**Web Application Development**

**Assignment: 02 – (B)**

***Step 1 — Removing previously installed Docker and updating the packages.***

1. sudo apt-get remove docker docker-engine docker.io containerd runc: Uninstall any older versions before attempting to install a new version
2. Update the apt package index and install packages to allow apt to use a repository over HTTPS

sudo apt-get update

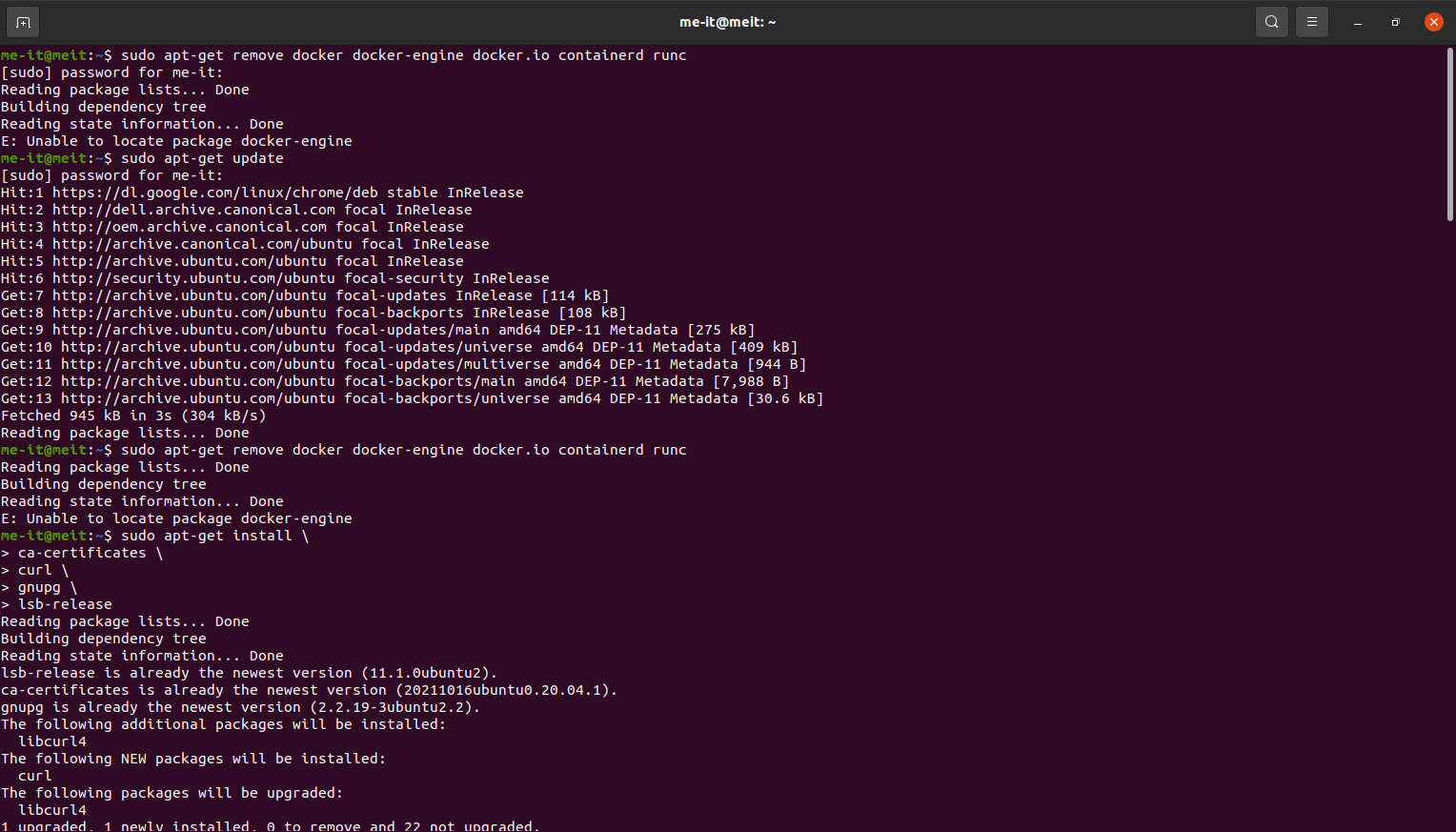
sudo apt-get install \

ca-certificates \

curl \

