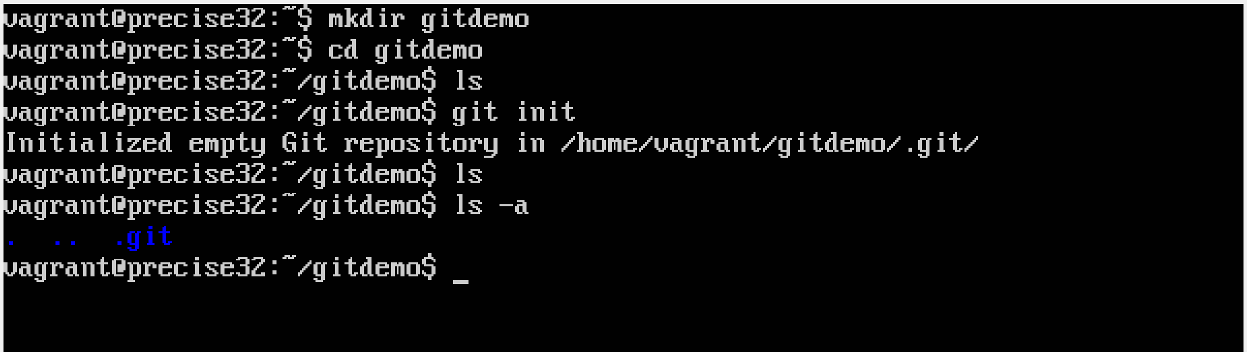
Git demo

1. Create local repository

Step 1: Bring up your vagrant VM and login. Check “1_setup_instructions” document

Step 2: Git init – Make git to track a particular folder

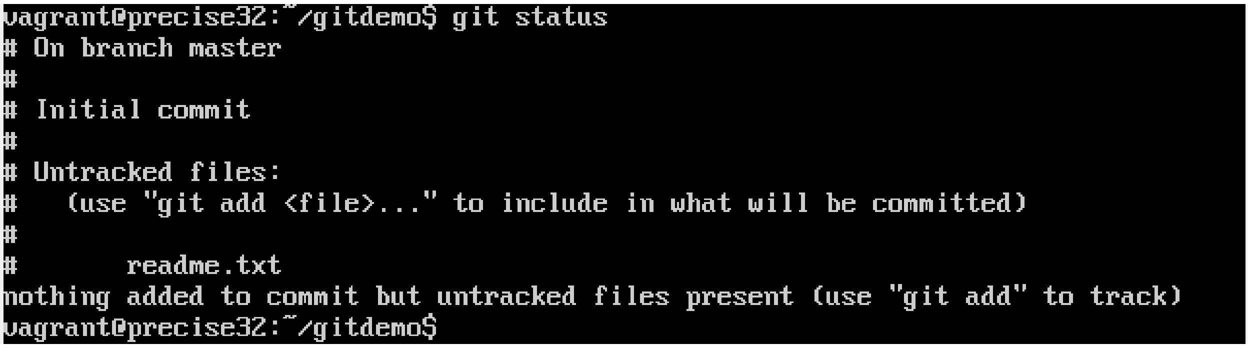
```
mkdir gitdemo
cd gitdemo
git init
```



```
vagrant@precise32:~$ mkdir gitdemo
vagrant@precise32:~$ cd gitdemo
vagrant@precise32:~/gitdemo$ ls
vagrant@precise32:~/gitdemo$ git init
Initialized empty Git repository in /home/vagrant/gitdemo/.git/
vagrant@precise32:~/gitdemo$ ls
vagrant@precise32:~/gitdemo$ ls -a
.  ..  .git
vagrant@precise32:~/gitdemo$ _
```

