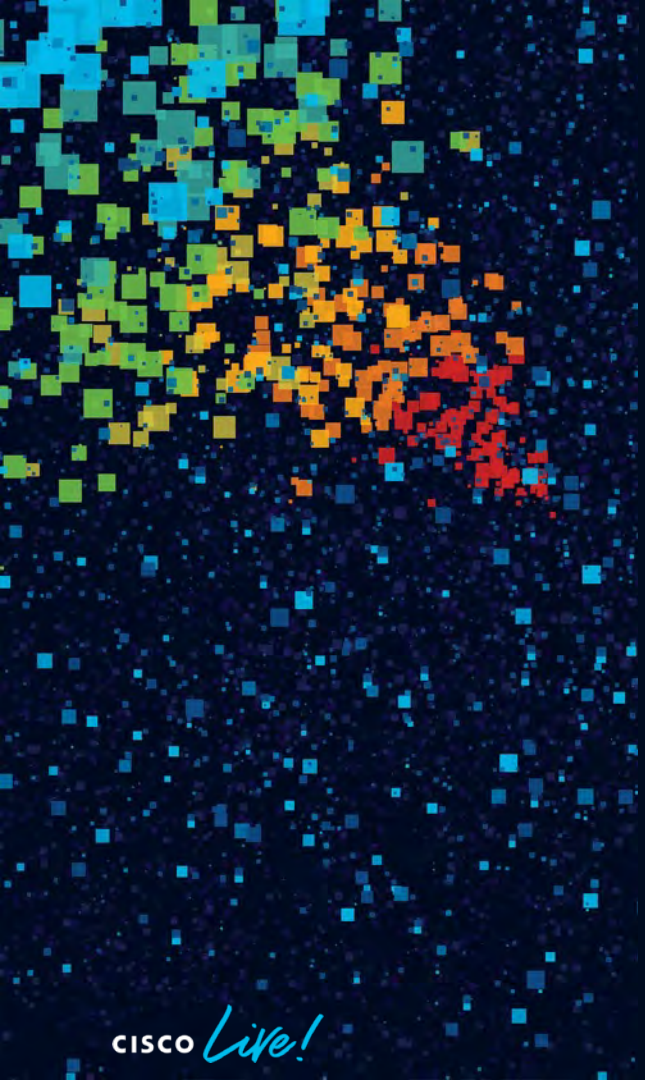
Intro to git

Kareem Iskander
@Kareem_Isk
DevNet-SN



Agenda

- Introduction
- Quick overview of Git
- Hands On


What is Version Control?

The Need for Version Control

How do I make incremental changes and share my work with others?

How do I go back to the version of this file from (yesterday, last week, last year, ...)?

What changed between version X and version Y of a file?

People have been making changes to the same file (or set of files)... How do I reconcile and merge all these changes?

What is Version Control?

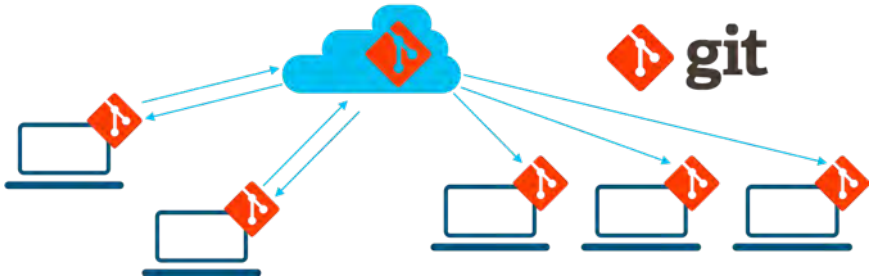
- A system that manages changes to a set files in order to keep a history of changes
- Version Control is similar to:
  - Snapshots of VMs
  - Incremental backups of files
  - Wiki versioning
- When you make a mistake or want to do some experimenting, you can do that in a safe way.


