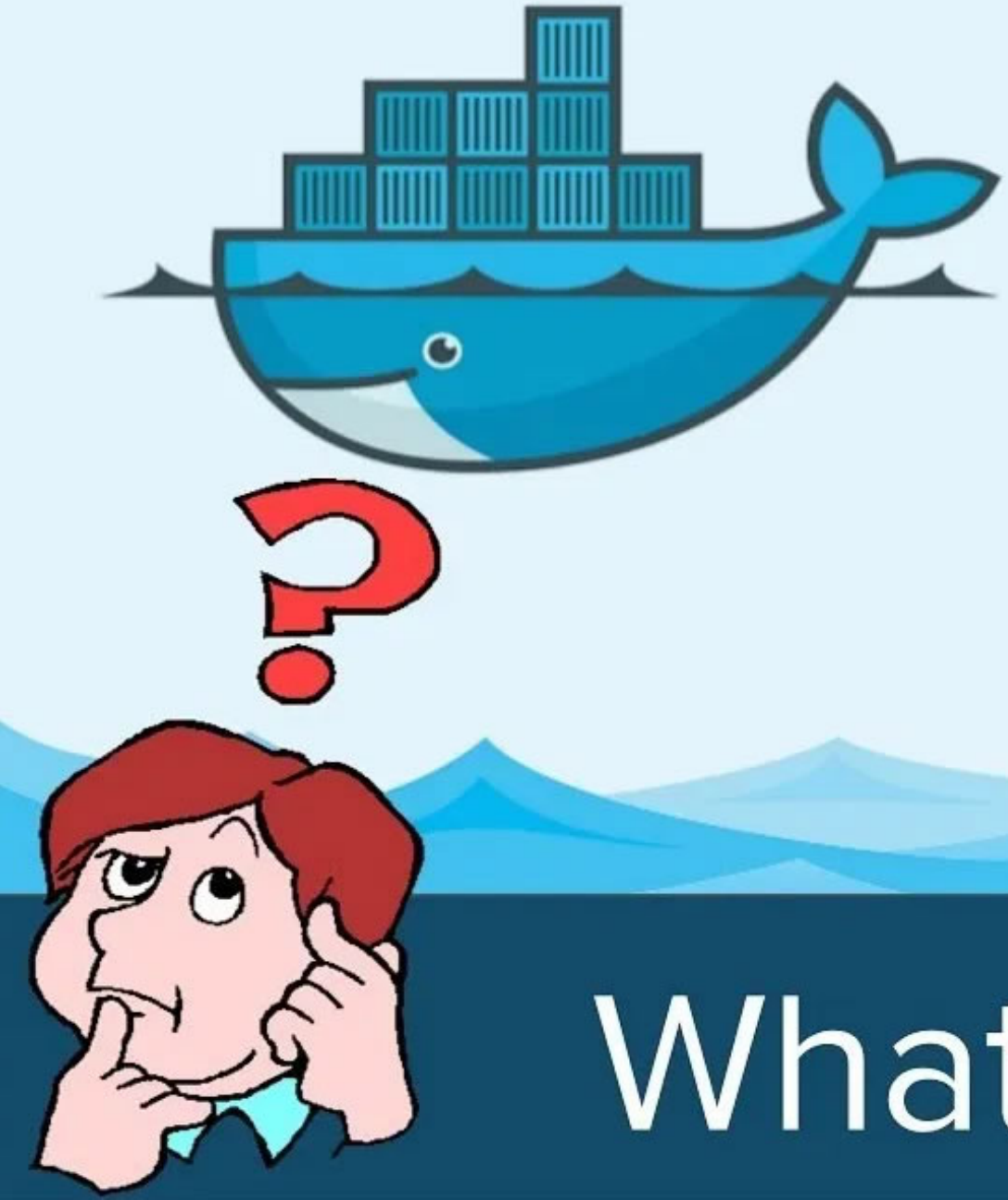


docker

What is DOCKER ?

