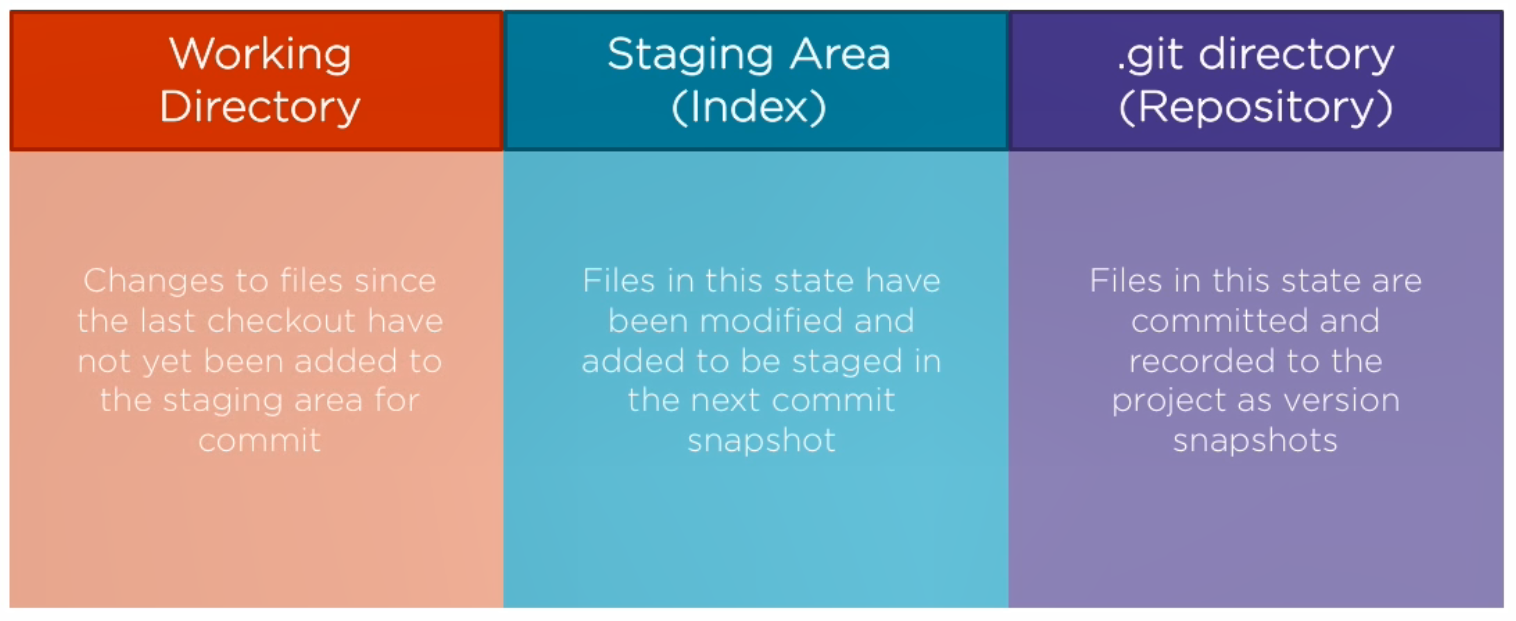
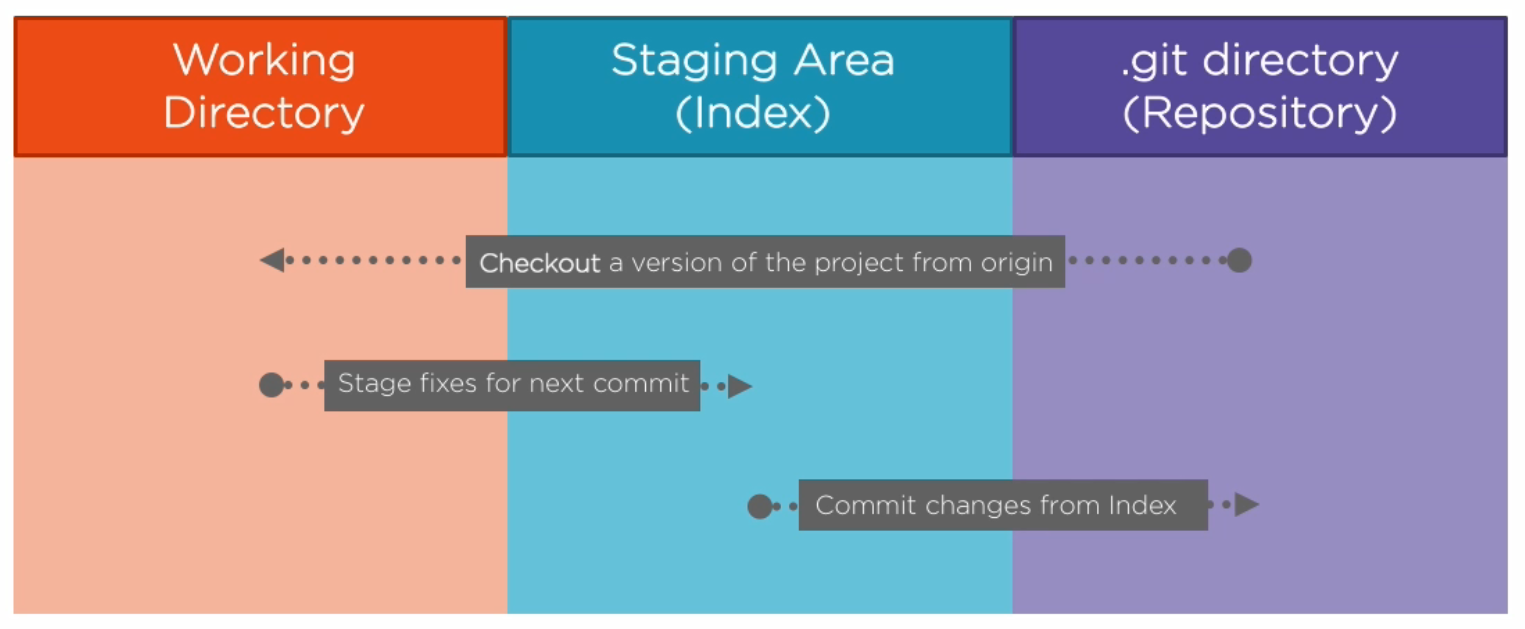