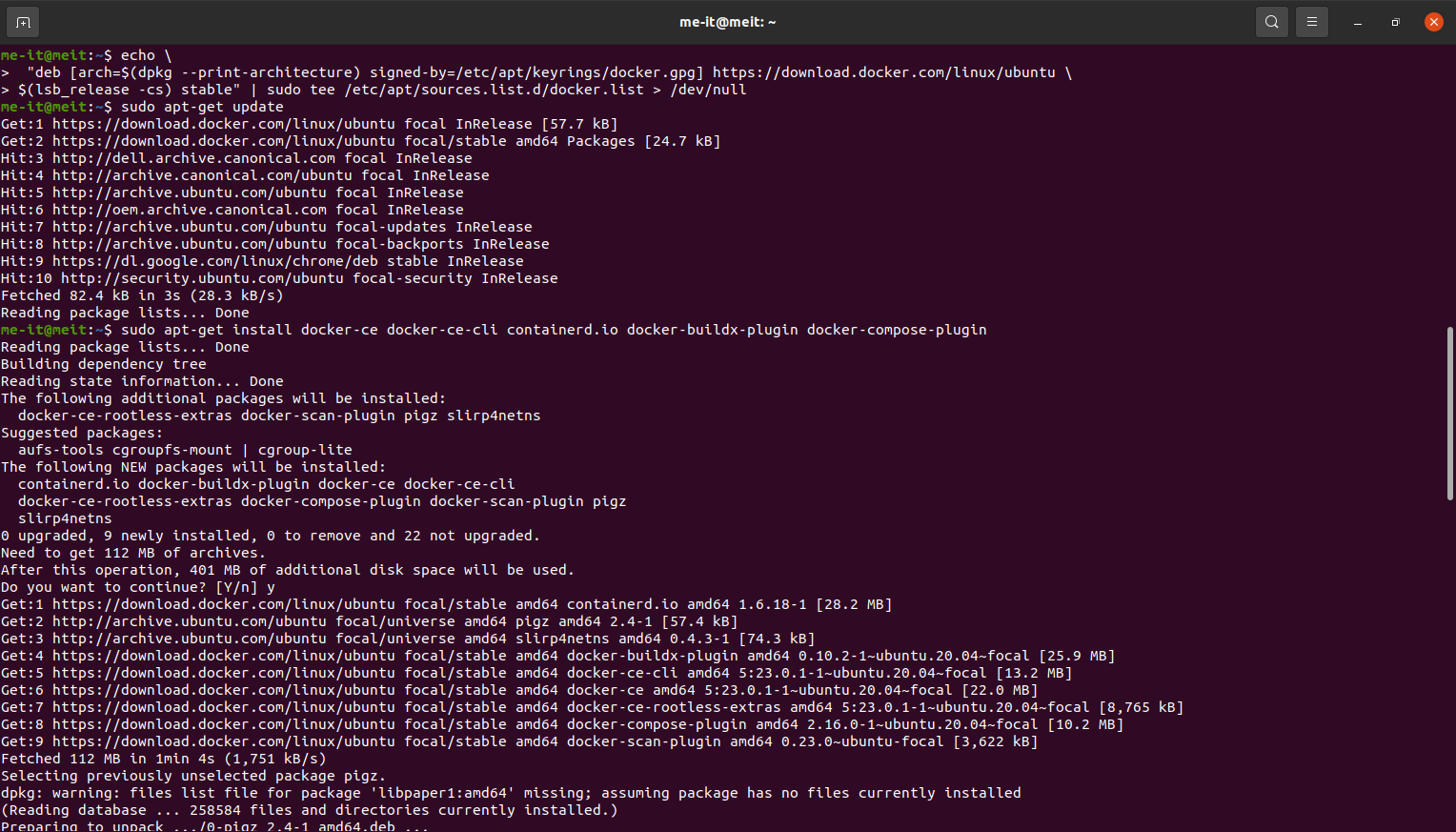
gnupg \

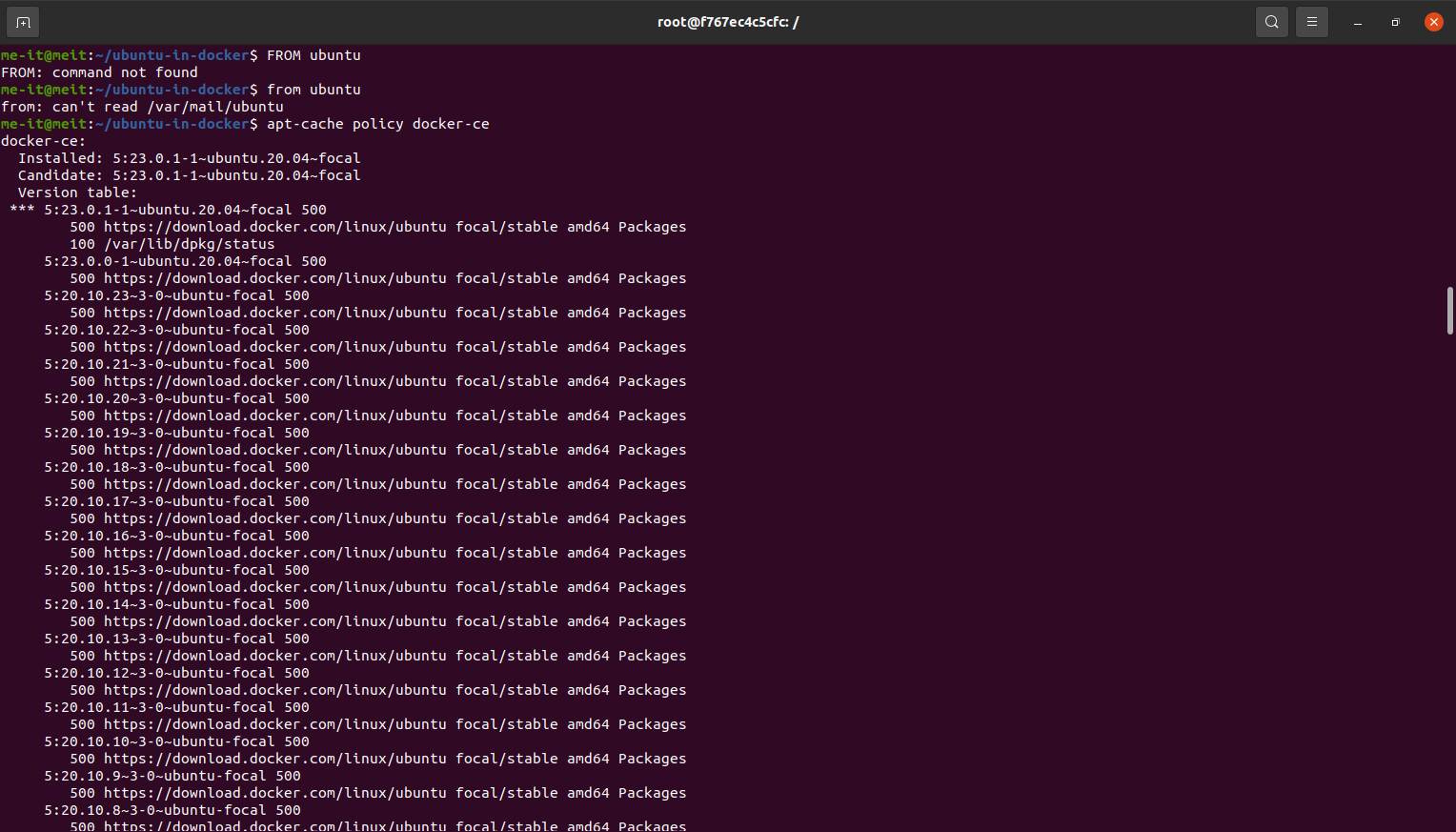
lsb-release

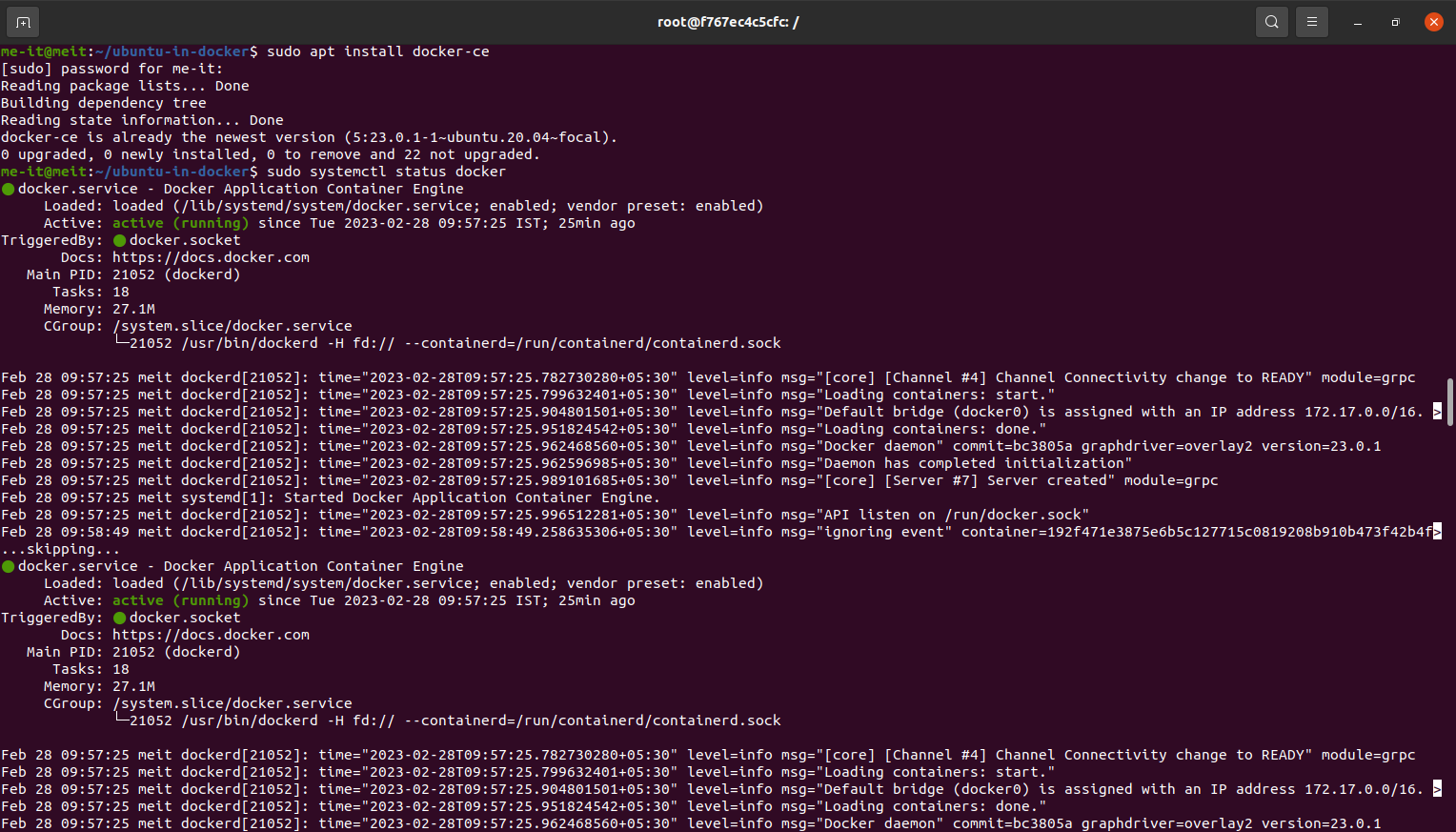


***Step 2 — Installing Docker***

