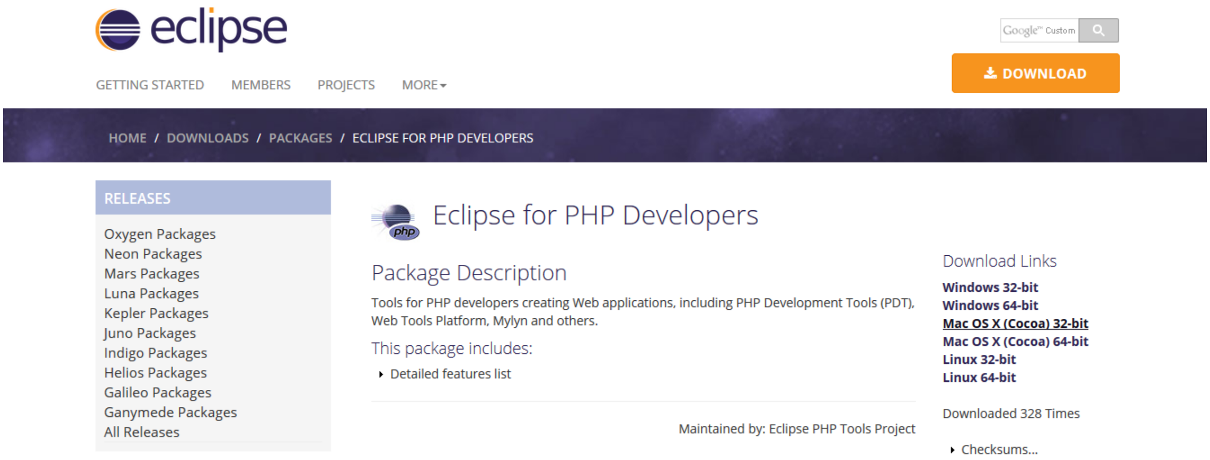
Installing Eclipse

1) Download Eclipse from the following link with your system requirement

   http://www.eclipse.org/downloads/packages/eclipse-php-developers/heliossr2



Eclipse for PHP Developers

Package Description

Tools for PHP developers creating Web applications, including PHP Development Tools (PDT), Web Tools Platform, Mylyn and others.

This package includes:

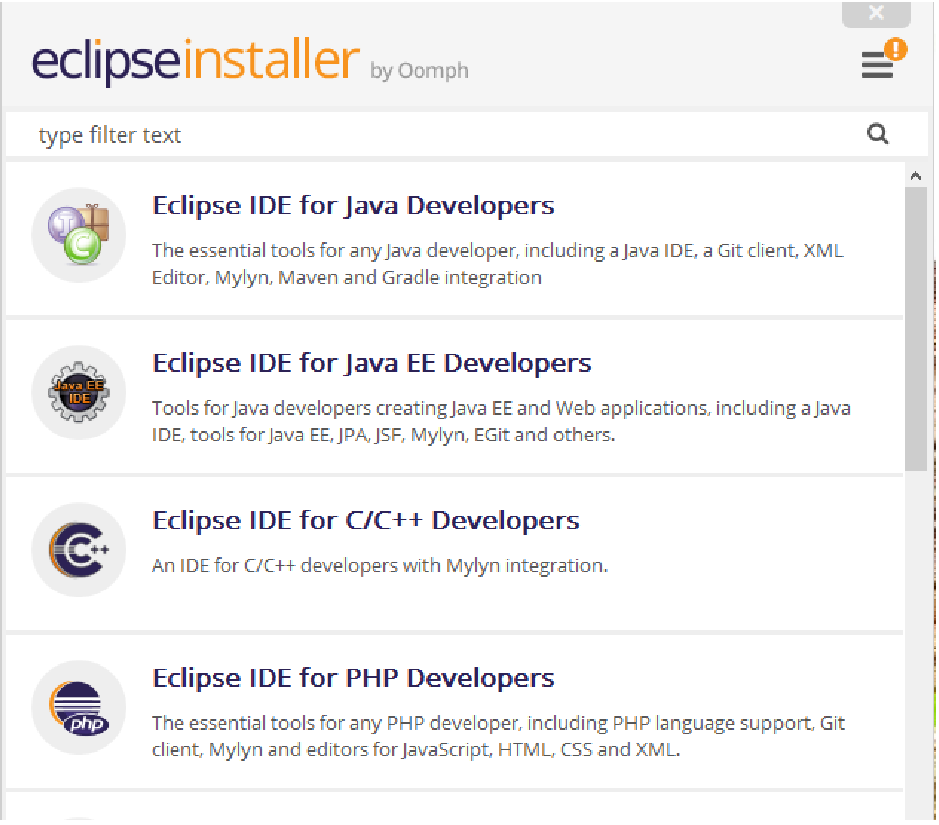
- Detailed features list

Maintained by: Eclipse PHP Tools Project

Downloaded 328 Times

2) After Download install the setup file by double click on it

3) Choose “Eclipse IDE for PHP Developers”



Eclipse IDE for Java Developers

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration

Eclipse IDE for Java EE Developers

Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn, EGit and others.

