Docker Workshop

day 1

Desambiguation

- Docker Inc: The company
- Docker: The Software

Docker: The Software

Docker is a **set of** platform as a service (PaaS) **products** that use OS-level **virtualization** to deliver software in packages called **containers**.

SIDE NOTE: in the future might not be docker:

https://www.tariqislam.com/posts/kubernetes-docker-dep/

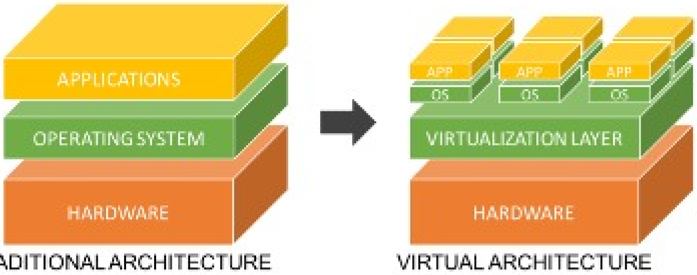
https://en.wikipedia.org/wiki/Docker_%28software%29

Agenda

- Virtualization
- Containers
- Set of products

Virtualization

Virtualization relies on software to simulate hardware functionality and **create a virtual computer system.** This enables more than one virtual system – and multiple operating systems and applications – on a single server.



https://www.vmware.com/se/solutions/virtualization.html

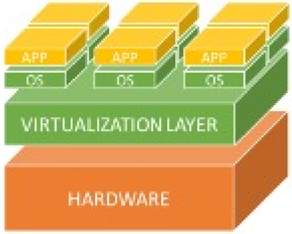
Key Properties of Virtual Machines

Partitioning
- Run multiple operating systems on one physical machine.
- Divide system resources between virtual machines.

Isolation
- Provide fault and security isolation at the hardware level.
- Preserve performance with advanced resource controls.

Encapsulation
- Save the entire state of a virtual machine to files.
- Move and copy virtual machines as easily as moving and copying files.

Hardware Independence
- Provision or migrate any virtual machine to any physical server.