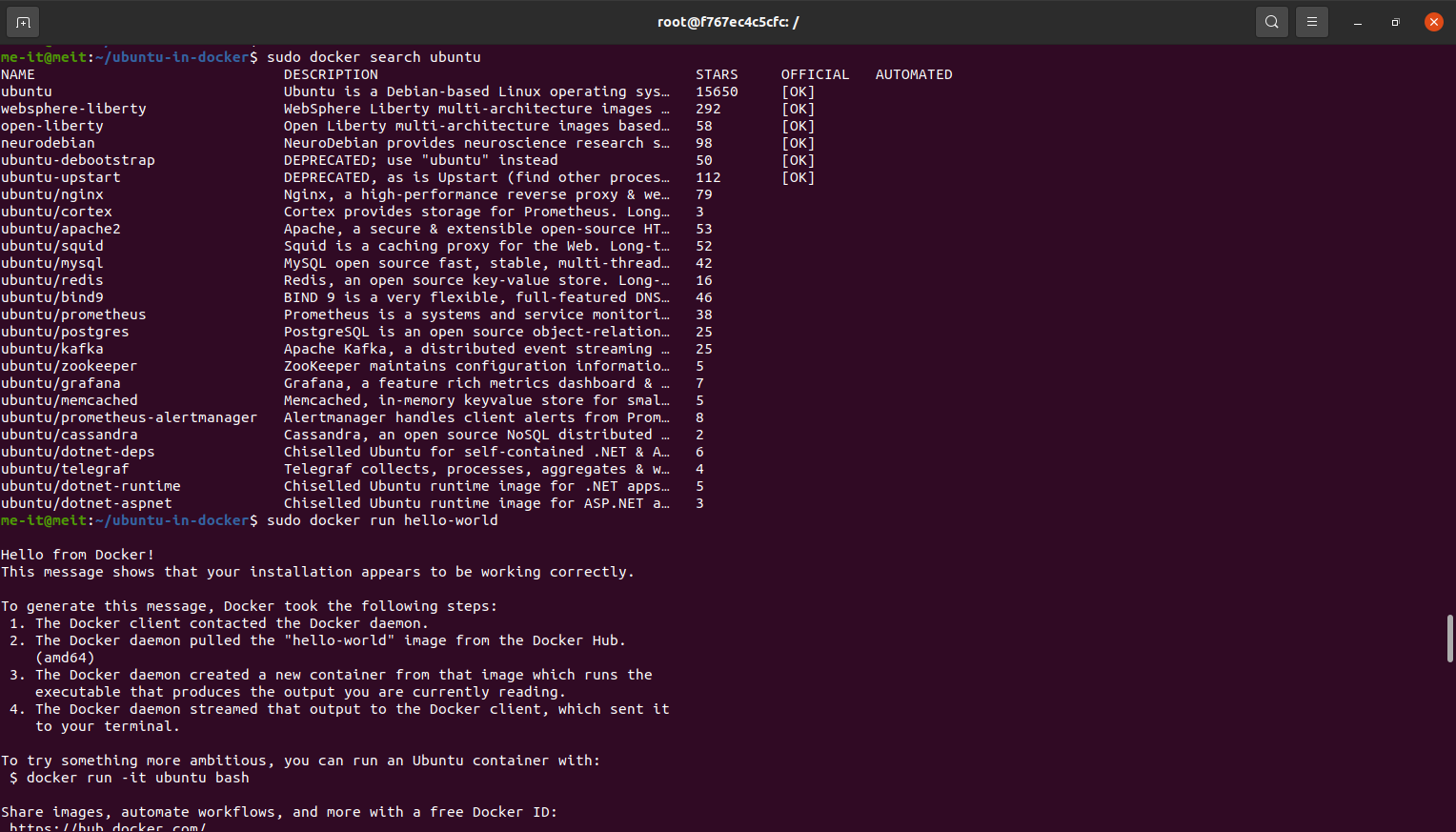
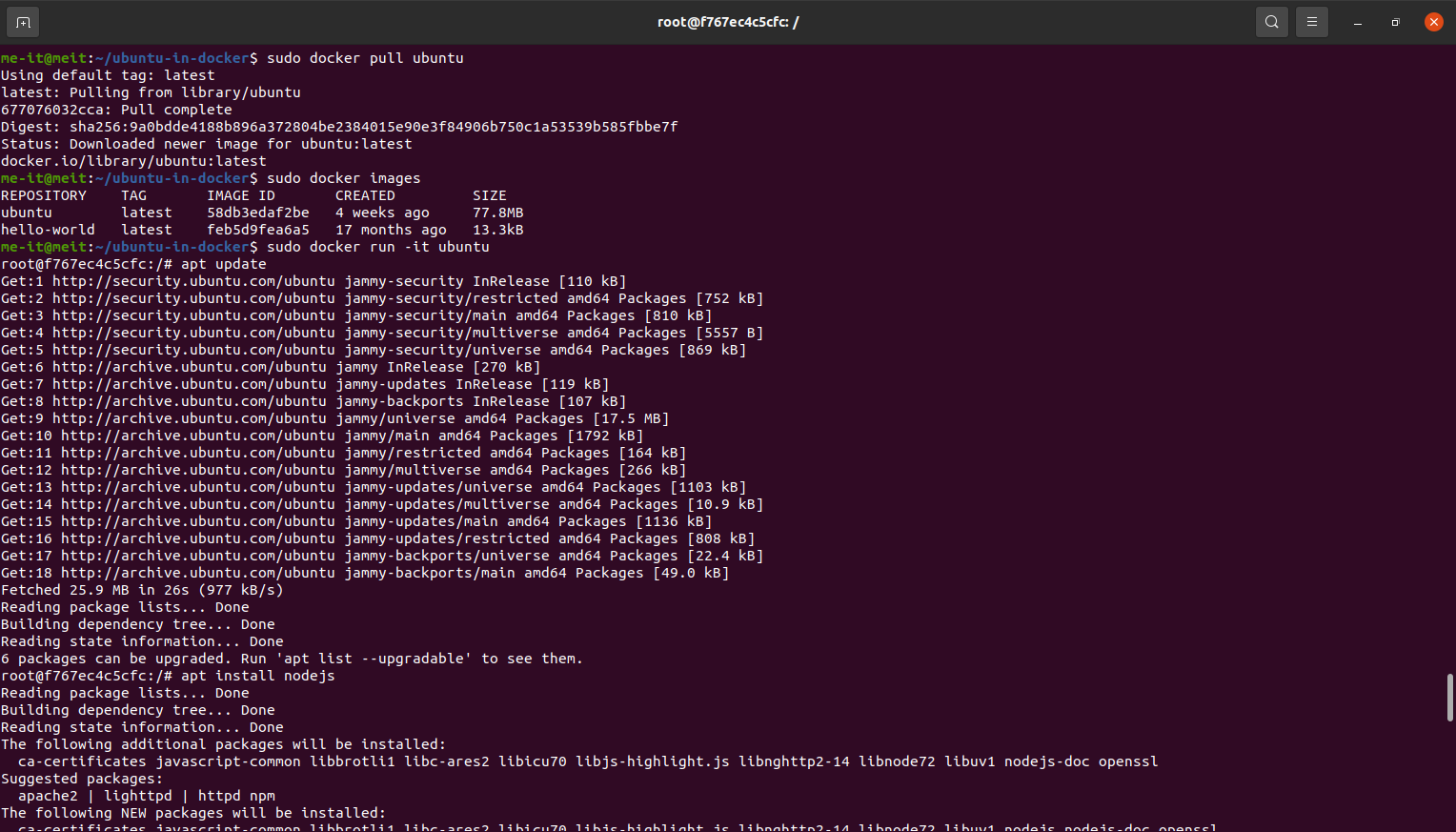
1. sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin: To install the latest version