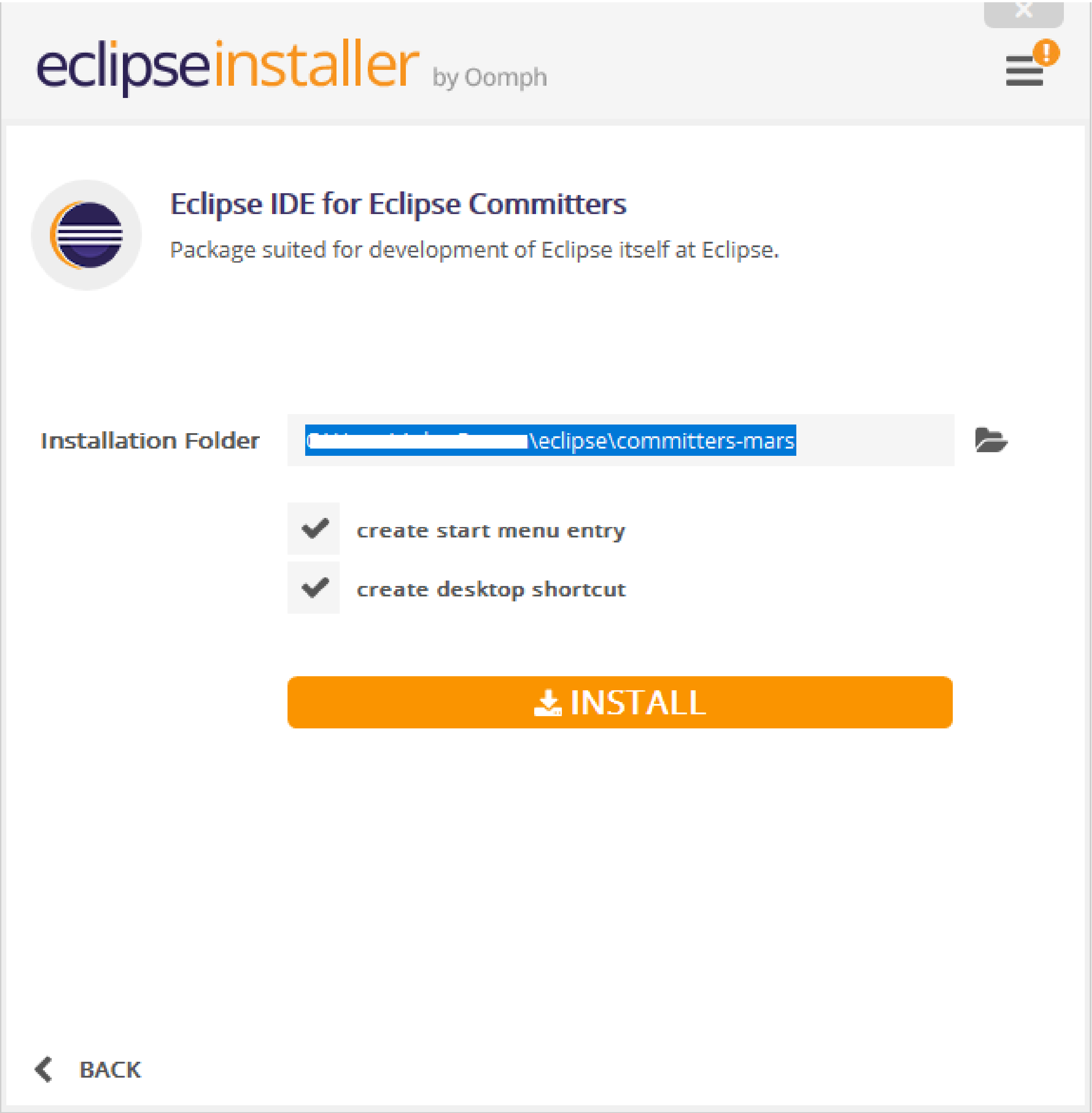
Eclipse IDE for C/C++ Developers

An IDE for C/C++ developers with Mylyn integration.

Eclipse IDE for PHP Developers

The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, HTML, CSS and XML.

4) Choose the installation path



Eclipse IDE for Eclipse Committers

Package suited for development of Eclipse itself at Eclipse.

