**Git Commands**

**Using Github – Basic Commands**

Git is the open source distributed version control system that facilitates GitHub activities on your laptop or desktop. This sheet is used Git command line instructions for quick reference.

**Install Git:-**

*Git* is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.GitHub provides desktop clients that include a graphical user interface for the most common repository actions and an automatically updating command line edition of Git for advanced scenarios.

**Check git version:-**

Currently which git version is running on our machine that can be check by using following commands:-

“git –version” .

**Repository:-**

There are two types of repository,

-Public Repository.

-Private Repository.

### **Steps and commands:-**

|  |  |
| --- | --- |
| **Command** | **Description** |
| Mkdir dir\_name | |  | | --- | | Create directory where you want to  create project. | |
| git clone Git\_project\_path  ls -lha | Create a local copy of a remote repository.  List of all hidden files and size of that files. |
| git status | Check status. |
| git add [file-name.txt] | Add a file to the staging area. |
| git add -A or --. | Add all new and changed files to the staging area |
| git config --global core.editor “nano” or (Short-cut command)  git commit -m "[commit message]" | Commit changes (Like save point) |
| git log | To know How many commit on git |
| git rm -r [file-name.txt] | Remove a file (or folder) |

### 

### **Branching & Merging**

|  |  |
| --- | --- |
| **Command** | **Description** |
| git branch | List branches (the asterisk denotes the current branch) |
| git branch -a | List all branches (local and remote) |
| git branch [branch name] | Create a new branch |
| git branch -d [branch name] | Delete a branch |
| git push origin --delete [branchName] | Delete a remote branch |
| git checkout -b [branch name] | Create a new branch and switch to it |
| git checkout -b [branch name] origin/[branch name] | Clone a remote branch and switch to it |
| git checkout [branch name] | Switch to a branch |
| git checkout - | Switch to the branch last checked out |
| git checkout -- [file-name.txt] | Discard changes to a file |
| git merge [branch name] | Merge a branch into the active branch |
| git stash | Stash changes in a dirty working directory |
| git stash clear | Remove all stashed entries |
| git merge [source branch] [target branch] | Merge a branch into a target branch |

### **Sharing & Updating Projects:-**

|  |  |
| --- | --- |
| **Command** | **Description** |
| git push origin [branch name] | Push a branch to your remote repository |
| git push -u origin [branch name] | Push changes to remote repository (and remember the branch) |
| git push | Push changes to remote repository (remembered branch) |
| git push origin --delete [branch name] | Delete a remote branch |
| git pull | Update local repository to the newest commit |
| git pull origin [branch name] | Pull changes from remote repository |
| git remote add origin ssh://git@github.com/[username]/[repository-name].git | Add a remote repository |
| git remote set-url origin ssh://git@github.com/[username]/[repository-name].git | Set a repository's origin branch to SSH |

### **Inspection & Comparison:-**

|  |  |
| --- | --- |
| **Command** | **Description** |
| git log | View changes |
| git log --summary | View changes (detailed) |
| git diff [source branch] [target branch} | Preview changes before merging |

