Università di Pisa

Dept. of Information Engineering

Course on Wireless Networks - 2020/2021

Virtualization (LAB)

Alessandra Fais – PhD Student

email: alessandra.fais@phd.unipi.it
web page: for.unipi.it/alessandra_fais/

LAB organization

- **PART I (theoretical)**
  - Introduction to SDN, NFV, MEC * concepts
  - Cloud computing and service-based architectures

* SDN = Software Defined Networking,
NFV = Network Function Virtualization,
MEC = Multi-access Edge Computing

- **PART II**
  - OpenStack cloud computing platform
  - OpenStack and NFV
  - Live session: OpenStack platform of the DII CrossLab project

LAB organization

* VM = Virtual Machine

- **PART III**
  - Virtualization overview and different approaches
    - VMs* on hypervisors, containers, alternative solutions
  - Hands-on session: VirtualBox + Ubuntu Linux VM creation

- **PART IV**
  - Containers -> Docker
  - Orchestrators -> Kubernetes
  - Hands-on session: Docker, docker-compose, Kubernetes

PART III

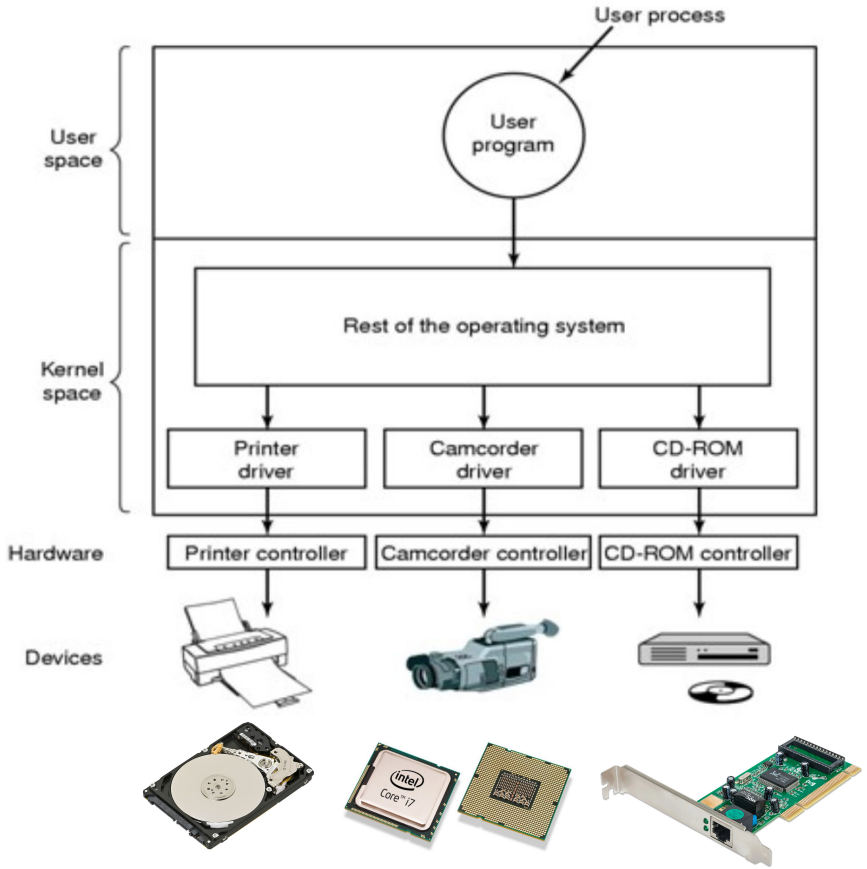
Outline of Part III

1) Virtualization overview
2) Virtualization approaches
    - Hypervisor-based solutions
    - Container-based solutions
    - Alternative solutions
3) Considerations on performance
4) Hands-on session with VirtualBox
    - Installation
    - Creation of a VM

Virtualization overview

Hardware vs Software

- **Hardware (HW)**: physical components of a computer
  - Monitor, CPUs, GPUs, hard drive, NICs, …
- **Software (SW)**: set of instructions written to alter the state of the hw
  - Device Drivers
  - Operating System
  - Application Software



Virtualization: an overview

- Run different **services** in the cloud
- Keep up with the **growth of exchanged data**
  - Increase the capability of *data centers* by means of *server virtualization*

