Version Control System

(Git & Github)

Trainer

Hi there! I am Agil Haykal, just call me Agil. I am a curious guy who end up involved in data technology.

I have experienced Data Science as a trainer, consultant, and developer. I have taught +300 Data Scientist, Engineer, and Business Intelligence in total.

I handled several industries from manufacturing, banking, telecommunication, government, and Insurance. Please feel free to contact me to discuss anything about data technology.

Linkedin: Agil Haykal

DigitalSkola
Uncover The World Of Digital Skills With Us

Quote of The Day

Great discoveries and improvements invariably involve the coperation of many minds.

- Alexander Graham Bell

DigitalSkola
Uncover The World Of Digital Skills With Us

Table of Content

What will We Learn Today?

1. Introduction
2. Version Control System (VCS)
3. Git
4. Git Hands on
5. Github
6. Github Hands-on

Introduction

Imagine you are a Data Scientist at an p2p lending company. Your first task is to build a model that can **predict credit scoring**. Your lead asks you to run the experiment on server since it has high computing power. You **put all of your resources inside it** (data, trained model, research papers).

On the first iteration, you do researches, data gathering and some extensive trials and model training. Now, your **model is not bad**, but simply does not good enough to deploy.

Introduction (2)

You need time to do more researches and testing. However, the company **reassigns you to do other stuffs** at another department because of the importance level.

Your **current project left unmaintained** because you are busy with this new assignment. All of your **knowledge** and **understanding** on the flow of your original project will **fade through time** as you don't actively do it.

Introduction (3)

Now here come the best part. The company **hires a new guy** to continue your original project, as the company **realizes the value to continue this.**

The new guy sees your work and **doesn't agree on how you approach the problem**. Their approach is totally different than you. They **delete your data** from the server, **give major changes on your code**, and have different ways of managing files (folders and stuff).

It turns out their approach was **unsuccessful**, but the deadline is approaching. You are asked to go **back to your old project to fix this issue**. However, once you look at it, it is **completely different** than what you did.

Introduction (4)

What should you do in this situation?

- Working with what is left from the new guy's work. Although, you don't know what they did or did not do.
- You could try to remember what you did previously and how you did it. Although, recreating something is harder than creating from scratch.
- Or… you can take this issue to upper management to avoid this kind of situation happen again and introduce the company to…

Version Control System

Version control systems (VCS) are a category of software tools that **help a software team manage changes to source code over time**. Version control software keeps track of every modification to the code in a special kind of database

- Atlassian

Why VCS?

- **Smooth collaboration** on a team or on part of large project.
- **Understand what happen** to your codebase.
- Powerful technology that **accepts any kind of computer project**.
- **Minimize disruption** on error or mistake.
- **Proper versioning**. No more "INSYA_ALLAH_FINAL_projek_ML (2).py“
- **Providing Back Up.**

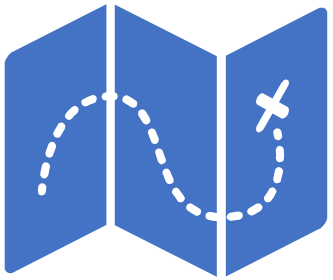
Git

Git is a free and open-source **distributed version control system** designed to handle everything from small to very large projects with speed and efficiency.

Git is so widely used then it become the **standard of VCS**.

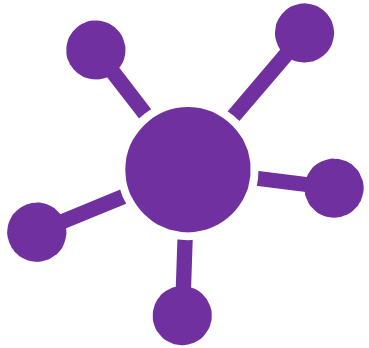
Fun Fact:
Creator of Git is also the creator of Linux!