Step 3: Create a file and execute git status

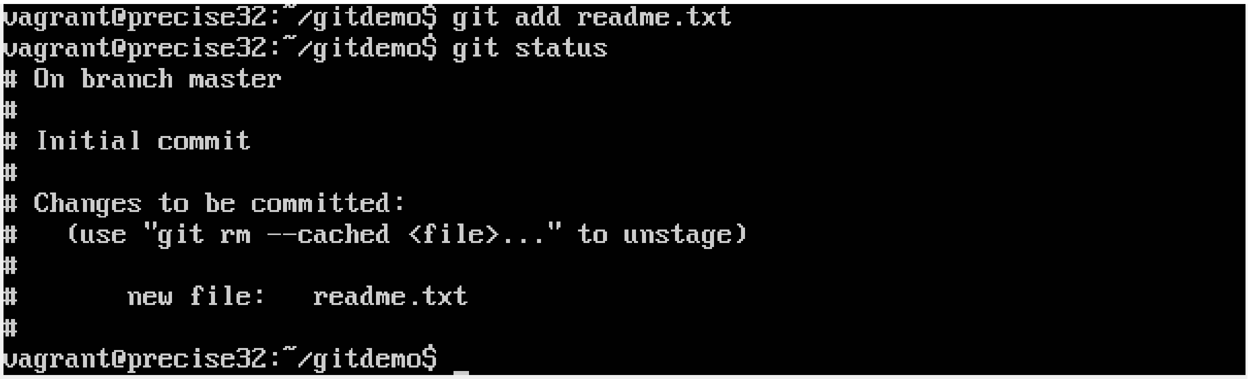
```
touch readme.txt
git status
```



```
vagrant@precise32:~/gitdemo$ git status
# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       readme.txt
nothing added to commit but untracked files present (use "git add" to track)
vagrant@precise32:~/gitdemo$
```

Step 4: Stage the newly created file using “git add” command

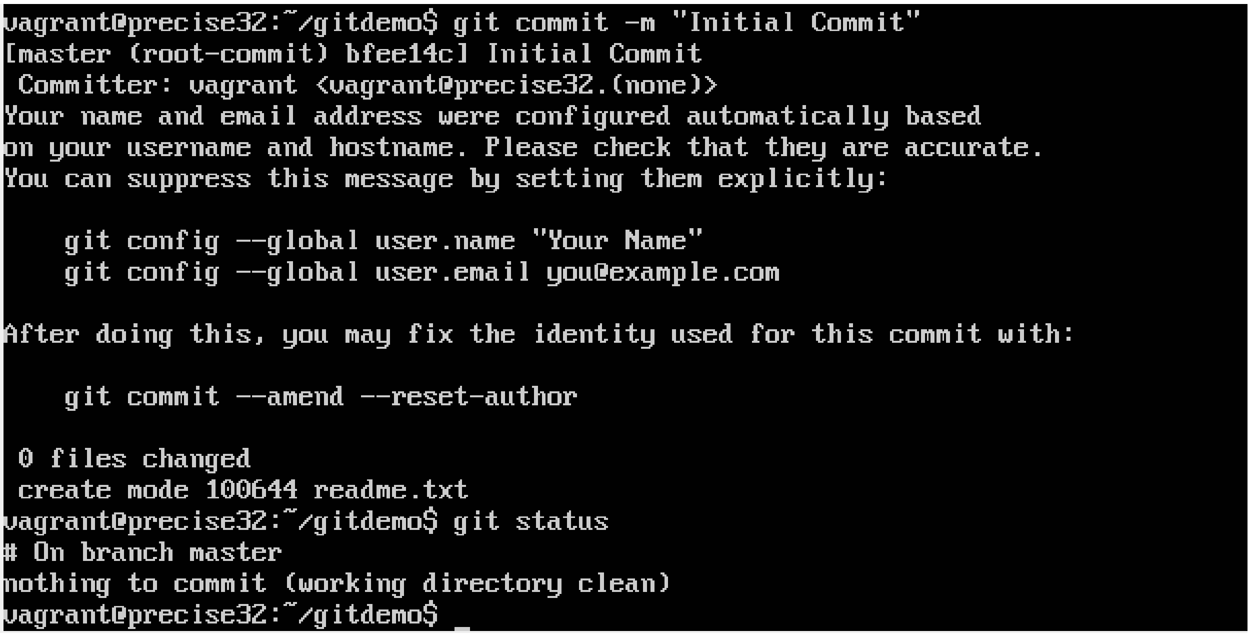
```
git add readme.txt
git status
```



```
vagrant@precise32:~/gitdemo$ git add readme.txt
vagrant@precise32:~/gitdemo$ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   readme.txt
#
vagrant@precise32:~/gitdemo$ _
```

