

A horizontal decorative bar consisting of a teal segment on the left and an orange segment on the right.

# Pragmatic Product Leader Course

Become a Product Manager with Superpowers



# Lesson 14N

# Git & Github



# Git & GitHub - Version





# Understanding Git

- Git is a VCS - version control system
- Git helps to manage project files
- Git keeps track of the entire history of things you are working on
- This is extremely helpful for developers who start by building the basic version and build on those versions
- While building on the basic version, you try out new features, and this can prove to be extremely error prone
- In such situations, Git helps you to revert to previous versions of code
- Git also allows for great collaboration
- Through Git multiple developers can work on the same project simultaneously



# Understanding Github

- Github is a web-based service for version control using Git
- It is similar to a social networking site for developers
- You have the ability to view everyone's code, identify issues in their codes and suggest changes
- In simple terms Github is a hosting service for Git repositories



# Important Terminology



## **Repository**

Git will track changes to any file in this specified folder called the repository.

## **Branch**

This is an independent line of development. It is similar to a new directory.

## **Fork**

A personal copy of another person's repository that resides in your account.

## **Clone**

A copy of your repository that lives on your computer instead of the server.

## **Commit**

Set of one or more changes to a file. Everytime you save, these changes are assigned an ID. This ID helps to track the changes.

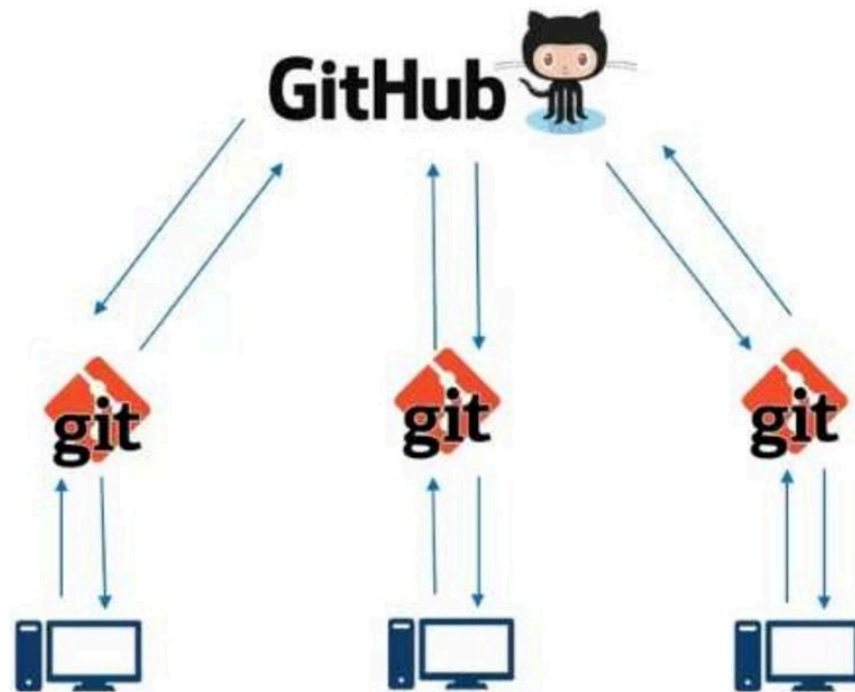
## **Master**

Default development branch



# Git & Github

- Github is the central repository
- Git is a tool that helps you create local repositories on your personal system





# Connecting to GitHub with SSH

- With the help of [SSH](#), you will be able to connect to GitHub without a username and password
- For this, you need to set up a SSH key and add that to your GitHub account
- Steps:
  - Generate SSH key
  - Add SSH key to SSh agent
  - Add SSH key to Github account



# Generate SSH key

Step 1: Open [Git Bash](#)

Step 2: Use the email associated with the Github account to write the following command:

```
$ ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

This command creates a new SSH key using your email as the label

```
Generating public/private rsa key pair.
```

Step 3: At this stage you will be asked to “Enter a file in which to save the key”. This sets the default file location

```
Enter a file in which to save the key (/c/Users/you/.ssh/id_rsa): [Press enter]
```

Step 4: Next you will be prompted to enter a secure passphrase

```
Enter passphrase (empty for no passphrase): [Type a passphrase]  
Enter same passphrase again: [Type passphrase again]
```



# Add SSH key to SSH-agent

Step 1: Ensure SSH-agent is running

Step 2: Replace *id\_rsa* in the command with the name of your private key file. (the name provided in step 3 in previous slide)

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

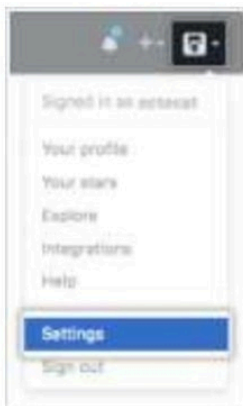


# Add SSH key to Github account

Step 1: Copy the SSH key to your clipboard using the following command

```
$ clip < ~/.ssh/id_rsa.pub  
# Copies the contents of the id_rsa.pub file to your clipboard
```

Step 2: In the upper right hand corner click on your profile photo and then click Settings





# Add SSH key to Github account - continued

Step 3: In the settings sidebar, click on SSH and GPG keys

Step 4: Click Add SSH key

Step 5: In the "Title" field, provide a descriptive label for you key

Step 6: Paste the key you copied to clipboard in the Key field

Step 7: Click "Add SSH key"

Step 8: You maybe prompted to confirm your GitHub password



# Let's Give it a Try!

Create your [GitHub Account](#)

```
generating public/private rsa key pair.
Enter file in which to save the key (/Users/testuser/.ssh/id_rsa):
Created directory '/Users/testuser/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/testuser/.ssh/id_rsa.
Your public key has been saved in /Users/testuser/.ssh/id_rsa.pub.
The key fingerprint is:
8a:7d:18:9c:8e:1e:06:1f:c4:7a:5c:17:3d:ef:8d:f8 testuser@franzers.io
The key's randomart image is:
---[ RSA 2048 ]---
+.  .0
..0... 0
+...+. 0
..++ .. .
.. 0 50 +
  0 0 E .
 . 0 .
 .
-----
bash-3.2$ ls ~/.ssh
id_rsa      id_rsa.pub
bash-3.2$ cat id_rsa.pub
cat: id_rsa.pub: No such file or directory
bash-3.2$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAgEjSxKLUVjDHRu9GqaFNIpTJ8h9KAJDOOPj186e567Huzs6DFHAKB39q0nc-WTDHFCy
mPKjy8RdczFjNadu02mF6aJfF9c389C3fF8wF8uPYhuc1p409CLpR0vUjT74TDLzrz73Uj0449V82jC7h0pY20dmA20LEFFAT/4dCzF
E2mYy8wYNGVY82228i:1y15wLl1h+J1zr6p0MALJ6K/+T2dfwFmGGHt38e118vFCuW7HfJQR7Nv8Z/8KjYk1A4830m6wLd081t/2e15
MVSj0B8B:81jA66/+E8wWT720K7C6d testuser@franzers.io
bash-3.2$
```



---

# To conclude

A horizontal decorative bar with a teal segment on the left and an orange segment on the right, positioned above the main text.

# Thank you.