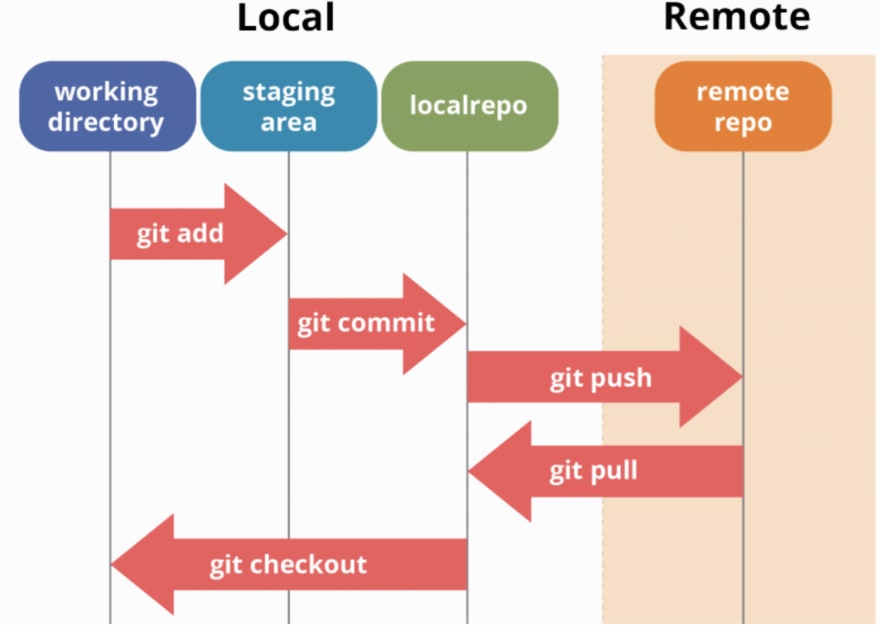
# GIT



