<image/nation>

Tech Training Series

Exclusive for

HKU BUSINESS SCHOOL 港大經管學院

Computing Virtualization

2023/2024

Using Crossed Operating System for Data Science Projects

by

Sunny NG

<image/nation>

In this workshop (3 hours)

- What is virtualization?
  - Running Windows, Mac and Linux on a single computer at the same time
  - You are using Mac? But target software only run on Windows? No problem.
- Getting Familiar with Hypervisor
- Hypervisor Selections
- OS Installer Images
- Introduction to Container
- Practical Works

Sunny Ng

- Founder / Master Trainer Image Nation
- Developer Web, Mobile, WeChat & IoT
- Content Creator Video producing / Live streaming
- AWS Solution Architect – Associate
- Alibaba Cloud Professional
- AWS Academy Educator
- Email: sunny.ng@imagenation.com.hk

github.com/ngsanluk

Virtualization

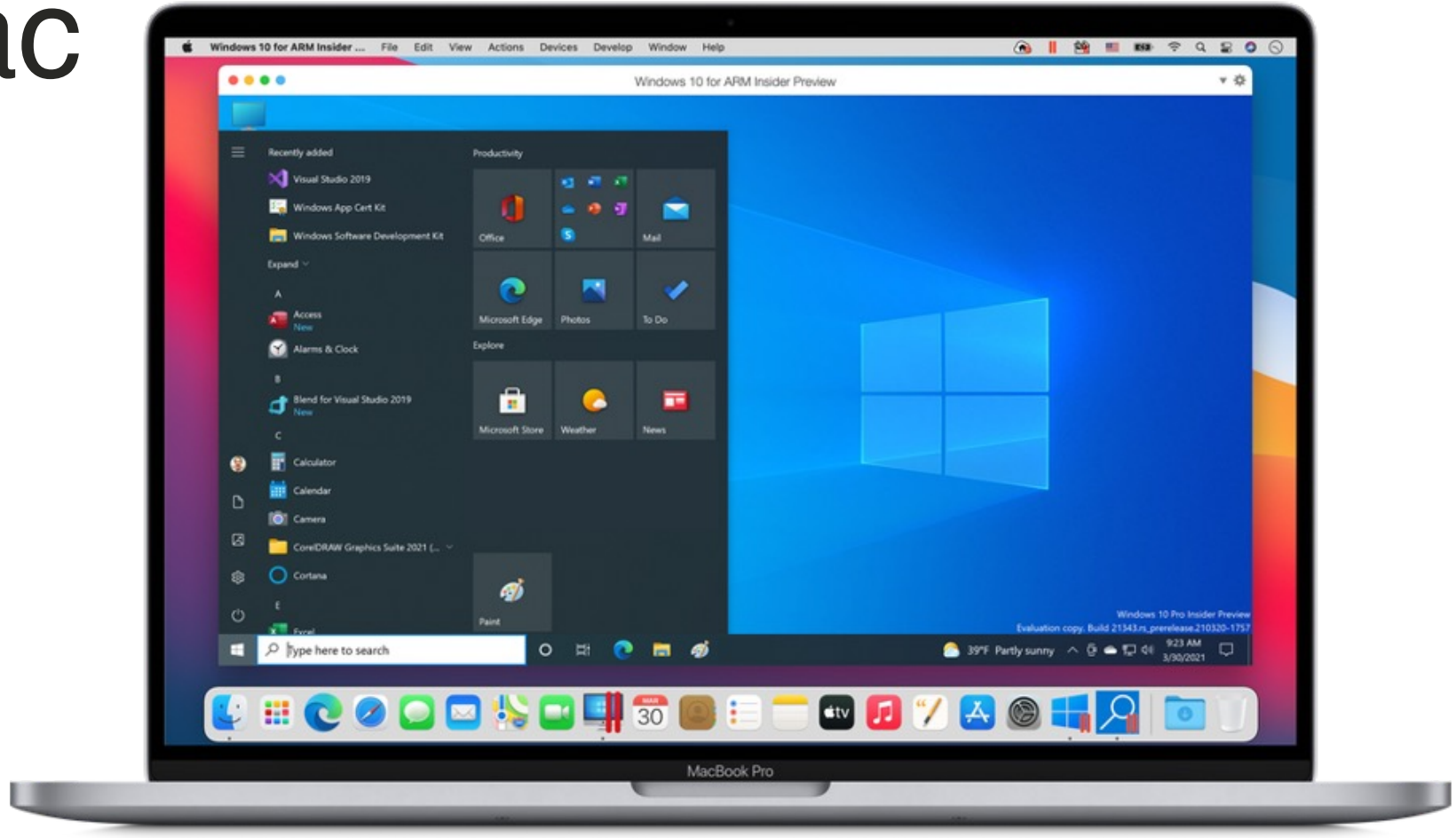
- Virtualized Computation Power
- Can easily confuse with VR (Virtual Reality)
- In computing, virtualization is the act of creating a **virtual version of computation resources**, including
  - virtual computer hardware platforms
  - storage devices
  - computer network resources
- When something is virtual, it means it's NOT real.

Virtual Machine

- Virtualized Computer
- Hardware virtualization refers to the creation of a virtual machine that **acts like a real computer** with an operating system.

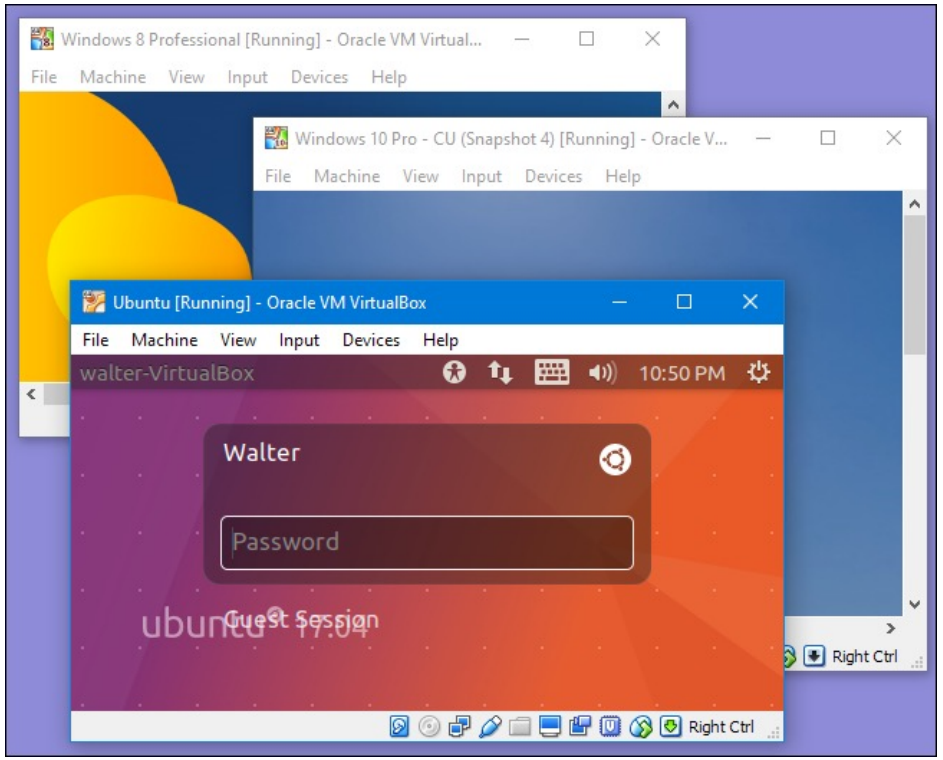
<image/nation>

You can run Windows on top of a Mac



You can run multiple Linux on top of Windows

- A computer that is running Microsoft Windows may host **many** virtual machines
  - such as Ubuntu Linux operating system



<image/nation>

Use cases for data scientist

- Certain software only run on particular operating system
- Example
  - You are using Mac and need to use PowerBI Desktop