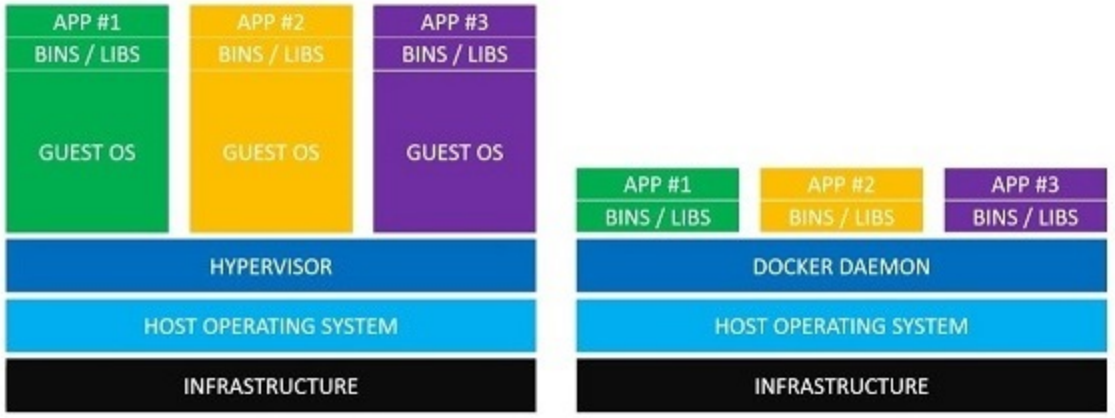
Docker

- Docker is an **Open Platform** for developing, shipping, and running applications.
- Docker is a set of platform as a service (*PaaS*) products that use OS-level virtualization to **deliver software in packages called containers**.
- Containers are **isolated** from one another and bundle their own software, libraries and configuration files. They communicate with each other through well-defined channels.
- Docker enables you to **separate your applications from your infrastructure** so you can deliver software quickly.
- With Docker, you can **manage your infrastructure** in the same ways you manage your **applications**.

Is Docker a virtual machine?

Docker is container based technology and containers are just user space of the operating system.

- In **Docker**, the containers running share the host OS kernel. **Isolates at the software level**
- A **Virtual Machine**, on the other hand, is **not** based on container technology. They are made up of user space plus kernel space of an operating system. **Isolates at the hardware level**



