

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

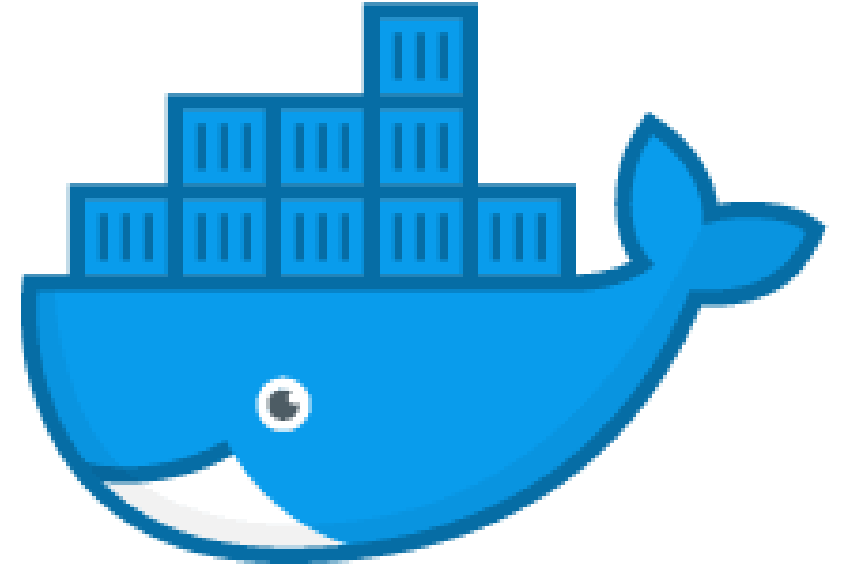


# Docker Platform

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

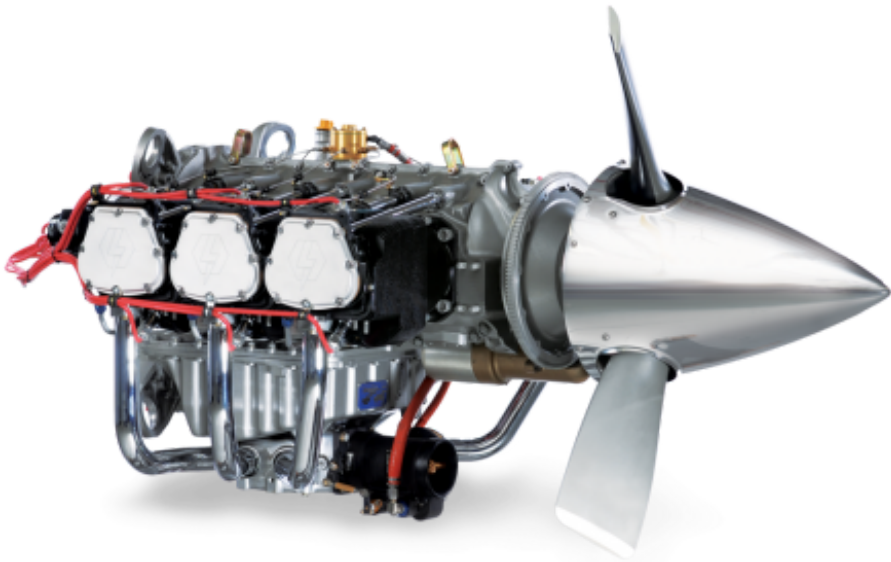
Docker Platform bundles code files and dependencies.

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



docker

# 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 -*

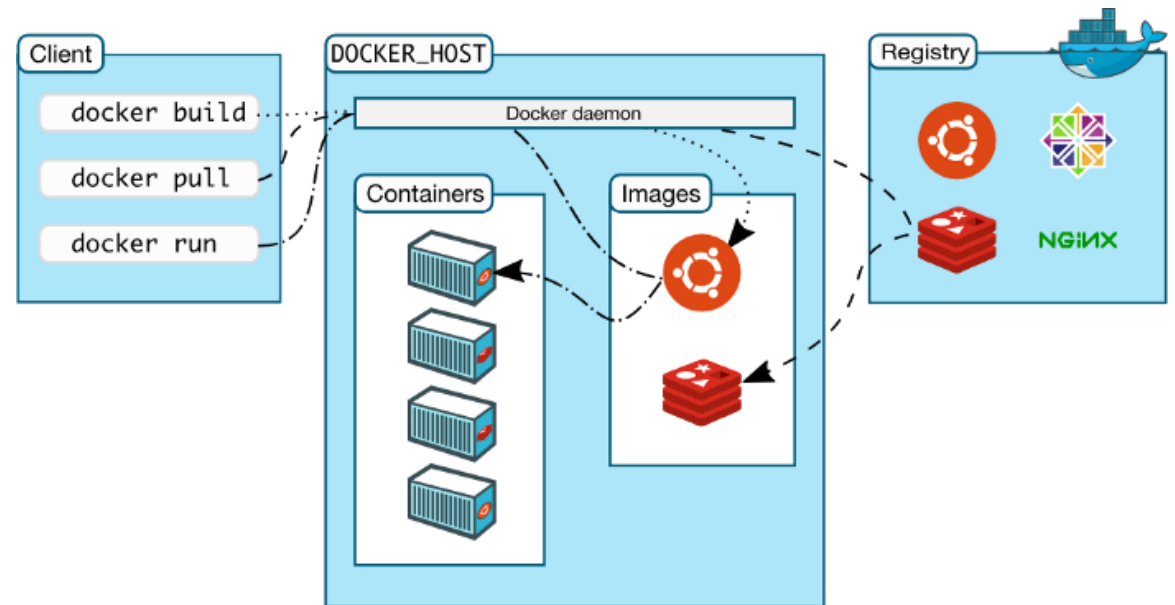
[www.docker.com/pricing](http://www.docker.com/pricing)