**Eg:-**

**=> git --version**

**=> mkdir mangesh**

**=> cd mangesh**

**=> git clone git\_repo\_path**

**=> ls -la**

**=> cd dir\_nm**

**=> ls**

**=> git checkout -b new\_nm**

**=> git status**

**=> git branch**

**=> git status**

**=> git add file\_nm.txt**

**=> git status**

**=> git config --global user.email “mail\_id”**

**=> git config --global user.name “mail\_id”**

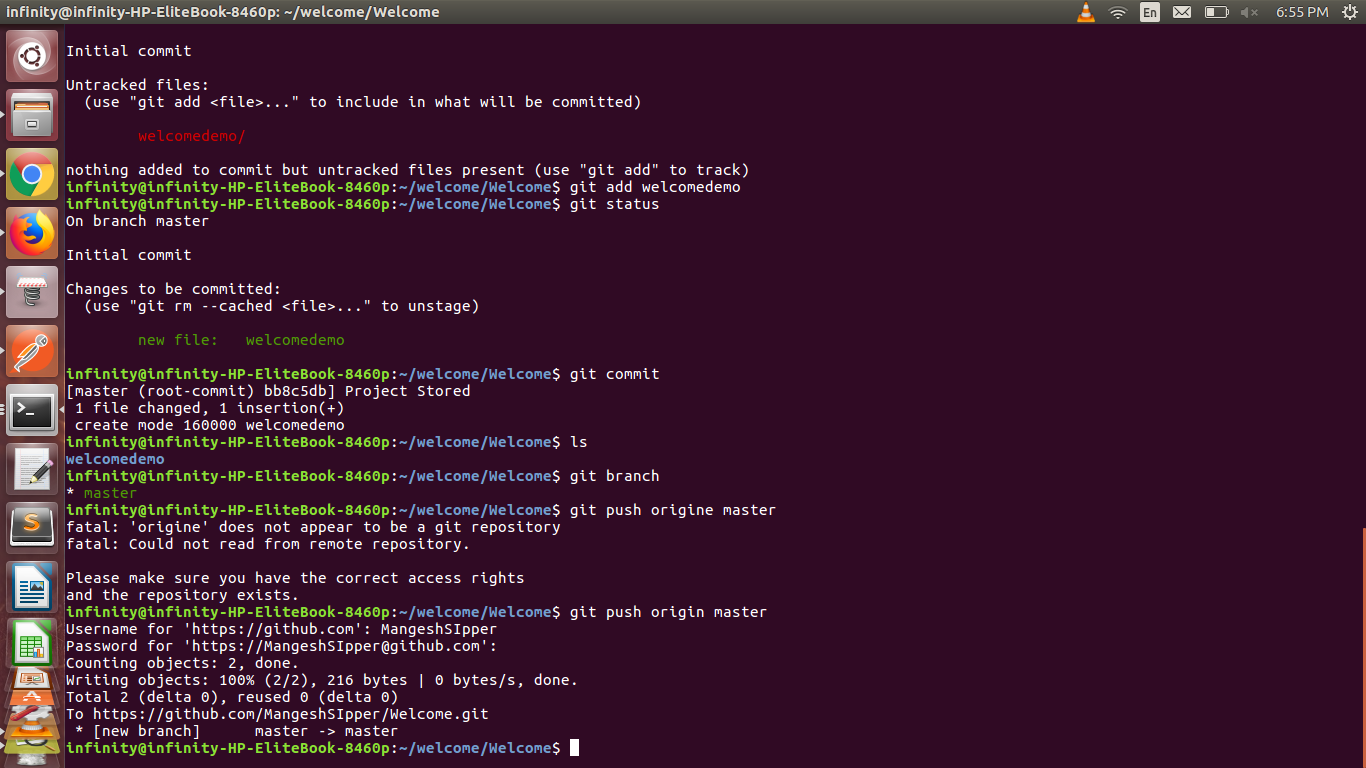
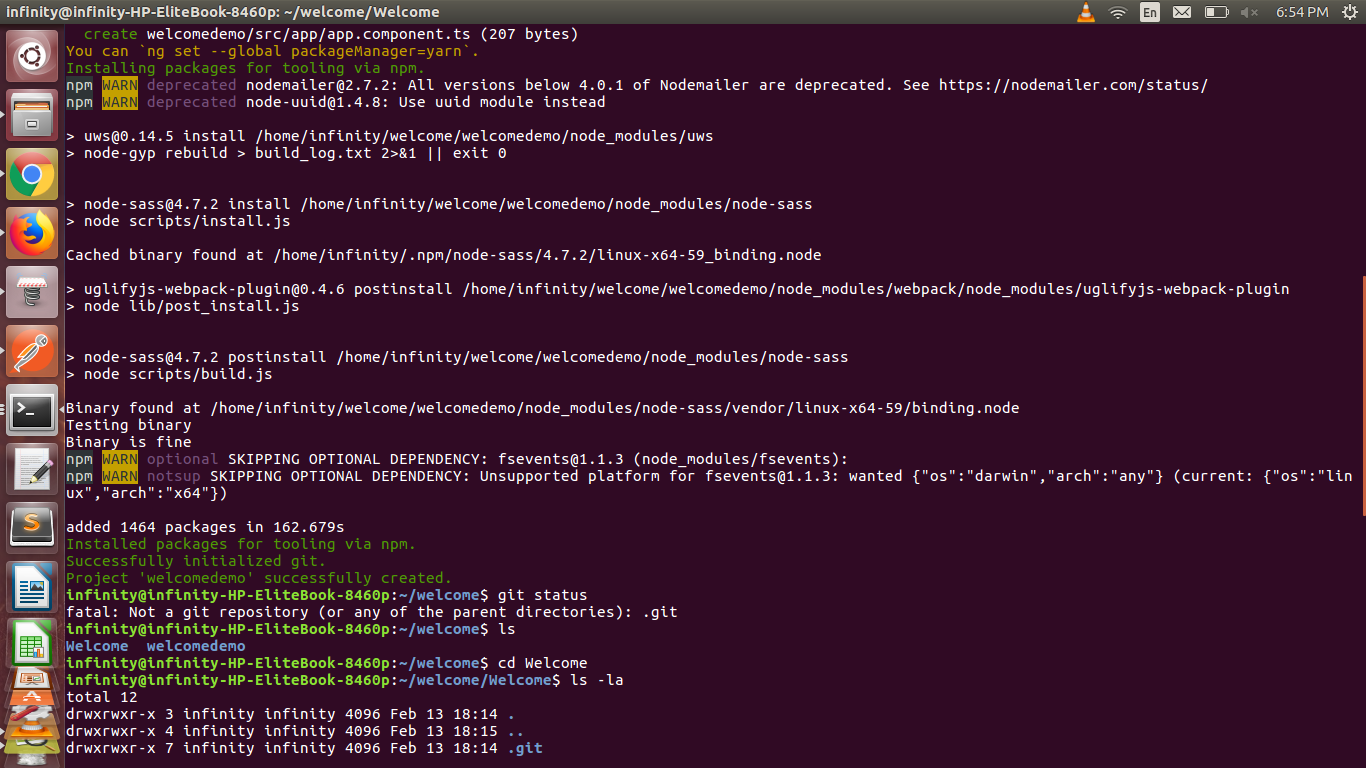
