**Command line cheat sheet**

Table of Contents

[Temp 1](#_Toc31635899)

[AWS (Amazon Web Services) 1](#_Toc31635900)

[Azure 1](#_Toc31635901)

[cURL 1](#_Toc31635902)

[Git 2](#_Toc31635903)

[Docker 4](#_Toc31635904)

[Node.js and npm (Node Package Manager) 4](#_Toc31635905)

[MongoDB 5](#_Toc31635906)

# Temp

* git config http.sslVerify false
* git config --global http.sslVerify false
* npm config set strict-ssl false

# AWS (Amazon Web Services)

On-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. In aggregate, these cloud computing web services provide a set of primitive abstract technical infrastructure and distributed computing building blocks and tools.

<https://docs.aws.amazon.com/cli/latest/index.html>

* Aws configure
* Aws ec2 describe-instances

# Azure

A cloud computing service for building, testing, deploying, and managing applications and services through Microsoft-managed data centers.

<https://docs.microsoft.com/en-us/cli/azure/reference-index>

# cURL

A computer software project providing a library (libcurl) and command-line tool (curl) for transferring data using various network protocols.

cURL commands

* Send request and return content, curl [url]
* Send request and return headers, curl [url] -I
* Send request and return content with headers, cur [url] -i

# Git

Git is a distributed revision control system, a stupid content tracker, a persistent map (vales and keys). Every object [blobs, trees, commits and tags] in Git has its own key SHA-1. Git maintains a config file as well that contains settings such as remote origin.

Git is a high level version file system built on top of your native file system.

<https://git-scm.com/docs>

A commit is a simple, short piece of text. Git generates and stores, contains meta data about the commit and including pointer to tree to recreate object database.

A merge is just a commit with one exception, it has two (or more?) parents. Merge preserves project history exactly as it happened.

Tag objects are a label for the current state of the project. A tag is like a branch but it will stick to the same object and doesn’t move. Regular and annotated tags. Annotated tags come with a message. Non-annotated tags (a.k.a. lightweight tags or regular tags)

A branch is just a reference to a commit. HEAD is just a reference to a branch; a pointer to a pointer. Asterix (\*) next to branch name denotes current branch. HEAD, there is only one HEAD object and marks the current position in the object model

Fast-forward is git being frugal and sparing commits in the object database ultimately making our project’s history less complicated.

Detached HEAD occurs when there is no current branch by checking out a commit instead of checking out a branch. A common way to use a detached HEAD, when you want to try something new or go down an experimental branch of your code, you can detach HEAD by checking out a commit, create more commits, create branch for those commits, then checkout to master or previous branch.

Rebasing is considered to be Git’s signature feature. Git looks for first commit in current Branch shared with rebase branch, current branch is detached (in simplest terms) and placed on top of rebase branch so it changes base of current branch. In actuality, objects in current branch are replicated except their parents change so they are updated to be new objects with new SHA-1 leaving old commits behind. Rebases help refactor project history so that it appears more streamlined (lose project history details) and looks cleaner but can have unwanted consequences. As a general rule, never rebase shared commits.

Three Rules

1. The current branch tracks new commits
2. When you move to another commit, Git updates your working directory
3. Any commit, blob or tree that cannot be reached from branch, HEAD or tag is considered dead and can be garbage collected

GitHub

* Fork
* Upstream
* Pull request

Git commands

* git --version

Git clone

* Takes the address of a git repository and downloads project and git directory/history locally, git clone [url]

Git status

* Get current status of git, git status

Git add

* Stage object, Git add ./
* Stage object, Git add [file name]

Git commit

* Commit object, git commit -m “First commit!”

Git log

* Show commit logs, git log

Git push

* Send local git (updated objects and branches) to remote git, git push
* Force push (generally, not a good idea), git push -f

Git branch

* List local branches, git branch
* List all the branches including remote branches, git branch --all
* Create new branch, git branch [branchName]

Git checkout

* Make branch current branch (move HEAD) and update working area, git checkout [branchName]

Git merge

* Merch branch to current branch, git merge [branchName]

Get fetch

* When remote changes and conflicts with local that needs to be reconciled, git fetch
* Now that we have the remote changes, we can merge local changes with remote history and push result

Git pull

* A fetch followed by a merge, git pull

Git rebase

* git rebase [branchName]

Git tag

* Create annotated tag, Git tag -a mytag -m “I love cheesecake.”
* List tags, Git tag

Git cat-file

* Show object type, Git cat-file [SHA-1] -t
* Print object contents, Git cat-file [SHA-1] -p

Git hash-object

* Get SHA-1 of object, echo "Apple Pie" | git hash-object --stdin

Git count-objects

* Returns the size and count of objects in the object database

Git show-ref

* Lists all the branches that have master in their names, git show-ref master

Git remote

* Git remote add origin [url]

# Docker

Docker is a set of platform as a service (Paas) products that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, ibraries and configuration files; they can communicate with each other through well-defined channels.

# Node.js and npm (Node Package Manager)

An open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting – running scripts server-side to produce dynamic web page content before the page is sent to the user’s web browser.

Node -version

Node start

Npm -version

# MongoDB

Mongo is