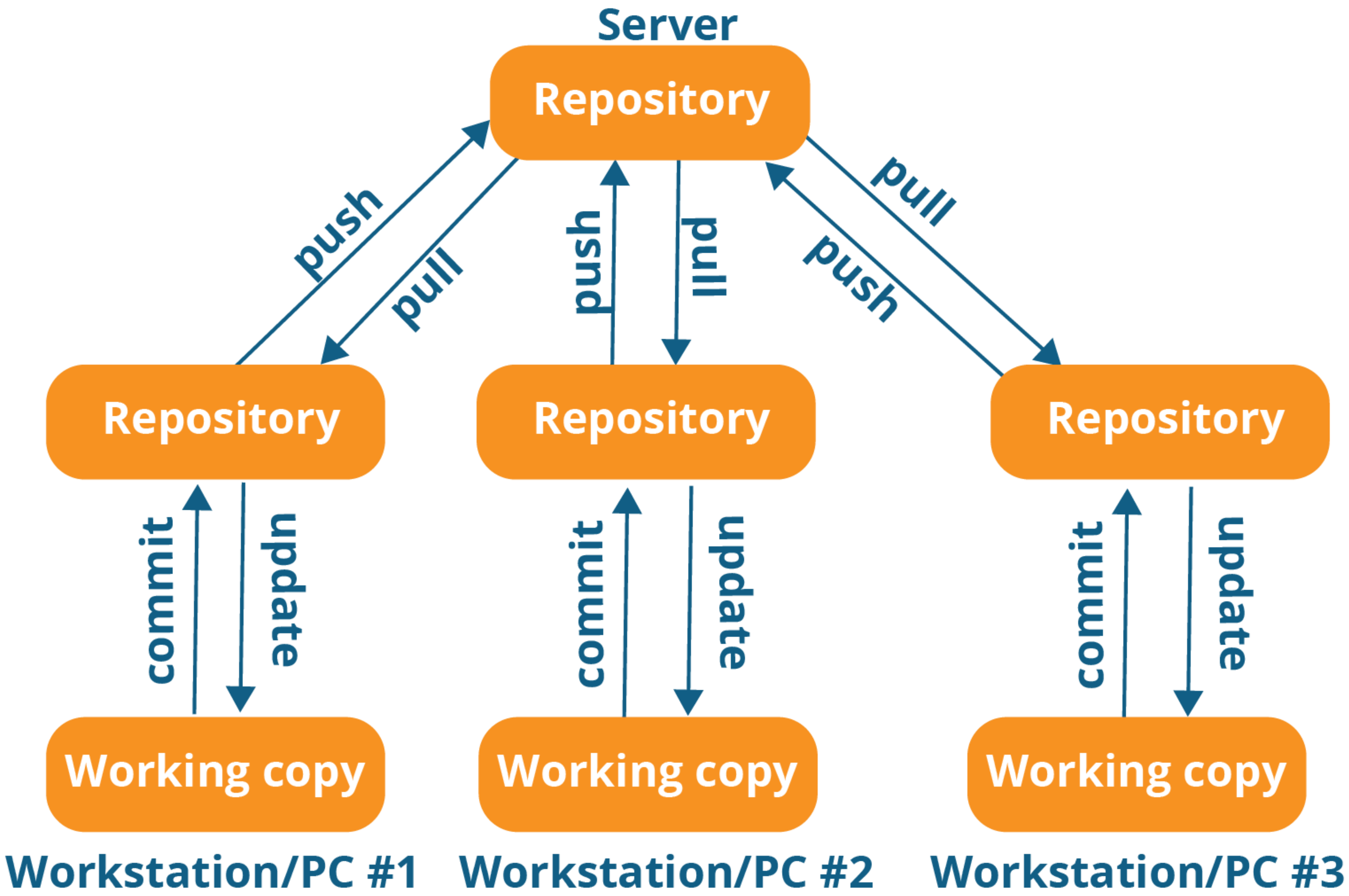
GIT

About Git

- Git is a free, open source distributed version control system tool
- It was created by Linus Torvalds in 2005 to develop Linux Kernel.
- Git has the great functionality, performance, security and flexibility.
- It the most popular and important distributed version-control DevOps tool.
- Git is primarily used as SCM manage your project, comprising a set of code/text files that may change.

Distributed version control system



Git Installation

- Git on Windows
  - Install visual studio code
  - Install git
  - Create a repo in github
  - Connect to github
- Git on Linux
  - sudo apt-get update
  - sudo apt-get install git

Github setup

- Create a repo
- Git config
- Initialize
- Add
- Commit
- Pull
- Push

Git Areas

Working Area

Staging Area

Commit Area

Git Command

- **git init**
  - Creates a new repository.
- **git config**
  - git config --global user.name “Your Name”
  - git config --global user.email “yourmail@yourdomain.com”
  - Sets the author name and email address respectively.
- **git clone**
  - git clone https://github.com/myrep/mycode.git
  - Copy of a repository from an existing URL.

Git Command

- **git status**
  - Displays the state of the working directory and the staging area.
  - You can see changes that are in the staging, those that are not staged and are not tracked by Git.
- **git add**
  - git add file.txt
  - Adds a file (file.txt) to the staging area.
  - git add .
  - git add *
  - Add all files to the staging area before commit

Git Command

- **git restore**
  - git restore file.txt
  - git restore .
  - Un-stage or even discard uncommitted local changes
- **git commit**
  - git commit -m “commit message”
  - Records the file permanently in the version history.
- **git show**
  - Displays the metadata and content changes of the commit.
- **git log**
  - Shows the commit logs
  - git log
  - git log –oneline
  - git log –name-only

Git Command

- **git rm**
  - git rm file.txt
  - git rm –f file.txt
  - git rm –cached file.txt
  - Deletes the file from your working directory and stages the deletion.
- **git push**
  - This command sends the committed changes of the master branch to your remote repository.
  - git push [variable name] master
  - git push [variable name] [branch]
  - git push origin master

Git Command

- **git pull**
  - Fetches and merges changes on the remote server to your working directory.
- **git branch**
  - **Lists** all the local branches in the current repository.
- **git branch develop_branch**
  - **Creates** a new branch.
- **git branch -d develop_branch**
  - **Deletes** the feature branch.
- git push --set-upstream origin feature/algo

Git Command

- **git checkout**
  - git checkout branch_name
    - Switch from one branch to another.
  - git checkout –b branch_name
    - Create a new branch and switch to the new branch.
- **git merge**
- git merge branch_name
  - Merges the specified branch’s history into the current branch.

Git revert and reset

- git revert
    - It is used for undoing changes to a repository's commit history
    - git revert HEAD
- git reset
    - It is used to undo local changes to the state of a Git repo
    - git reset --hard f414f31
    - git reset --soft HEAD@{1}
- git stash

.gitignore

- Git can specify which files or parts of your project should be ignored by Git using a .gitignore file.
- Git will not track files and folders specified in .gitignore.
- The .gitignore file itself is tracked by Git.
- Examples of ignored file
  - log files
  - temporary files
  - hidden files
  - personal files

Git Fork

- Forking is a git clone operation executed on a server copy of a projects repo
- A fork is a copy of a repository that you manage.
- Forks let you make changes to a project without affecting the original repository
- **Syncing a fork**
  - Sync a fork of a repository to keep it up-to-date with the upstream repository