accenture technology

Learning and Knowledge Management

Module 2: Introduction to Git

Module Objectives

At the end of this module, you will be able to:

- Describe Git and its key features.
- Compare Git with other SCMs.
- Explain key concepts in Git.
- Apply the steps to install Git.
- Describe GitHub and creation of a repository.

Topic List

- Introduction to Git
- Key Concepts in Git
- Installation
- Introduction to Git Hub
- Exercise 2.1: Install Git

Topic List

- **Introduction to Git**
- Key Concepts in Git
- Installation
- Introduction to Git Hub
- Exercise 2.1: Install Git

Introduction to Git (1)

What is Git?

Git is free SCM software provided under the GNU general public license.

Each working directory of Git is a repository which includes the history with revision tracking thus maintaining versions.

It is independent of network/central server.

Editions can be done and saved without the same.

Introduction to Git (2)

What is Git?

Git – In Distributed Development

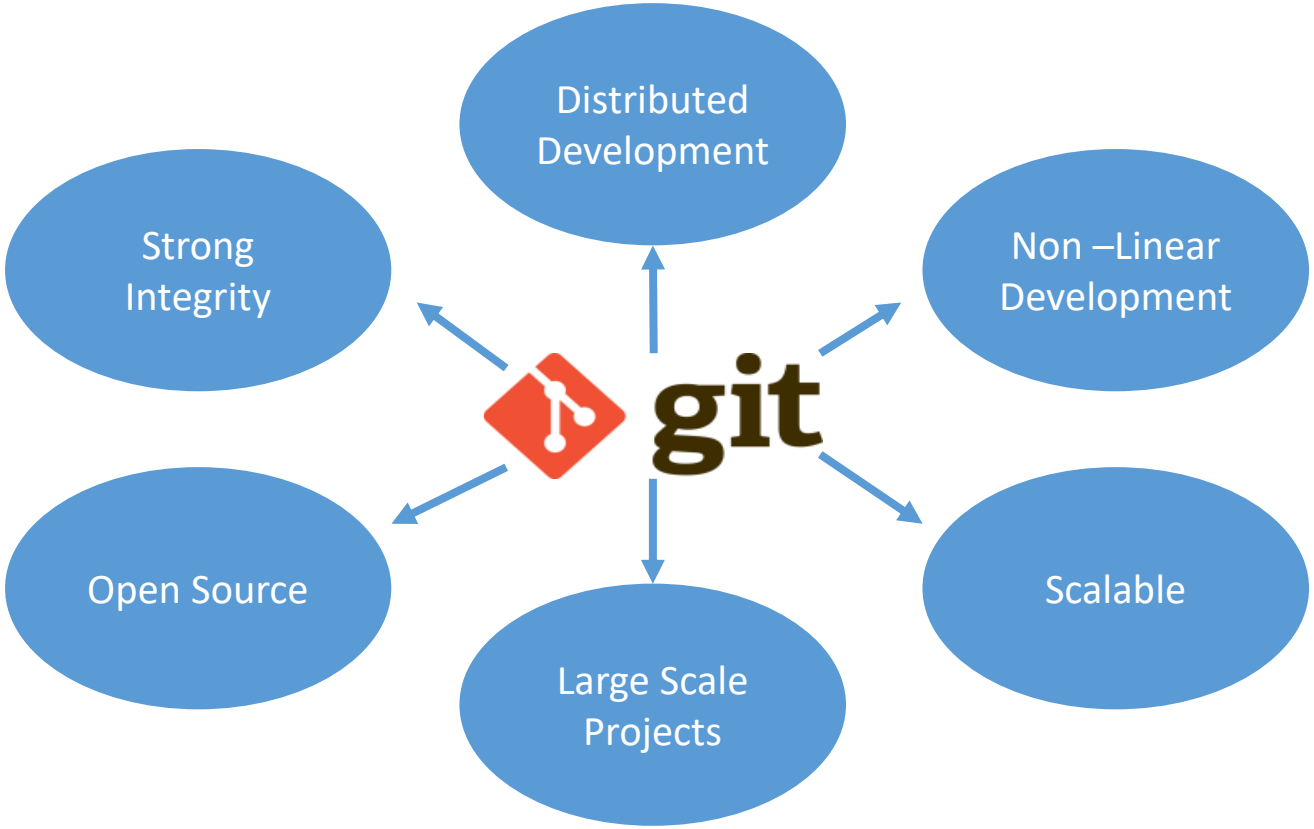
- Each developer has a local copy of the development history.
- Changes can be pushed or pulled from repository to repository copies.
- Servers can be replicated and distributed around the world if necessary.

