbucket and Git correctly and can use verison control repository correctly.

2. A description of the inputs, outputs, and behavior of the six aforementioned Git commands.

(a) git init

The git init command will create a new local repository with the

specified name that is provided. So if one writes the following command into the terminal:

```
git init test
```

Then one will create a local repository named test.

(b) `git status`

The `git status` command will list all of the new or modified work in a workspace that has not been committed.

(c) `git add`

The `git add` command will add file contents to the index. This will not put the files into the repository. However, it will put the files to pre-staging area to be later committed to the repository.

(d) `git commit`

The `git commit` command will be able to record changes to the repository. If one writes the command:

```
git commit -m "added files to the repository"
```

Then the changes made to the repository will be recorded and on BitBucket or GitHub, one will be able to see the comment "added files to the repository" by the specific user if one has read access to the repository.

(e) `git push`

The `git push` command will be able to push all of the modified or new work to the repository so everybody else having access to the repository will be able to see the changes made. In order for that to happen, one would need be able to add to the pre-staging area and record the changes that were made to the repository. After those steps, then a user will be able to push the changes onto the repository for everybody to see.

(f) `git pull`

The `git pull` command will pull all of the new or modified files that were made to the repository. If one is working in a group and is working on a specific file, before starting to edit, one would need to pull the changes to successfully be able to modify any files in the repository.