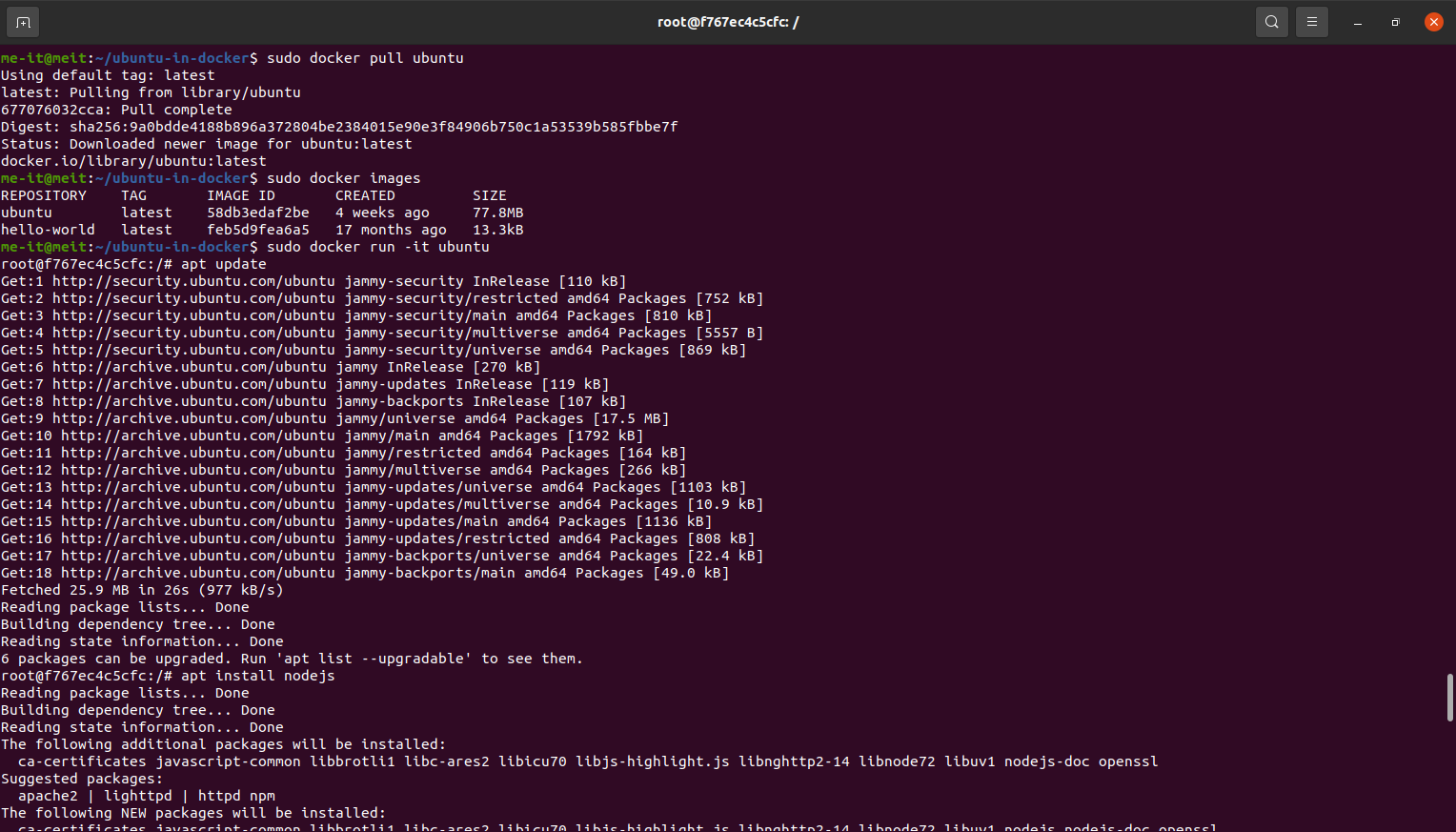
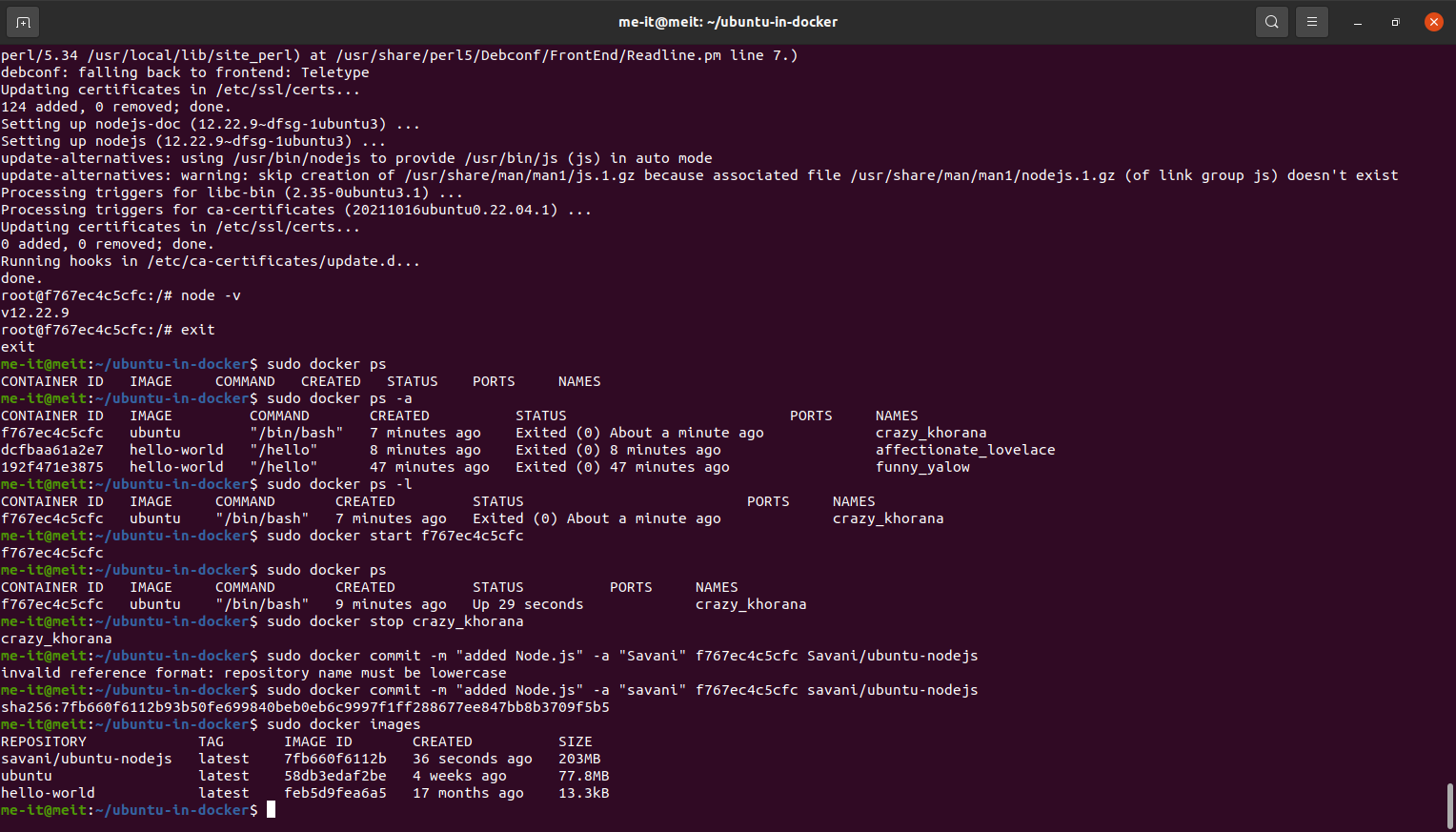
1. apt-cache policy docker-ce: to install from the Docker repo instead of the default Ubuntu repo
2. sudo systemctl status docker: Docker installed, the daemon started, and the process enabled to start on boot. Check that it’s running.



