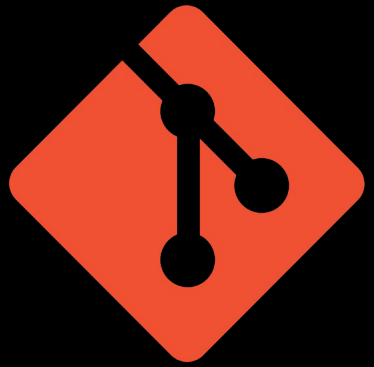


git

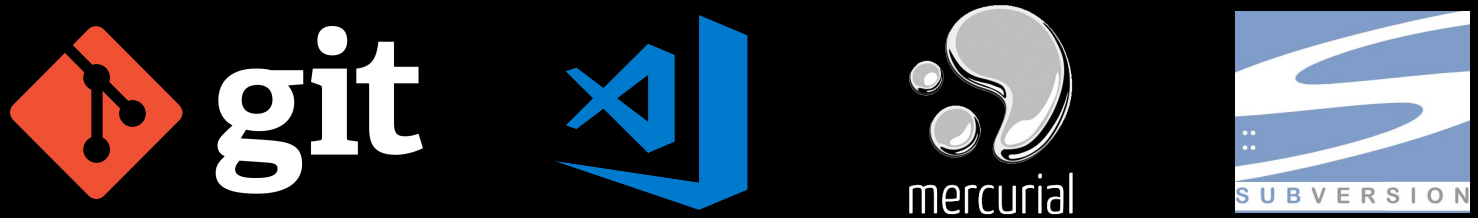
PRESENTED BY

SAID ZIANI

VCS (Version Control System)

VCS allows you to revert selected files back to a previous state, revert the entire project back to a previous state, compare changes over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more.

Some famous VCS



BENEFITS OF VCS

- A complete long-term change history of every file
- Going back to previous versions
- See who last modified something that might be causing a problem
- Traceability and bug tracking
- Branching and merging
- Collaborative Team work and Conflict resolution
- Remote work
- etc...

WHAT IS GIT ?

Git is a mature, actively maintained open source project originally developed in 2005 by **Linus Torvalds**. Developers who have worked with Git are well represented in the pool of available software development talent and it works well on a wide range of operating systems and IDEs.

PERFORMANCE:

- **Git focuses on the file content itself, Fully Distributed, Fast and Reliable**

SECURITY:

- **Relationships between files and directories, versions, tags and commits, all of these objects are secured with SHA1.**

FLEXIBILITY:

- **Support for nonlinear development workflows, Efficient in both Small and Large projects, Compatibility with many existing systems.**

COMPANIES AND PROJECTS USING GIT

INSTALL GIT

Linux :

```
$ sudo apt install git
```

Windows :

```
Download: https://git-scm.com/download/win
```

Mac OS :

```
Download: https://git-scm.com/download/mac
```

GIT configuration

```
$ git config --global user.name "Foo Boo"
```

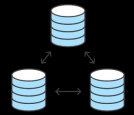
```
$ git config --global user.email fooboo@host.com
```