Installation Folder [redacted]\eclipse\committers-mars

☑ create start menu entry

☑ create desktop shortcut

5) Install the setup

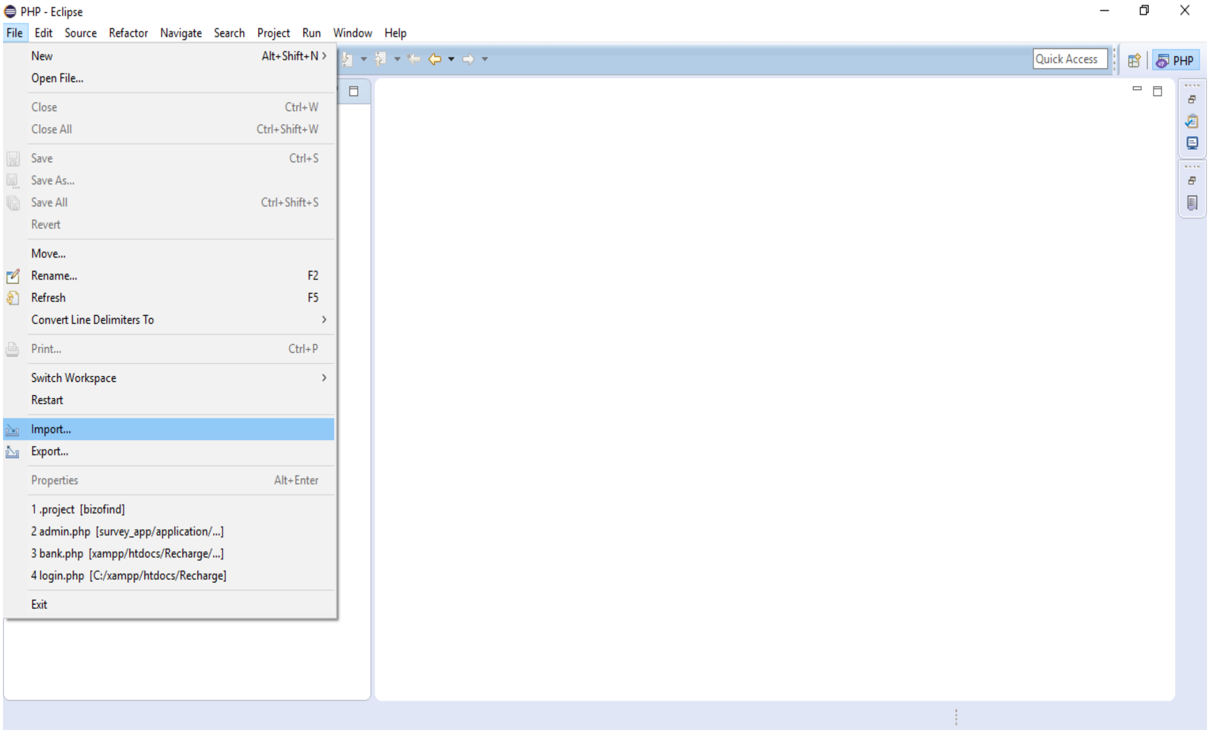
6) Now you are successfully installed eclipse

Git Operations through Eclipse

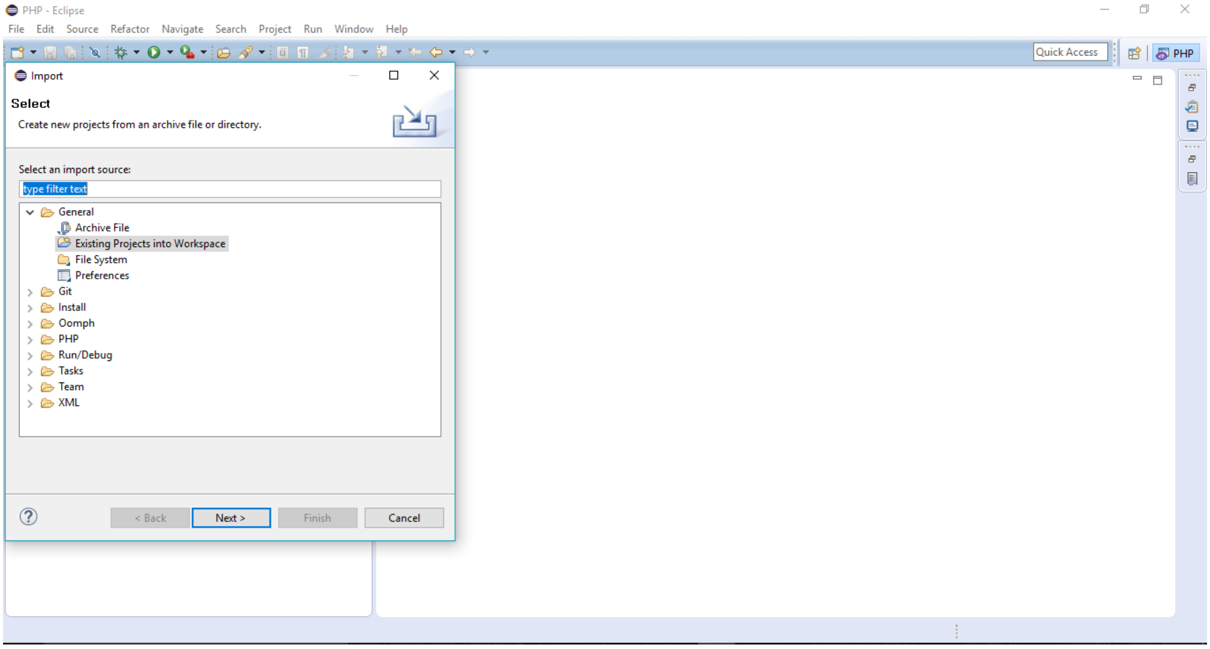
1) Open Eclipse and Import project to eclipse, you need .project file to import project to eclipse