- Performance tests shows that Git is much faster than other revision control systems.
- With increase in historic data there is no hampering effect on the project.

Introduction to Git (2)

Key Features of Git

Following are the key features of Git:



Introduction to Git (3)

Characteristics of Git

Following are the characteristics of Git:

Local Operation

- Git operations are done on local resources and files.

Integrity

- Before storing, all files in Git are check-summed.
- Files are referenced by checksum.
- For check summing, Git uses SHA-1 hash mechanism.
- Checksum is a string, generated based on the contents of a file or directory structure in Git.
- It consists of 40 hexadecimal characters (0–9 and a–f). Example of SHA-1 hash: 24b9da6552252987aa493b52f8696cd6d3b00373.

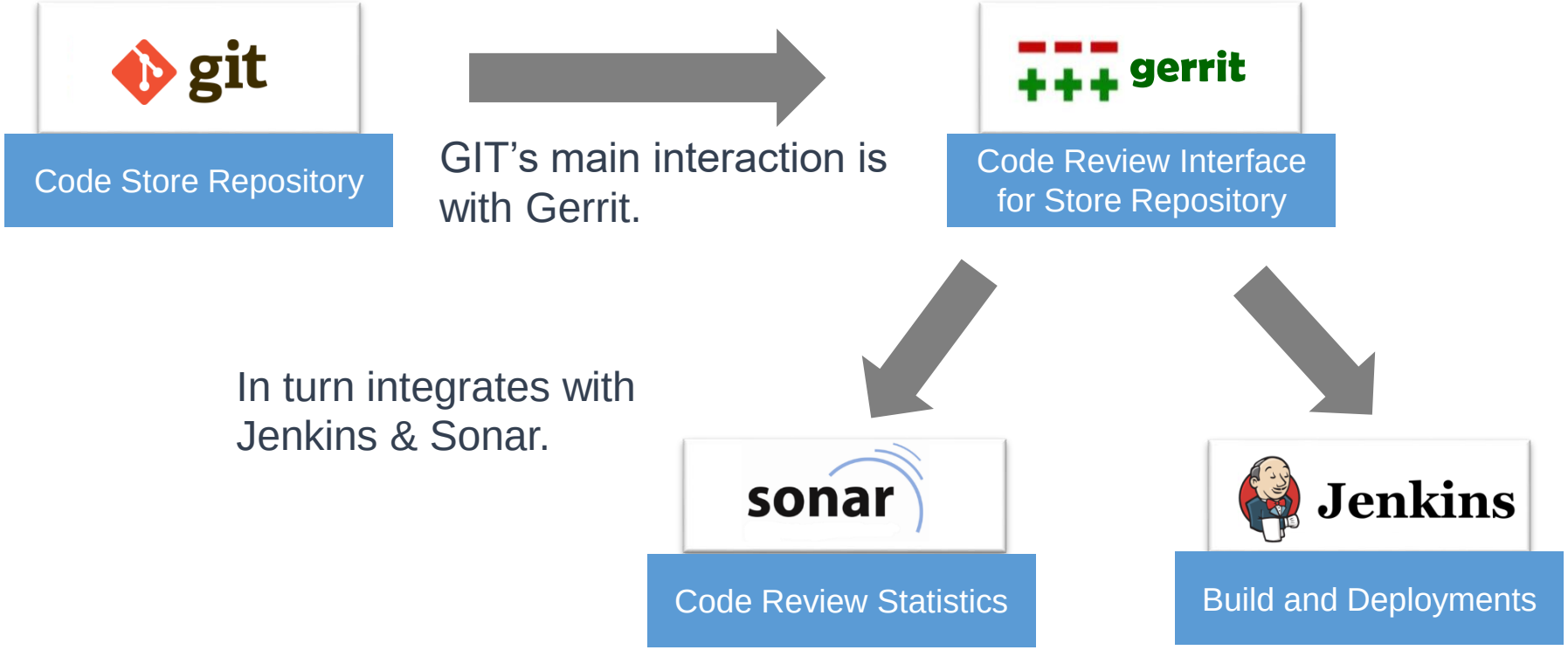
Git adds data

- Git avoids data mess up for the changes that are yet to be committed unlike other VCS;
- Once committed, a snapshot commits into Git.
- Thus data can’t be lost, when there are frequent pushes of content into original or another repository.