Start a new repository

After creating a new project directory,

```
$ git init
```

Clone an existing project



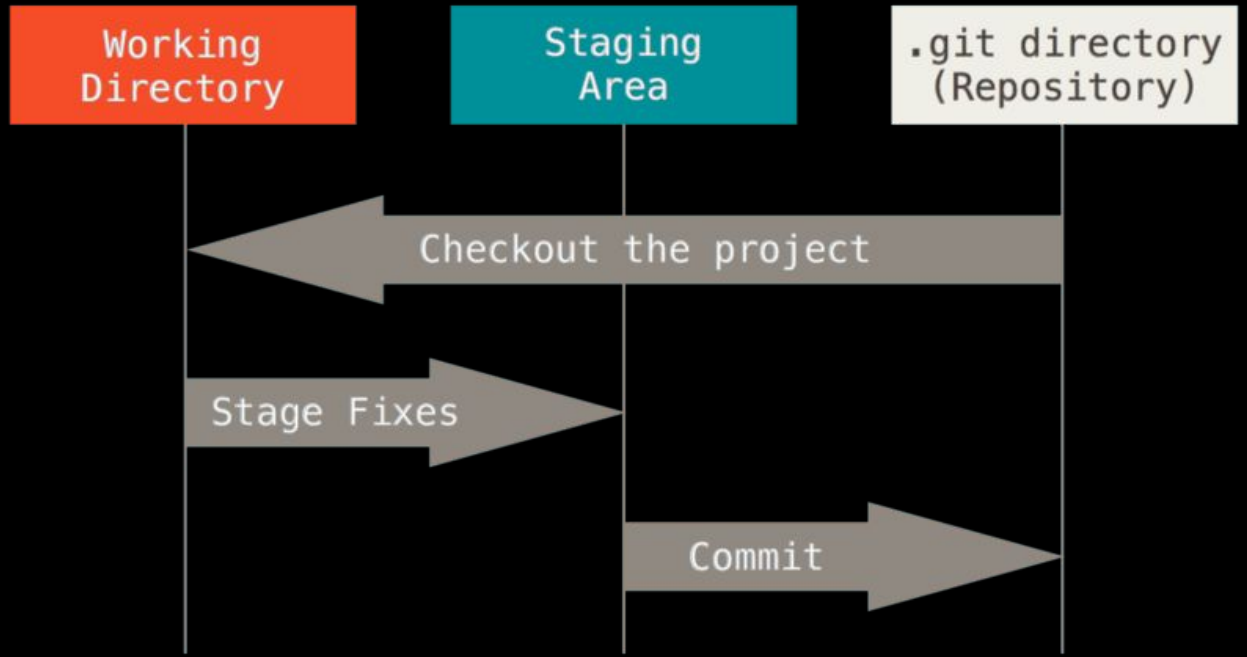
Locally:

```
$ git clone /path/myawesomeproject
```

Remote server:

```
$ git clone username@host:/path/myawesomeproject
```

GIT workflow



Saving changes

Add changes to the index:

```
$ git add myfile.ext
```

Commit changes with *UNDERSTANDABLE* and *EXPLICIT* message

```
$ git commit -m “add myfile.ext to the repo”
```

Inspecting repository

State of working directory

```
$ git status
```

List the project history

```
$ git log [-p]
```

or

```
$ git log --oneline
```

Inspecting repository

History of all changes on specific file

```
$ git blame myfile.ext
```

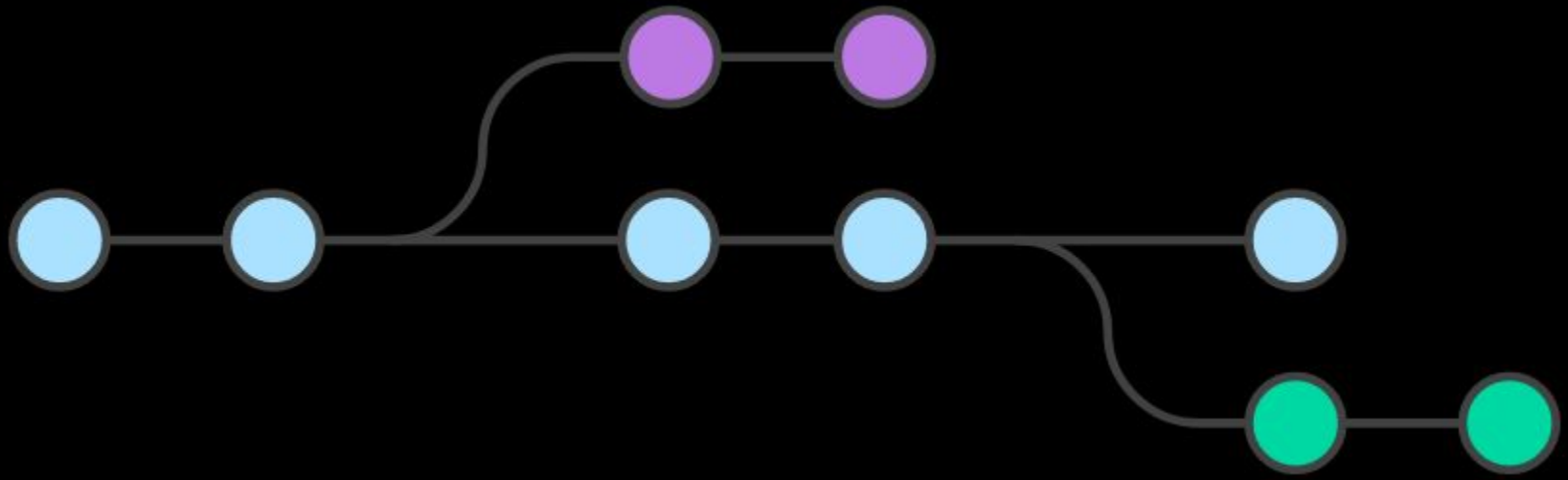
History of all changes on specific commit

```
$ git show commit_hash
```