***Step 3 — Working with Docker Images:***

1. docker search ubuntu: You can search for images available on Docker Hub by using the docker command with the search subcommand.
2. docker run hello-world: To check whether you can access and download images from Docker Hub.
3. docker pull ubuntu: to download the official ubuntu image to your computer.
4. docker images: To see the images that have been downloaded to the computer.

***Step 4 — Running a Docker Container***

1. docker run -it ubuntu: to run a container using the latest image of Ubuntu. The combination of the -i and -t switches gives you interactive shell access into the container
2. apt update: update the package database inside the container.
3. apt install nodejs: install Node.js
4. node -v: When the installation finishes, verify that Node.js is installed

***Step 5 — Managing Docker Containers***

1. docker ps: to see active (running) and inactive containers on the computer. In this case, active.
2. docker ps -a: To view all containers — active and inactive, run docker ps with the -a switch
3. docker ps -l: To view the latest container you created, pass it the -l switch:
4. docker start ID: To start a stopped container, use docker start, followed by the container ID or the container’s name.
5. Text

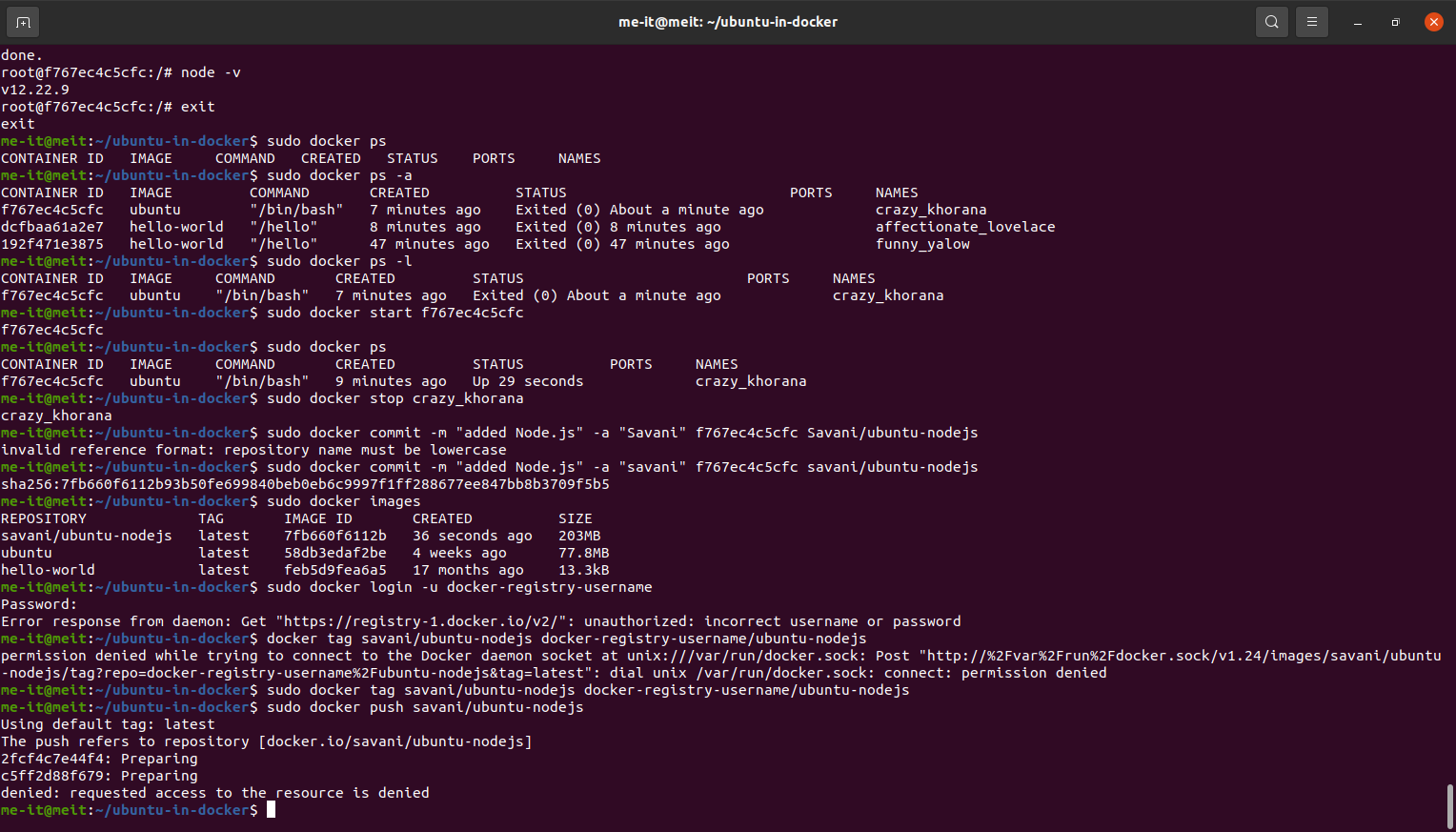
   Description automatically generateddocker stop container ID or name: To stop a running container, use docker stop, followed by the container ID or name

***Step 6 — Committing Changes in a Container to a Docker Image***

1. docker commit -m "added Node.js" -a "savani" d9b100f2f636 savani/ubuntu-nodejs: commit the changes to a new Docker image instance using the command.
2. Text

   Description automatically generateddocker images: Listing the Docker images again will show the new image, as well as the old one that it was derived.

***Step 7 — Pushing Docker Images to a Docker Repository***

1. docker login -u docker-registry-username: To push your image, first log into Docker Hub.
2. docker tag sammy/ubuntu-nodejs docker-registry-username/ubuntu-nodejs : If your Docker registry username is different from the local username you used to create the image, you will have to tag your image with your registry username.
3. docker push savani/ubuntu-nodejs: To push the ubuntu-nodejs image to the savani repository.