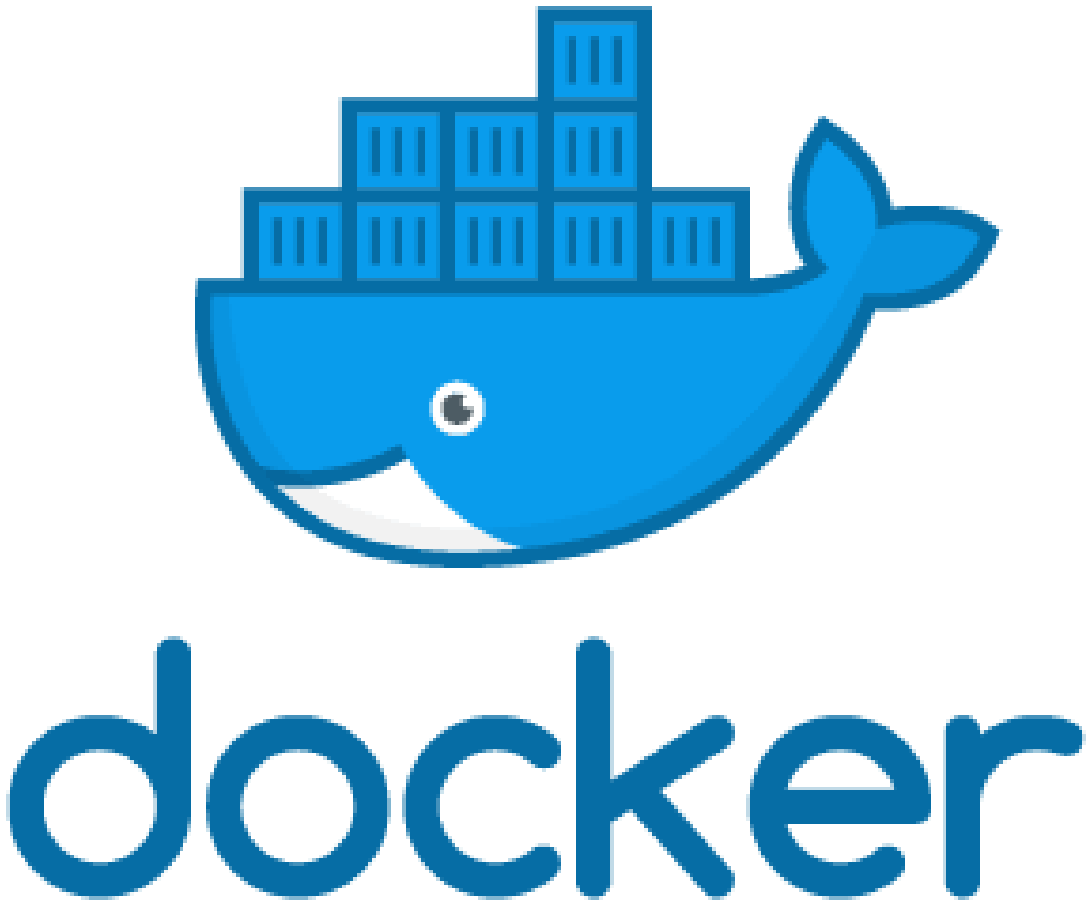
Docker ecosystem

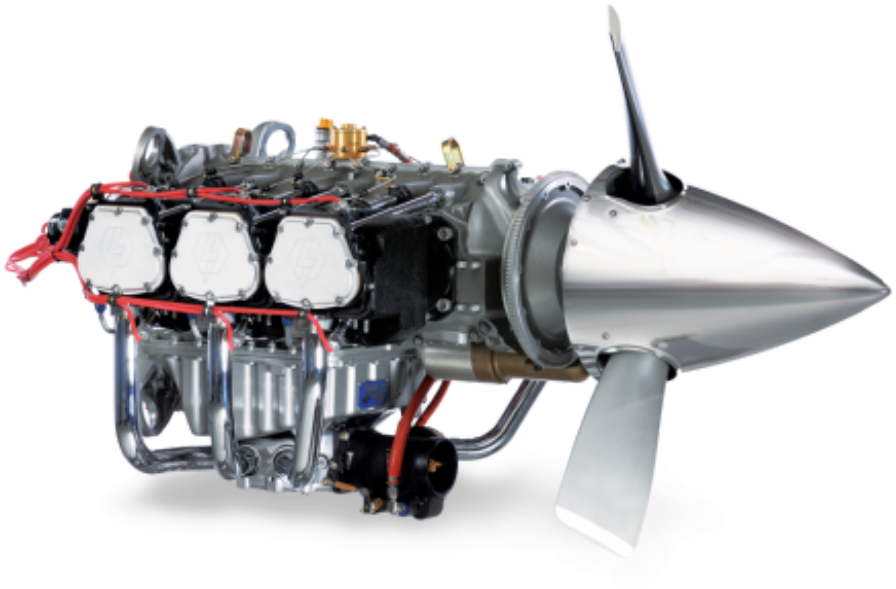
Docker Platform

Docker Platform is Docker's software that provides the ability to package and run an application in a container on any computer platform.

Docker Platform bundles code files and dependencies.

It promotes **easy scaling** by enabling portability and reproducibility.