What is Git?

Git

- An open source distributed version control system

Git vs. GitHub

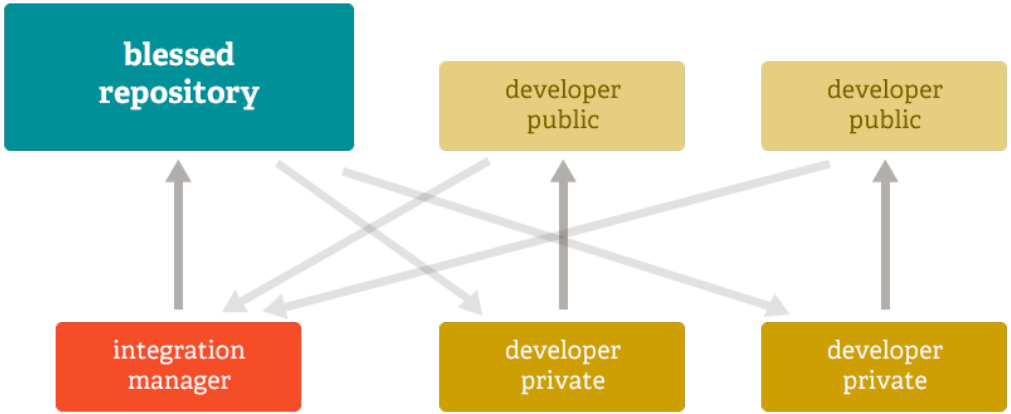


Git is an open source Distributed Version Control System

GitHub is a commercial company, that runs GitHub.com based on Git Version Control System

DISTRIBUTED VERSION CONTROL

- Opens up to new workflows: git flow
- Each system has an exact replica of the repo as other collaborators.



https://git-scm.com/images/about/workflow-b@2x.png

Git: Technical Overview

Basic Git Terminology

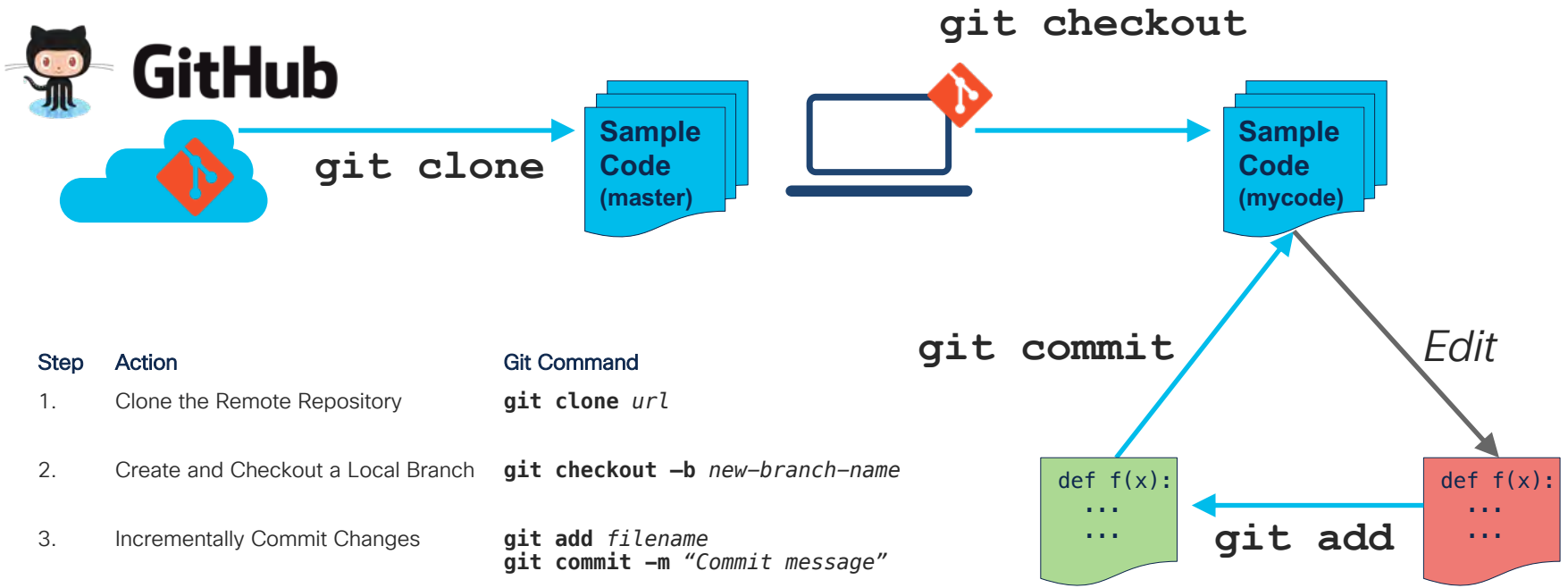
- **Repository (Repo)** – A vault for storing version controlled files
- **Working Directory** – The visible directory and its contents
- **Versioned Files** – Files you have asked Git to track
- **Un-Versioned Files** – Files in your working directory not tracked by Git
- **Commit** – Snapshot in time (of your version controlled files)
- **Branches** – A safe place for you to work