[.project file]()

```
<?xml version="1.0" encoding="UTF-8"?>
<projectDescription>
    <name>Project Name</name>
    <comment></comment>
    <projects>
    </projects>
    <buildSpec>
        <buildCommand>
            <name>org.eclipse.wst.common.project.facet.core.builder</name>
            <arguments>
            </arguments>
        </buildCommand>
        <buildCommand>
            <name>org.eclipse.wst.validation.validationbuilder</name>
            <arguments>
            </arguments>
        </buildCommand>
        <buildCommand>
            <name>org.eclipse.dltk.core.scriptbuilder</name>
            <arguments>
            </arguments>
        </buildCommand>
    </buildSpec>
    <natures>
        <nature>org.eclipse.php.core.PHPNature</nature>
        <nature>org.eclipse.wst.common.project.facet.core.nature</nature>
    </natures>
</projectDescription>
```



2) Click on import option then choose ‘existing project into workspace’ option from opened window



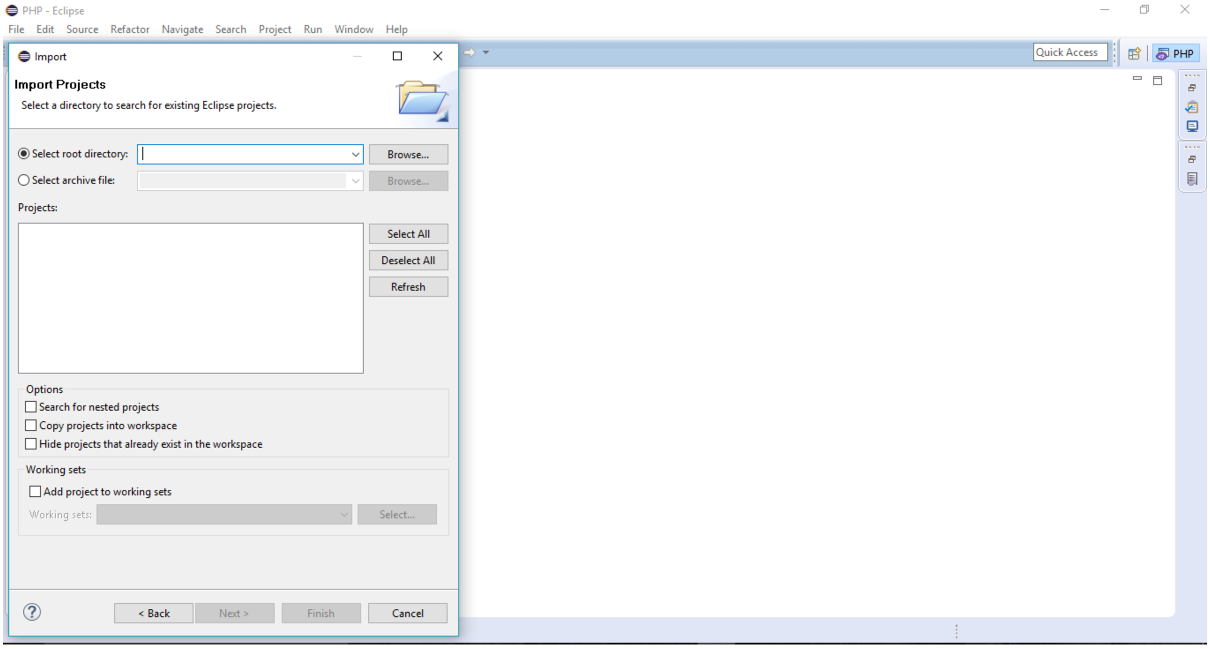
Select

Create new projects from an archive file or directory.

Select an import source:

type filter text

3) Browse and select the project you want to import


PHP - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Import
Import Projects
Select a directory to search for existing Eclipse projects.
● Select root directory:
Browse...
○ Select archive file:
Browse...
Projects:
Select All
Deselect All
Refresh
Options
□ Search for nested projects
□ Copy projects into workspace
□ Hide projects that already exist in the workspace
Working sets
□ Add project to working sets
Working sets:
Select...
< Back
Next >
Finish
Cancel
Quick Access
PHP