Cloud services are
virtualized computation power

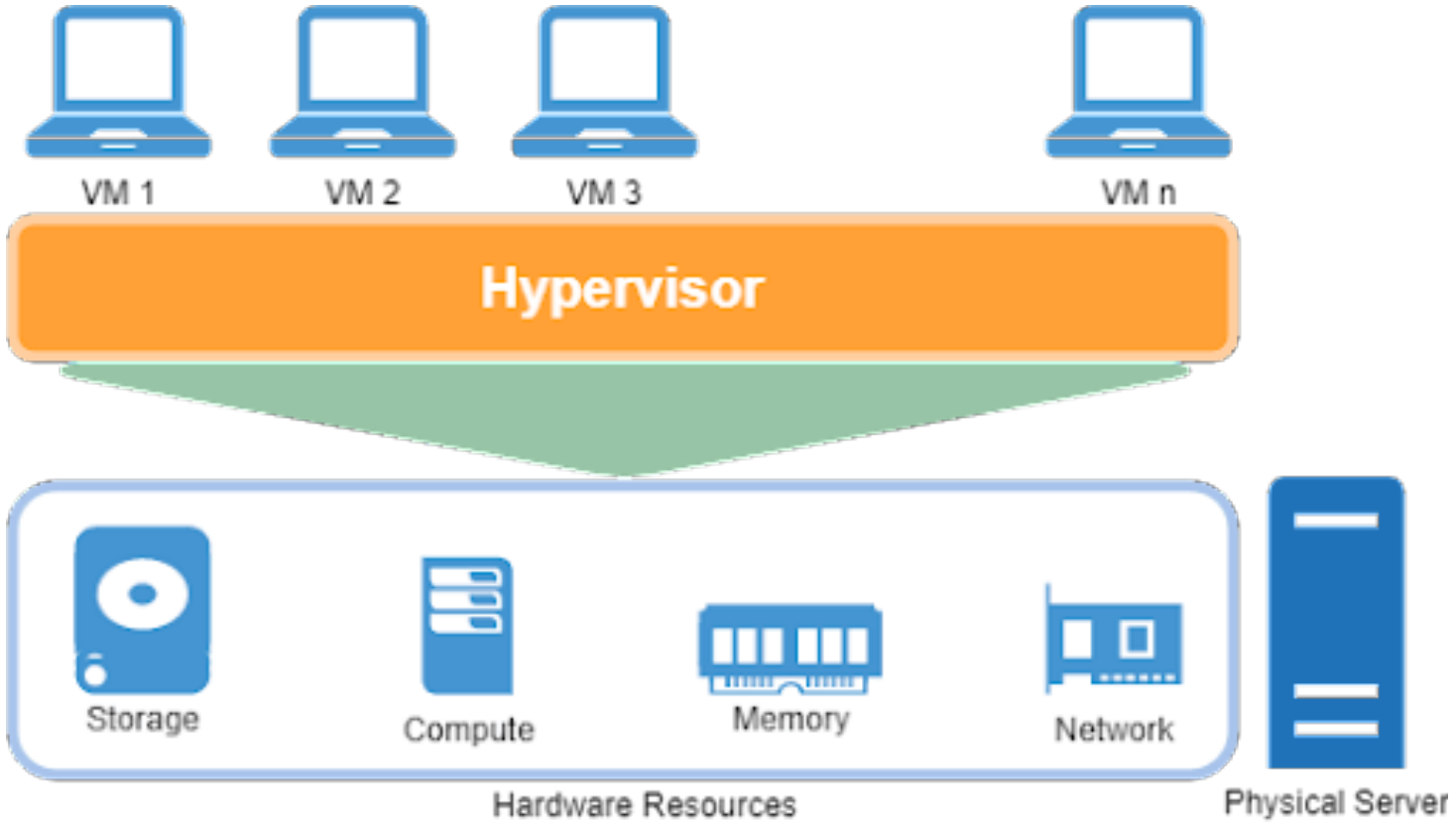
<image/nation>

Flexible Cloud Computing built on top of virtualization

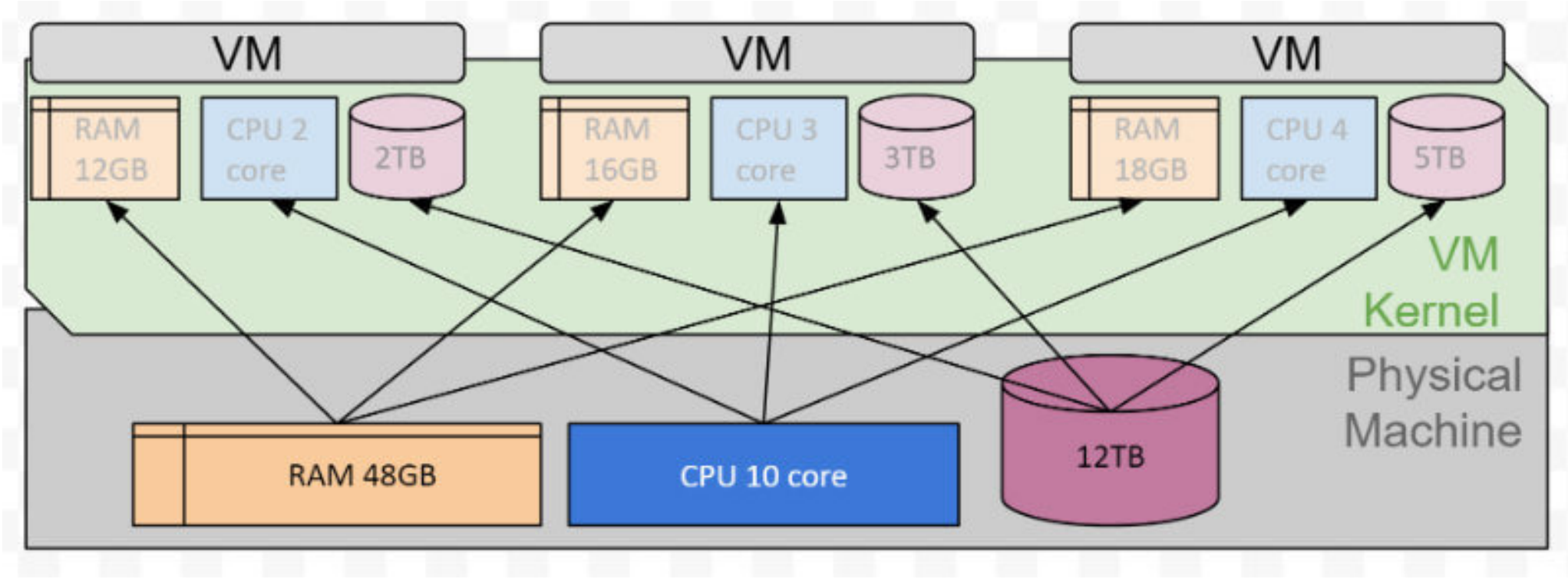
- Imagine a very powerful computer that comes with excellent hardware configuration (CPU, RAM, HD and etc.)
- Many virtual machines are running on top this very powerful hardware
- Each virtual machine is rented to different client/tenant.
- Each virtual machine is configured based on client's needs