Step 5: Commit the file

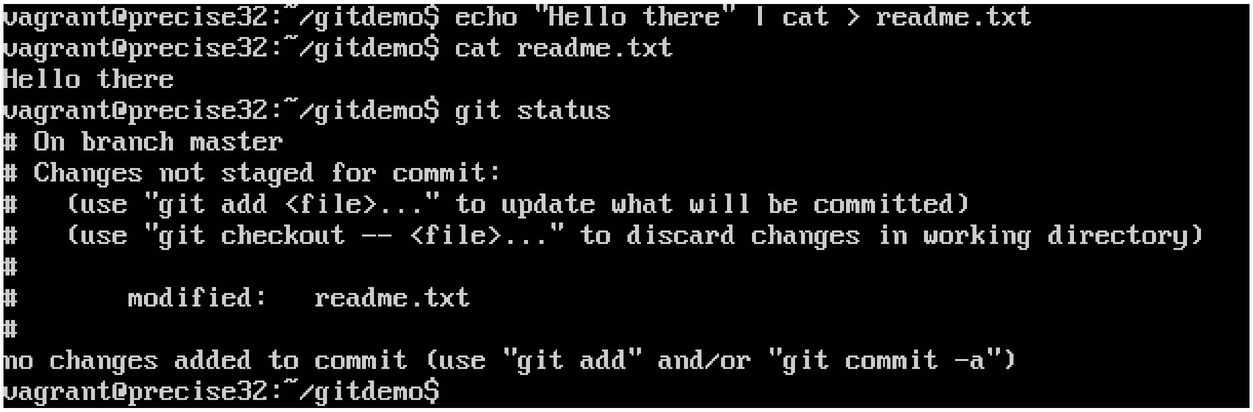
```
git commit –m “Initial Commit”
git status
```



```
vagrant@precise32:~/gitdemo$ git commit -m "Initial Commit"
[master (root-commit) bfee14c] Initial Commit
 Committer: vagrant <vagrant@precise32.(none)>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 0 files changed
 create mode 100644 readme.txt
vagrant@precise32:~/gitdemo$ git status
# On branch master
nothing to commit (working directory clean)
vagrant@precise32:~/gitdemo$ _
```

Step 6: Make some random change to the file readme.txt

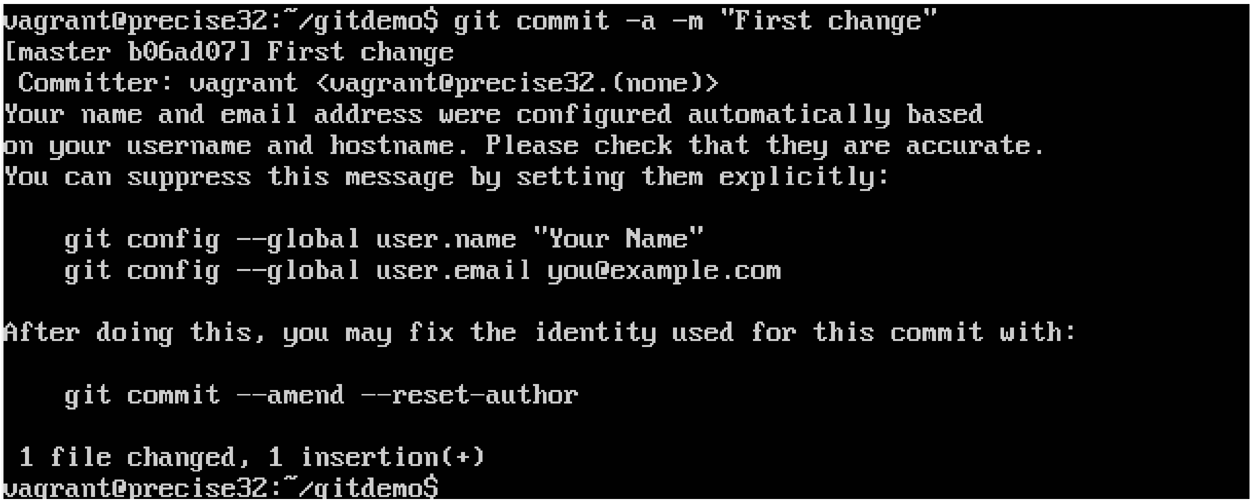
```
echo “Hello there” | cat > readme.txt
cat readme.txt
git status
```



```
vagrant@precise32:~/gitdemo$ echo "Hello there" | cat > readme.txt
vagrant@precise32:~/gitdemo$ cat readme.txt
Hello there
vagrant@precise32:~/gitdemo$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   readme.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
vagrant@precise32:~/gitdemo$
```

Step 7: Commit with “-a” option.