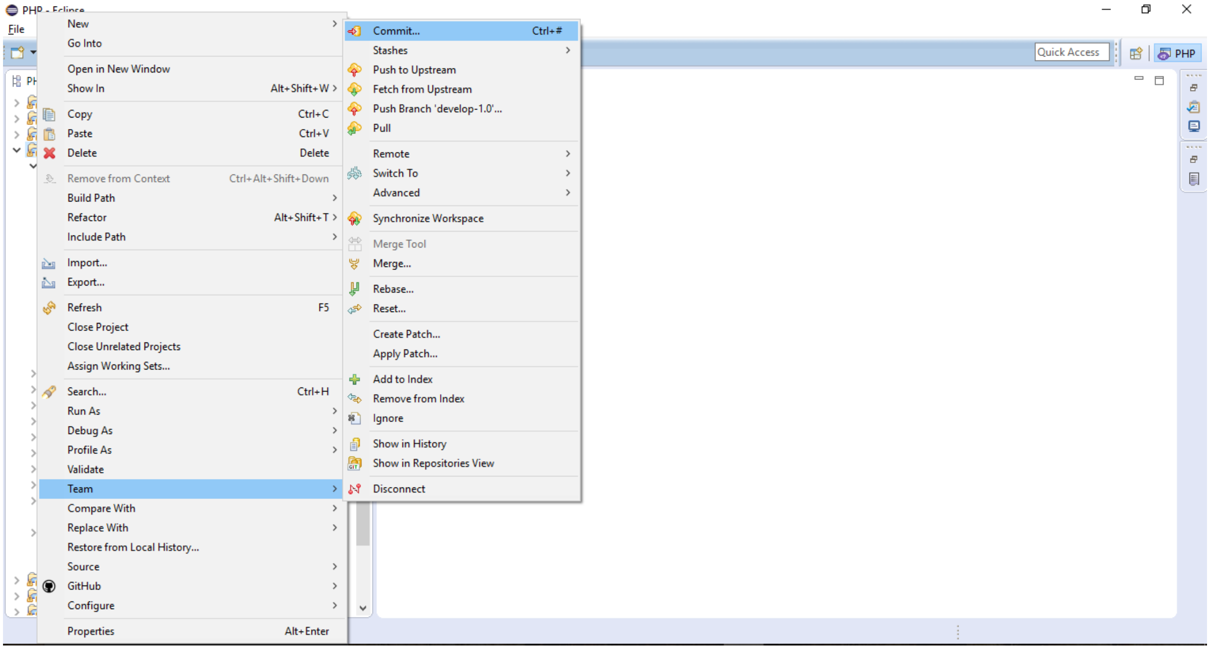
4) Now the project will be on eclipse

5) After make changes push the changes to the repository were we created on git

(That git settings already included in the .git folder under the project directory)

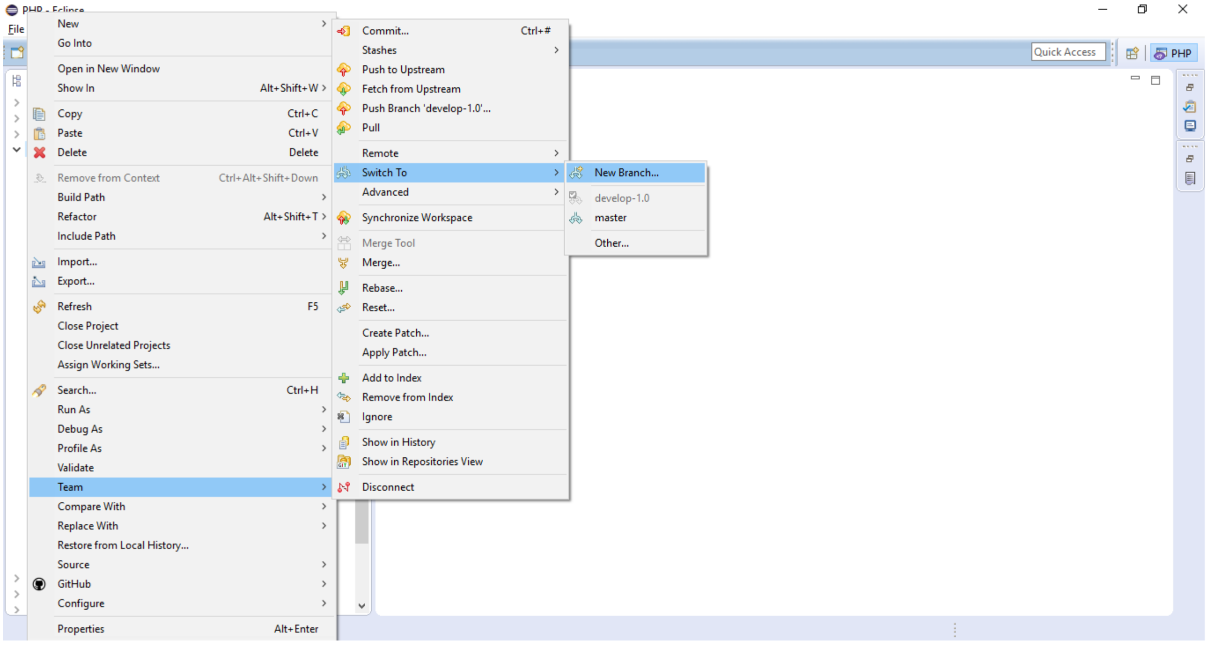
6) Commit the changes

a) If branch created in the git and push the code first time then directly commit and push the codes