DigitalSkola
Uncover The World Of Digital Skills With Us



Traceability

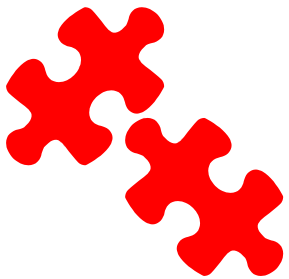
History of Every File



Distributed System

Git Features

Branch Workflow

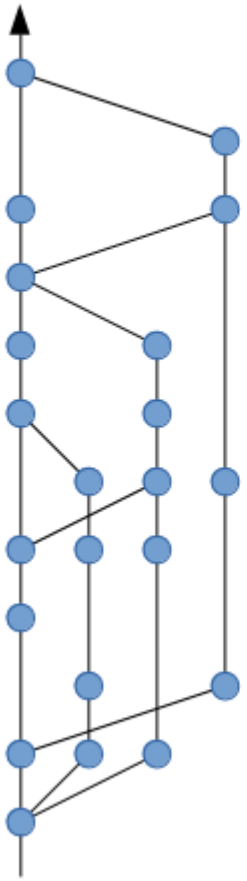


Merging

History of every files

Starting from the project is created, all changes are tracked by Git. This includes creation and deletion of files. This enables us to revert to older version of the project.

Non-linear history



Linear history

Traceability

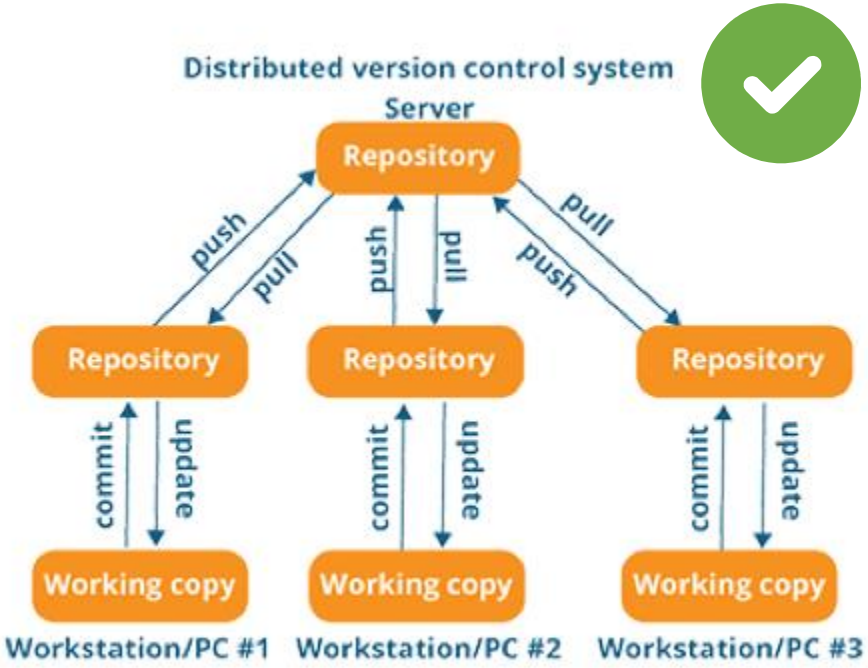
Every character change is tracked with the detail of it (author, reason, time, etc.). This is beneficial to understand the flow of development and to prepare for future development.

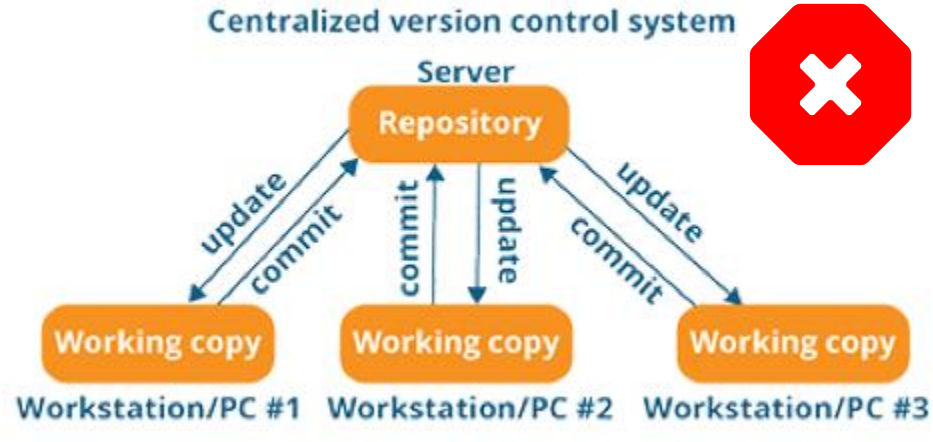
```
$ git log
commit bcb792dcc7dfbfcfd620ee73ed7422295f3d50ca (HEAD -> computer_player, origin/computer_player)
Author: lpenzey <lucaspenzeymoog@gmail.com>
Date:   Fri Jul 27 15:19:27 2018 -0500

    cleaned formating with rubocop

commit e953f0fdbfcf8038afec2a50f72c9d65601d346c
Author: lpenzey <lucaspenzeymoog@gmail.com>
Date:   Fri Jul 27 14:55:41 2018 -0500

    updated script

commit d443cc147cf543bc2892a82143e3b0ab016f7847
Author: lpenzey <lucaspenzeymoog@gmail.com>
Date:   Fri Jul 27 14:53:12 2018 -0500

    added travisci

commit bf0c6b5362ea8e675a6c866d388aee7867b816ea
Author: lpenzey <lucaspenzeymoog@gmail.com>
Date:   Fri Jul 27 14:49:45 2018 -0500

    added travisci

commit ed75abbca061c2c93834d1ee606a1b665175e10d
Merge: 08aa658 a098936
Author: lpenzey <lucaspenzeymoog@gmail.com>
Date:   Thu Jul 26 10:15:16 2018 -0500

    Merge branch 'save_resume_game' of https://github.com/lpenzey/Mastermind into save_resume_game
```