<image/nation>

One Hardware Runs Many OS



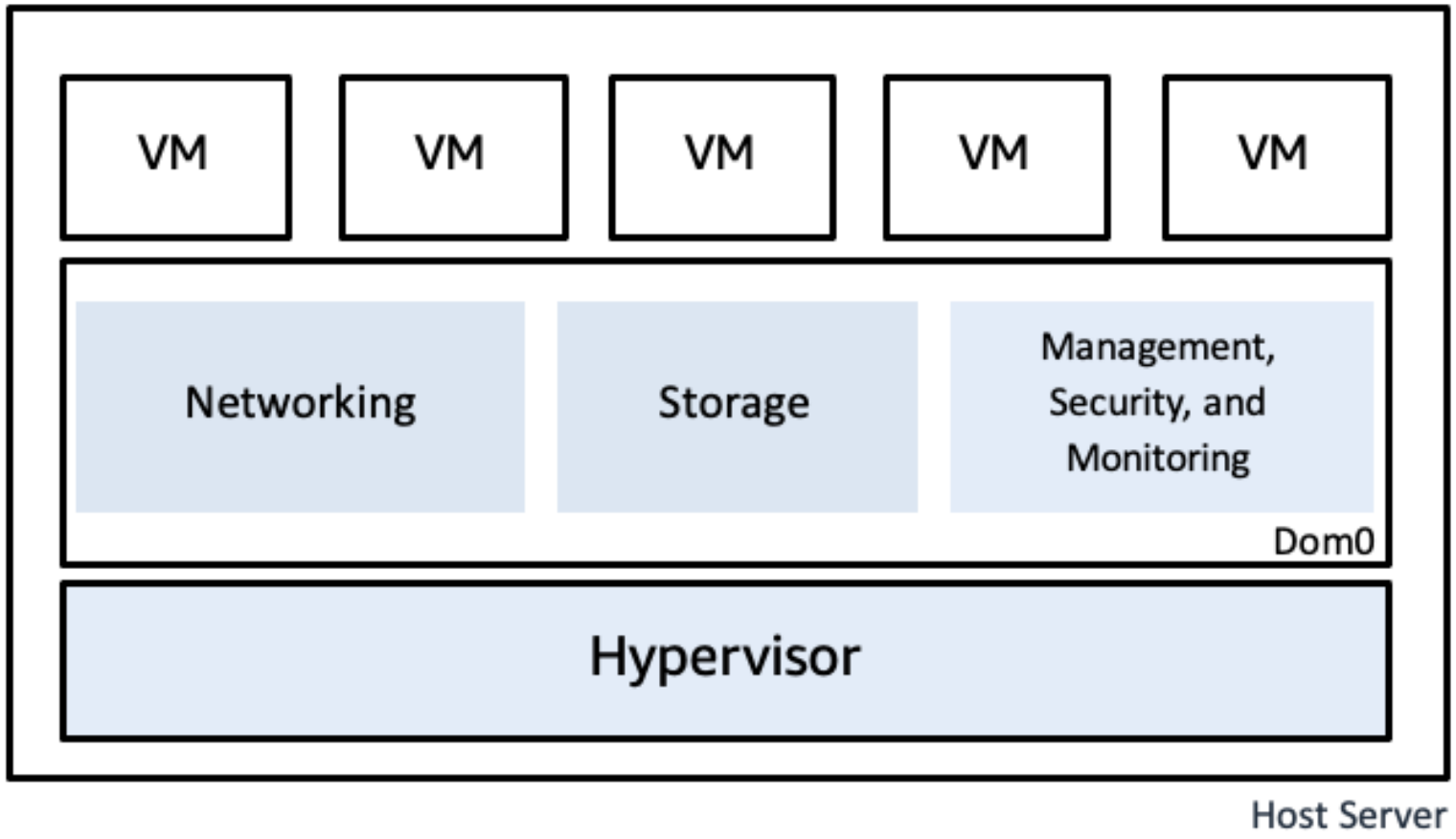
What is happening underneath?



<image/nation>

AWS Cloud Server

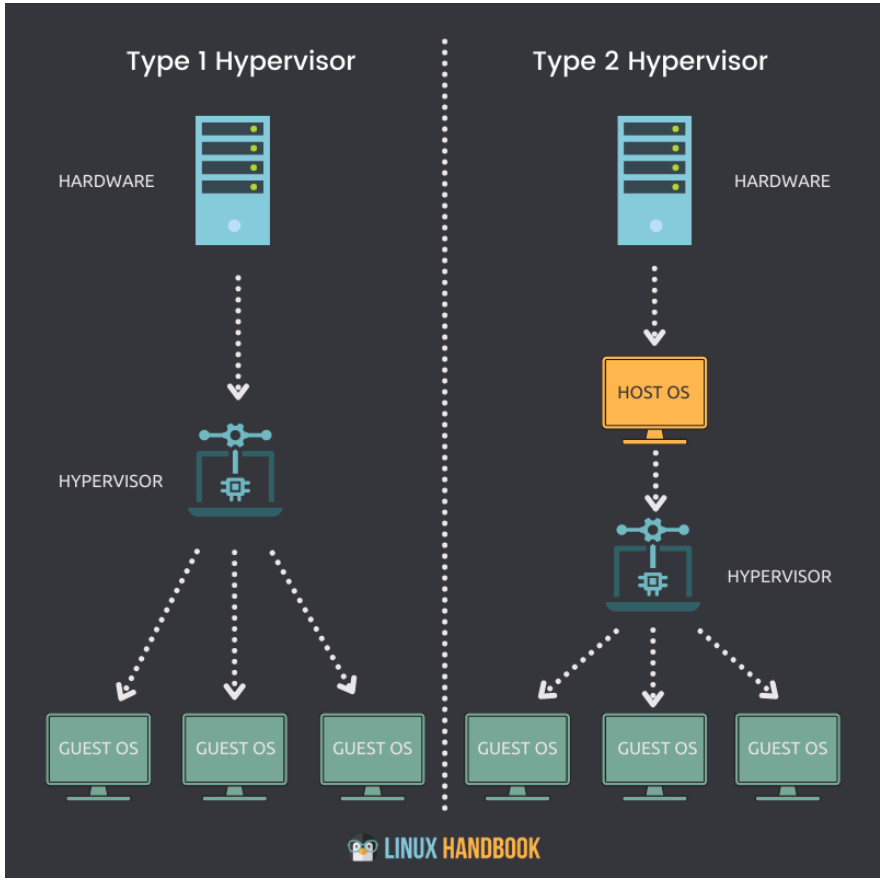
In cloud, VM server is widely referred as Instance



<image/nation>

Popular Hypervisor

- A hypervisor is a computer software, or hardware that creates and runs virtual machines.
- It's also known as Virtual Machine Monitor (VMM)
- Famous choices
  - Parallels Desktop
  - VMWare
  - VirtualBox



Parallels Desktop

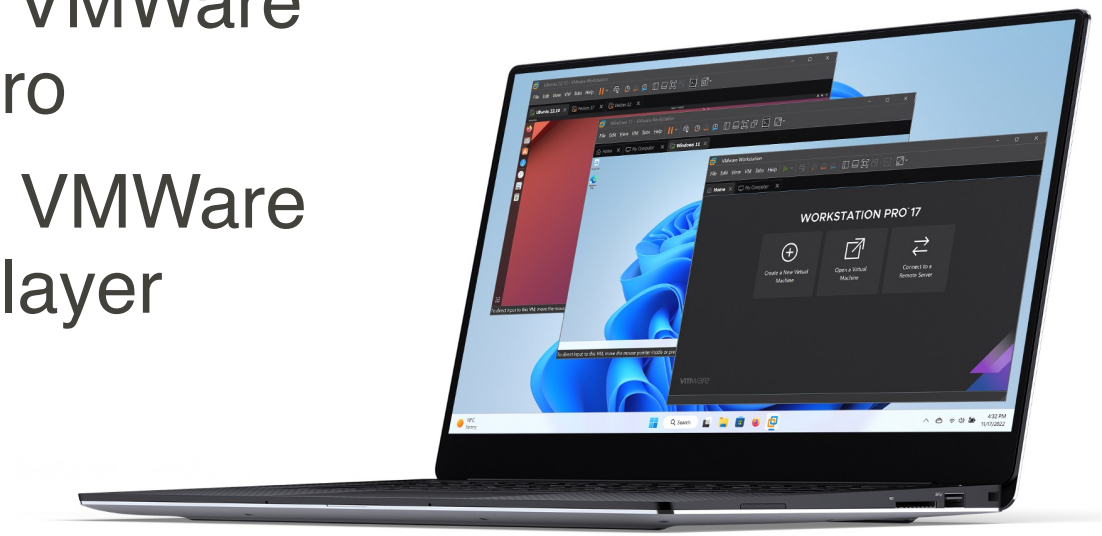
- For **Mac** only
- Supports both Apple M-series CPU and Intel CPU
- Paid by one-off purchase or paid by yearly subscription
  - One month FREE trial
- Ready to download OS images
- Super easy to install Windows 11 to run on top of Mac