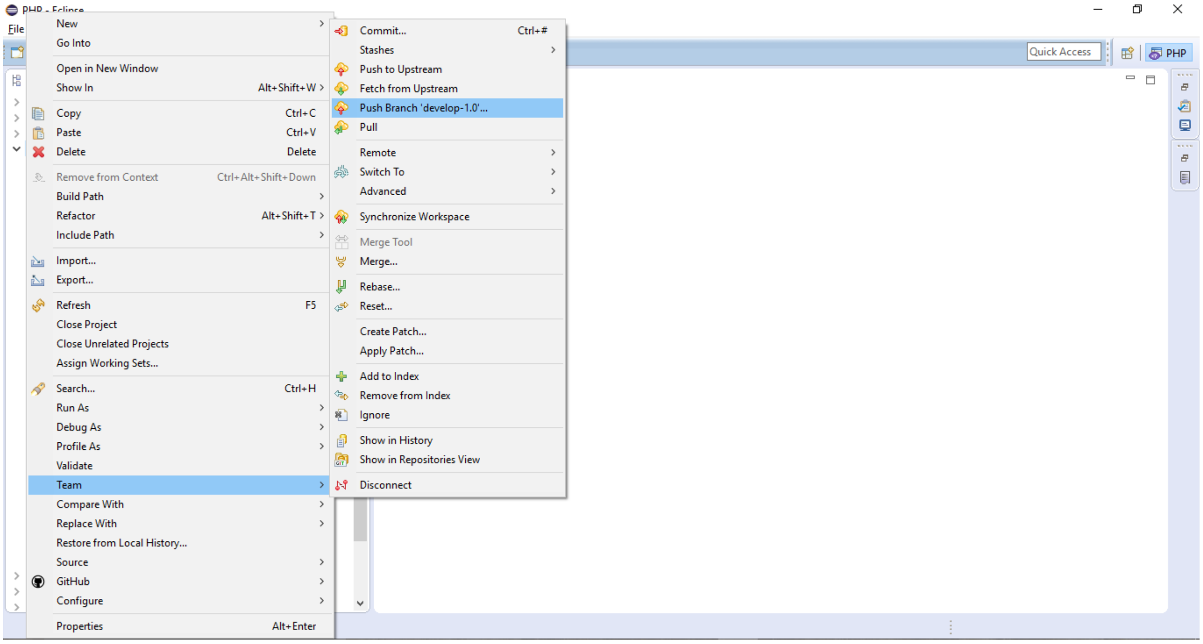


Create branch from local machine and push branch & code to git

1) Create branch (When creating a branch the whole code automatically check out to the newly created branch, then the project's branch will be the newly created branch)



2) After Create branch ,push created branch to git



Generating Public key for Git Authentication

Generating Your SSH Public Key using Keygen

Git servers authenticate using SSH public keys. In order to provide a public key, each user in your system must generate one if they don't already have one. This process is similar across all operating systems. First, you should check to make sure you don't already have a key. By default, a user's SSH keys are stored in that user's **`~/.ssh`** directory. You can easily check to see if you have a key already by going to that directory and listing the contents

```
$ cd ~/.ssh
$ ls
authorized_keys2  id_dsa       known hosts
config            id_dsa.pub
```

1) If you have this .ssh directory then you were already created a public key, check if that folder contain key pairs here,

   You're looking for a pair of files named something like **id_dsa** or **id_rsa** and a matching file with a `.pub` extension. The `.pub` file is your public key, and the other file is your private key. If you don't have these files (or you don't even have a **`.ssh`** directory), you can create them by running a program called **`ssh-keygen`**, which is provided with the SSH package on Linux/Mac systems and comes with Git for Windows:

   a) Open command prompt, then run the command

   ```
   $ ssh-keygen
   ```

   b) Give path where your private key wants to save, normally its save under the user's directory with .ssh name folder

   ```
   Generating public/private rsa key pair.
   Enter file in which to save the key
   (/home/User/.ssh/id_rsa):
   Created directory '/home/User/.ssh'.
   ```

   c) Enter passphrase, which you can leave empty if you don't want to type a password when you use the key.