Useful Git Commands

Setup	Tell git who you are *one-time setup*	**git config --global user.name** *"your name"* **git config --global user.email** *your@email.com*
Clone	Clone ("download") a git repository	**git clone** *url*
Status	Check the Status of your local repository	**git status**
Checkout A Branch	Create and Checkout a local **Branch** Creates a "safe place" for your changes	**git checkout -b** *new-branch-name*
Add	Add a file to your next commit.	**git add** *filename*
Commit	Commit your changes.	**git commit -m** *"Your commit message."*
Checkout A File	Checks-out a file from the last commit. Reverts any changes you have made, and restores the last committed version of a file.	**git checkout** *filename*

Learn More: **git --help** and **man git**

DevNet Sample-Code Workflow

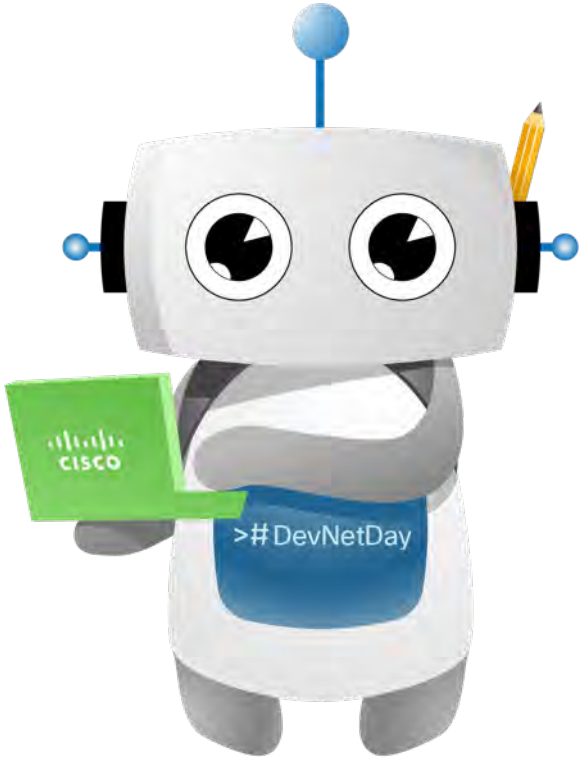


Step	Action	Git Command
1.	Clone the Remote Repository	**git clone** *url*
2.	Create and Checkout a Local Branch	**git checkout –b** *new–branch–name*
3.	Incrementally Commit Changes	**git add** *filename* **git commit –m** *“Commit message”*

La Démo

Explore More

- DevNet Start Now
- DevNet on GitHub



Thank you