Introduction to Git (4)

Git with Other Tools in Development Environment

Following figure represents the Git's interactions with other tools:



Git compliments the additional tools to form a fast, agile developer environment.

Introduction to Git (5)

Difference between Git and Other SCM

Following tables compares Git with other SCMs:

Git	Other SCM
Data is stored in the form of a stream of snapshots.	Stores data as a list of file based changes.
Takes a snapshot whenever a change is performed or saved. If there are no changes made, instead of taking snapshot and storing the file again, Git stores a reference to the previous identical file.	These systems (CVS, Subversion, Perforce, Bazaar, and so on) consider the information maintained as a set of files Tracks the changes made to each file over time.

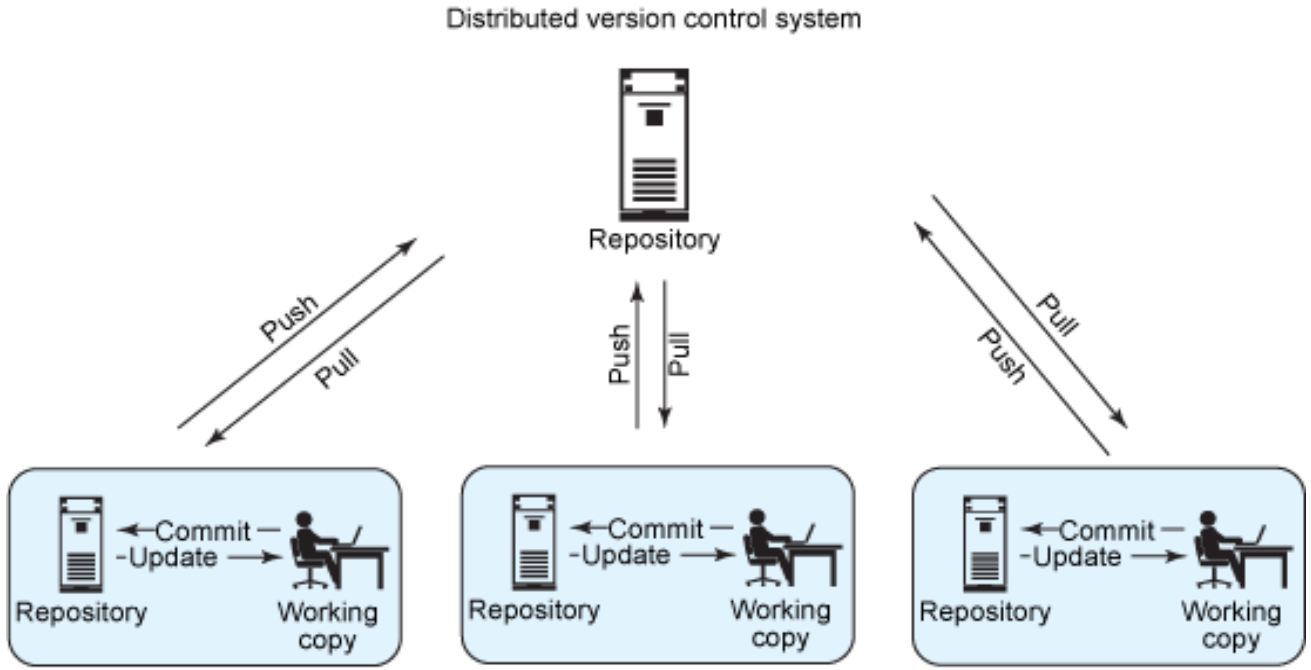
Topic List

- Introduction to Git
- **Key Concepts in Git**
- Installation
- Introduction to Git Hub
- Exercise 2.1: Install Git

Key Concepts in Git (1)