Distributed System

Each person gets their own local repository. This minimizes network connection and blocking on error, unlike the centralized system.

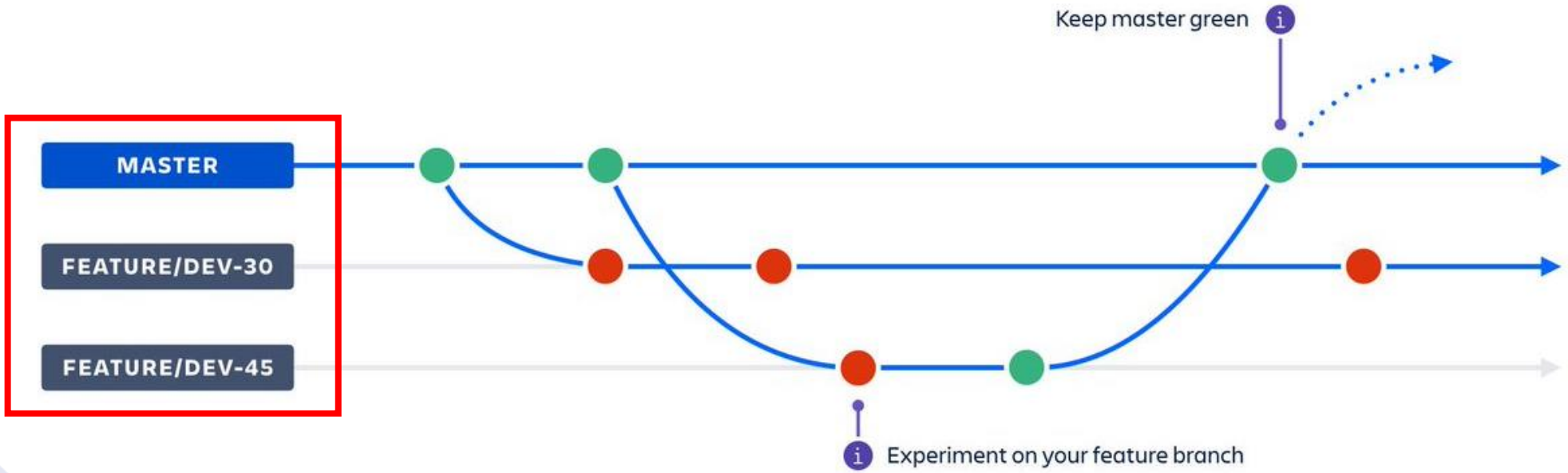
Distributed system also enables easier backup. If someone delete or messing up with their repository, they can simply copy from another person and start again.