If you are using Mac with M-Series (M1/M2, aka Apple Silicon)

Parallels Desktop is the best choice

VMWare Workstation

- For Windows
- Latest version - 17
- Paid version - VMWare Workstation Pro
- Free version - VMWare Workstation Player



VMWARE WORKSTATION PRO™ 17

VMWARE WORKSTATION PLAYER™ 17

VMWare Fusion

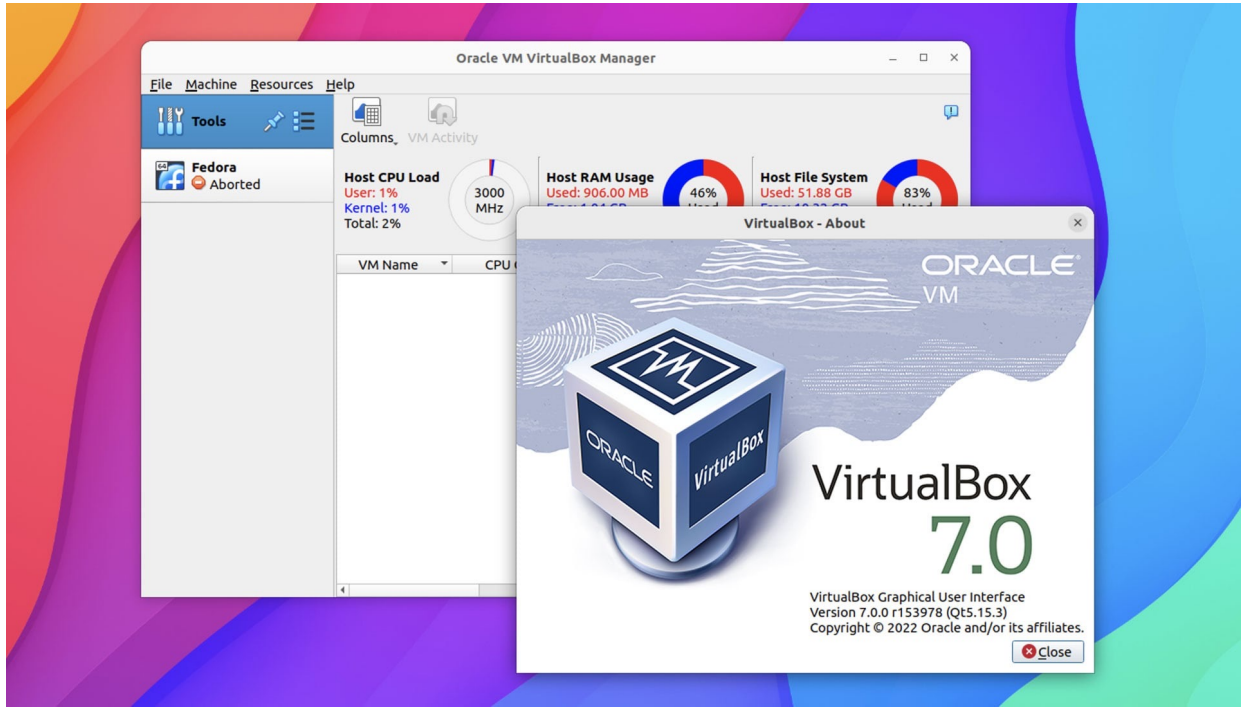
VMWARE FUSION® 13 PRO

VMWARE FUSION® 13 PLAYER

- For Mac
- Latest version - 13
- Paid version - VMWare Fusion Pro
- Free version - VMWare Fusion Player

VirtualBox

- For Windows and Mac (Intel CPU only)
- Latest version - 7
- Free
- Ful virtualization features
- Harder to use



OS disk images

- You need to download **O**perating **S**ystem disk image file
- A disk image file is special computer file that contains operating system folders, sub-folders and file for installation
- Before we have disk image file, we had to burn CD/DVD for OS installation
- Images can be downloaded for free and used to install **Guest Operating System** on your host computer

Popular OS Images

- **Windows 11** (Licensing required. Can install for trial.)
- **Ubuntu Linux** (Free. Popular for testing, dev and production)
- **Kali Linux** (Free. Popular for ethical hacking and penetration testing)

Different images for different CPU

Intel CPU vs. ARM CPU (e.g. Apple M-series)

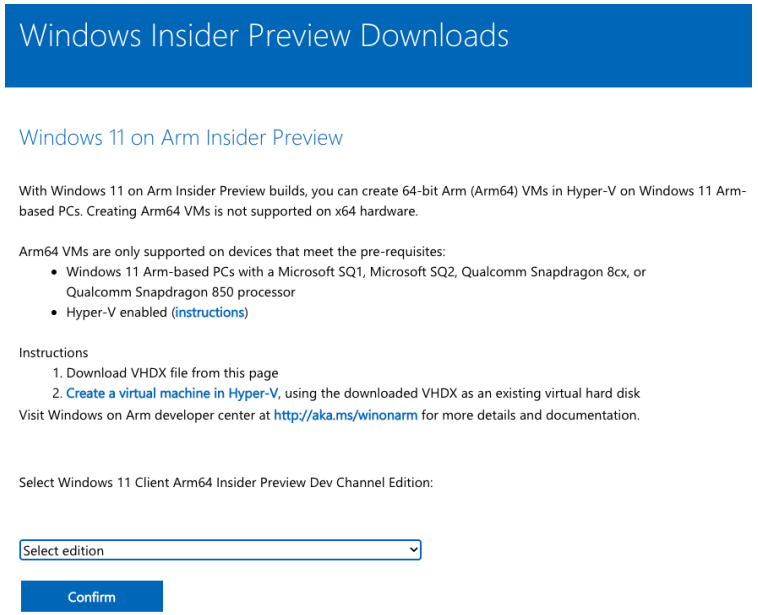
Make sure you download the right images

Windows image download

- Windows 11 for Intel CPU
  - https://www.microsoft.com/en-in/software-download/windows11

Windows image download

- Windows 11 for ARM CPU (e.g. Apple M-Series)
  - Become Windows Insider (free)
  - Go to Windows Insider Preview Downloads
  - https://www.microsoft.com/en-us/software-download/windowsinsiderpreviewARM64
  - A bit complicated. Not for beginner.
  - Easier approach
- Use Parallels Desktop to download windows images