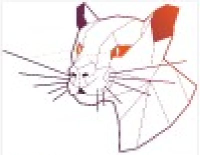
VIRTUAL ARCHITECTURE

https://www.vmware.com/se/solutions/virtualization.html

Virtual Machines

www.virtualbox.org

Oracle VM VirtualBox

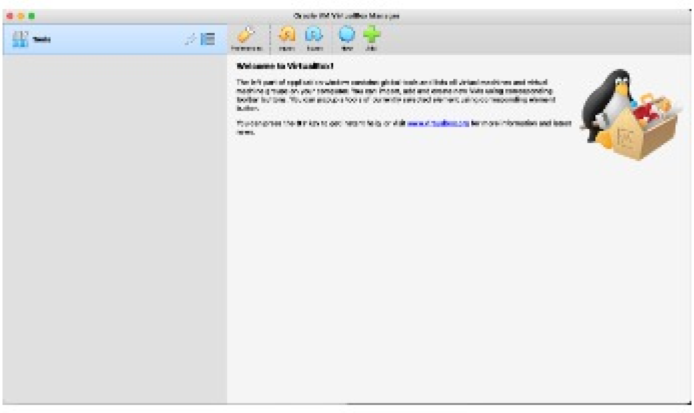


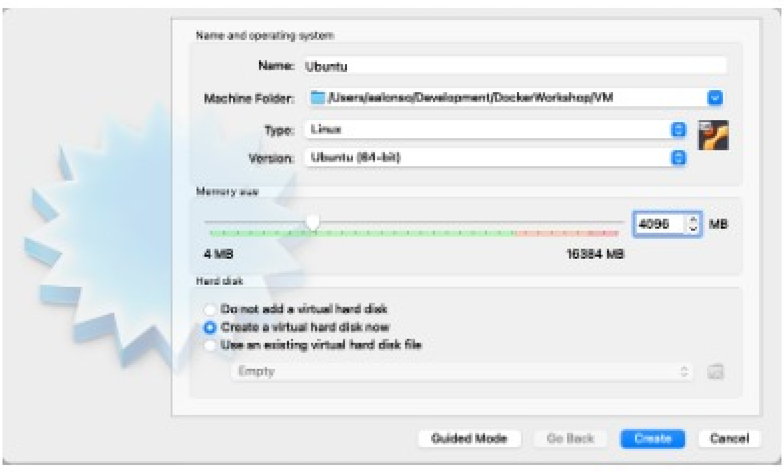
ubuntu.com

Get Ubuntu | Download | Ubuntu

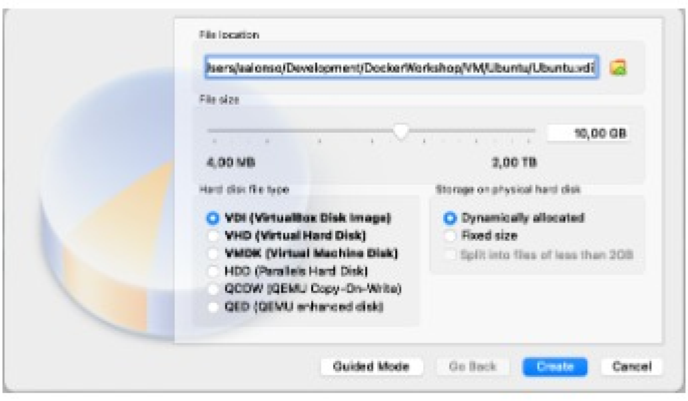
Download Ubuntu desktop, Ubuntu Server, Ubuntu for Raspberry Pi and IoT devices, Ubuntu Core and all the Ubuntu flavours. Ubuntu is an open-source software platform that runs everywhere from the PC to the server and the cloud.

https://www.vmware.com/se/solutions/virtualization.html

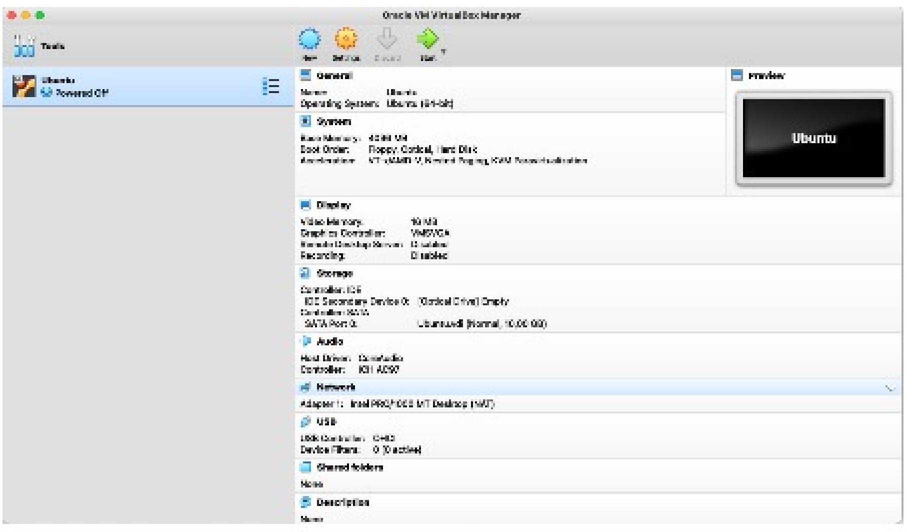




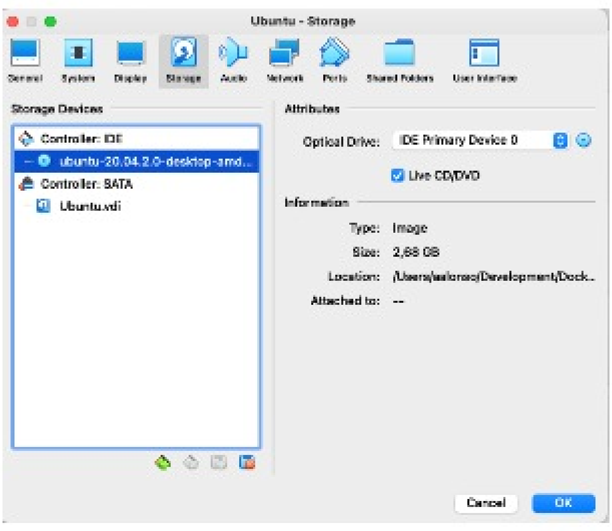
Name: Ubuntu
Machine Folder: /Users/aalonso/Development/DockerWorkshop/VM
Type: Linux
Version: Ubuntu (64-bit)
Memory size: 4096 MB (4 MB – 16384 MB)
Hard disk: Do not add a virtual hard disk / [✓ Create a virtual hard disk now] / Use an existing virtual hard disk file
Empty