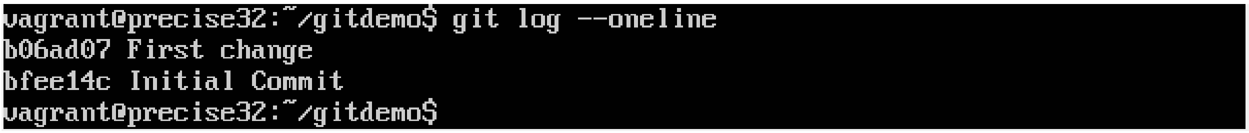
```
git commit –a –“First Change”
```



```
vagrant@precise32:~/gitdemo$ git commit -a -m "First change"
[master b06ad07] First change
 Committer: vagrant <vagrant@precise32.(none)>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+)
vagrant@precise32:~/gitdemo$
```

Step 8: Git log

```
git commit –a –“First Change”
```



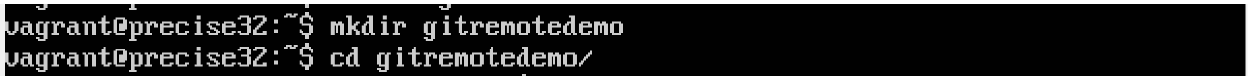
```
vagrant@precise32:~/gitdemo$ git log --oneline
b06ad07 First change
bfee14c Initial Commit
vagrant@precise32:~/gitdemo$
```

2. Git Remote repository

Create and use git remote repository

Step 1: Create a new directory called gitremotedemo under you home directory

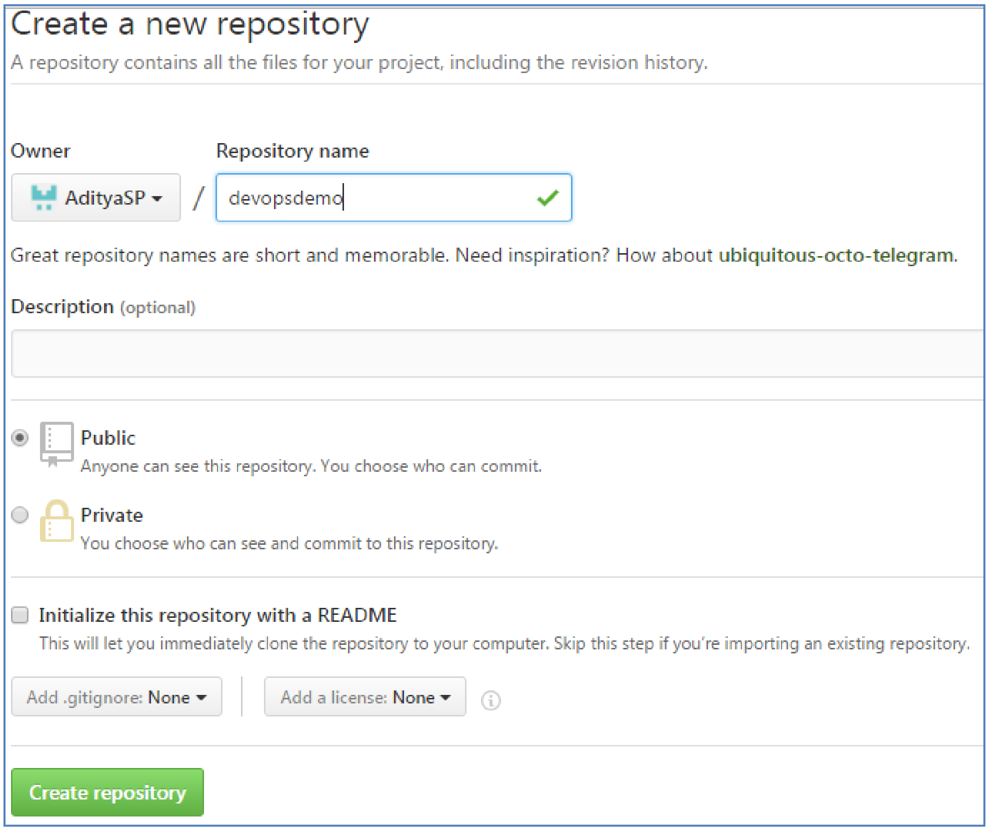
```
cd
mkdir gitremotedemo
cd gitremotedemo
cp -r ~/devopsfoundation/code/* .
```



```
vagrant@precise32:~$ mkdir gitremotedemo
vagrant@precise32:~$ cd gitremotedemo/
```