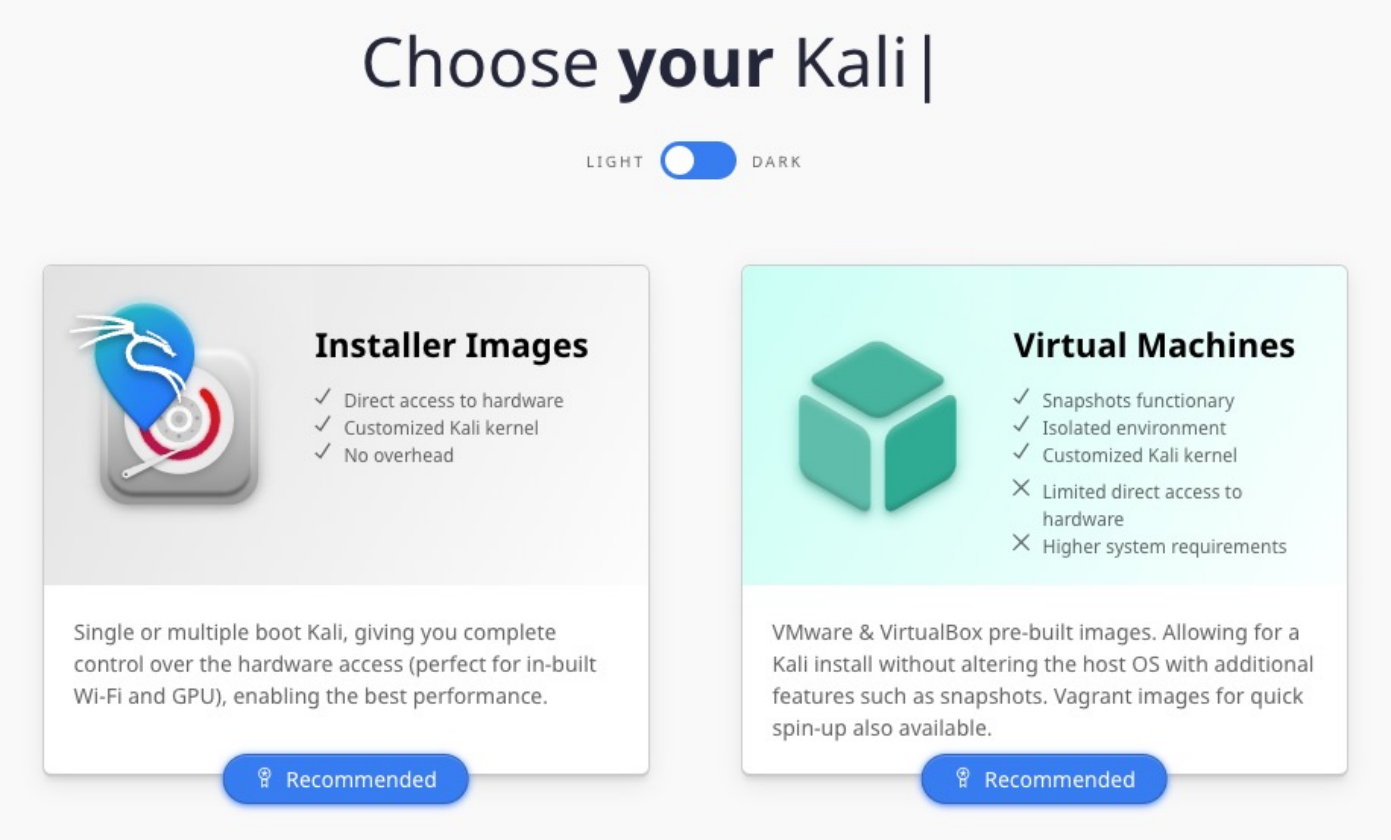


Ubuntu image download

- **Desktop** version vs. **Server** version
  - Desktop version is graphical UI based. For personal use.
  - Server version is command based. For production server use.
- https://ubuntu.com/download/desktop

Kali image download

https://www.kali.org/get-kali/#kali-platforms



What is container?

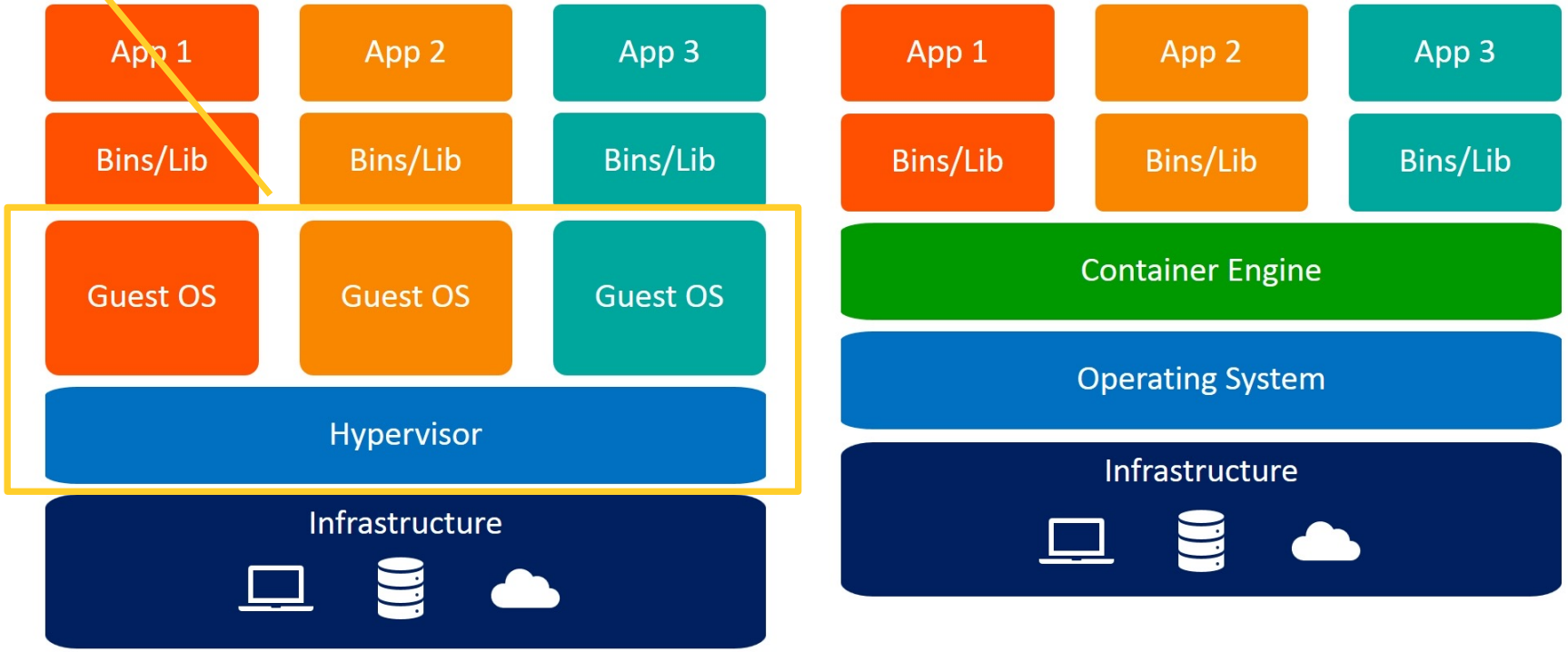
Containerization is a software deployment process that bundles an application's code with all the files and libraries it needs to run on any infrastructure.

- Too difficult to understand? No problem. Let's talk about its benefit and you will love it.
- Just remember, the ultimate goal of these virtualization technologies is to **get software to run effortlessly**.

<image/nation>

Container is light-weight

No hypervisors and guest OS are needed



Virtual Machines

Containers

<image/nation>

With container, software **dependency** is not a pain

These sound familiar to you?

- You download a software. When you try to run it, it says you also need to download/install another software before it can work (one software **depends** on another)
- You download some Python codes for your projects, when you try to run the codes, it says some libraries are missing (your python codes **depend** on these libraries.

Software dependency is very challenging !!!

Container - Use cases for data scientist

- Easily setting up data science development environment just by a few command (copy and paste)
- You can easily have different versions of software environment (server or libraries) running at the same computer at the same time

Most used container

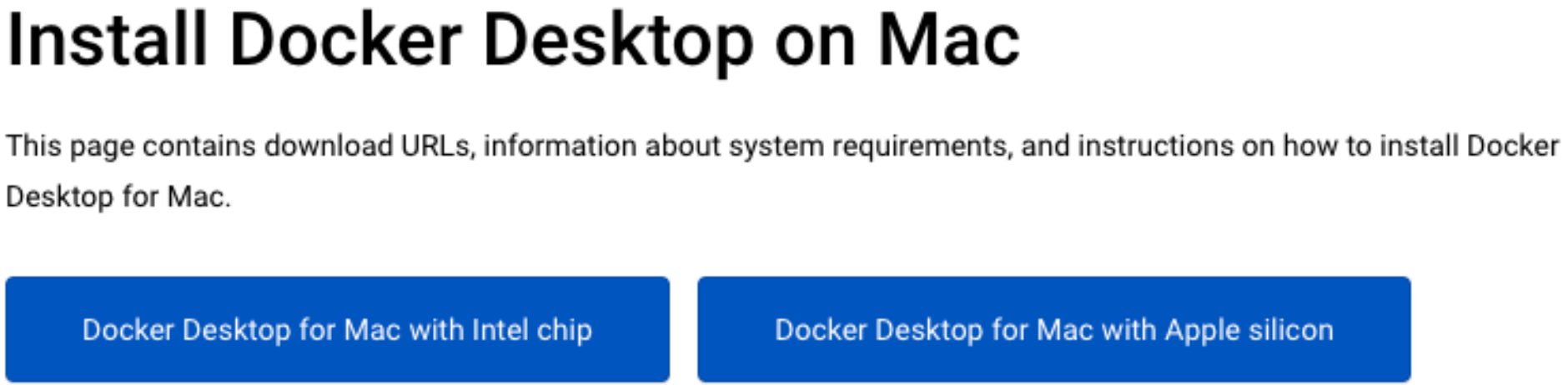
Installing docker for Windows

- Docker Desktop for Windows
- https://docs.docker.com/desktop/install/windows-install/

