(Linux distribution)

Using Rosylinux as almalinux was not available due to some reason

Objective → Professor gives us a code (Prous) via github and my job is to add up parallelisation for the projet made

↳ for faster Processing of the code e.g. CUDA (NVIDIA)
GPU parallelisation

(Cloud)

Machine Configeration → 2 Core
30 Gb RAM
2 CPU

Note → This machine configeratons happens on the Cloud, it isn't the

machine infrastructure assigned on my machine (written link copy)

↳ Adding the ssh key so prof. can see our project

SSh Key → Remote Connection set up

I can think of it like anydesk where professor can see and edit the machinery but it can be called more secure (as it includes encryption & key)

All this is basically called the Virtual Machine

Questions What's SSh, key, cores RAM, what kinda RAM infrastructure they use there?

Virtual Machine

↳ Can be Cloud or own computer system based

Virtual Machine → I have my physical machine and I make a separate new operating system which is isolated from my original Operating system. e.g. Here, we made a separate operating system (Rocky Linux here) on the Cloud. We asked computer on the cloud to allocate us a particular space (30Gb disk etc)

→ If I setup a linux operating for example I did WSL and then almalinux setting up a separate environment on

top windows → This is also a Virtual Machine

Now, next we made a directory using mkdir command in the ssh terminal (after adding the key, we clicked on ssh button on the Projects screen)

Open directory, click upload files & upload the files.

→ files uploaded and we see them there now.

Now, move these files to IMAP containers folder using mv command

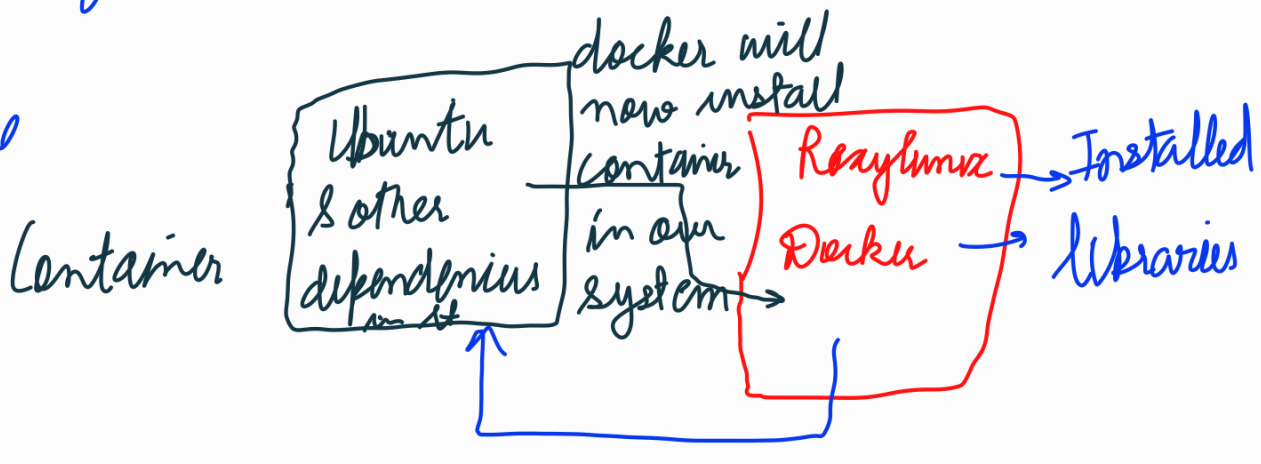
Next step : Install Relevant libraries to Run the C++ code via docker.

① update the system

docker container contains all the Relevant file dependencies

Another way of seeing this docker is we are installing another VM on top of our already Running Roxy linux VM.

Purpose → We want to Run our C++ in a predefined docker container - Ubuntu Machine



Now, Install docker in our Rockylinux Virtual Machine

↓

Later we will use this docker to get Relevant dependencies in order to Run our C++ code.

Command to chek if docker is installed already

```
docker --version
```

IMPORTANT

Now we need to enable docker so we could use docker commands.

5 shitty Commands

↓

Write them Now!!!

Now, We need dockerfill.prod and we will install this docker fill in our system now.

dockerfile is a text file which has all the instructions in it and then, it decides what our container will look like.

Opening docker file → Install nano library and use nano command

nano dockerfile.prod

cat dockerfile.prod

Docker Container docker container will be built as per the instruction in dockerfile.

Building docker command

docker build -t .... command

→ Run It

(Basically docker container is building up)

Docker Image (docker container)

docker <images>

Note → docker image is immutable, i.e. once created we cannot edit.

↳ we cannot add more libraries in it

→ We can only build a new docker if we forgot to install something.

Next step: Run the docker container now using the relevant command.

(Much like it happens with ROOT similar to that)

New docker container terminal will open

Now, we will now be in a new VM → That is our docker container environment which has relevant files already installed in it which can be used to Run C++ code.

Note → When we are in a new VM, we are independent of our previous system now → We cannot access our offline files from here

Now, Run the C++ code, we get the output :-

Grain Size Time

↓

No of parallel processes that will happen

(No. of divisions we made) → Performance keeps on improving with more Multithreading we implements but saturates at some point.

e.g. 8 → Figure will be divided into 8 different parts processed parallelly by 4 cores

Now we made an application and we will use our new docker file (Production docker) to Run the application

Now we will use separate docker container to Run this application.

we will create new docker to Run the application as application will require only libraries required to Run the application

Obvious → dependies used to build-develop application must be different from what are need to Run the application.

e.g. Now we don't need to compile the code etc.

Next Step : Build, Run etc the new docker and now Run the application with the production docker.

What is mandelbrotk based on which I have put my file Name.

nano dockerfile.prod

Used to open the test files, then we edit it do Ctrl + O and then hit enter to save the file. Then, Ctrl + X to exit.