

# *Version Control and Git Basics*

Integration Engineer's  
Workshop

***Day 01 - Session 02***

11.00am to 01.00pm  
September 16, 2014

*Team: VLEAD, Virtual Labs  
IIIT-H*

# Agenda

- Understand Version Control
- GitHub for Version Control
- GitHub – Basic Commands

***Lets Git this party started!***

---

# Objectives

*At the end of the session you will be able to:*

- Realize the importance of version control
  - Have an account in GitHub
  - Have GitHub commands at finger tips
-

# Our Environment

Timely code releases

Multiple developers working together

Code Reverts and Changes

***Who is keeping track of changes?***

---

# Version Control

Version\_01

Ver. 5.0

The\_Final

Rough\_Copy

Final\_Final

Final\_copy

---

# My Search!

- We have “N” files to look
  - The file containing the required is deleted and doesn't exist anymore
-

# Version Control

- Log every version
  - Know the difference between each
  - Revert to required version
-

# What Else?

- Better Collaboration
  - Easy Transfer
  - Backup
-



# Activity 01:

Which of the following are used for version control?



# GitHub

- Since 2008
  - A code sharing and publishing service
  - A social networking site for programmers
-

# Get An Account



<http://www.github.com/>

---

# Get Git On your machine



`yum install git`  
(From your Terminal)

---

# Tell Git your name

```
git config --global user.name "YOUR NAME"
```

---

# Tell Git your Email

```
git config --global user.email "YOUR EMAIL  
ADDRESS"
```

**Note:**

The email you specify should be the same one you used to sign up for GitHub.

---

# Authenticate Your Account

- Generating an SSH key
  - Add the public key to your GitHub account
-

# Check for SSH keys

```
ls -al ~/.ssh
```

Lists the files in your .ssh directory, if they exist

---



# Generate a new SSH key

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

Creates a new ssh key, using the provided email as a label

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/you/.ssh/id\_rsa):

[Press enter]

---

# Enter the Passphrase

Enter passphrase (empty for no passphrase): [\[Type a passphrase\]](#)

Enter same passphrase again: [\[Type passphrase again\]](#)

---

# What you will see

Your identification has been saved in `/c/Users/you/.ssh/id_rsa`.

Your public key has been saved in `/c/Users/you/.ssh/id_rsa.pub`.

The key fingerprint is:

`01:0f:f4:3b:ca:85:d6:17:a1:7d:f0:68:9d:f0:a2:db your_email@example.com`

---

# Add your Key to ssh agent

start the ssh-agent in the background

```
eval "$(ssh-agent -s)"  
Agent pid 59566
```

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

---

# Add ssh key to GitHub

Copy the contents of the

`id_rsa.pub`

file to your clipboard

---

# Add ssh key to GitHub

`ssh -T git@github.com`

Put below lines in a file named config in .ssh directory

Host github.com

User git

Hostname ssh.github.com

PreferredAuthentications publickey

IdentityFile ~/.ssh/id\_rsa

Port 443

---

# Activity 02:

Create a public repository called 'My First Repo'.

How does the name appear?

Initialize a read me for the repo

Update the README

---

# Activity 03:

Add a Collaborator to your created repo

Restrict editing to collaborators only

Ask a question!

---



# Activity 04:

## Predict the commands

git status

git get

git command

git set

git push

git init

git post

git contribute

git clean

git pull

git commit

git help

git branch

git check

git log

git add

git end

git fetch

---

# Activity 04:

Predict the commands

git status

git get

git command

git set

git push

git init

git post

git contribute

git clean

git pull

git commit

git help

git branch

git check

git log

git add

git end

git fetch

---

# Activity 05:

Raise a issue.

Other contributors can help you in solving it!

---

# Activity 06:

Clone the created repo to your machine and start working

---

# What did we learn?

I don't know!

You got to tell me!

---

# Thank you.

== End of the Session ==

---