<image/nation>

Installing docker for Mac

- Docker Desktop on Mac
- https://docs.docker.com/desktop/install/mac-install/

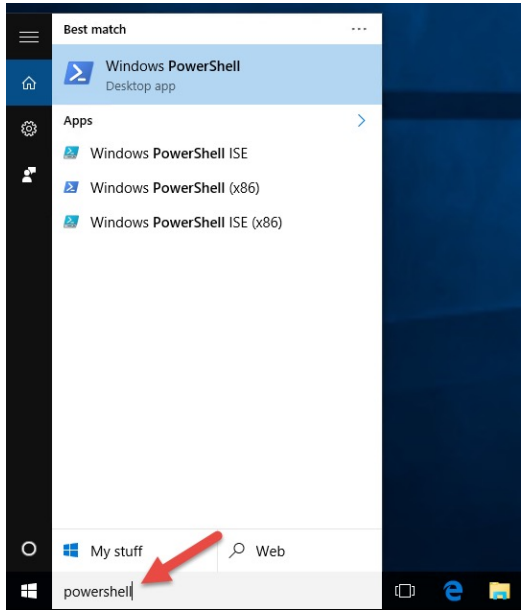


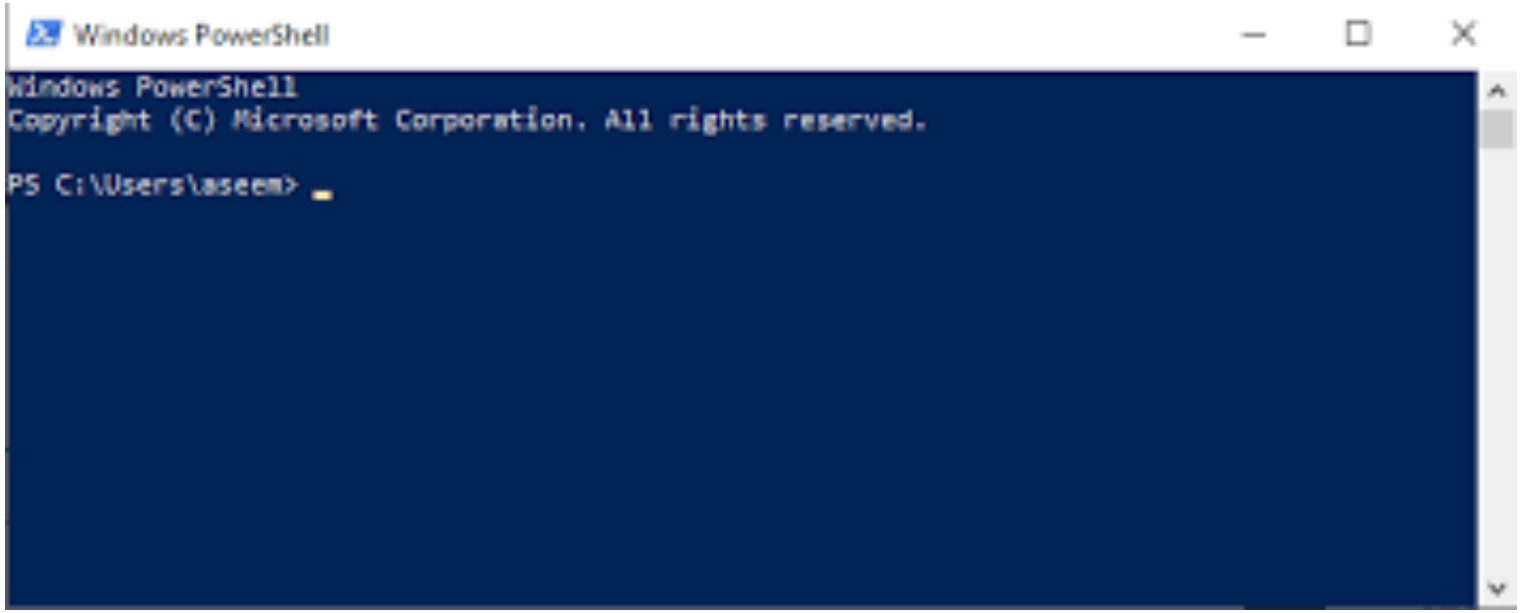
It requires a little bit command line typing

Command Line Interface for Windows

PowerShell

Click Start and search “PowerShell”

