

Git Command Line Interface (CLI)

Business Science

11/26/2019

Contents

Git Command Line Interface (CLI)	1
How Git Works	1
Definitions	2
Installation	2
Git Workflow	3
Git Commands	4

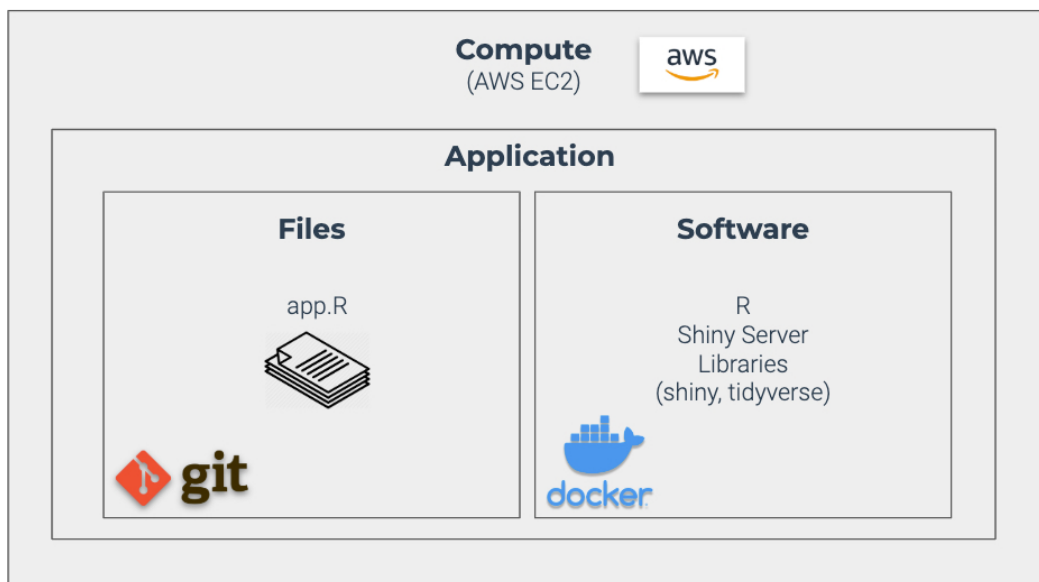
Git Command Line Interface (CLI)

This document covers the basic `git` commands used frequently developing and deploying software.

How Git Works

A Shiny App run on AWS EC2 consists of:

1. **Files** - Controlled by `git` version control
2. **Software Environment** - Controlled by `docker` image



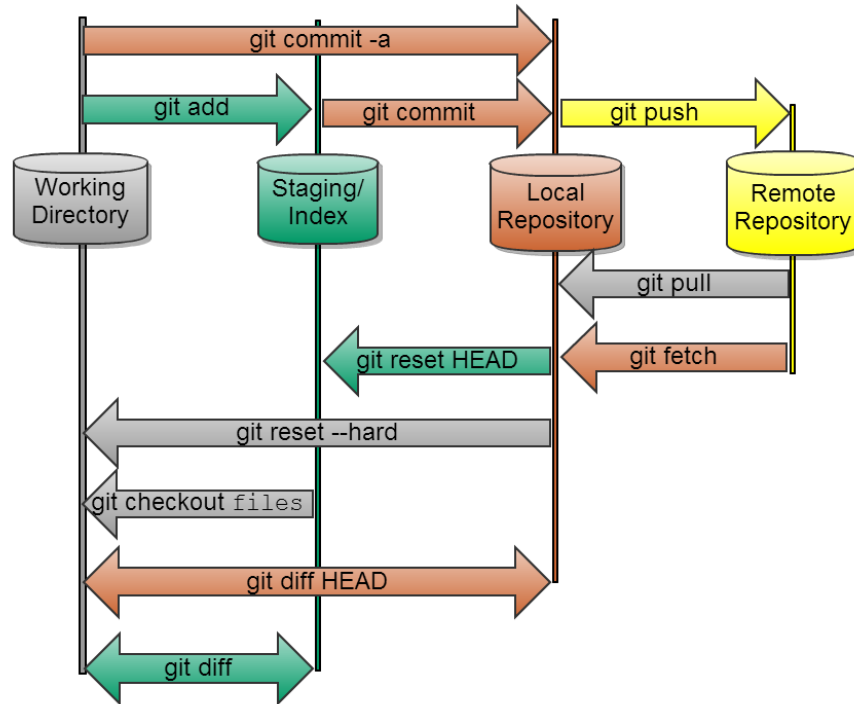
Definitions

1. **GitHub** - An online community for storing and sharing version-controlled software files. Has Public and Private repositories.
2. **Repository (repo)** - A group of files that are version controlled by **git**. Every change made to any of the files is tracked and can be differenced from the working version and any previous version that was committed.
 - **Remote Repository** - A remote repository is one that is stored in a shared location typically on GitHub (or BitBucket, GitLab, Azure, etc).
 - **Local Repository** - The local repository is one that is stored on your computer or your EC2 Server.
3. **commit** - Creates a checkpoint for version controlling your file. As you develop, you should commit often because committing creates a safety harness (rock climbing analogy). The more often you commit, the shorter your fall when you screw something up.
4. **push** and **pull** - These actions are used to send files to (**push**) and from (**pull**) your remote repository.
5. **clone** - Used to setup a local repo from a remote repo. We do this commonly when setting up software on EC2 servers.
6. **branch** - Branches are used for creating prototype software (development version) without modifying the working (production version) until you are ready. Once ready, we **merge** the **branch** (development version) with the **master** (production version).