   ```
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   ```

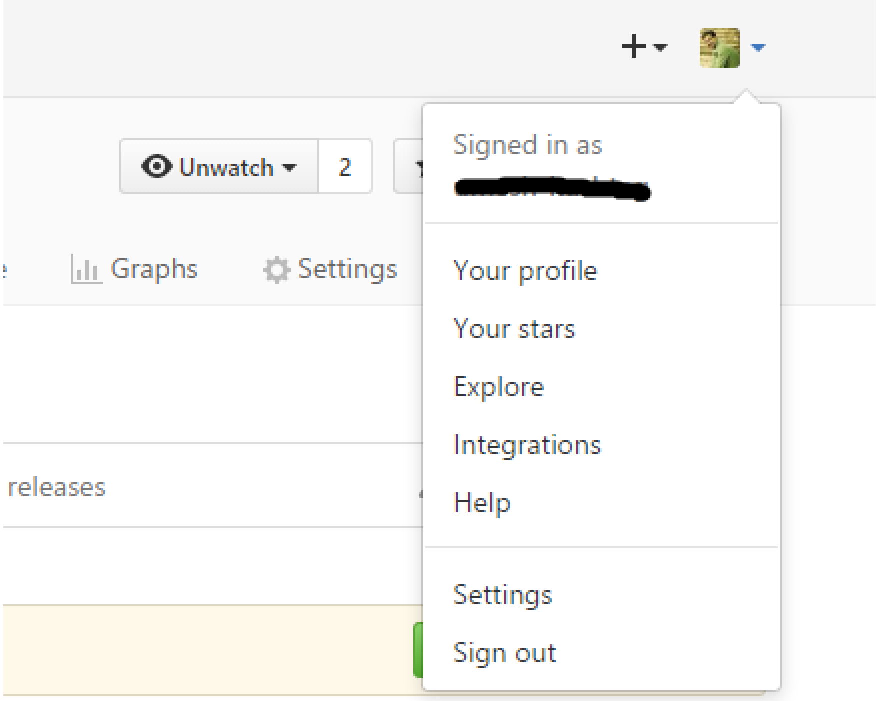
   d) Then your private/public keys has been saved to your preffered location

```
Your identification has been saved in /home/User/.ssh/id_rsa.
Your public key has been saved in /home/User/.ssh/id_rsa.pub.
The key fingerprint is:
d0:82:24:8e:d7:f1:bb:9b:33:53:96:93:49:da:9b:e3 username@ipaddress
```

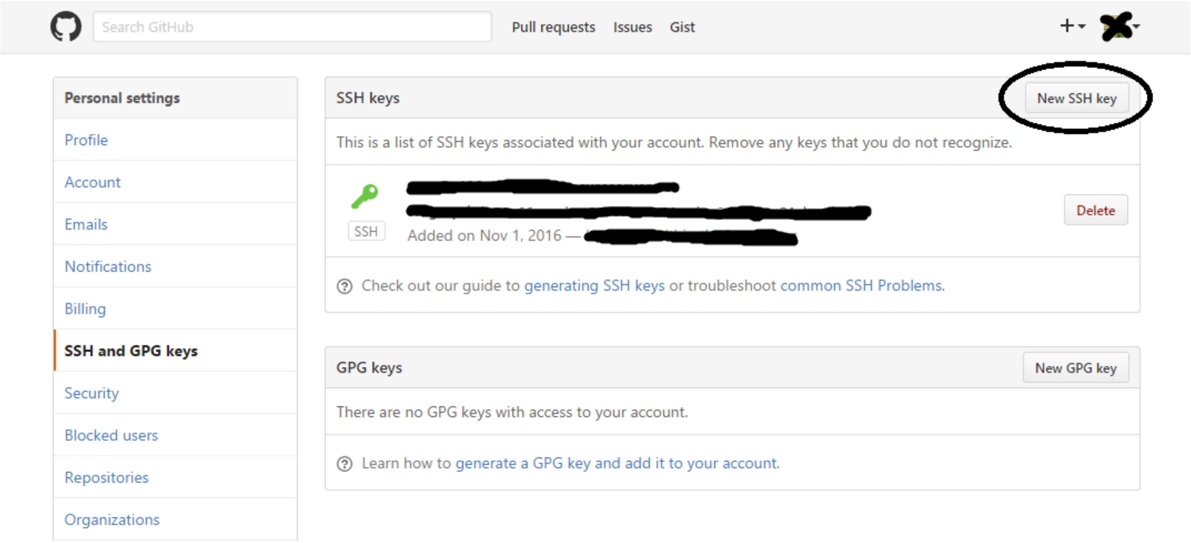
   e) All they have to do is copy the contents of the .pub file and email it. The public keys look something like this:

```
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAklOUpkDHrfHY17SbrmTIpNLTGK9Tjom/BWDSU
GPl+nafzlHDTYW7hdI4yZ5ew18JH4JW9jbhUFrviQzM7xlELEVf4h9lFX5QVkbPppSwg0cda3
Pbv7kOdJ/MTyBlWXFCR+HAo3FXRitBqxiX1nKhXpHAZsMciLq8V6RjsNAQwdsdMFvSlVK/7XA
t3FaoJoAsncM1Q9x5+3V0Ww68/eIFmb1zuUFljQJKprrX88XypNDvjYNby6vw/Pb0rwert/En
mZ+AW4OZPnTPI89ZPmVMLuayrD2cE86Z/il8b+gw3r3+1nKatmIkjn2so1d01QraTlMqVSsbx
NrRFi9wrf+M7Q== username@ipaddress
```

d) Open settings of your git account



e) Add New ssh key



This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

[redacted]

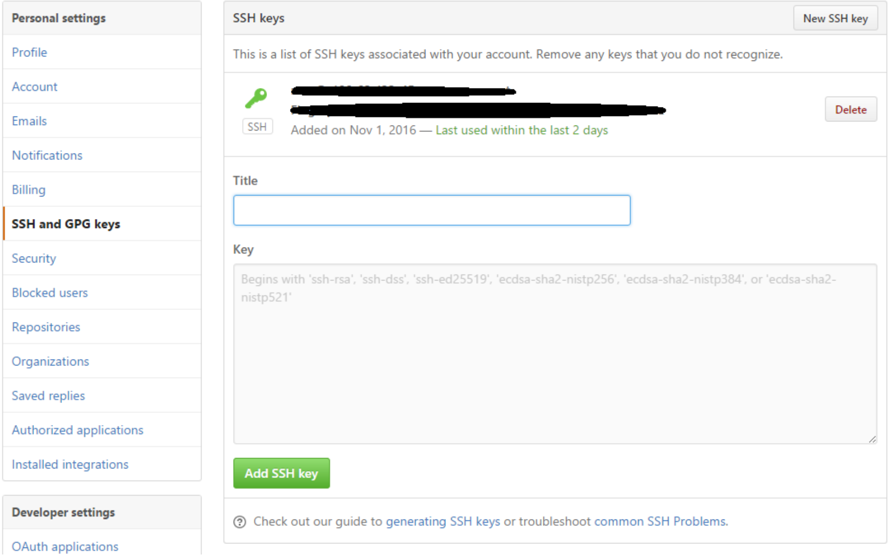
Added on Nov 1, 2016 — [redacted]

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

There are no GPG keys with access to your account.

Learn how to generate a GPG key and add it to your account.

f) Add Key



This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

[redacted]

Added on Nov 1, 2016 — Last used within the last 2 days

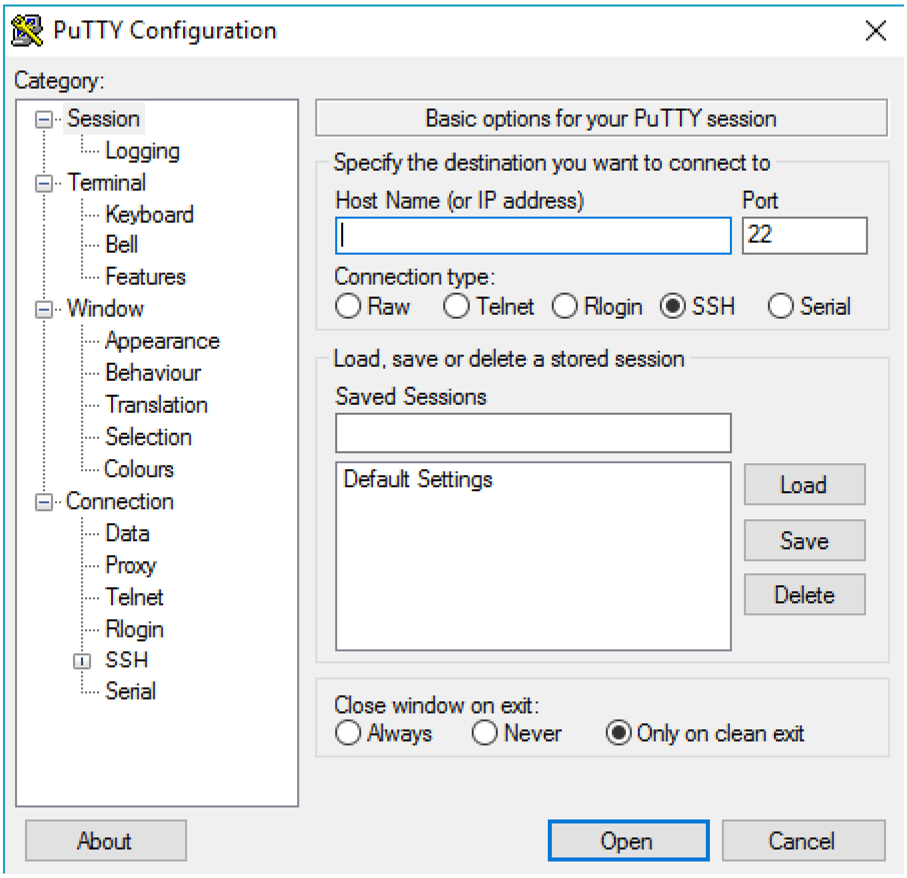
Key: Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

g) Replace the **http url** with **SSH url** on git directory in the project folder

Server Access via PUTTY

1) Download putty from the following link based on your system requirements
   http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
2) After download open putty.exe



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

3) Give the hostname (server IP address) other all options are same, no more change

4) Then you will be get into the server interface, then you can manage project codes

5) Now pushed codes from local can be pull down on the server using git command : **git pull**

Work Flow : When push the code to git from local ,it will go to the git repository and then after we sign in to the putty and pull the code on server, then code taken from git repository and pull to the server machine