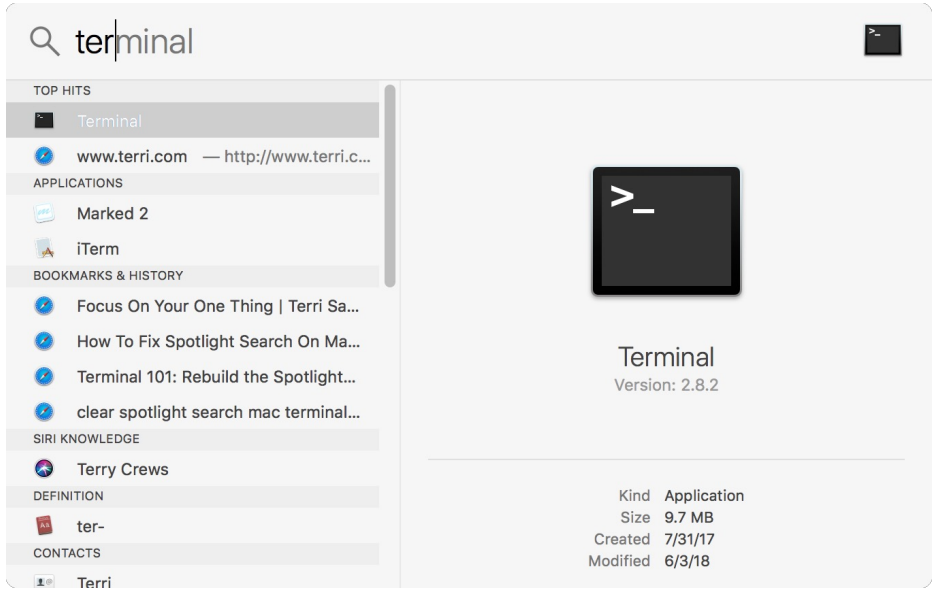


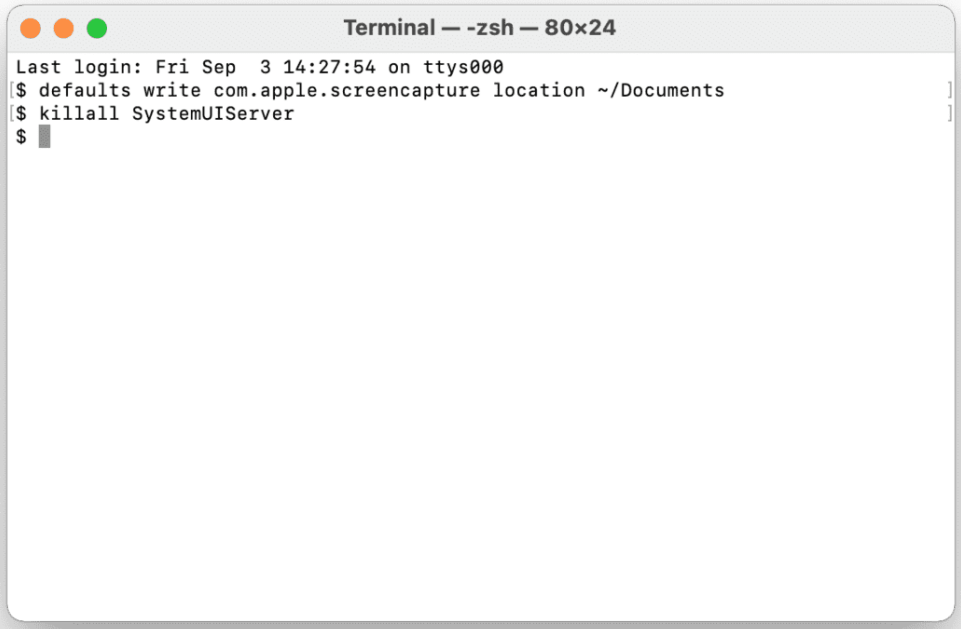


Command Line Interface for Mac

Terminal

Click Spotlight and search “Terminal”





Check your docker installation

Make sure you have started **Docker Desktop**, in the command line interface, type

```
docker -v
```

If the installation is correct, you should see the installed docker version. e.g.

```
Docker version 20.10.20, build 9fdeb9c
```

Some Docker Commands

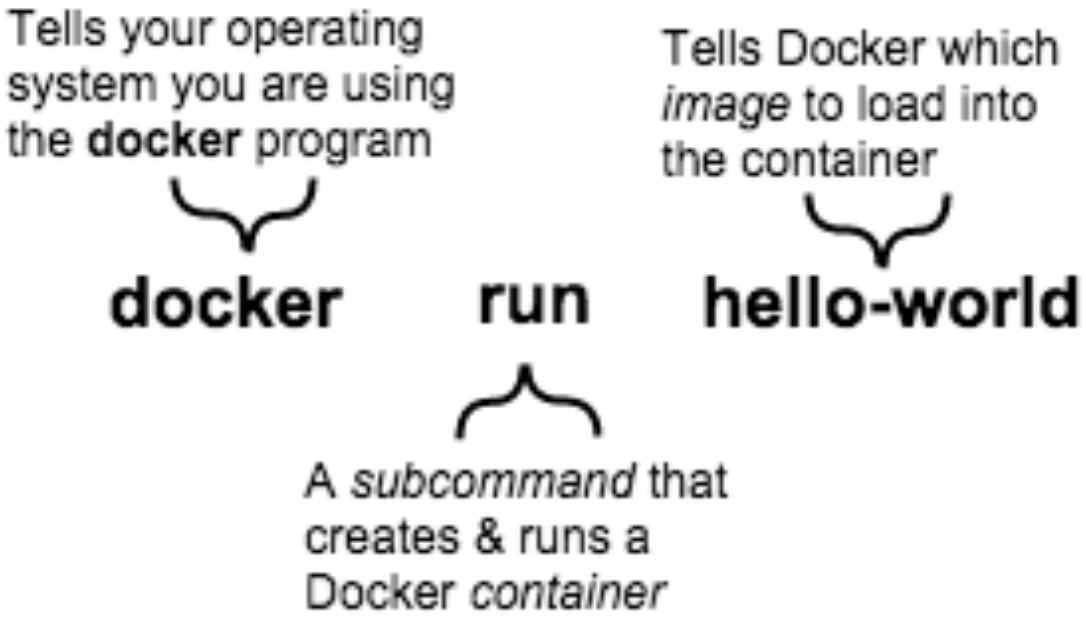
Command/Sub-command	Usage/Examples
`docker images`	Shows docker images available on your computer. Docker image can consume a lot of storage.
`docker ps`	Shows containers that you‘ve run or are running
`docker run`	To run a container (Remember: a container is like a virtual computer. Example: `docker run hello-world`
`docker stop`	To stop a container. Example: `docker stop hello-world`

Running a container

This example uses an existing container **images** named “hello-world” to run a container

The ”hello-world” images is automatically downloaded form docker repository to produce a replica and run

There are many docker images on the internet (repository)