Step 2: Create a remote git repository

1. Open a browser on your windows machine
2. Create an account on www.github.com and login
3. Create a new repository by clicking on + on the top right
4. Give the repository a name . say "devopsdemo"



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: AdityaSP / Repository name: devopsdemo ✓

Great repository names are short and memorable. Need inspiration? How about **ubiquitous-octo-telegram**.

Description (optional)

● Public
Anyone can see this repository. You choose who can commit.

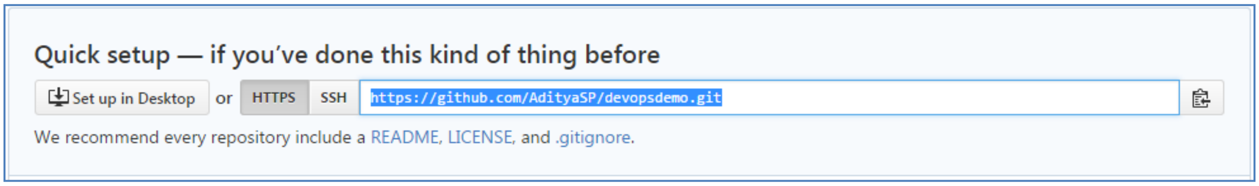
○ Private
You choose who can see and commit to this repository.

□ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None ▾ | Add a license: None ▾ ⓘ

Create repository

Step 3: Copy the https url of your repository



Quick setup — if you've done this kind of thing before

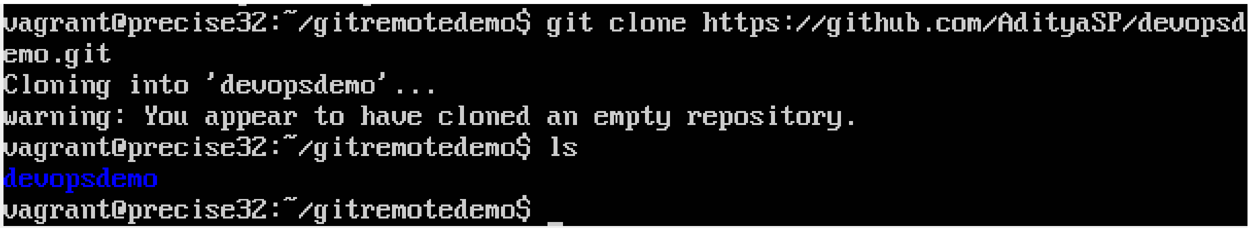
Set up in Desktop or HTTPS SSH https://github.com/AdityaSP/devopsdemo.git

We recommend every repository include a README, LICENSE, and .gitignore.