Installation

- **Mac and Linux** - **git** comes pre-installed on Mac and Linux.
- **Windows** - Visit the git website (<https://git-scm.com/download/win>) to install on Windows.

Git Workflow & Commands



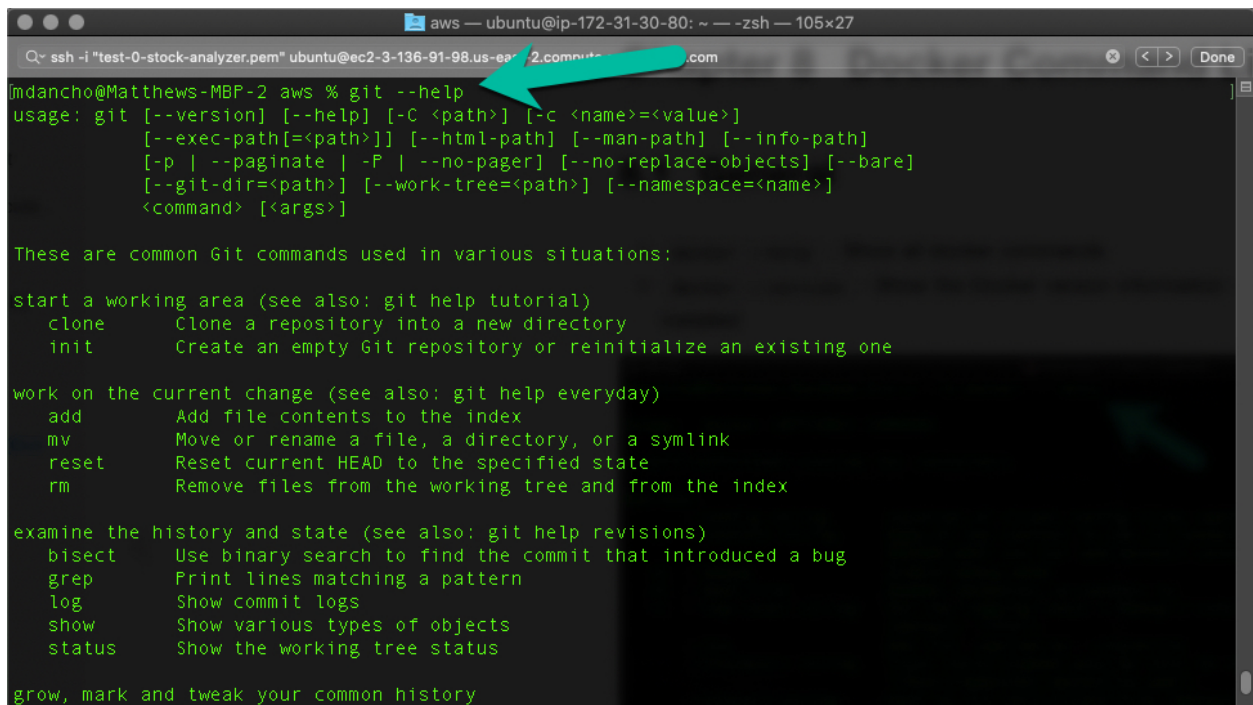
Git Commands

These are the primary commands for working with git from the commandline.

Important Note - We will use RStudio IDE to abstract away much of the git workflow. RStudio is integrated tightly with git and GitHub, which is a massive productivity booster.

General Commands

- `git --help` - Use help to investigate the available commands.
- `git --version` - Useful for getting the version and checking git installation.

A terminal window titled 'aws — ubuntu@ip-172-31-30-80: ~ — zsh — 105x27'. The prompt is 'mdancho@Matthews-MBP-2 aws %'. The command 'git --help' has been executed. The output shows the usage of git with various options like --version, --help, -C, -c, --exec-path, --html-path, --man-path, --info-path, --paginate, --no-pager, --no-replace-objects, --bare, --git-dir, --work-tree, and --namespace. It also lists common Git commands categorized into: start a working area (clone, init), work on the current change (add, mv, reset, rm), examine the history and state (bisect, grep, log, show, status), and grow, mark and tweak your common history. A red arrow points to the 'git --help' command in the terminal.

```
mdancho@Matthews-MBP-2 aws % git --help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
    clone      Clone a repository into a new directory
    init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
    add        Add file contents to the index
    mv         Move or rename a file, a directory, or a symlink
    reset      Reset current HEAD to the specified state
    rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
    bisect     Use binary search to find the commit that introduced a bug
    grep       Print lines matching a pattern
    log        Show commit logs
    show       Show various types of objects
    status     Show the working tree status

grow, mark and tweak your common history
```

Local Repo Setup

No Remote Repo set up?

- `git init` - Initialize a Local Repo
- `git commit` - Done to initialize (add) your files to the Local Repo.

Remote Repo already set up?

- `git clone` - Clones a remote repo to your local machine. We do this to get software onto EC2 Servers.

GitHub Repo Setup

Important: Make sure a blank repo exists. Then:

- `git remote add origin https://github.com/user_name/repository.git` - Used to link a local repository with a
- `git push -u origin master` - Pushes your initial commit on your Local Repo to the Remote Repo (e.g. GitHub).

Git Workflow

- `git commit` - Adds snapshots of changed files to your Local Repo.
- `git push` - Pushes committed files from your Local Repo to your Remote Repo (e.g. GitHub)
- `git pull` - Pulls the remote repository files to your local repository

Branches

- `git status` - Find the branch you are currently on
- `git branch` - Creates a new branch
- `git checkout` switches the Local Repository to the specified branch
- `git merge` - Merges a branch with the master. This is typically done via “Pull Requests”.