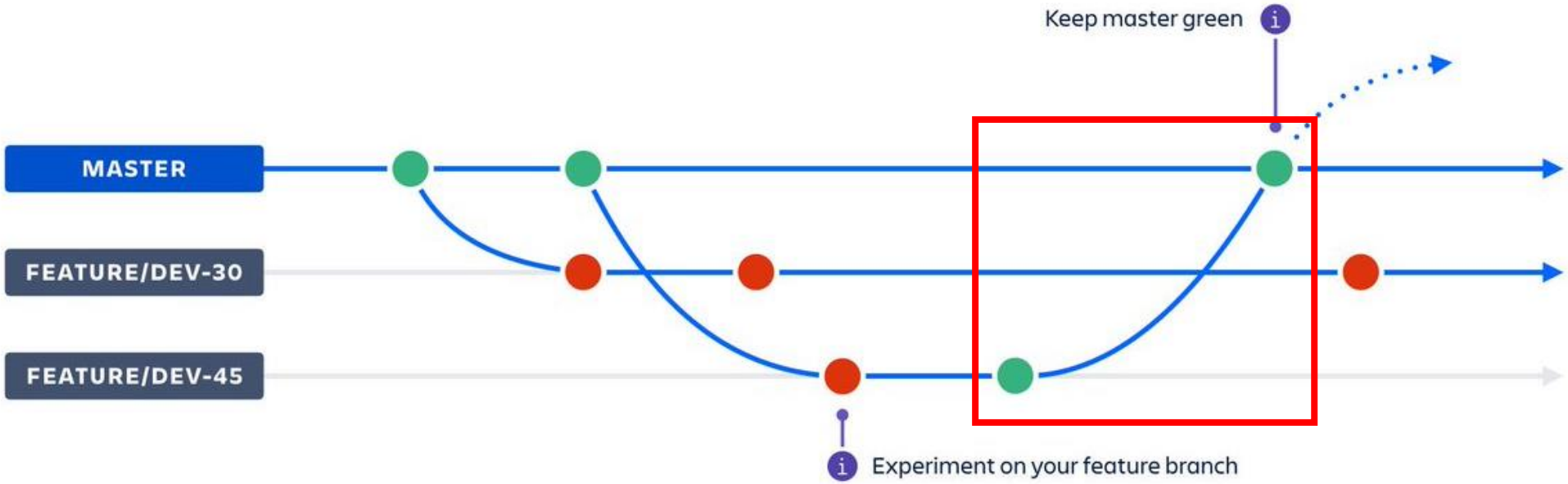
Branching Workflow

Git enable us to create a branch or another stream of independent work. This what makes concurrent work and collaboration using Git so easy.



Merging

Once work is done on branch. Git can join branch to the main project by merging it. Merging is done by opening Pull-Request to merge the branch to repository. This makes easier for Project Leader to check the code before integrating it.



DigitalSkola
Uncover The World Of Digital Skills With Us

Git Configuration Commands

Verify you already installed the correct Git version

git –version

Configure Git username and email

git config --global user.name “Name Name"

git config --global user.email “email@email.com"

Verify the changes

git config user.name

git config user.email

Git Repo Commands

Create your first repository (create folder first)

git init

Check status of changes

git status

Commit to historical changes

git log

git log –p **(See detail changes) *press “q” to exit**

DigitalSkola
Uncover The World Of Digital Skills With Us

Git Execution Commands

Add file to staging

git add “filename” (Per file)

git add * (All Files)

Remove file in staging

git rm –cached “filename” (Per file)

git rm –cached * (All Files)

Commit to changes

git commit -m “message”

Commit and add all changed files

git commit -a -m “message”

DigitalSkola
Uncover The World Of Digital Skills With Us

Git Branching Commands

Creating a new branch

git checkout -b “new_branch”

Verify all branches and the active one

git branch

Switch to another branch

git switch “new_branch”

