Working with Git in the Command Line

Introduction

Git is a very popular distributed version control system. In this lab, you will be asked to perform basic Git-related tasks in the command line, which includes installing Git, initializing a repository, adding and committing files, and creating and merging branches.

Solution

Log in to the lab server using the credentials provided:

```
ssh cloud_user@<PUBLIC_IP_ADDRESS>
```

Install and Configure Git

1. Install Git:

   ```
   sudo yum install git -y
   ```

2. Clear your screen:

   ```
   clear
   ```

3. Update the Git user email setting to be `cloud_user@mybusiness.com`:

   ```
   git config --global user.email "cloud_user@mybusiness.com"
   ```

4. Update the Git username setting to be `cloud_user`:

   ```
   git config --global user.name "cloud_user"
   ```

Initialize the Git Repository

1. Clear your screen:

   ```
   clear
   ```

2. Write the full pathname of the current working directory:

```
pwd
```

3. Create the `alpha` directory in the home directory:

```
mkdir alpha
```

4. List the names of the files in the current directory:

```
ll
```

5. Change directories to `/home/cloud_user/alpha`:

```
cd alpha/
```

6. Initialize the repository:

```
git init
```

7. `clear` your screen.

Add and Commit Files to the Git Repository

1. Create two files called `artifact01` and `artifact02`:

```
touch artifact01 artifact02
```

2. Add the newly created files to the staging area:

```
git add .
```

3. Check the status:

```
git status
```

4. `clear` your screen.
5. Commit the files to the repository and include a message:

```
git commit -m "added two files"
```

6. Check the status:

```
git status
```

7. `clear` your screen.

Create Branches of the Git Repository

1. Create a new branch called `hot_fixes`:

```
git branch hot_fixes
```

2. Check the branch:

```
git branch
```

3. `clear` your screen.
4. Create and checkout a branch called `feature01`:

```
git checkout -b feature01
```

5. Check the branch:

```
git branch
```

6. Create a directory called `feature01`:

```
mkdir feature01
```

7. Create a file called `manifest.txt` in the `feature01` directory:

```
touch feature01/manifest.txt
```

8. Add the changes to the `feature01` branch:

```
git add .
```

9. Check the status:

```
git status
```

10. Commit changes to the `feature01` branch with a message:

```
git commit -m "adding feature01 directory with manifest"
```

11. Check the status:

```
git status
```

12. `clear` your screen.

Merge a Branch with the Master

1. Check the branch:

```
git branch
```

2. Checkout the `master` branch:

```
git checkout master
```

3. Confirm you're in the `master` branch:

```
git branch
```

4. Merge the `feature01` branch with `master`:

```
git merge feature01
```

Conclusion

Congratulations — you've completed this hands-on lab!