## What is GIT?

* Distributed version control system
  + Your local copy of code is a complete version control repository
  + Commit your work locally, and then sync your copy of the repository with the copy on the server
  + Differs from centralized version control where clients must synchronize code with a server before creating new versions of code
* Created in 2006 to manage Linux kernel
* Most commonly used version control system today
  + Quickly becoming the standard for version control

## Benefits

* Everyone has their own local copy of the code
  + Work on your own branch
  + Work offline
  + Faster since you don’t have to communicate with the server
* Everyone has a backup of the entire project
* Everyone maintains the full history of the project
* Everyone can experiment with new additions before committing to master

## Git vs GitHub vs Azure Devops

* Git is the plumbing/underneath “language”
* Github is a Web based repo
* Azure DevOps is that plus project management/ DevOps tools
* Visual Studio (Code) has great support for both Azure DevOps and GitHub
* You can use Git or TFVC with an Azure DevOps project

## Git Commands

### Initialize Commands

|  |  |  |
| --- | --- | --- |
| Command |  | Description |
| git init | Create repo |  |
| copy con | Create a new empty file |  |
| mkdir | Create a new empty folder |  |

### Navigation & Status Commands

|  |  |  |
| --- | --- | --- |
| Command |  | Description |
| cd | Change working directory | Go up: cd..  Go to subfolder: cd <name of subfolder>  Go to beginning: cd ~ |
| pwd | Print working directory | Lets you know where you are inside your terminal |
| git status | Shows modified files which are not yet commited |  |
| dir | List files in a directory |  |
| git log | Log of all commits |  |

### Moving Files Commands

|  |  |  |
| --- | --- | --- |
| Command |  | Description |
| git add <filename>  git add \*.docx  git add . | * Include/add files to the index * git add . moves all files in the current directory * git add <file name> moves just this certain file |  |
| git commit -m “Message”  git commit -a | Creates a new snapshot in the repo |  |
| git checkout <commit code> | Puts version of a certain commit to file system |  |

|  |  |  |
| --- | --- | --- |
| Command |  | Description |
| Git config - - list (view configurations) | Configuration settings/ view |  |
| git diff <commit code1> <commit code2> | Shows differencies (changes) between 2 commits |  |

## GIT Started

### Create a repository

* A folder that you’ve told Git to help you track file changes in.
* A Git repo contains every version of every file saved in the repo
* You can have any number of repos on your computer, each stored in their own folder.

|  |  |  |
| --- | --- | --- |
| 1 | Open GIT CMD |  |
| 2 | Type path of the folder you want to create your repo of | *cd + <path>* |
| 3 | Create repo | *git init* |
| 4 | Change files |  |
| 5 | Check status | *git status* |

### Add files to staging area

|  |  |  |
| --- | --- | --- |
| 1 | Check status | Git status |
| 2 | Add files to statging | Git add <file> |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |

### Connect to GitHub repo

|  |  |  |
| --- | --- | --- |
| 1 | Create GitHub repository in GiHub |  |
| 2 | Connect git repo with GitHub | *git remote add origin <https:GitHub\_rep.git>* |
| 3 | Check connection | *git remote -v* |
| 4 | Push repo to GitHub | *git push origin master* |
| 5 | Refresh GitHub and see changes |  |
| 6 | Pull actual version from GitHub | *Git pull origin master* |

### Clone an existing repo

### Save work with commits