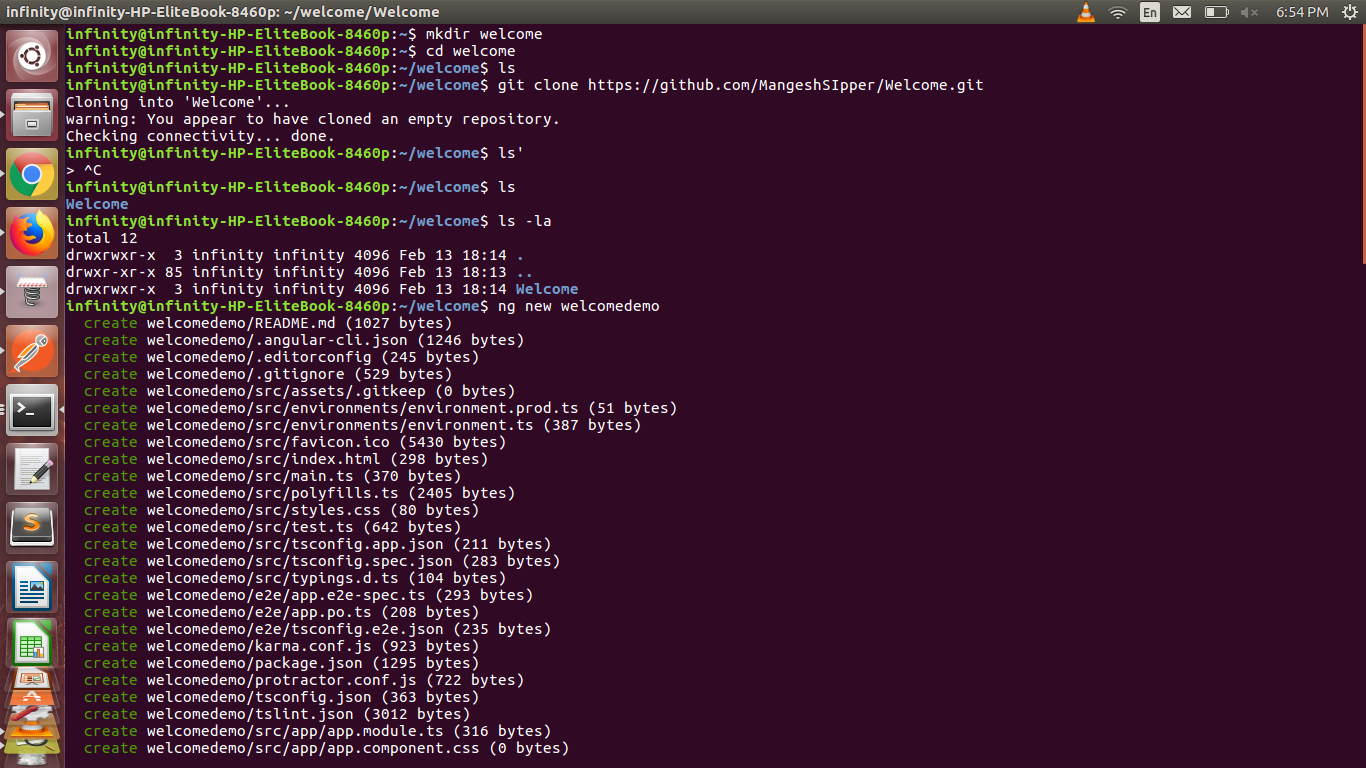
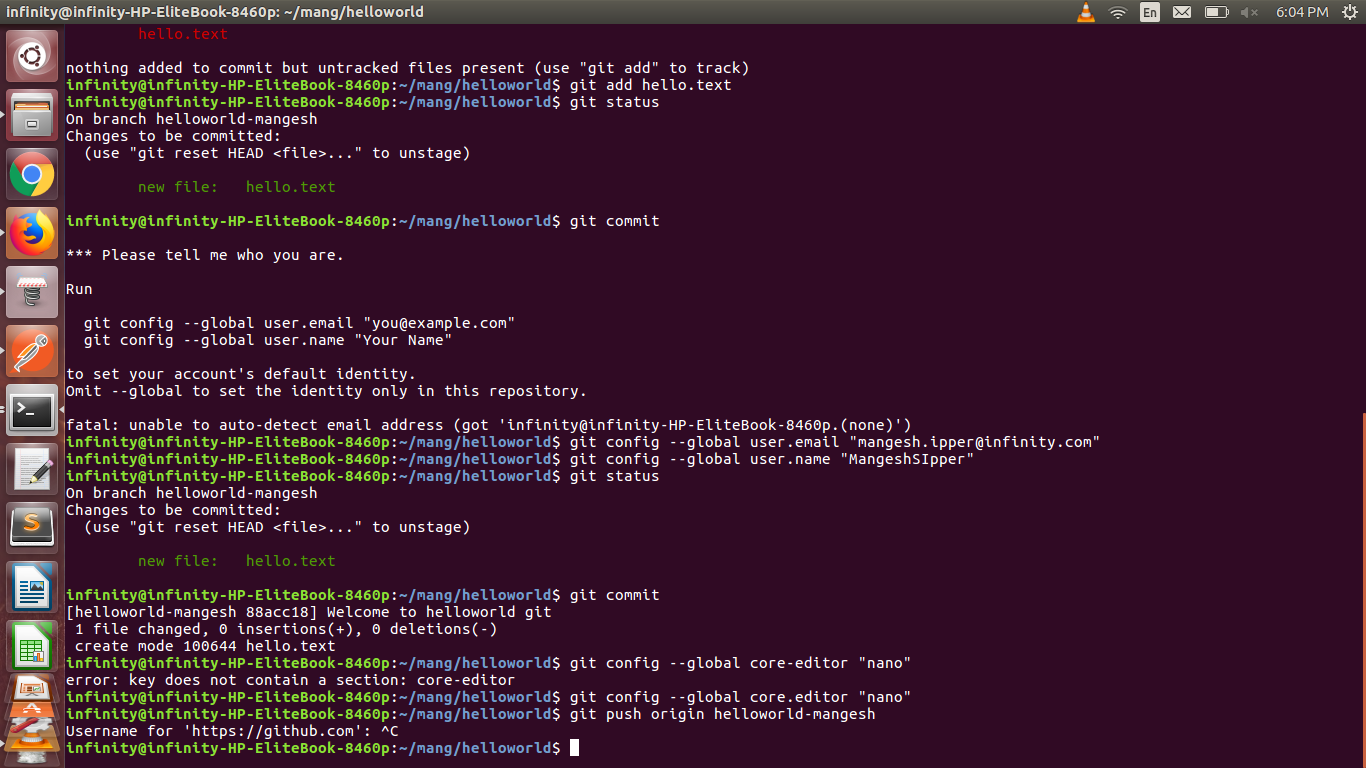
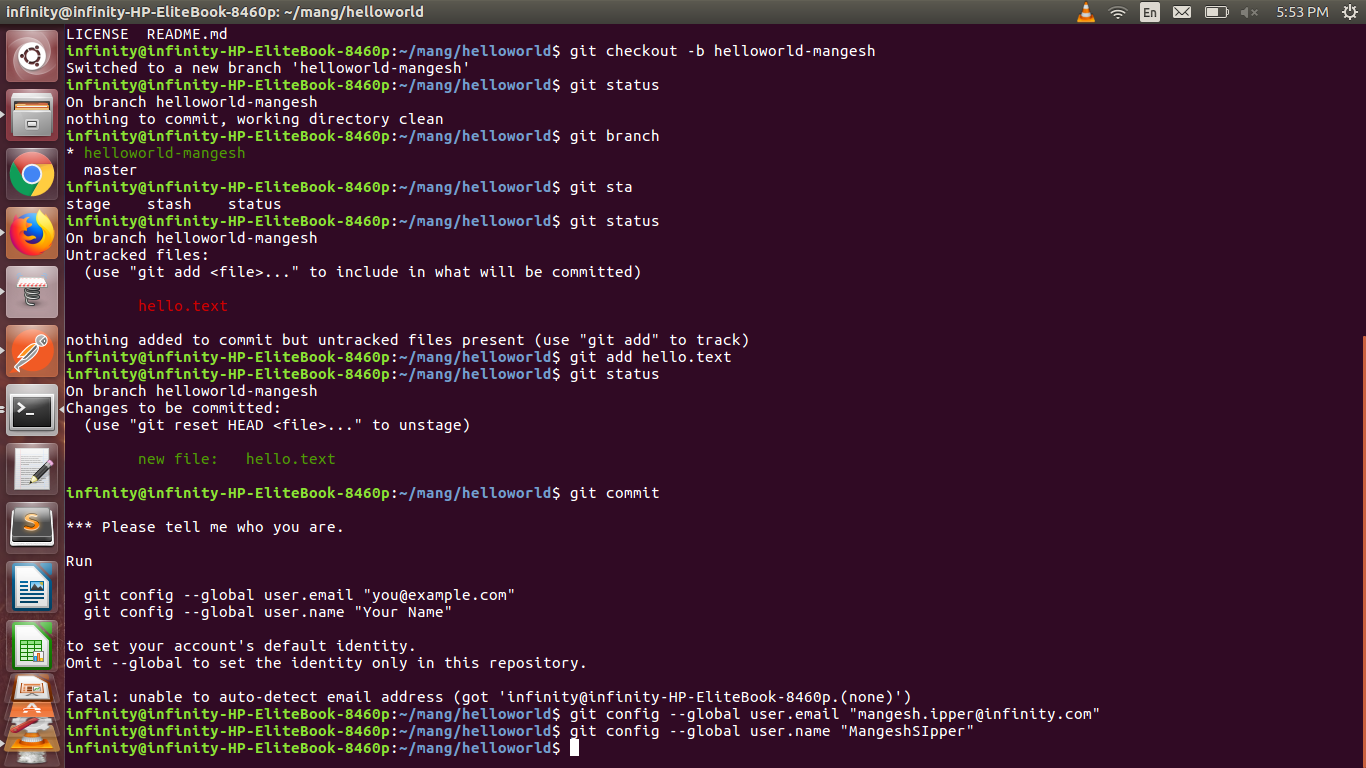
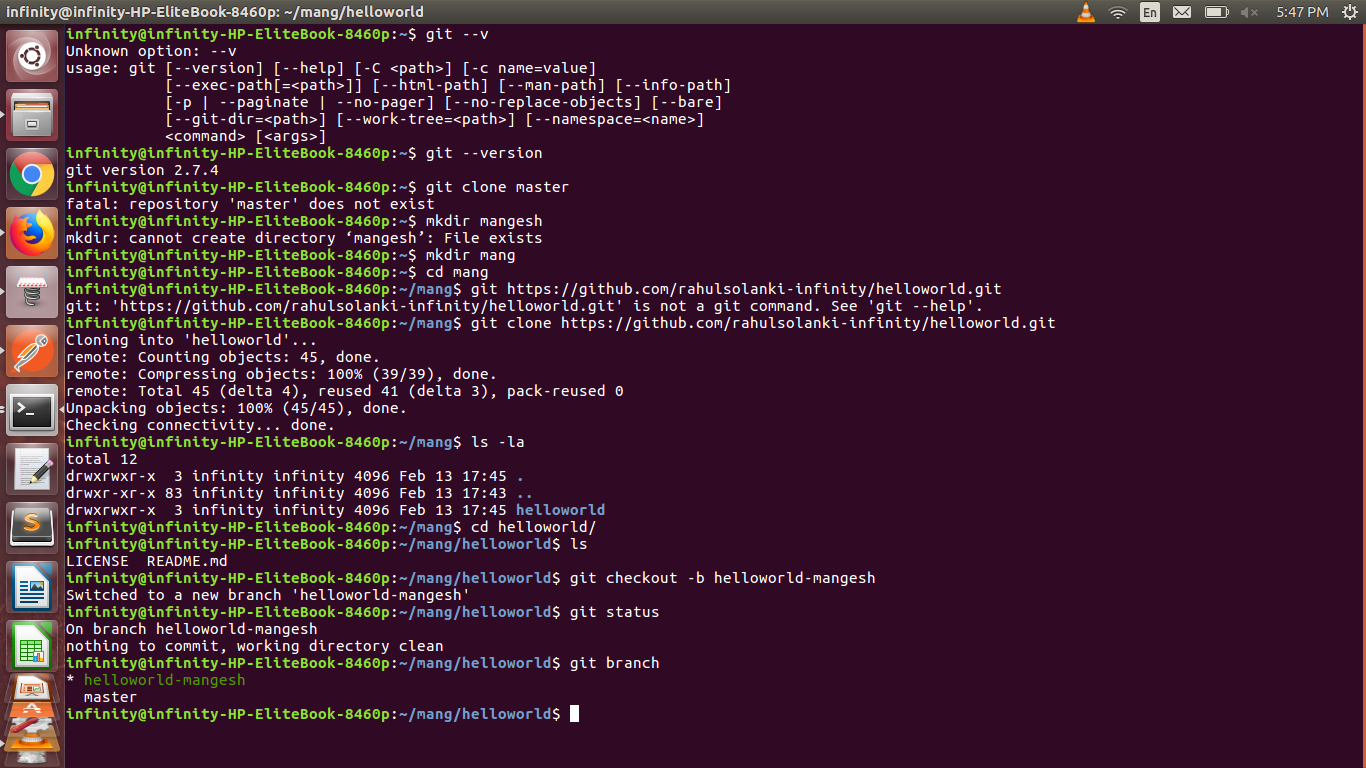
**=> git status**

**=> git config --global core.editor “nano”**

**To save commit ctr+o, Enter, ctr+x**

**=> git push origin master**

**Linux OS:**

****

**Windows OS:**