Git Clone Principle

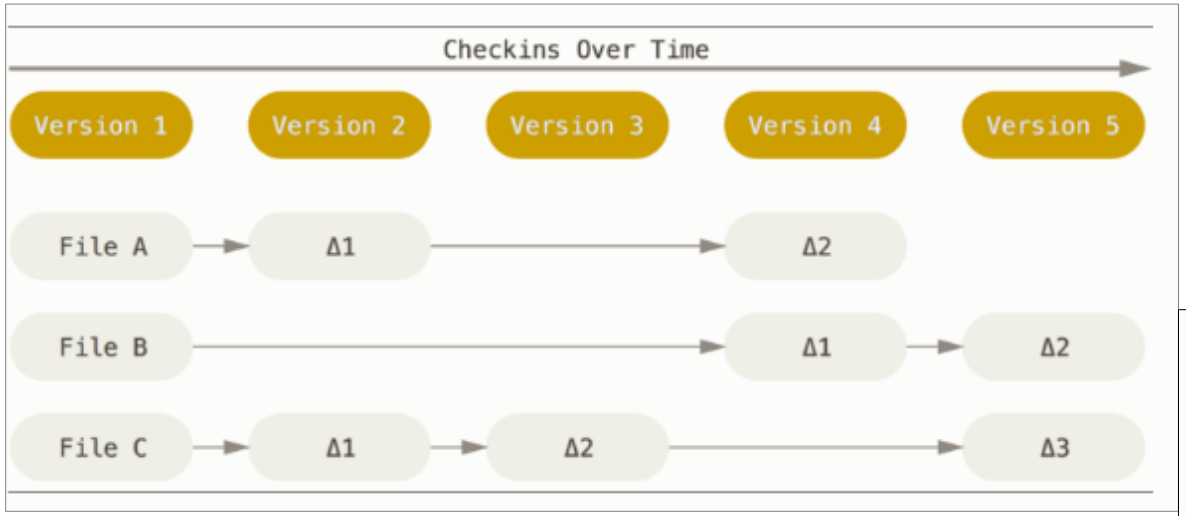
1) Every developer has a copy or clone of the central repository that contains a full copy of the repository including the history.
2) File changes and commits are performed locally.
3) Later, commits and changes are pushed to central repository.

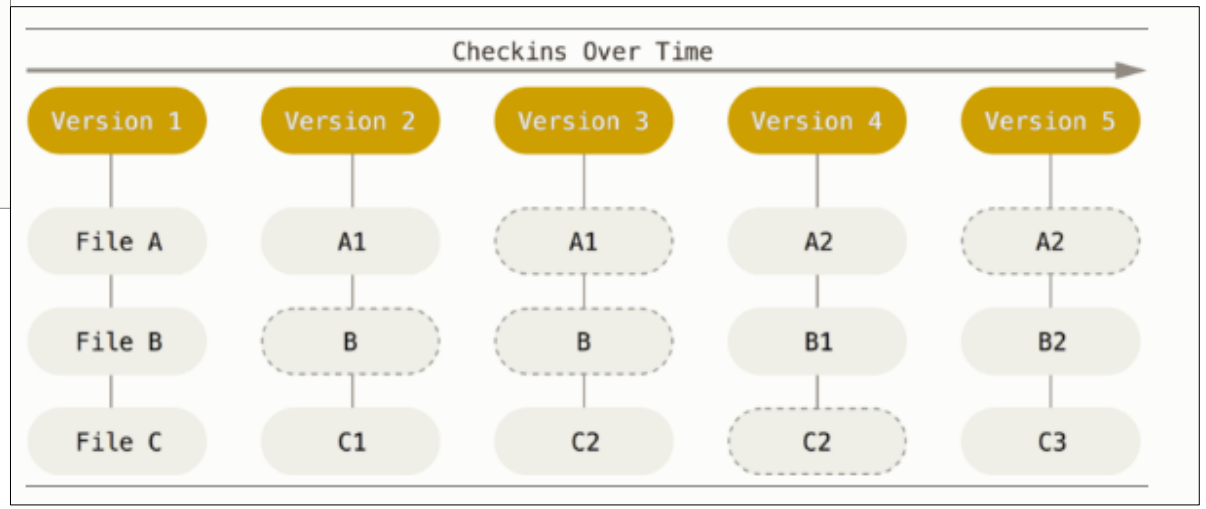


Key Concepts in Git (2)

Representation of Git Checkins

Following figure shows the representation of Git Checkins over time:





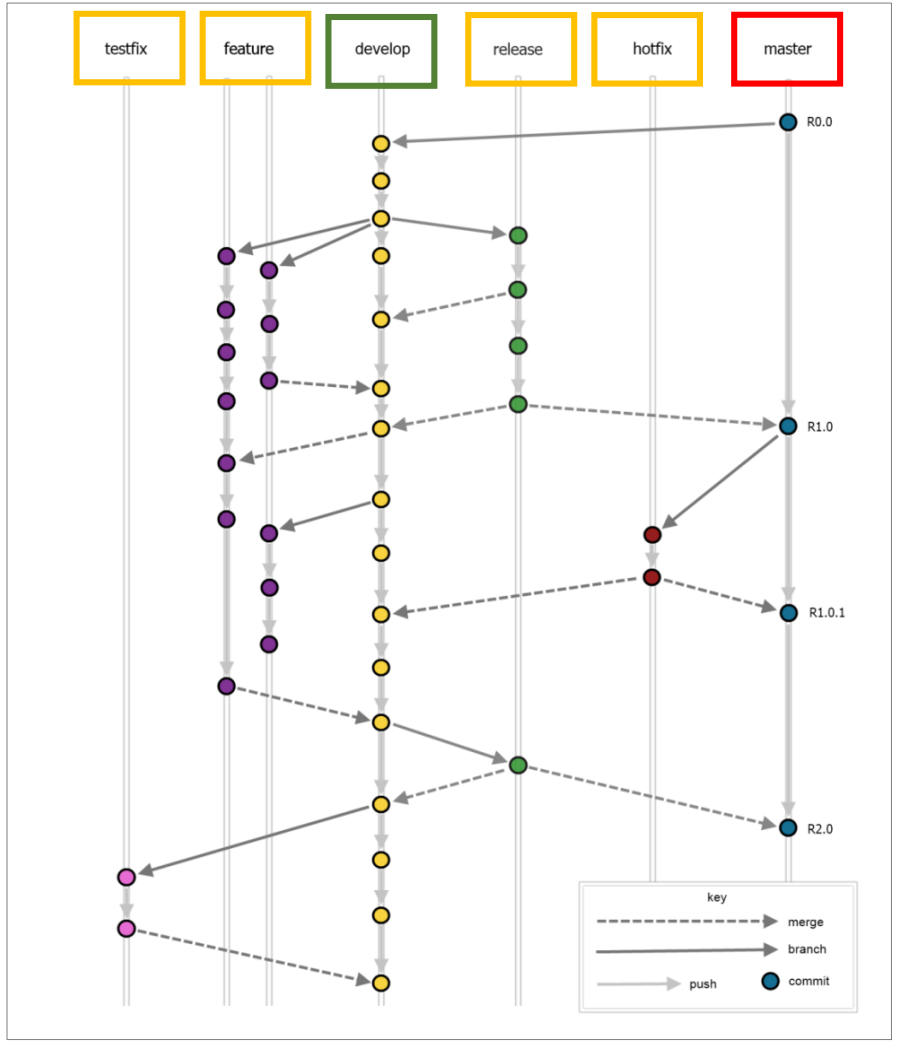
Key Concepts in Git (3)

Git Branching Technique

- The branching model is based around the principle that there are two permanent branches in a repository and they are:
  - **Develop**: Captures all development work.
  - **Master**: Maintains the current live state (or at least live-ready releases).

- Branching model also involves the use of temporary or support branches such as;
  - Feature
  - Release
  - Testfix
  - Hotfix

- Each branch, merge or rebase in the overview can be performed with a single Git flow command.