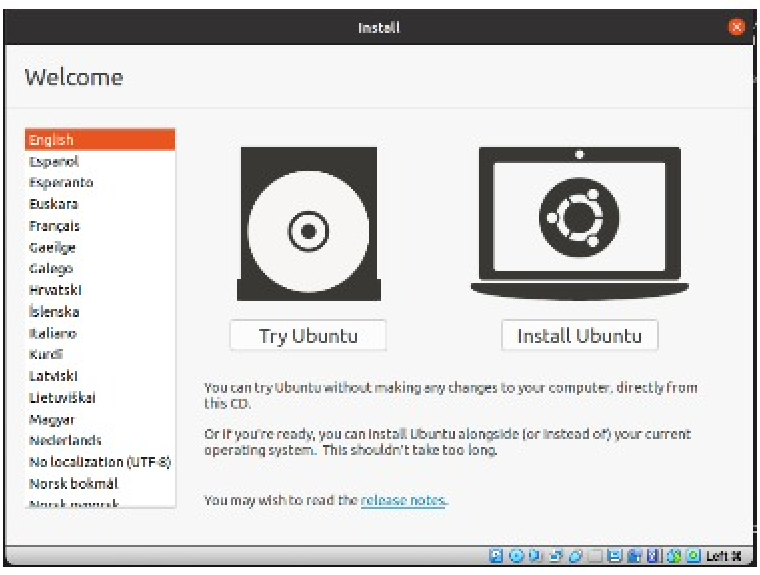
File location: sers/aalonso/Development/DockerWorkshop/VM/Ubuntu/Ubuntu.vdi
File size: 10,00 GB (4,00 MB – 2,00 TB)
Hard disk file type: [✓ VDI (VirtualBox Disk Image)] / VHD (Virtual Hard Disk) / VMDK (Virtual Machine Disk) / HDD (Parallels Hard Disk) / QCOW (QEMU Copy-On-Write) / QED (QEMU enhanced disk)
Storage on physical hard disk: [✓ Dynamically allocated] / Fixed size / Split into files of less than 2GB



General
Name: Ubuntu
Operating System: Ubuntu (64-bit)
System
Base Memory: 4096 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization
Display
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled
Storage
Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Ubuntu.vdi (Normal, 10,00 GB)
Audio
Host Driver: CoreAudio
Controller: ICH AC97
Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)
USB
USB Controller: OHCI
Device Filters: 0 (0 active)
Shared folders
None
Description
None



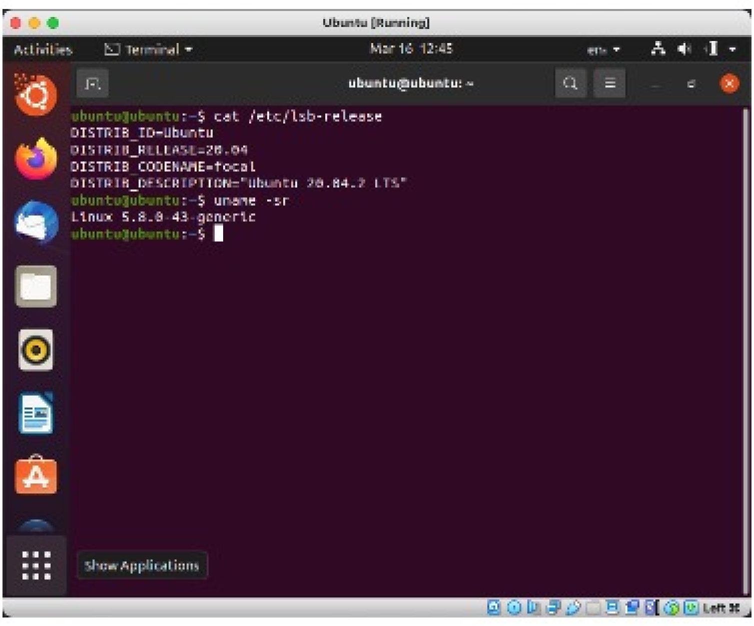
Controller: IDE
ubuntu-20.04.2.0-desktop-amd...
Controller: SATA
Ubuntu.vdi
Optical Drive: IDE Primary Device 0
[✓] Live CD/DVD
Type: Image
Size: 2,68 GB
Location: /Users/aalonso/Development/Dock...
Attached to: --



You can try Ubuntu without making any changes to your computer, directly from this CD.

Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.

You may wish to read the release notes.

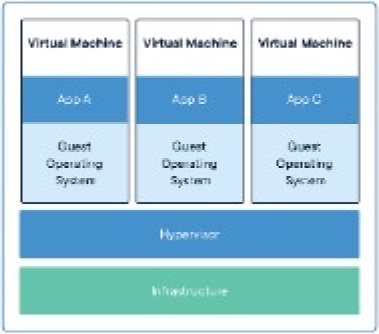


```
ubuntu@ubuntu:~$ cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=20.04
DISTRIB_CODENAME=focal
DISTRIB_DESCRIPTION="Ubuntu 20.04.2 LTS"
ubuntu@ubuntu:~$ uname -sr
Linux 5.8.0-43-generic
ubuntu@ubuntu:~$ 
```

Containers

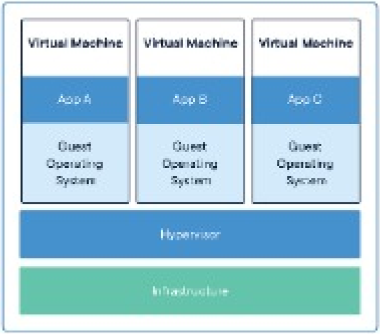
A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

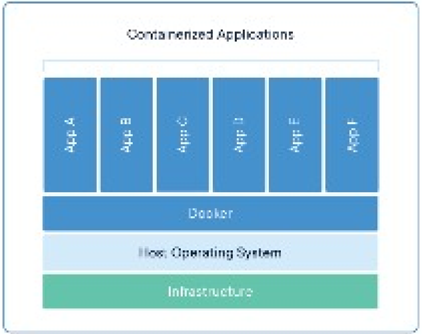
This is how it looks using VMs?



Containers

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

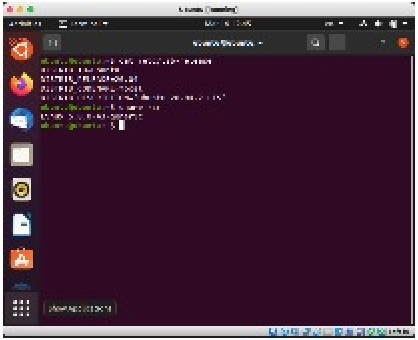




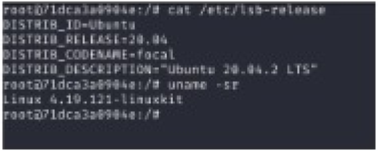
https://www.docker.com/resources/what-container

Containers

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.



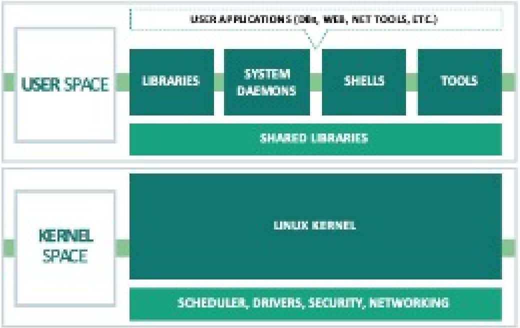
```
[illegible]
```



```
root@71dca3a0904e:/# cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=20.04
DISTRIB_CODENAME=focal
DISTRIB_DESCRIPTION="Ubuntu 20.04.2 LTS"
root@71dca3a0904e:/# uname -sr
Linux 4.19.121-linuxkit
root@71dca3a0904e:/#
```

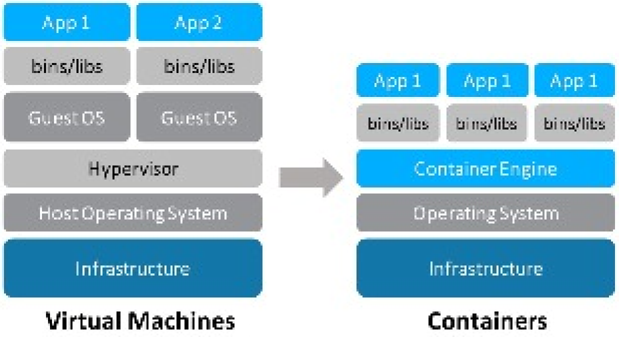
User vs Kernel Space

An application interacts only (in most cases) with the "User Space" of the OS so why don't we share the "Kernel Space" ??