Step 4: Pull from remote git

Switch back to your VM and execute the below command

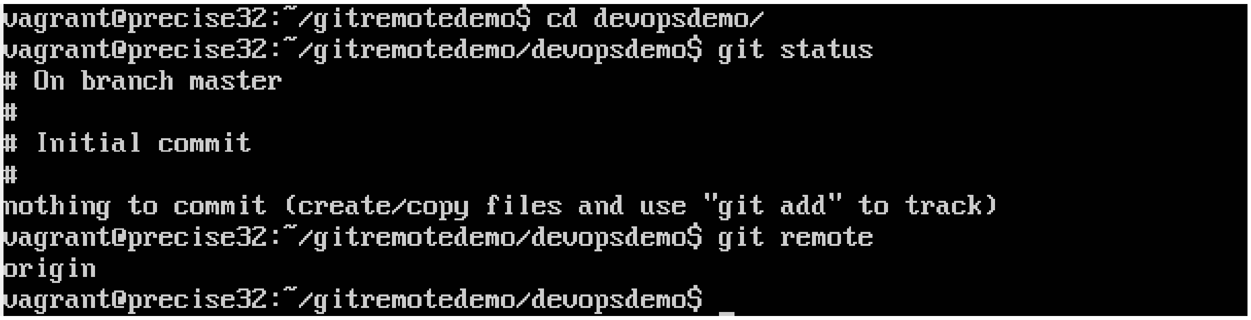
```
git clone https://github.com/<youraccount>/devopsdemo.git
ls
```



```
vagrant@precise32:~/gitremotedemo$ git clone https://github.com/AdityaSP/devopsd
emo.git
Cloning into 'devopsdemo'...
warning: You appear to have cloned an empty repository.
vagrant@precise32:~/gitremotedemo$ ls
devopsdemo
vagrant@precise32:~/gitremotedemo$ _
```

Step 5: cd into devopsdemo and execute git status

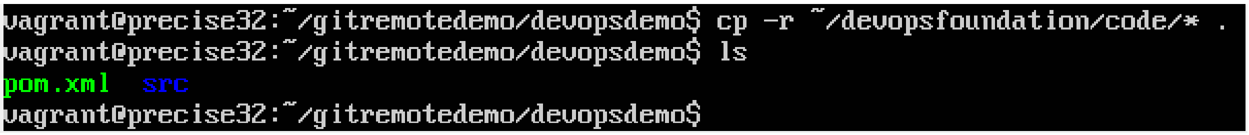
```
cd devopsdemo
git status
git remote
```



```
vagrant@precise32:~/gitremotedemo$ cd devopsdemo/
vagrant@precise32:~/gitremotedemo/devopsdemo$ git status
# On branch master
#
# Initial commit
#
nothing to commit (create/copy files and use "git add" to track)
vagrant@precise32:~/gitremotedemo/devopsdemo$ git remote
origin
vagrant@precise32:~/gitremotedemo/devopsdemo$ _
```

Step 6 : Copy code given. In the home directory of the vagrant vm there is a folder called devopsfoundation which contains material required for this session. Copy the contents of code into the git remote repository folder

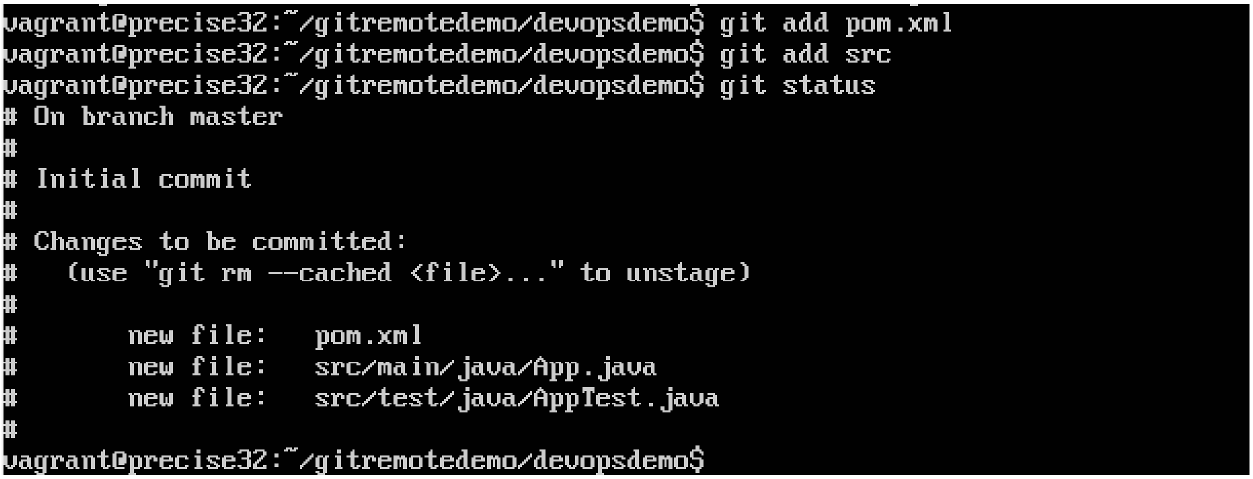
```
cp -r ~/devopsfoundation/code/* .
ls
```



```
vagrant@precise32:~/gitremotedemo/devopsdemo$ cp -r ~/devopsfoundation/code/* .
vagrant@precise32:~/gitremotedemo/devopsdemo$ ls
pom.xml  src
vagrant@precise32:~/gitremotedemo/devopsdemo$
```

