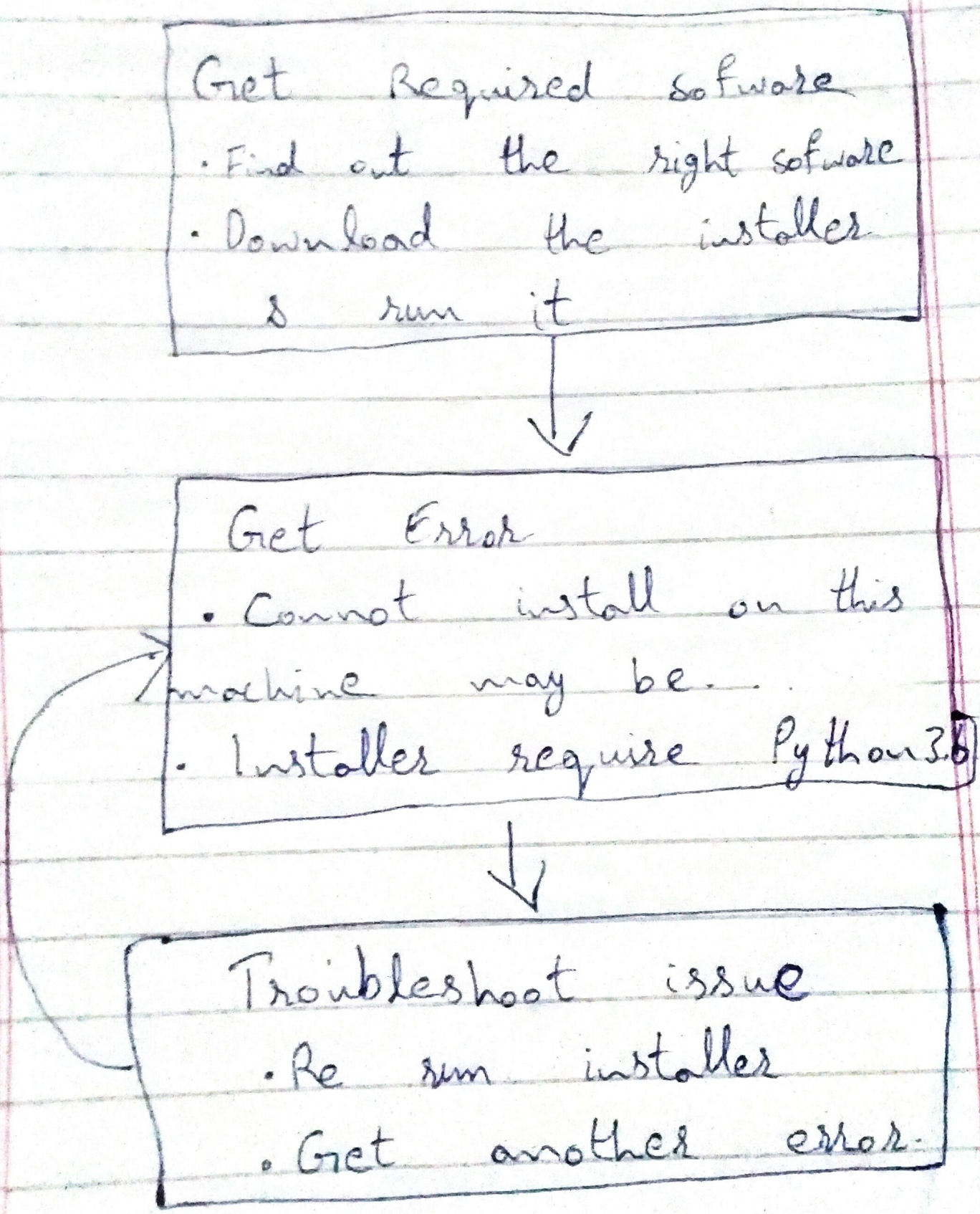
Why Should we use Docker:



Docker

Docker is an open platform for developing, shipping and running applications. Docker enables you to separate your application from your infrastructure so you can deliver software quickly.

The Docker platform:-

Docker provides the ability to package and run application in loosely isolated environment called a container. Container are lightweight and can contain everything needed to run application, so you do not neede to rely on what is currently installed on the host.