# 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/pr  
oducts/docker-desktop](https://www.docker.com/products/docker-desktop)





# Hello World

```
docker run hello-world
```

## Example of minimal Dockerization

[https://hub.docker.com/\\_/hello-world](https://hub.docker.com/_/hello-world)

```
Kommandoprompt
C:\Users\Tue Hellstern>docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
b8dfde127a29: Pull complete
Digest: sha256:308866a43596e83578c7dfa15e27a73011bdd402185a84c5cd7f32a88b501a24
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

C:\Users\Tue Hellstern>
```

# 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`

```
Vælg docker exec -it 4410bd9677d7ad1b8f016f5aab278838c188123ae991de5bab1d7178d1592834 /bin/sh
r_cookie_secret
[I 2021-03-22 15:50:07.655 ServerApp] nbclassic | extension was successfully linked.
[I 2021-03-22 15:50:07.700 LabApp] JupyterLab extension loaded from /usr/local/lib/python3.8/dist-packages/jupyterlab
[I 2021-03-22 15:50:07.700 LabApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[I 2021-03-22 15:50:07.706 ServerApp] jupyterlab | extension was successfully loaded.
[I 2021-03-22 15:50:07.711 ServerApp] nbclassic | extension was successfully loaded.
[C 2021-03-22 15:50:07.712 ServerApp] Running as root is not recommended. Use --allow-root to bypass.
# jupyter lab --allow-root
[I 2021-03-22 15:50:57.029 ServerApp] jupyterlab | extension was successfully linked.
[I 2021-03-22 15:50:57.283 ServerApp] nbclassic | extension was successfully linked.
[I 2021-03-22 15:50:57.323 LabApp] JupyterLab extension loaded from /usr/local/lib/python3.8/dist-packages/jupyterlab
[I 2021-03-22 15:50:57.323 LabApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[I 2021-03-22 15:50:57.328 ServerApp] jupyterlab | extension was successfully loaded.
[I 2021-03-22 15:50:57.333 ServerApp] nbclassic | extension was successfully loaded.
[I 2021-03-22 15:50:57.333 ServerApp] Serving notebooks from local directory: /
[I 2021-03-22 15:50:57.334 ServerApp] Jupyter Server 1.4.1 is running at:
[I 2021-03-22 15:50:57.335 ServerApp] http://localhost:8888/lab?token=48ffe0df79eea7e391733337bedd5b3e3ff5f6de68fe57
[I 2021-03-22 15:50:57.336 ServerApp] or http://127.0.0.1:8888/lab?token=48ffe0df79eea7e391733337bedd5b3e3ff5f6de68fe57
[I 2021-03-22 15:50:57.337 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 2021-03-22 15:50:57.343 ServerApp] No web browser found: could not locate runnable browser.
[C 2021-03-22 15:50:57.344 ServerApp]

To access the server, open this file in a browser:
file:///root/.local/share/jupyter/runtime/jpserver-1262-open.html
Or copy and paste one of these URLs:
http://localhost:8888/lab?token=48ffe0df79eea7e391733337bedd5b3e3ff5f6de68fe57
or http://127.0.0.1:8888/lab?token=48ffe0df79eea7e391733337bedd5b3e3ff5f6de68fe57
```

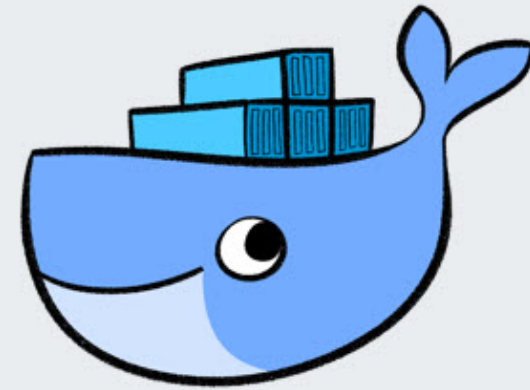
# MongoDB

MongoDB document databases provide high availability and easy scalability.

Demo including network access



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



# Play with Docker

A simple, interactive and fun playground to learn Docker

Login ▼

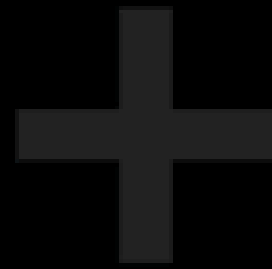
# Docker 101 Tutorial

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

# Docker 101 Tutorial

#LearnDocker





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
```

```
root@ubuntu:~# docker commit aba5aac12d7bca3f9c7f0267d80c7033ca4b1174d1d4ba49c8f4d503cc623364 nginx-  
template
```

```
sha256:8b42a0ec75eb79f1f3509883b87bc29a8a675b453ec8a3c1a45fd73c042083ef
```

```
root@ubuntu:~# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nginx-template	latest	8b42a0ec75eb	About a minute ago	390.1 MB
nginx	latest	5766334bdaa0	2 weeks ago	182.5 MB



# 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](http://www.docker.com)
- [www.docker.com/101-tutorial](http://www.docker.com/101-tutorial)
- [docs.docker.com/get-started](http://docs.docker.com/get-started)
- [hub.docker.com/\\_/mysql](http://hub.docker.com/_/mysql)
- [hub.docker.com/\\_/mongo](http://hub.docker.com/_/mongo)
- [fastapi.tiangolo.com/deployment/docker/](http://fastapi.tiangolo.com/deployment/docker/)