Commands And Their Description

- **sudo apt update:** command is used to download package information from all configured sources.

- **sudo apt install apt-transport-https ca-certificates curl software-properties-common -y:** the command to allow your operating system to access the Docker repositories over HTTPS and download the dependencies.

  - **apt-transport-https** allows the package manager to transfer files and data over https.
  - **ca-certificates** allows the system to check security certificates.
  - **curl** is a tool for transferring data.
  - **software-properties-common** adds scripts for managing software

- **curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add :** add GPG keys to ensure the software we're installing is authentic.

- **sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable" :** Here **$(lsb_release –cs**) scans and returns the codename of our Ubuntu installation – In our case, its bionic and the **stable** is the type of docker release.

- **sudo apt install docker-ce -y :** the latest version of docker community edition.

- **sudo systemctl status docker :** the dockers need to be run at startup.

- **sudo docker pull nginx :** This downloads all the necessary components for the container. Docker will cache these, so when we run the container we don't need to download the container image(s) each time.

- **sudo docker run --name docker-nginx -p 80:80 nginx**

  - run is the command to create a new container
  - The --name flag is how we specify the name of the container (if left blank one is assigned for us, like nostalgic_hopper from Step 2)
  - -p specifies the port we are exposing in the format of -p local-machine-port:internal-container-port. In this case we are mapping Port 80 in the container to Port 80 on the server
  - nginx is the name of the image on dockerhub.

- **sudo docker container run -d -p 3306:3306 --name <Container Name> -e MYSQL_ROOT_PASSWORD=yes mysql** : Run a commond in new container.
  - -d run the container in detached mode (in the background)
  - -p map port 80 of the host to port 80 in the container
  - -e Set environment variable

- **sudo docker container logs <Container Name>** : Fetch the logs of container.
  - --details Show extra details provided to logs
  - -f Follow log output
  - --since Show logs since timestamp (e.g. 2013-01-02T13:23:37Z) or relative (e.g. 42m for 42 minutes)
  - -t Show timestamps
  - --until Show logs before a timestamp (e.g. 2013-01-02T13:23:37Z) or relative (e.g. 42m for 42 minutes)

- **sudo vim <File name>** : Creating a file.

- **sudo docker container logs <Container Name>** > <File name> : appending logs to file.

- **sudo docker container run -d -p 8080:80 --name apache httpd** : Used to run apache web server.

- **sudo docker container run -d -p 80:80 --name web nginx** : Used to run nginx web server

- **sudo docker container ls** : List containers

- **sudo docker container ls -a** : Show all containers (default shows just running)
  - -f Filter output based on conditions provided
  - -n Show n last created containers (includes all states)
  - -l Show the latest created container (includes all states)
  - --no-trunc Don't truncate output
  - --quiet , -q Only display container IDs
  - --size , -s Display total file sizes
  - --format Pretty-print containers using a Go template

- **sudo docker container rm web apache <Container Name>** : Remove one or more containers
    - --force , -f Force the removal of a running container (uses SIGKILL)
    - --link , -l Remove the specified link
    - --volumes , -v Remove anonymous volumes associated with the container

- **sudo docker container stop web apache <Container Name>** : Stop one or more running containers.
    - --time , -t Seconds to wait for stop before killing it

- **sudo docker images** : List images
    - --all , -a Show all images (default hides intermediate images)
    - --digests Show digests
    - --filter , -f Filter output based on conditions provided
    - --format Pretty-print images using a Go template
    - --no-trunc Don't truncate output
    - --quiet , -q Only show image IDs
- **sudo docker container top** : Display the running processes of a container.

- **sudo docker container inspect** : Display detailed information on one or more containers
    - --format , -f Format the output using the given Go template
    - --size , -s Display total file sizes

- **sudo docker container stats** : Display a live stream of container(s) resource usage statistics
    - --all , -a Show all containers (default shows just running)
    - --format Pretty-print images using a Go template
    - --no-stream Disable streaming stats and only pull the first result
    - --no-trunc Do not truncate output