Images:

An image is a read-only template with instruction for creating docker container. For example you may build image ubuntu image, but install Apache web server and your application, as well as the configured details needed to make your ~~image~~ application run.

Containers:

A container is a runable instance of an image. you can create, start, stop, move, or delete a container using docker client. A container is defined by its image as well as any configuration ~~optic~~ option provide to it when you create or start it.

The Docker daemon:

The Docker daemon listen for Docker client request and manage Docker objects like images, container etc.

The Docker client:

The client is primary way that many Docker user interect with Docker. When you use commands like "docker run" the client send these command to Docker daemon

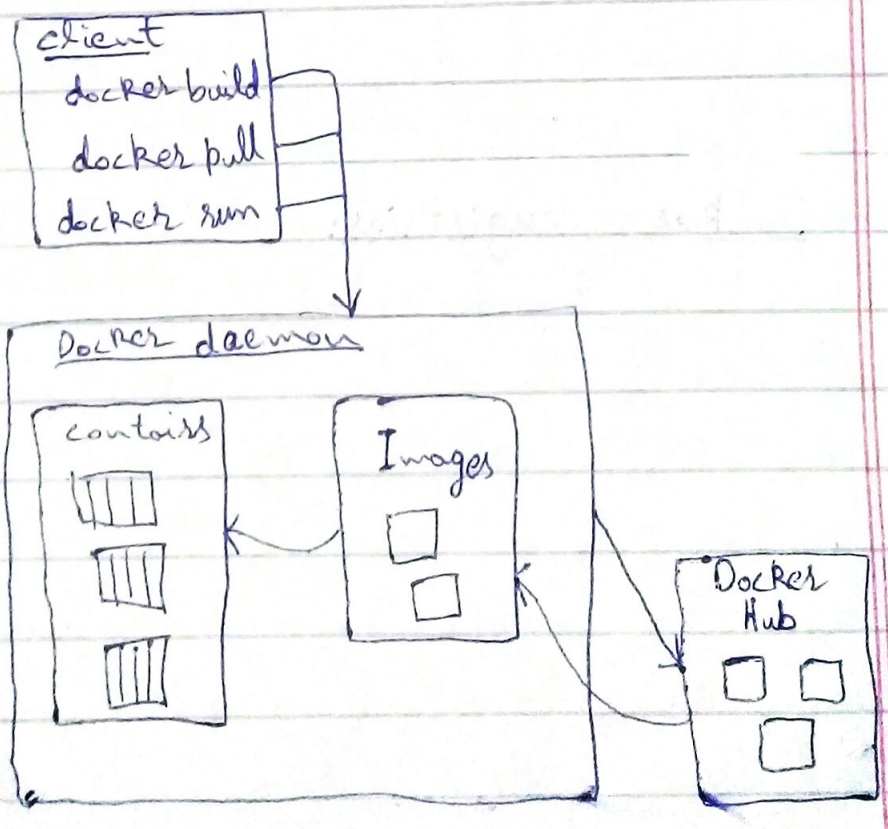
Docker registries:

A Docker registry stores Docker images. ~~containers, networks~~ etc. Docker Hub is public registry that anyone can use. When you use "docker pull" or "docker run" command

the required images are pulled from your configure registory. When you use docker push command the required images are push from the configured registory.

Docker architecture

Docker uses client server architecture. The docker client talks to the docker daemon which does the heavy lifting of building, running and distributing your Docker containers. The Docker client an deamon run on the same system, or you can connect a docker client with a remote Docker deamon



Example "docker run" command:

The following command runs an ubuntu container, attach interectively to your command line

```
$ docker run -i -t
ubuntu /bin/bash
```

when you run this command following happens:

1. if you dont have ubuntu image it pull it from Docker hub registory
2. Docker create a new container, as though you had run "docker container create" command.
3. Docker allocated a read write file system to the container
4. Docker create a network interface to connect the container to default network.

5- Docker start the container and execute bin/bash, and attach to the terminal because of "-i" and "-t" flag.

6- When you type "exit" to terminate the /bin/bash command, the container stops but not removed.