Step 7: Add the newly added code files and add them to git by using the already familiar command “git add”

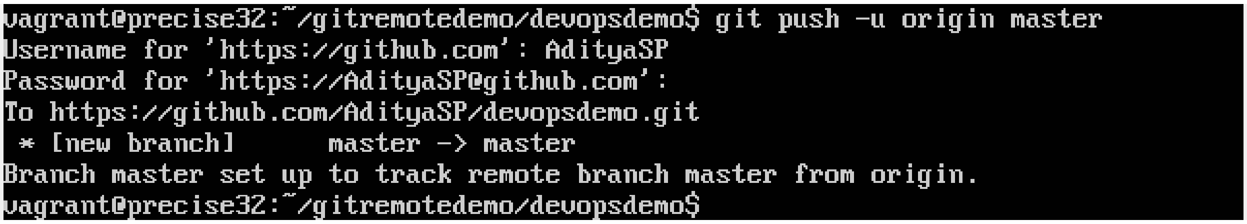
```
git add pom.xml
git add src
git status
```



```
vagrant@precise32:~/gitremotedemo/devopsdemo$ git add pom.xml
vagrant@precise32:~/gitremotedemo/devopsdemo$ git add src
vagrant@precise32:~/gitremotedemo/devopsdemo$ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   pom.xml
#       new file:   src/main/java/App.java
#       new file:   src/test/java/AppTest.java
#
vagrant@precise32:~/gitremotedemo/devopsdemo$
```

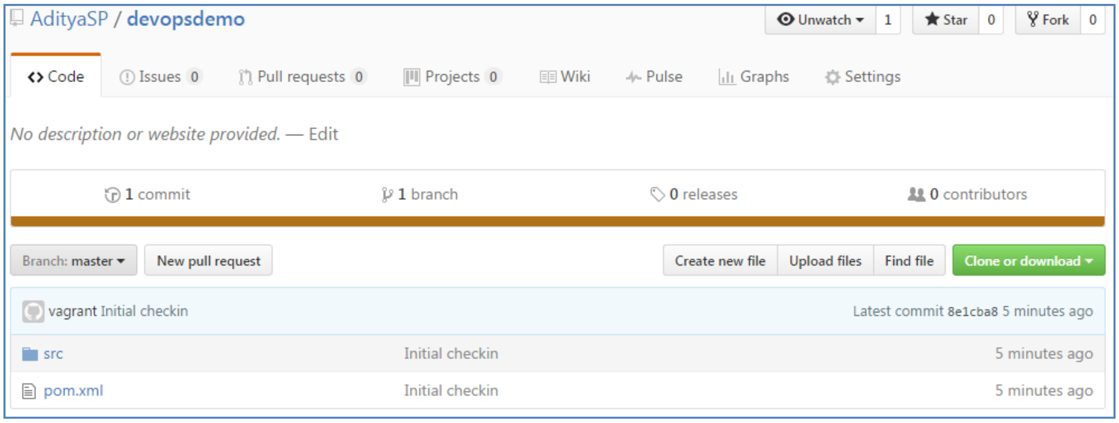
Step 8 : Execute the below commands

```
git commit -m “Initial Commit”
git push -u origin master
```



```
vagrant@precise32:~/gitremotedemo/devopsdemo$ git push -u origin master
Username for 'https://github.com': AdityaSP
Password for 'https://AdityaSP@github.com':
To https://github.com/AdityaSP/devopsdemo.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
vagrant@precise32:~/gitremotedemo/devopsdemo$
```

Step 9: check your repository on github by going to https://www.github.com/<your account>/devopsdemo . You should find the files that you uploaded in step 8



AdityaSP / devopsdemo

No description or website provided. — Edit

1 commit 1 branch 0 releases 0 contributors

vagrant Initial checkin Latest commit 8e1cba8 5 minutes ago

src	Initial checkin	5 minutes ago
pom.xml	Initial checkin	5 minutes ago