Docker Engine

Docker Engine is the client-server application. The Docker company divides the Docker Engine into two products:

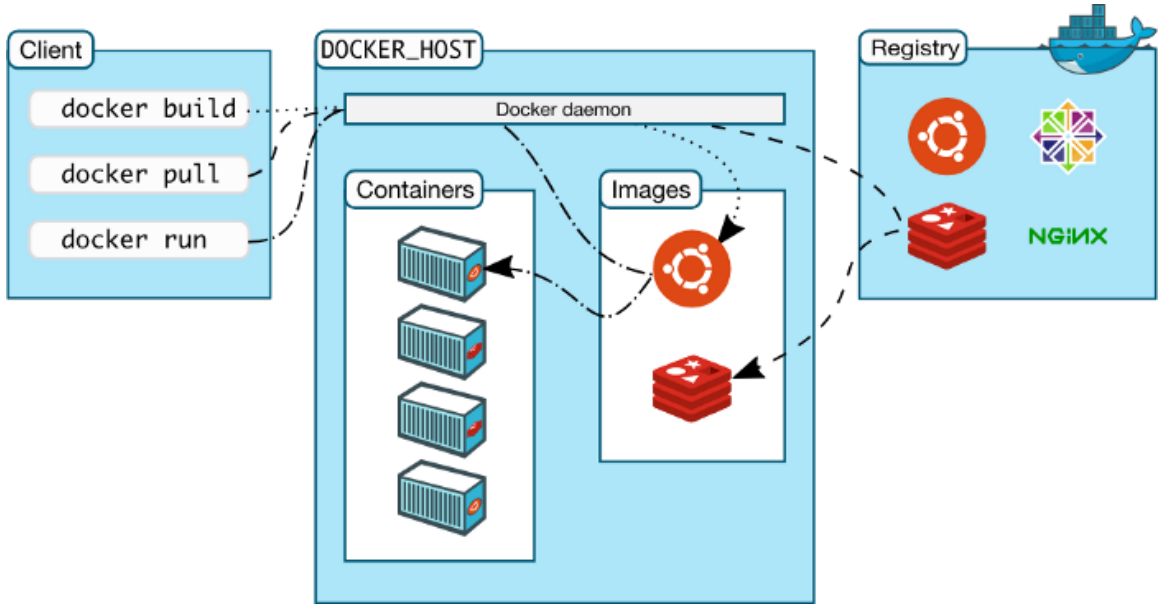
- Docker Community Edition (*CE*) is free and largely based on open source tools.
- Docker Enterprise comes with additional support, management, and security features. *Enterprise is how Docker earns money* - www.docker.com/pricing