USER APPLICATIONS (DBs, WEB, NET TOOLS, ETC.)
USER SPACE
LIBRARIES
SYSTEM DAEMONS
SHELLS
TOOLS
SHARED LIBRARIES
KERNEL SPACE
LINUX KERNEL
SCHEDULER, DRIVERS, SECURITY, NETWORKING

https://cumulusnetworks.com/blog/linux-architecture/

User vs Kernel Space



A set of tools

- **Docker Engine:** Running in the host and what runs the containers
- **Docker CLI:** The interface we will use to control docker
- **Other tools:** Other tools that we might use directly or will be used in the background for us

Docker engine

- systemd service unit:
    - systemctl status docker
- docker cli to list containers:
    - docker ps
- docker system information:
    - docker system info

Docker CLI

- help:
  - docker help
  - docker run --help
- Running containers:
  - docker run hello-world

https://docs.docker.com/engine/reference

Docker Hello-world

```
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/
```

Docker CLI

- All the containers:
    - `docker ps -a`
- Removing containers:
    - `docker rm [ID]`
- Removing containers when they are stopped:
    - `docker run --rm`

https://docs.docker.com/engine/reference

Docker CLI

- Runnning ubuntu 20.04
  - docker run -it ubuntu:20.04
  - check the kernel
  - check the OS version
  - check top
- Check the running/stopped containers
  - docker ps -a
  - A container run while it has active tasks. CONTAINERS DON'T IDLE

Q: Why containers stop after the last task?

Docker Containers vs Docker Image

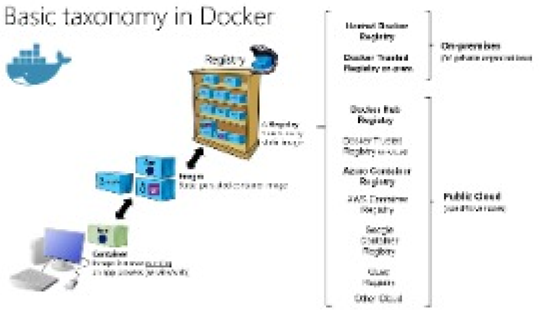
- Try installing zsh inside the container:
  - `docker run -it --rm ubuntu:20.04`
  - `apt update`
  - `apt install zsh`
  - `zsh`
  - `ps -p $$`

Q: What happens after you stop and start the container?

Docker Images

Q: What is a docker image?

- When we ran ubuntu:20.04 that is:
    - an image: ubuntu
    - tagged: 20.04
- This image is coming from docker hub registry



https://docs.microsoft.com/en-us/dotnet/architecture/microservices/container-docker-introduction/docker-containers-images-registries

Docker Images

Q: What if don't find the image we need? or How to make our own image?

- Dockerfile: describe the steps to create a new image
- docker build is the cli that we use to build that image

Dockerfile

Let's try to install zsh again
Create a file called: Dockerfile
with this:

```
FROM ubuntu:20.04
RUN apt update
RUN apt install -y zsh
CMD zsh
```

docker build

- `docker build -t myubuntu .`
- `docker run -it --rm myubuntu`
- `ps -p $$`

Files dependencies

Instead we could have a list of packages we want to install.

- `docker run -it --rm myubuntu xargs -a packages.txt apt install -y`

  xargs take a txt file and pass each line as arguments to apt

Dockerfile

Let's add the packages.txt
In the same Dockerfile
with this:

```
FROM ubuntu:20.04
RUN apt update
RUN apt install -y zsh
COPY $PWD/packages.txt packages.txt
CMD zsh
```

docker build

- `docker build -t myubuntu .`
- `docker run -it --rm myubuntu xargs -a packages.txt apt install -y`

Q:What happens if we modified the file packages.txt?

Docker Volumes

We can overwrite the data present in an image using volumes. Let's update the content of packages.txt and use volumes to overwrite the original content in the image

- docker run -it --rm -v "$(pwd)/packages.txt:/packages.txt" myubuntu xargs -a packages.txt apt install -y

The end of day 1

- containers vs VMs
- Isolation and kernel sharing
- Docker engine and CLI
- docker container and images
- Dockerfiles
- Volumes

Next:

- docker cli is getting too complicated. Docker compose
- how do we communicate with a container. Networking