Git >

# Git Cheat Sheet

1. go to dir that you want to use as a repository
```
$> cd /srv/salt
$> ls
top.sls
curl
python
vmware
etc...
```
2. add a Read Me file that describes your code
```
echo "# Ubuntu configuration using SaltStack" >> README.md
```
3. initialize Git
```
$> git init
```
4. add a README file
```
$> git add README.md
```
5. mark all your current files in /srv/salt to be pushed to Origin (github)
```
$> git add *
```
6. add a Commit message
```
$> git commit -m "my first commit"
```
7. add the remote ORIGIN repo thats on GitHub
```
$> git remote add origin https://github.com/username/name-of-repo.git
```
8. Check remote origins that are configured
```
git remote -v
origin ssh://github...somerepo.git
```
9. Delete origin if necessary
```
git remote rm origin
```
10. configure Global variables
```
$> git config --global user.email "your.email@email.com"
$> git config --global user.name "your-github-username"
```
11. Pull Github files down to your local repo (update your local repo with latest from Github)
```
$> git pull origin master
```
12. Push all your local files to Github
```
$> git push -u origin master
```
13. Make some changes to your local files, commit them and push again to Origin
```
$> git commit -a -m "updated files"
$> git push origin master
```

Rename Git branch

```
git branch -m old_branch new_branch # Rename branch locally

git push origin :old_branch # Delete the old branch

git push --set-upstream origin new_branch # Push the new branch, set local branch to track the new remote
```

# ADVANCED

Delete local branch
```
git branch -D <branchName>
```

Delete branch on origin (github)
```
git push origin --delete <branchName>
```

See all commits in a branch
```
git show-branch <branchName>
```

show all commits on origin
```
git log --oneline --no-merges origin/master --format="format:* %s" | sort
```

unstage added files (opposite of 'git add')
```
git reset
```

**Squash Commits**

```
git rebase -i origin/master

This will bring up your text editor (-i is for "interactive") with a file that looks like this:


pick 16b5fcc Code in, tests not passing
pick c964dea Getting closer
pick 06cf8ee Something changed
pick 396b4a3 Tests pass
pick 9be7fdb Better comments
pick 7dba9cb All done

Change all the pick to squash or "s" except the first one


pick 16b5fcc Code in, tests not passing
squash c964dea Getting closer
squash 06cf8ee Something changed
squash 396b4a3 Tests pass
squash 9be7fdb Better comments
squash 7dba9cb All done

or use 'delete' to delete a commit entirely
```

Save your file and exit your editor.

Then another text editor will open to let you combine the commit messages from all of the commits into one big commit message.
comment out the commit messages you dont want, leave 1 uncommented

Save and exit, **:wq**

To modify commit msg again,
git commit --amend

change commit message
git commit --amend

show all commits and messages in repo
```
git log --oneline
```

wack the commit that youre sitting on, revert to previous commit
```
git reset --hard HEAD~1
```

purge all unstaged changes on a branch (discard changes)

```
git clean -df
git checkout -- .
```

disable SSL verify

```
git config --global http.sslVerify false
```

Stash changes - if you made changes on Master and need to pull latest Master commits from remote, but dont want to lose changes,

```
(master) $> git stash
Saved working directory and index state WIP on master: d464b3a added image
HEAD is now at d464b3a added image

switch to new branch
(master) $> git checkout -b "new_branch"

pop stashed changed into this branch
(new_branch) $> git stash pop
```

**Pull a specific branch from remote to local**

*show all branches including hidden*

```
git branch -a
```

*activate remote branch as a local branch*

```
git checkout <name of remote branch>
```

# EXAMPLE FLOW

Git squash previous commits into 1, force push

```
git add --all
git commit -m "update"
git rebase -i origin/master  (select "s" for all commits other than 1st top one, :wq, comment out all commit messages other than 1st)
git push -f origin mybranch
```

Delete a remote branch

git push origin --delete <name of remote br>