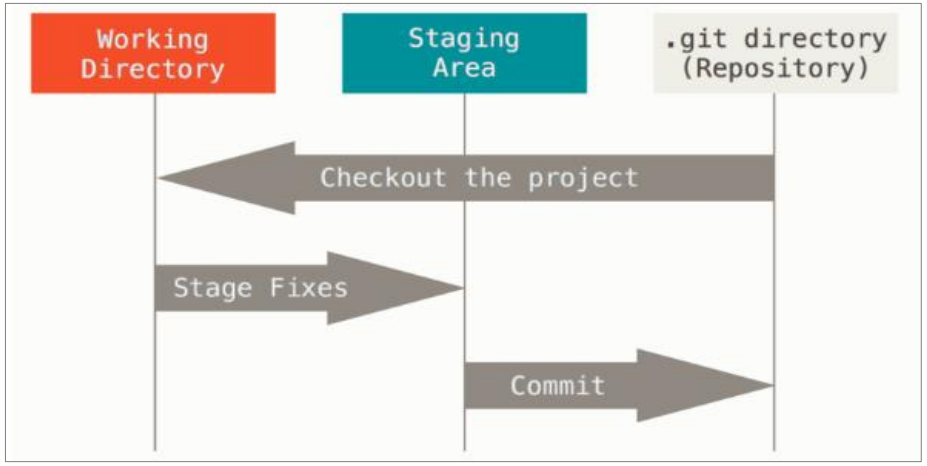


Key Concepts in Git (4)

Git Stages

Following are the different states in Git where the edited files can reside:

Committed	The data is saved and stored in the local database.
Modified	File(s) changed but not committed it to your database yet.
Staged	For the ones changes mark the changes and save it in staging so that it will go through the next a commit operation is performed.



Topic List

- Introduction to Git
- Key Concepts in Git
- **Installation**
- Introduction to Git Hub
- Exercise 2.1: Install Git

Installation (1)

Installation Commands

Following table lists the commands to install Git in different operating systems:

Operating System	Command to Install
Linux	Run the following command: sudo yum install git-all
Debian-based like Ubuntu	Run the following command: sudo apt-get install git-all
Windows	Access the following link: http://git-scm.com/download/win The download will start automatically.

Installation (2)

Initial Configuration Setup

GIT_CONFIG is a tool in Git used to set configuration variables to control all operations of Git.

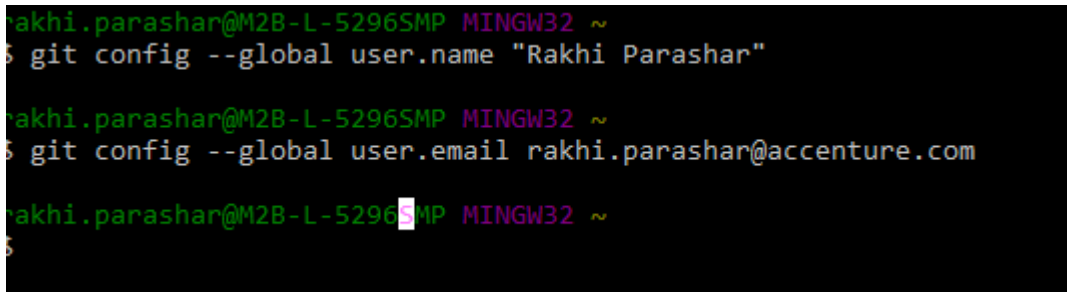
To set an identity, use the following commands:

```
git config --global user.name

git config --global user.email
```

Example

```
git config --global user.name “Rakhi Parashar"
git config --global user.email rakhi.parashar@accenture.com
```



Installation (3)

Viewing Configurations

To view all the configurations use the following command:

```
git config –list
```

```
rakhi.parashar@M2B-L-5296SMP MINGW32 ~
$ git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
rebase.autosquash=true
http.sslcainfo=C:/Users/rakhi.parashar/AppData/Local/Programs/Git/mingw32/ssl/certs/ca-bundle.crt
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.required=true
filter.lfs.process=git-lfs filter-process
pack.packsizelimit=2g
gui.recentrepo=C:/Users/rakhi.parashar/develop
gui.recentrepo=C:/Devops/TestRakhi
gui.recentrepo=C:/Devops/TestRakhi
gui.recentrepo=C:/Data/DevopsAcademy/ExampleProject
user.name=Rakhi Parashar
user.email=rakhi.parashar@accenture.com

rakhi.parashar@M2B-L-5296SMP MINGW32 ~
```

The configuration variables are stored in any of these three locations:

1. /etc/gitconfig file
2. ~/.gitconfig or ~/.config/git/config file
3. config file in the Git directory (that is, .git/config)

Installation (4)

Initializing Git Repository

When initializing a Git repository the system creates a ‘GIT’ directory in a new or in an existing project. To initialize Git repository execute the following command:

```
git init
```

Verifying the Git Repository Initialization

After executing the initialization command, you can verify it by performing the following:

1. Execute cd c:/<user>/.git/ to get into the Git directory.
2. Enter the command “dir” to list all files and directories.