Virtualization: an overview

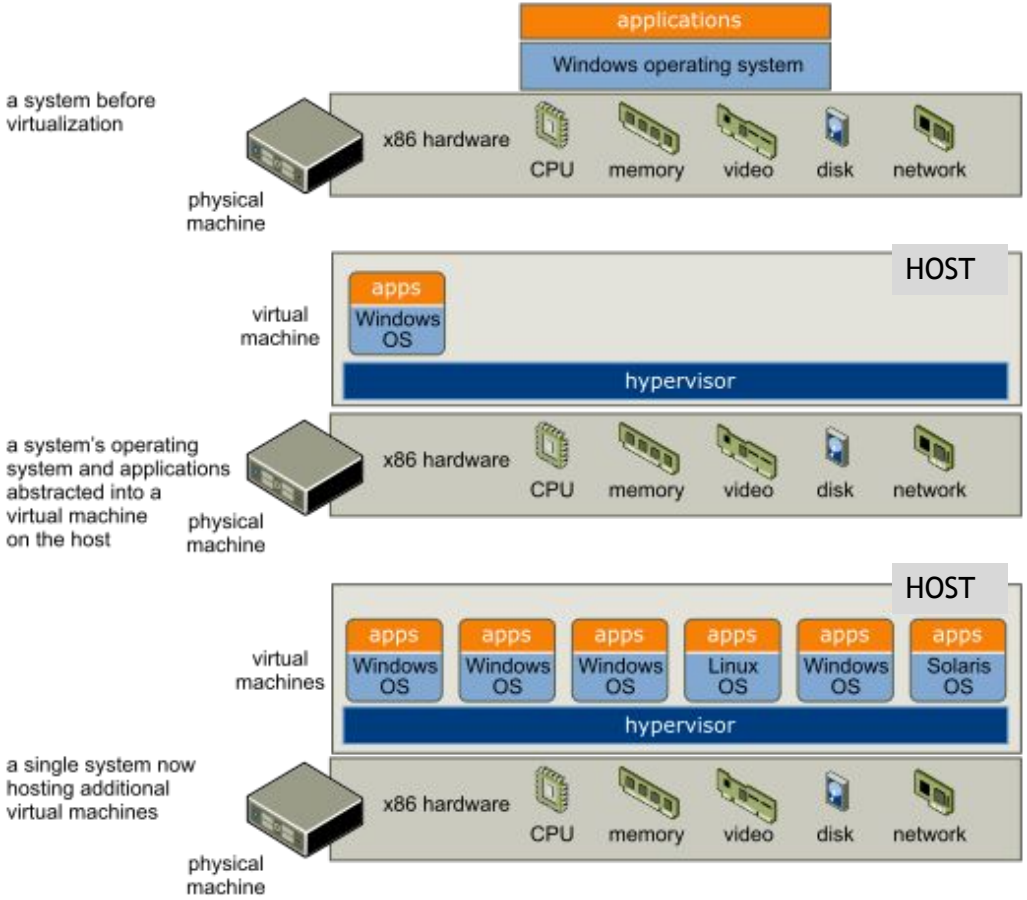
- Considerations on **availability** and **cost**

Modern machines come with a lot of hardware resources

(many CPUs, large memories, …)

- Partition the resources of a **physical server** among several virtual machines, each running a **virtual server**
- Less expensive than having a separate machine for each server
  - Operating **costs** (space, energy, heat, failures, …) **reduced**
- Running inside a virtual machine has a performance cost

