The Linux Command Line

Managing packages

```
apt update
apt list
apt install nano
apt remove nano
```

Navigating the file system

```
pwd          # to print the working directory
ls           # to list the files and directories
ls -l        # to print a long list
cd /         # to go to the root directory
cd bin       # to go to the bin directory
cd ..        # to go one level up
cd ~         # to go to the home directory
```

Manipulating files and directories

```
mkdir test               # to create the test directory
mv test docker           # to rename a directory
touch file.txt           # to create file.txt
mv file.txt hello.txt    # to rename a file
rm hello.txt             # to remove a file
rm -r docker             # to recursively remove a directory
```

Editing and viewing files

```
nano file.txt        # to edit file.txt
cat file.txt         # to view file.txt
less file.txt        # to view with scrolling capabilities
head file.txt        # to view the first 10 lines
head -n 5 file.txt   # to view the first 5 lines
tail file.txt        # to view the last 10 lines
tail -n 5 file.txt   # to view the last 5 lines
```

Searching for text

```
grep hello file.txt        # to search for hello in file.txt
grep -i hello file.txt     # case-insensitive search
grep -i hello file*.txt    # to search in files with a pattern
grep -i -r hello .         # to search in the current directory
```

Finding files and directories

```
find                # to list all files and directories
find -type d        # to list directories only
find -type f        # to list files only
find -name “f*”     # to filter by name using a pattern
```

Managing environment variables

```
printenv             # to list all variables and their value
printenv PATH        # to view the value of PATH
echo $PATH           # to view the value of PATH
export name=bob      # to set a variable in the current session
```

Managing processes

```
ps                   # to list the running processes
kill 37              # to kill the process with ID 37
```

Managing users and groups

```
useradd -m john      # to create a user with a home directory
adduser john         # to add a user interactively
usermod              # to modify a user
userdel              # to delete a user

groupadd devs        # to create a group
groups john          # to view the groups for john
groupmod             # to modify a group
groupdel             # to delete a group
```

File permissions

```
chmod u+x deploy.sh     # give the owning user execute permission
chmod g+x deploy.sh     # give the owning group execute permission
chmod o+x deploy.sh     # give everyone else execute permission

chmod ug+x deploy.sh    # to give the owning user and group
                        # execute permission

chmod ug-x deploy.sh    # to remove the execute permission from
                        # the owning user and group
```

Images

Dockerfile instructions

```
FROM          # to specify the base image
WORKDIR       # to set the working directory
COPY          # to copy files/directories
ADD           # to copy files/directories
RUN           # to run commands
ENV           # to set environment variables
EXPOSE        # to document the port the container is listening on
USER          # to set the user running the app
CMD           # to set the default command/program
ENTRYPOINT    # to set the default command/program
```

Image commands

```
docker build -t <name> .
docker images
docker image ls
docker run -it <image> sh
```

Starting and stopping containers

```
docker stop <containerID>
docker start <containerID>
```

Removing containers

```
docker container rm <containerID>
docker rm <containerID>
docker rm -f <containerID>        # to force the removal
docker container prune            # to remove stopped containers
```

Volumes

```
docker volume ls
docker volume create app-data
docker volume inspect app-data
docker run -v app-data:/app/data <image>
```

Copying files between the host and containers

```
docker cp <containerID>:/app/log.txt .
docker cp secret.txt <containerID>:/app
```

Sharing source code with containers

```
docker run -v $(pwd):/app <image>
```

Containers

Running containers

```
docker run <image>
docker run -d <image>              # run in the background
docker run —name <name> <image>    # to give a custom name
docker run —p 3000:3000 <image>    # to publish a port HOST:CONTAINER
```

Listing containers

```
docker ps                          # to list running containers
docker ps -a                       # to list all containers
```

Viewing the logs

```
docker logs <containerID>
docker logs -f <containerID>       # to follow the log
docker logs —t <containerID>       # to add timestamps
docker logs —n 10 <containerID>    # to view the last 10 lines
```

Executing commands in running containers

```
docker exec <containerID> <cmd>
docker exec -it <containerID> sh   # to start a shell
```

Starting and stopping containers

```
docker stop <containerID>
docker start <containerID>
```

Removing containers

```
docker container rm <containerID>
docker rm <containerID>
docker rm -f <containerID>          # to force the removal
docker container prune              # to remove stopped containers
```

Volumes

```
docker volume ls
docker volume create app-data
docker volume inspect app-data
docker run -v app-data:/app/data <image>
```

Copying files between the host and containers

```
docker cp <containerID>:/app/log.txt .
docker cp secret.txt <containerID>:/app
```

Sharing source code with containers

```
docker run -v $(pwd):/app <image>
```

Multi-containers apps

Docker Compose commands

```
docker-compose build
docker-compose build --no-cache
docker-compose up
docker-compose up -d
docker-compose up —build
docker-compose down
docker-compose ps
docker-compose logs
```