Docker Client

Docker Client is the **primary way** you'll interact with Docker.

When you use the Docker Command Line Interface (*CLI*) you type a command into your terminal that starts with docker.

Docker Client then uses the Docker API to send the command to the Docker Daemon.



Docker Docker Desktop

Windows and Mac

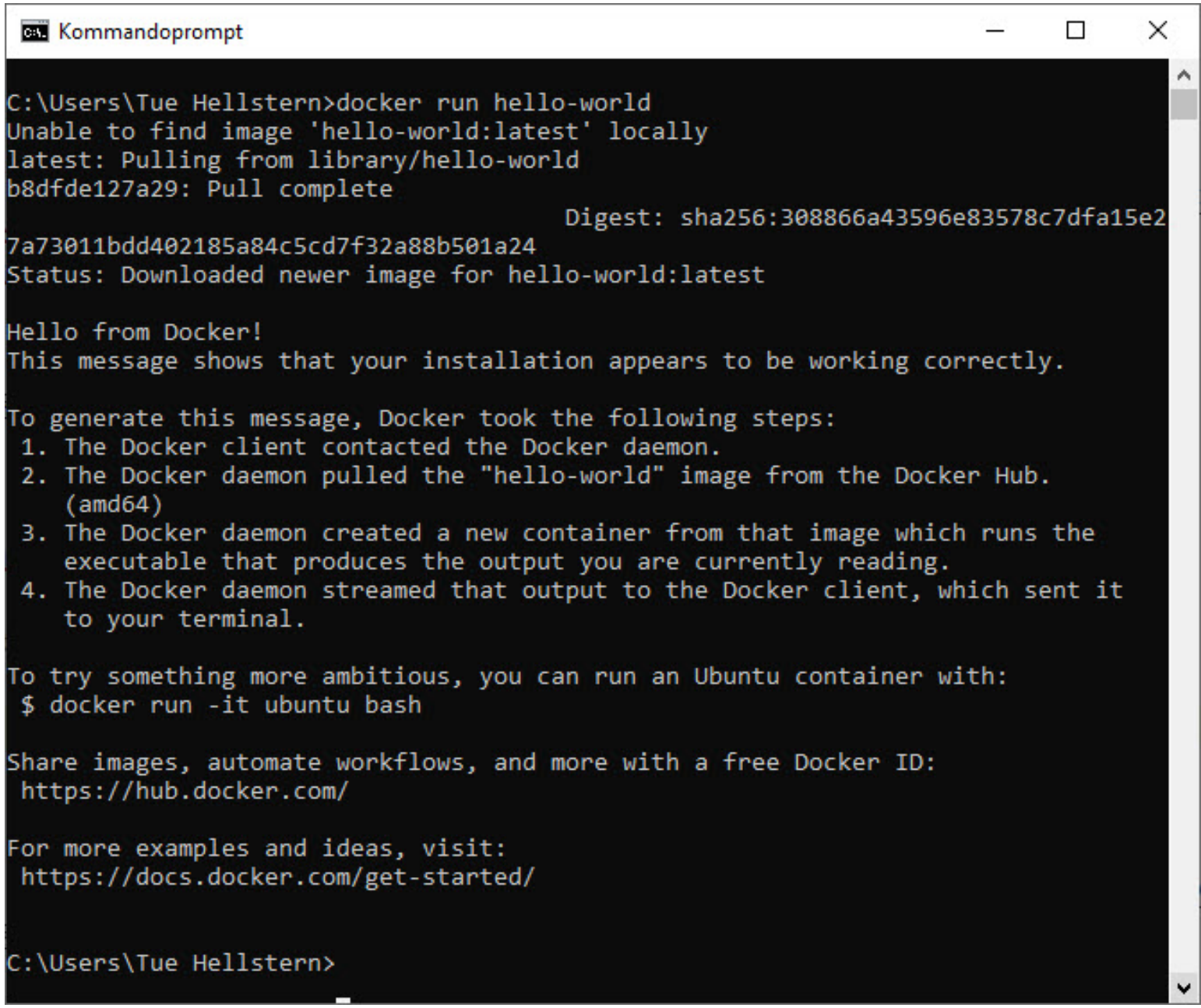
The fastest way to containerize applications on your desktop