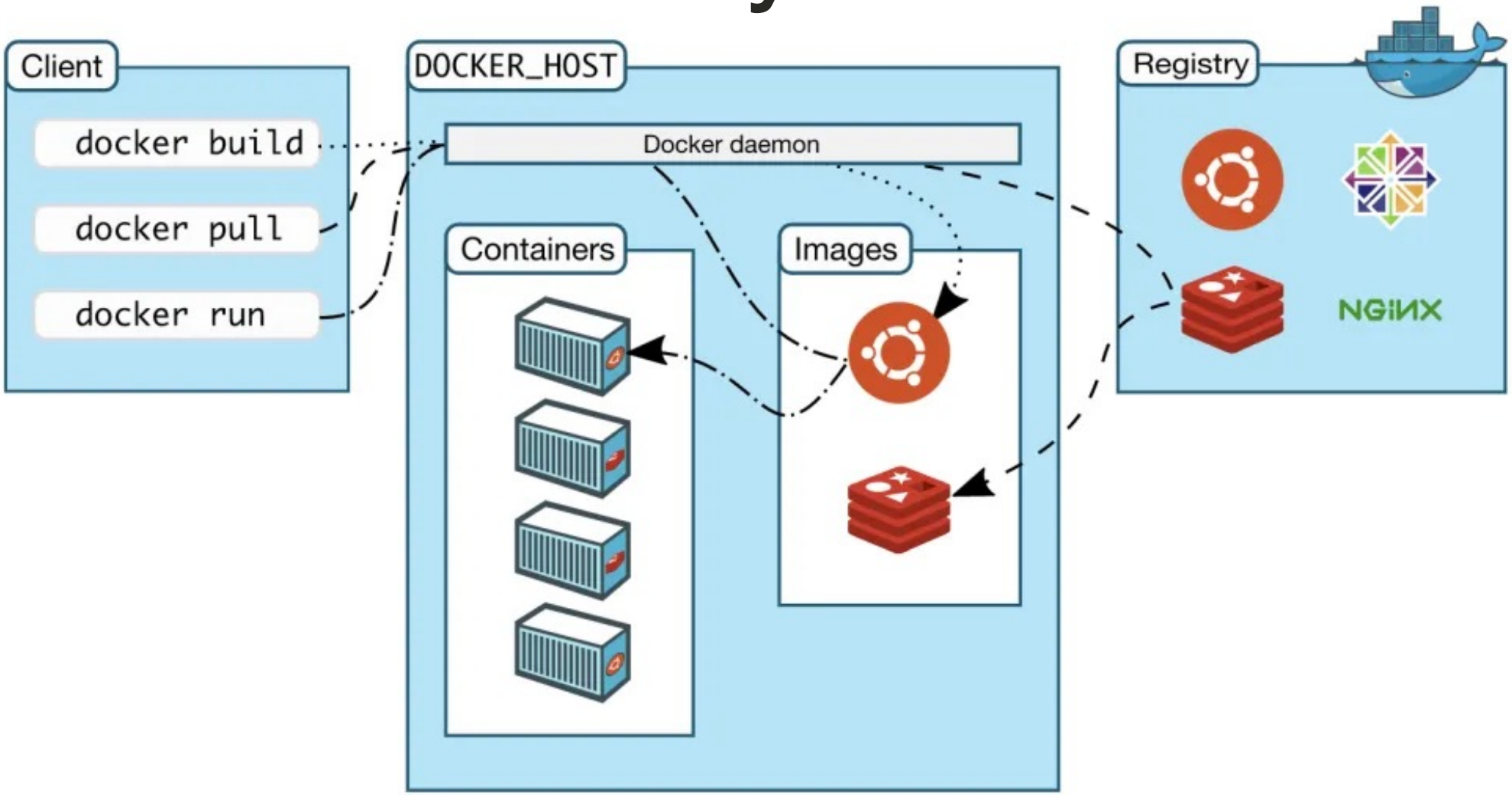
Type in

docker run hello-world

And you should see the following output

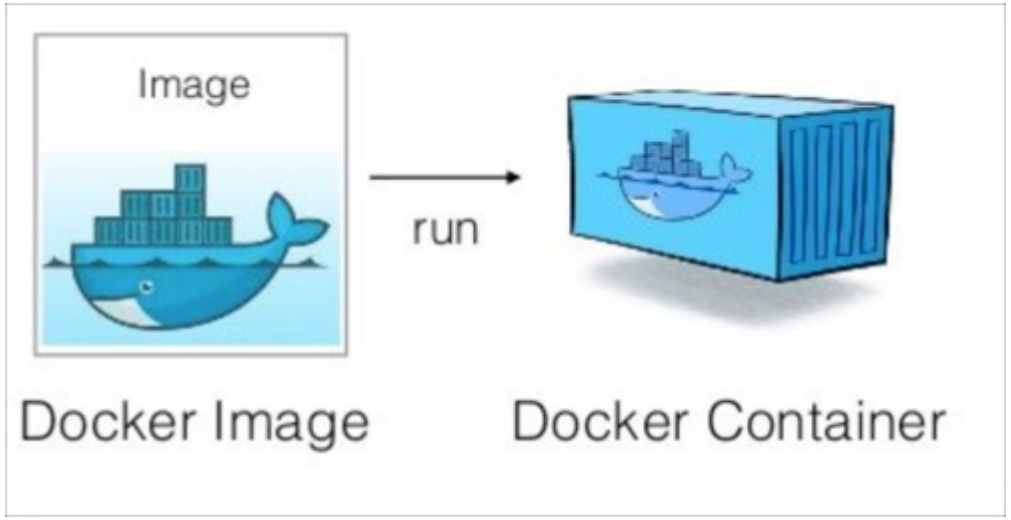
```
Hello from Docker!
This message shows that your installation appears to be working correctly.

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.
    (arm64v8)
 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/
```

Docker Eco-system



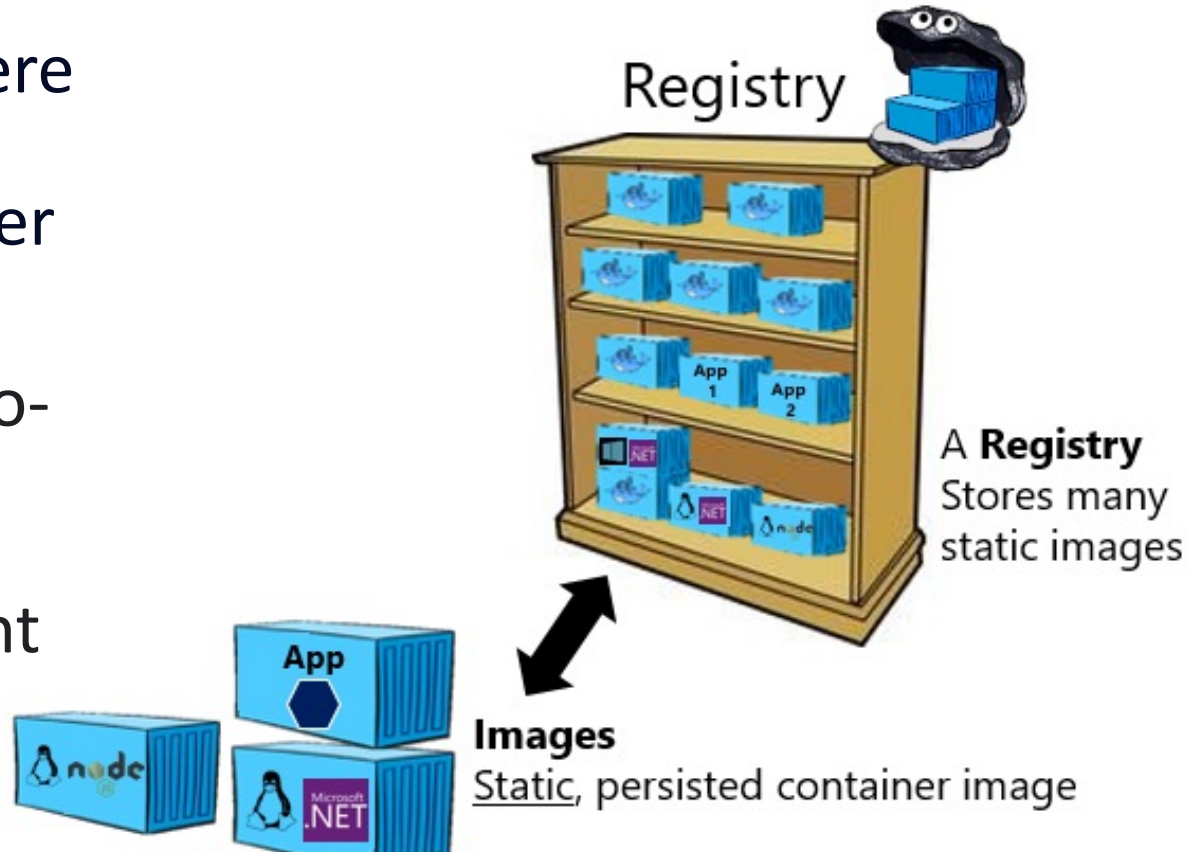
Docker images

- A Docker image is a file used to execute code in a Docker container.
- Docker images act as a set of instructions to build a Docker container, like a template.



Docker images repository

- A Docker repository is where you can store 1 or more versions of a specific Docker image.
- You can download ready-to-use docker images from public repository and run them for your development purpose.



Practical Works

Docker Stacks
for
Data Scientist

Jupyter Docker Stacks

https://jupyter-docker-stacks.readthedocs.io/en/latest/index.html

Jupyter Docker Stacks

Jupyter Docker Stacks are a set of ready-to-run Docker images containing Jupyter applications and interactive computing tools. You can use a stack image to do any of the following (and more):

- Start a personal Jupyter Server with the JupyterLab frontend (default)
- Run JupyterLab for a team using JupyterHub
- Start a personal Jupyter Server with the Jupyter Notebook frontend in a local Docker container
- Write your own project Dockerfile

<image/nation>

Let's start a Jupyter Server

- Type in the following command in your CLI windows

```
docker run -p 10000:8888 jupyter/scipy-notebook
```

- This command pulls the jupyter/scipy-notebook image from Docker Hub if it is not already present on the local host.
- It then starts a container running a Jupyter Server with the JupyterLab frontend and exposes the container's internal port **8888** to port **10000** of the host computer

To visit your Jupyter Server

- In your preferred browser, open the page like below
  `http://<hostname>:10000/?token=<token>`
  - **hostname** is the name of the computer running Docker
  - **token** is the secret token printed in the CLI console
- For example:
  - **`http://localhost:10000/?token=XXXXXXXXX`**

The container remains intact

- To quit the container, in command line windows, press **CTRL+C**
- The container remains intact for restart after the Server exits.
- Meaning whatever you have done to the container don’t persist.
- The files (e.g. Notebooks file that you create in the previous run) will all be gone after container restart.

What if you need to keep files?

- You can map your host computer folder to the container
- So, when you add files or changes file, it actually saves the files to your host computer folder and therefore your updates (codes or data) persists
- Type in the following command

```
docker run -p 10000:8888 -v
"${PWD}":/home/jovyan/work  jupyter/scipy-notebook
```

Wide selections of images

- https://jupyter-docker-stacks.readthedocs.io/en/latest/using/selecting.html

Selecting an Image

- Core Stacks
- Image Relationships
- Community Stacks

Using one of the Jupyter Docker Stacks requires two choices:

1. Which Docker image you wish to use
2. How you wish to start Docker containers from that image

This section provides details about the first.