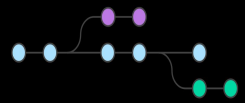
Difference between two commits

```
$ git diff commit_hash_1 commit_hash_2
```

GIT branching model



GIT branching model



Create a new branch and switch to it

```
$ git checkout -b feature_name
```

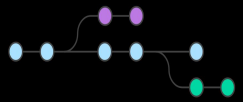
Switch to another branch

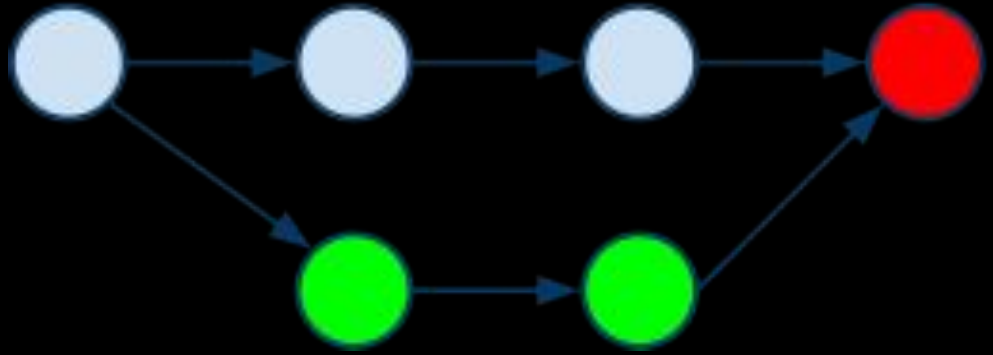
```
$ git checkout another_feature_name
```

Delete an existing branch

```
$ git branch -d feature_name
```

Merging Branches





```
$ git checkout feature_name

$ git merge another_feature_name
```

Undoing commits and changes

Checkout commit

```
$ git checkout commit_hash
```

Revert commit

```
$ git revert commit_hash
```

Reset files to specific commit

```
$ git reset --hard commit_hash
```

Working on a single central repository

Connect local repository to a remote server

```
$ git remote add origin remote_server
```

Push changes to a remote server

```
$ git push origin branch_name
```

Pull change from a remote server

```
$ git pull origin branch_name
```

.gitignore

A *.gitignore* file specifies intentionally untracked files that GIT should ignore. Files already tracked by GIT are not affected.

What to add to .gitignore ?

- Operating System Files (exp : thumbs.db)
- IDE project files (exp : .idea)
- Language and Framework Files (exp : npm-debug.log)
- Credentials (exp : wp-config.php) 

Introduction to Github

GitHub is a web-based hosting service for VCS using git. It is mostly used for computer code. It provides access control and several collaboration features such as bug tracking, feature requests, task management, etc.

Thank you!