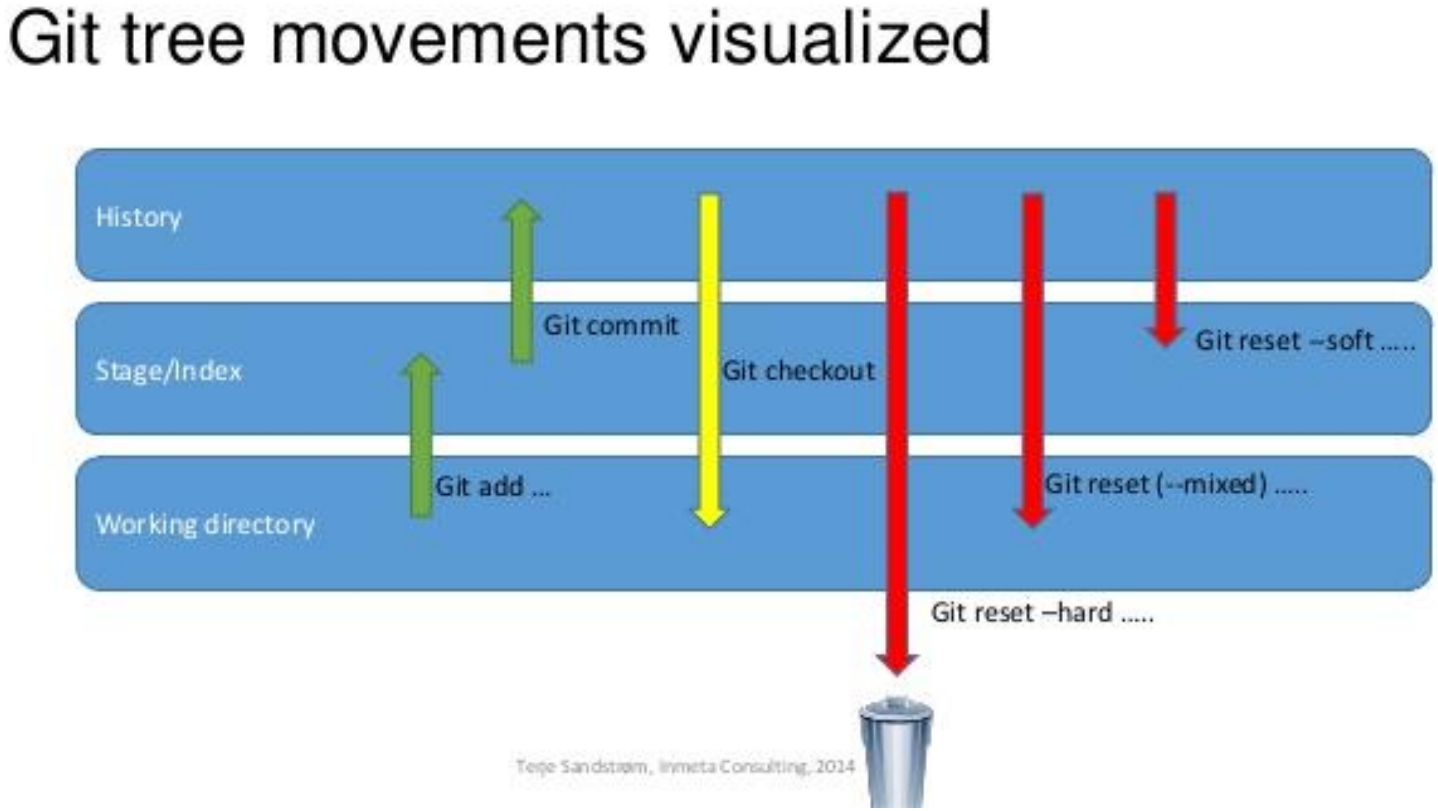
Merge the new branch to master

git merge “new_branch”

Back to previous commit

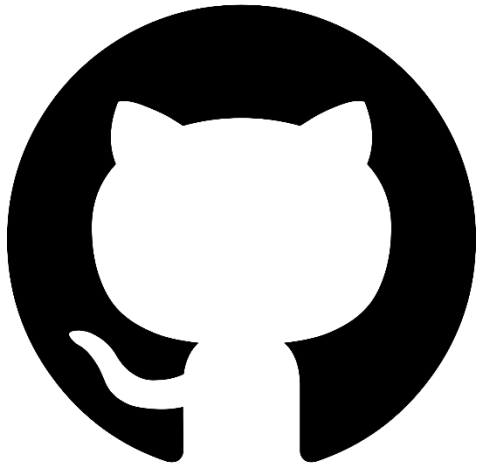
git reset –hard HEAD~1

Git Reset



Github

- Community that collaborates (most developers are here)
- Lot of services to integrate on its Marketplace.
- Support academics
- You have to pay for private repositories.



DigitalSkola
Uncover The World Of Digital Skills With Us

Create Github Account

Create GitHub Account, go to github.com

Create new repo in your account

Clone the repo to your computer

Push and Pull the code

Github Commands

Create a new branch

git clone <link>

Set Github account to your local Git

git remote set-url origin <link>

Push all commit to Github Repo

git push

Pull all commit to Github Repo

git pull

Gitignore

- Helps you to untrack files you don’t want to sync between locals and remote.
- Use file .gitignore on your repo. Helpful when you have huge amount dependencies packages on your project (NodeJS) or when your dataset is included on your git repository.
- Each local can download or initiate it separately.

DigitalSkola
Uncover The World Of Digital Skills With Us

Summary

Git and Github are systems that essentially needed by programmers, data scientist, or anyone who work with code. So do not forget to keep practicing git and github.

1 Introduction

2 VCS

3 Git

4 Git Hands on

5 Github

6 Github Hands on

DigitalSkola

Uncover The World Of Digital Skills With Us

Thank YOU