The .git directory contains:

- COMMIT_EDITMSG
- Branches
- config
- description
- gitweb
- HEAD
- Hooks
- index
- info/
- Logs
- Objects
- ORIG_HEAD
- packed-refs
- refs

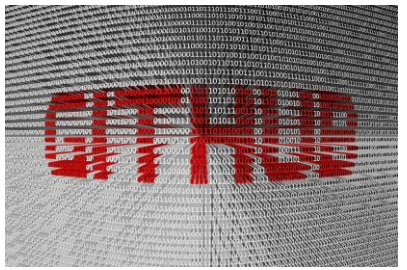
Topic List

- Introduction to Git
- Key Concepts in Git
- Installation
- **Introduction to Git Hub**
- Exercise 2.1: Install Git

Introduction to GitHub (1)

Git Vs GitHub

Following table compares Git with GitHub:

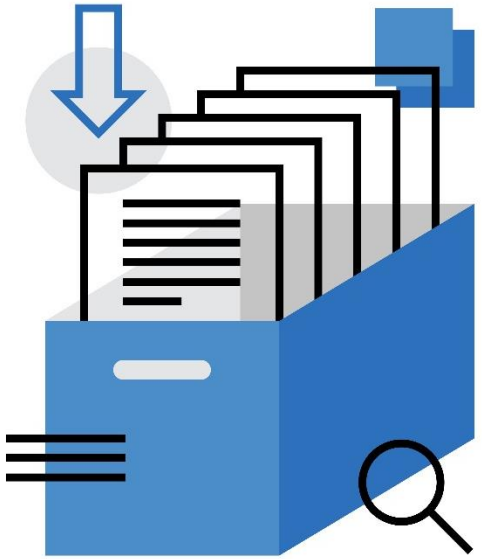


Git	GITHub
• Git is a command-line tool for effective operations. • Work and changes can locally be saved and eventually be committed and pushed to repositories created in Git Hub or elsewhere.	• GITHub is a hosting service for Git repositories. • GITHub is a web-based Git repository hosting services. Apart from adding its own features, it offers all of the distributed version control and source code management functionality of Git. • Developers can store their projects in GitHub.com. • Also used to network with other Git users.

Introduction to GitHub (2)

Repository in GitHub

- A repository is a location to store all the files for a particular project.
- Apart from storing files, repository also stores history.
- Repository is retrieved over a network where it acts as a server and version control tool acts as a client.
- Clients can connect to the repository, and then they can store/retrieve their changes to/from repository.
- Client saves the changes and makes it available to other people and other people's changes are taken by the client as a working copy on his retrieval of changes.
- Each project has its own repository, with a unique URL.

