GIT

===

git clone ADDRESS

la = ls -la

# Header

## Sub-Header

git status

git add file1 file2

git add .

git commit -m "Message"

git commit -m "Subject-Message" -m "Description"

{

ssh-keygen -t rsa -b 4096 -C "EMAIL"

...

}

git push origin master

git init

git remote add NAME ADDRESS

git remote

git remote -v

verbose

{

git push -u origin master

-u = --set-upstream

az in be bad faghat bezan 'git push'

}

git branch

git checkout EXIST-BRANCH-NAME

git checkout -b NEW-BRANCH-NAME

git diff BRANCH-NAME

git merge BRANCH-NAME

git branch -d BRANCH-NAME

delete

git commit -am "Message"

add, it's only for 'modified' files not 'created' files.

git add + git commit

git reset

soft

reset to Un-Staged

git reset FILE-NAME

git reset HEAD~1

one previous commit

git reset COMMIT-ID

git reset --hard COMMIT-ID

all changes will be Un-Staged and Removed

{

changes = Un-Staged

add = Staged

commit = commit

}

git log

========

restore

git config --global user.name "NAME"

.email

.editor "code --wait"

git config --global core.autocrlf input

for linux and mac

true

for windows

git config --global -e

open the config file in the default editor

git COMMAND --help

git COMMAND -h

gives a short summery of this command

git help COMMAND

opens a browser to the document/COMMAND page

git restore --staged FILE-NAME

ls -a

'git rm' is just like the 'rm' command in linux.

git add PATTERN

echo TEXT >> FILE.EXT

append

git rm --cached FILE-NAME

to unstage

git mv FILE-NAME-1 FILE-NAME-2

git rm FILE-NAME

.gitignore

git ls-files

same as 'git status' but listed and clean

\*{

vaghti yek file/folder ra add+commit mikoni bad mizarish to .gitignore file ama taghirat track mishe, che baiad kard:

git rm --cached FILE

git rm -r --cached FOLDER

}

git status -s

short

git diff --staged

chia gharare commit beshan ke add shodan, che taghirati

git diff

chia gharare add/stage beshan, che taghirati

setting git diff configs:

git config --global diff.tool vscode

git config --global difftool.vscode.cmd = "code --wait --diff $LOCAL $REMOTE"

'git difftool' bejaye 'git diff'

git log --oneline

on line description

--reverse

in reverse order

git show COMMIT-ID

git show HEAD

git show HEAD~1

git show COMMIT-ID:FILE-OR-FOLDER

for see the changes on a special file/folder/path

git ls-tree COMMIT-ID

file and folders at that COMMIT-ID

git reset --soft COMMIT-ID

--hard

undo the 'add' operation (Un-Stage):

git restore --staged FILE

git clean -fd

f: force

d: directories

for discarding local changes; to remove all new un-tracked file/folder

git restore FILE

get the previous version of that FILE

restore a file to the previous version:

git restore --source=HEAD~1 FILE

=====================================

DOCKER

======

when you run an image, then a container will be created-running.

docker run ubuntu COMMAND INPUT-OPTIONS-FOR-THE-COMMAND

docker run ubuntu echo helllllllllo

execute a command on my docker container:

docker exec CONTAINER-IMAGE-NAME COMMAND INPUT-OPTIONS-FOR-THE-COMMAND

=================================

balaye docker compose bayad version ra moshakhas konid:

"

version: 2

services:

redis:

image: redis

"

==============================

vaghti az version 'alpine' estefade mikoni dige 'bash' nadari chon 'alpine' kochike vali bejash 'sh' ke hamon shell-e default hast ro dari.

docker run -it IMAGE-NAME bash

for 'alpine' :

docker run -it IMAGE-NAME sh

# hazfe tamame container haye stop shode:

1:

docker container prune

# hazfe tamame image haye bedardnakhor va bi nam:

2:

docker image prune

docker run -p bironi:daroni

docker run -p (HOST)3000:(CONTAINER)5000

docker run -p 3000:5000

Execute a command in a running container:

docker exec CONTAINER-ID ANY-LINUX-OR-WINDOWS-COMMANDS

docker exec -it CONTAINER-ID ANY-LINUX-OR-WINDOWS-COMMANDS

docker exec -it CONTAINER-ID sh

docker exec -it CONTAINER-ID bash

'docker start' vs 'docker run' :

docker start :

start mikone yek container-e stop shode

docker run :

yek container jadid az 'IMAGE'-e delkhah ejra mikone

VOLUMES:

docker volume create VOLUME-NAME

docker run -v VOLUME-NAME:/app/a-name-for-directory IMAGE-NAME

docker run -v NEW-VOLUME:/app/a-name-for-directory IMAGE-NAME

NOTE: you have to mkdir that directory in Dockerfile with 'RUN mkdir dir-name' to prevent any permission issues.

copy from a container to host(folder):

docker cp CONTAINER-ID:/app/path/file.txt .

in noghte yani poshe ee ke Dockerfile toshe

from host to container:

docker cp file.txt CONTAINER-ID:/app/path

docker cp ./file.txt CONTAINER-ID:/app

bind/binding/link-using-volumes:

When we want to link a path/folder in host(where project and/or Dockerfile is) to a container:

docker run -v $(pwd):/app IMAGE-NAME

docker run -v .:/app IMAGE-NAME

====================================

docker compose up

# manande 'run' ast

# docker compose baraye kar ba multi container ast

docker compose up --build

docker compose up -d

detach

docker compose ps

docker compose down

going in with a special user:

docker exec -it -u root 243hfg9gfd sh

docker exec -it -u root 243hfg9gfd bash

docker exec -it -u amir 243hfg9gfd sh

docker compose logs

docker compose logs 2j3h44hg23

docker compose logs 2j3h44hg23 -f

follow