https://www.docker.com/products/docker-desktop

Hello World

```
docker run hello-world
```

Example of **minimal Dockerization**



https://hub.docker.com/_/hello-world

Ubuntu Docker

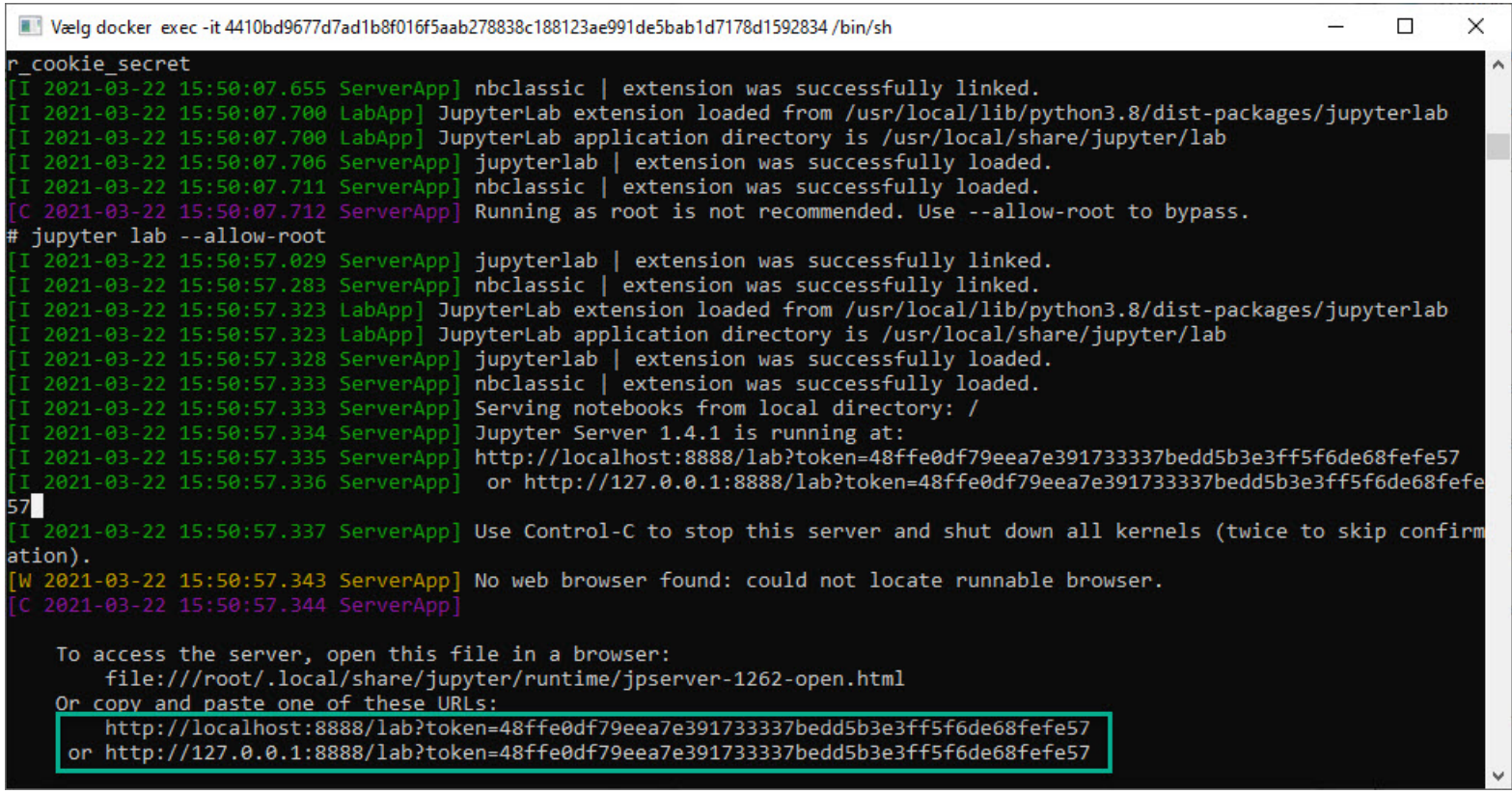
```
docker pull ubuntu
```

Install Jupyter Lab with pip

```
apt update
apt upgrade
apt install python3

python3 --version

apt install python3-pip

pip3 install jupyterlab
```

Run Jupyter Lab

jupyter lab --allow-root



MongoDB

MongoDB document databases provide high availability and easy scalability.

Demo including network access

https://hub.docker.com/_/mongo

https://labs.play-with-docker.com

Docker 101 Tutorial

https://www.docker.com/101-tutorial

docker

Docker Engine - Ubuntu

Docker Engine - Ubuntu (Community) is the best way to install the Docker platform on Ubuntu Linux environments.

Simplify provisioning and setup of Docker and accelerate your time to value in building and deploying container based applications.

Install Docker Engine on Ubuntu - https://docs.docker.com/engine/install/ubuntu/

Install Docker on Ubuntu

Update Software Repositories

It’s a good idea to update the local database of software to make sure you’ve got access to the latest revisions.

```
sudo apt-get update
```

Install Docker

To install Docker on Ubuntu

```
sudo apt-get update
```

Start and Automate Docker

The Docker service needs to be setup to run at startup.

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

Verifying the Installation

To verify that Docker has been successfully installed and that you can execute the docker command.

```
docker container run hello-world
```

Check Docker Version

To verify the installed Docker version number, enter:

```
docker --version
```

Uninstalling Docker

The following commands stops all running containers and remove all docker objects

```
docker container stop $(docker container ls -aq)
docker system prune -a --volumes
```

You can now uninstall Docker as any other package installed with apt.

```
sudo apt purge docker-ce
sudo apt autoremove
```

Docker Commands

Docker commands :: Developing

- **docker create [image]**: Create a new container from a particular image
- **docker login**: Log into the Docker Hub repository.
- **docker pull [image]**: Pull an image from the Docker Hub repository
- **docker push [username/image]**: Push an image to the Docker Hub repository
- **docker search [term]**: Search the Docker Hub repository for a particular term
- **docker tag [source] [target]**: Create a target tag or alias that refers to a source image

Docker commands :: Running

- **docker start [container]**: Start a particular container
- **docker stop [container]**: Stop a particular container
- **docker exec -ti [container] [command]**: Run a shell command inside a particular container
- **docker run -ti — image [image] [container] [command]**: Create and start a container at the same time, and then run a command inside it
- **docker run -ti — rm — image [image] [container] [command]**: Create and start a container at the same time, run a command inside it, and then remove the container after executing the command
- **docker pause [container]**: Pause all processes running within a particular container

Docker commands :: Utilities

- **docker history [image]**: Display the history of a particular image
- **docker images**: List all of the images that are currently stored on the system
- **docker inspect [object]**: Display low-level information about a particular Docker object
- **docker ps**: List all of the containers that are currently running
- **docker version**: Display the version of Docker that is currently installed on the system

Docker commands :: Cleaning

- **docker kill [container]**: Kill a particular container
- **docker kill $(docker ps -q)**: Kill all containers that are currently running
- **docker rm [container]**: Delete a particular container that is not currently running
- **docker rm $(docker ps -a -q)**: Delete all containers that are not currently running

Links

- www.docker.com
- www.docker.com/101-tutorial
- docs.docker.com/get-started
- hub.docker.com/_/mysql
- hub.docker.com/_/mongo
- fastapi.tiangolo.com/deployment/docker/