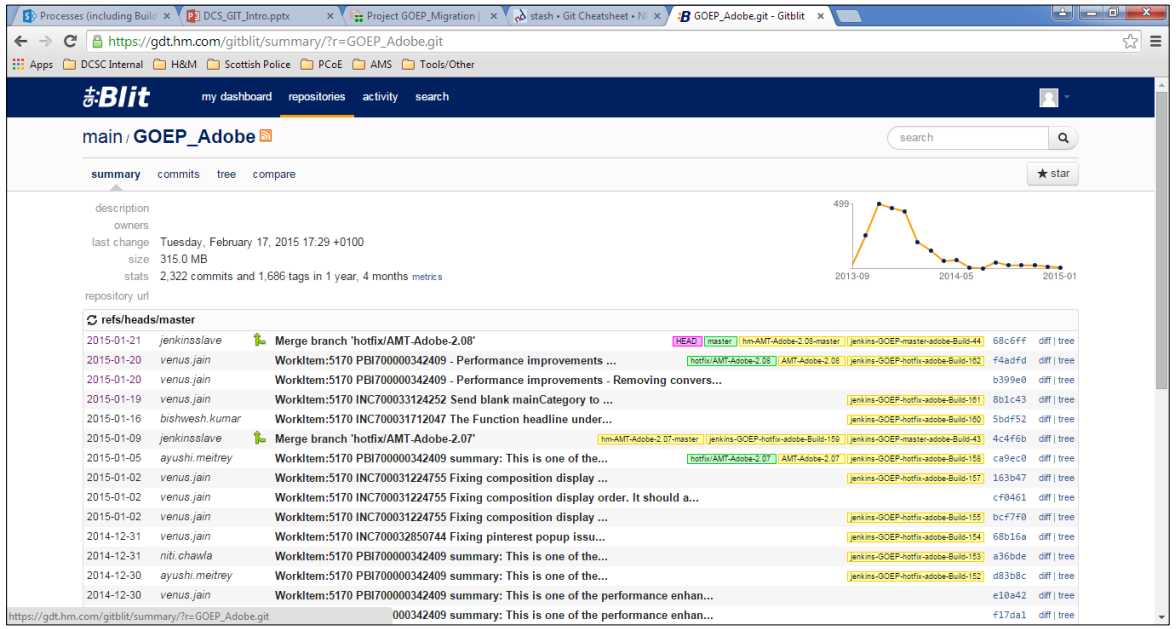


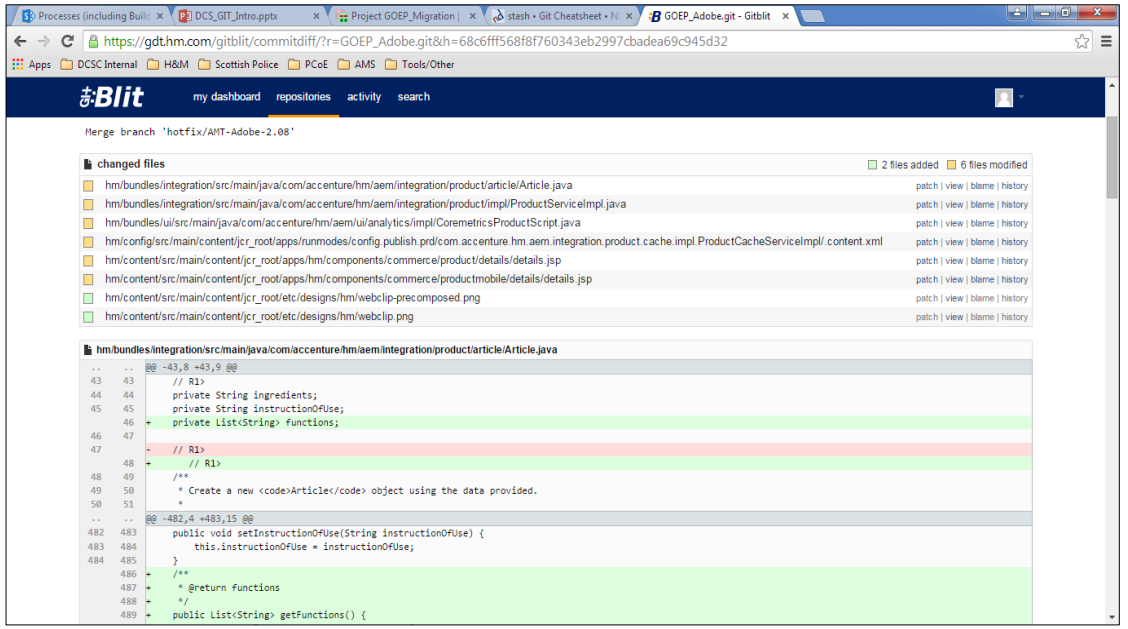
Introduction to GitHub (3)

Gitblit

Gitblit is the browser repository used in the project. Though it is used to display what appears on the remote repository, it is a visual history of the GIT repository.

Following screen capture shows the Gitblit repositories:





Topic List

- Introduction to Git
- Key Concepts in Git
- Installation
- Introduction to Git Hub
- **Exercise 2.1: Install Git**

Exercise 2.1: Install Git

Scenario

To get started hands on for Git, we need to install GIT based on OS we use.

In this exercise, you need to practice the following tasks in your machine:

- Install Git
- Perform initial configuration setup
- View configurations
- Initializing Git repository

Knowledge Check

Select the right answer

Key features of GIT are:

- ✔ a) Strong Integrity
- ✔ b) Open Source
- c) Code Review Statistics
- d) Build and Deployments

Module Summary

Now, you should be able to:

- Describe Git and its key features.
- Compare Git with other SCMs.
- Explain key concepts in Git.
- Apply the steps to install Git.
- Describe GitHub and creation of a repository.

Reference

https://help.github.com/articles/github-glossary/#commit

https://**git**-scm.com/book/en/**Get**ting-Started-**Git**-Basics

Thank You