

**Docker Commands:-**

Basic commands to know versions and installations

* docker --version
* docker-compose --version
* docker-machine --version
* docker-machine ip
* docker version
* docker info
* docker images(to get know the list of images that having )
* docker login
* docker logs
* docker ps(to get the list of containers which are actually running)
* docker ps-a(to get know the which container is running latest.)
* docker rm

docker rmi [options] container [image …]

--force, -f false Force removal of the running containers

--link, -l false remove the specified link.

--volumes, -v false remove the volumes associated with containers.

* docker rmi

docker rmi [options] Image [image …]

--force, -f false Force removal of the image

--no-prune false don’t delete untagged parents.

Note:--Either container id or image name

* docker rmi $(docker images -a -q) (to remove all the images)
* docker stop $(docker ps -a -q) (to stop all containers in one step)
* docker rm $(docker ps -a -q) (to remove all stopped containers in one step)
* docker run [OPTIONS] IMAGE[:TAG|@DIGEST] [COMMAND] [ARG...]
* docker-machine ls (To get know running vm)
* docker volume ls (to know the volume and host)
* -**d**will create a container with the process detached from our terminal
* **-P** will publish all the exposed container ports to random ports on the Docker host
* **-e** is how you pass environment variables to the container
* **--name** allows you to specify a container name
* AUTHOR is the environment variable name and Your Name is the value that you can pass
* docker network ls (To know the network)

**Git Commands:-**

* git –version
* git init
* git status
* git add .

git add <file>

* git commit –m “comments”
* git log
* git push

git push origin [branch name]

git pull origin [branch name]

* git pull
* git clone -b <branch> <remote\_repo>

git clone -b develop [git@github.com:user/myproject.git](mailto:git@github.com:user/myproject.git)

* git config –-global user.name

(To get the current working user)

* git config --global user.email

(To get the current email of user)

* git config –global.alias.<alias-name> <git-commnad>

(To define the author name to be used for all commits by the current user)

* git config –system core.editor <editor>
* git config -–global core.autocrlf true

(To avoid spaces)

* git config -–global core.autocrlf false

CRLF (carriage return line feed)

**Git Stash:-**

* git stash
* git stash pop <stash@{0}> (To reapply previous stashed changed)
* git stash apply
* git stash save “<comments>”
* git stash list
* git stash show
* git stash show -p
* git stash –p (partial stashes)
* git stash brach <branchname> stash@{1} (creating branch from your stash)
* git stash drop stash@{1} (To cleanup you stash)
* git stash clear (to clean all stashes)

Linux Commands:--

* ls (Listing Files)
* **ls –al (gives the detailed information about files )**
* **ls –a (will give u the list of hidden files )**
* **cat > file (to create the file )**
* **cat file (to view the file )**
* **cat file1 file 2> sample (To combine the files)**
* **cat sample to view the files**

**O/p: file1 file 2**

* **rm file (To delete a file)**
* **mv filename “new\_file\_location“ (To move the file path Note : mv needs root permission)**
* **sudo command\_you\_want\_to\_execute (sudo give the security permissions if mv not works use with sudo and type command.)**
* **mv filename new\_file\_name (for renaming the file && directory)**
* **directory manipulations**
* **mkdir directory\_name (to create a new directory)**
* **rmdir directory\_name (to remove the directory)**
* **man (gives help information on the command)**
* **history (history command will show the all the commands u have executed )**
* **clear (clears the terminal)**
* **apt-get (command used to update and install packages)**
* **ctrl+c (copy)**
* **ctrl+shift+p (paste)**
* **pr (used for formatting the file for printing in the terminal)**
* **pr –x (Divides the file into columns)**
* **pr –h(assigns a header to the file)**
* **pr –n (Denotes the file with line numbers)**
* **owner permissions**
* **ls –l (gives in the terminal about file permissions)**
* **-rw- rw-r- - (file name ,first rw give the user and second rw is group ,r- - is others)**
* **r read**
* **w write**
* **x execute permission**
* **Note :- ‘-’ first symbol give the file and rw- (give the no permission to execute the file)**
* **chmod permissions filename (change mode )**
* **sudo chown chaitu sample.txt (we can change the permissions of an user from root to a specific user )**
* **sudo chown root : root sample.txt (changing the ownership to root from the owner)**
* **chgrp (stands for change the group)**
* **ls -dl smaple.txt (to check the current owner of the file)**
* **sudo chgrp root sample.txt (to change the owner to root for file)**
* **grep (to search string in a file)**
* **grep - i a (matches the string with both upper and lower case having a)**
* **pwd (gives the current user and directory)**
* **editing the file using vi is vi filename**

|  |  |
| --- | --- |
| **Keystrokes** | **Action** |
| i | Insert at cursor **(goes into insert mode)** |
| a | Write after cursor **(goes into insert mode)** |
| A | Write at the end of line **(goes into insert mode)** |
| ESC | Terminate insert mode |
| u | Undo last change |
| U | Undo all changes to the entire line |
| o | Open a new line **(goes into insert mode)** |
| dd 3dd | Delete line Delete 3 lines. |
| D | Delete contents of line after the cursor |
| C | Delete contents of a line after the cursor and insert new text. Press ESC key to end insertion. |
| dw 4dw | Delete word Delete 4 words |
| cw | Change word |
| x | Delete character at the cursor |
| r | Replace character |
| R | Overwrite characters from cursor onward |
| s | Substitute one character under cursor continue to insert |
| S | Substitute entire line and begin to insert at the beginning of the line |
| ~ | Change case of individual character |

Make sure you press the right command otherwise you will end up making undesirable changes to the file. You can also enter the insert mode by pressing a, A, o, as required.

**Moving within a file**

You need to be in the command mode to move within a file. The default keys for navigation are mentioned below else; You can **also use the arrow keys on the keyboard**.

|  |  |
| --- | --- |
| **Keystroke** | **Use** |
| k | Move cursor up |
| j | Move cursor down |
| h | Move cursor left |
| l | Move cursor right |

**Saving and Closing the file**

You should be in the **command mode to exit the editor and save changes** to the file.

|  |  |
| --- | --- |
| **Keystroke** | **Use** |
| Shift+zz | Save the file and quit |
| :w | Save the file but keep it open |
| :q | Quit without saving |
| :wq | Save the file and quit |