Virtualization: an overview



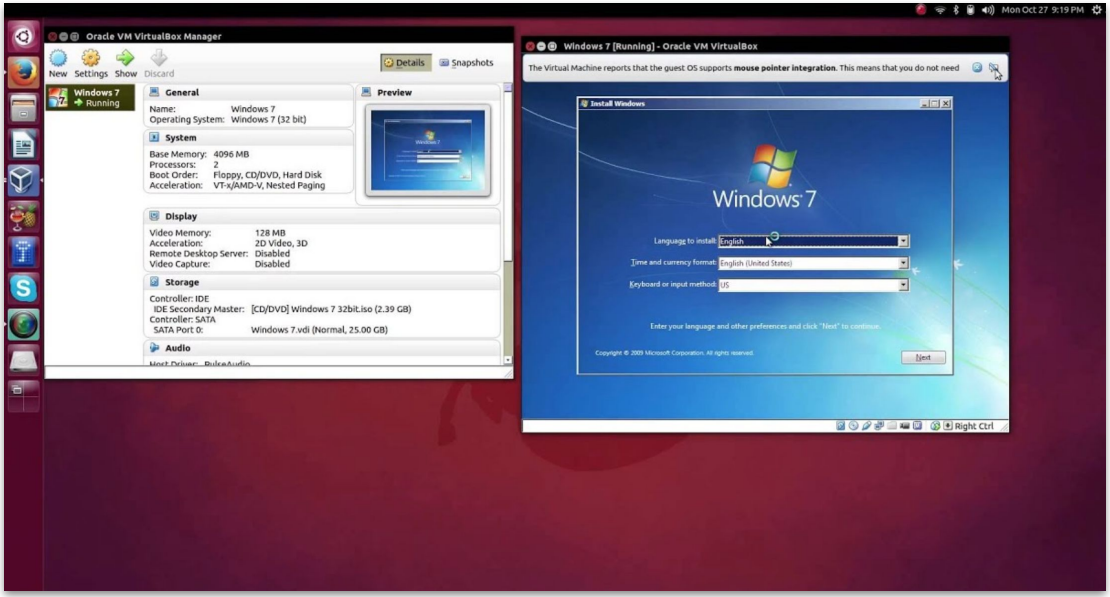
Virtualization: an overview

Considerations on **flexibility**

- More flexible **management of resources**
- **Dynamic provisioning**
  - Easily add/remove persistent storage, memory, processors
- Improved **utilization of resources**
  - Available resources multiplexed among the active users
- **Live migration**
  - Move an entire running system from a physical machine to another
- **Ease of deployment**

Virtualization: an overview

- Considerations on **flexibility**



- Use of applications that run on different OSes on the same desktop

Virtualization: where and why

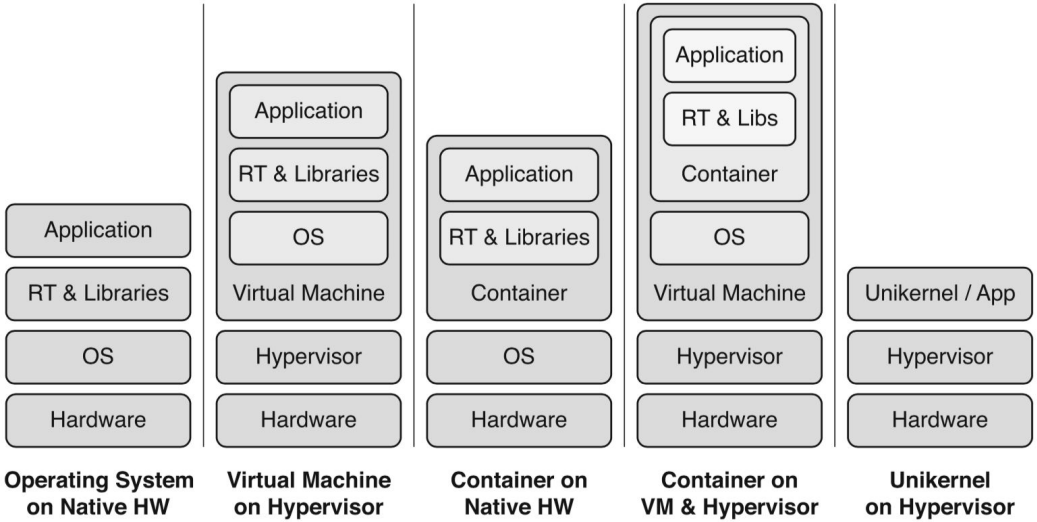
- Context
  - Cloud environment
  - Internet of Things (IoT)
  - Network Function Virtualization (NFV)

- Benefits
  - Hardware independence
  - Isolation
  - Secure user environment
  - Increased scalability

Virtualization approaches

Virtualization: different approaches

- Traditional virtualization
  - Whole system virtualization
  - *Hypervisor-based*
- Lightweight virtualization
  - *Container-based* (OS level)
  - *Unikernel-based* (library OS on a hypervisor level)



Virtualization approaches

Hypervisor-based virtualization architecture

Hypervisor-Based Virtualization Architecture

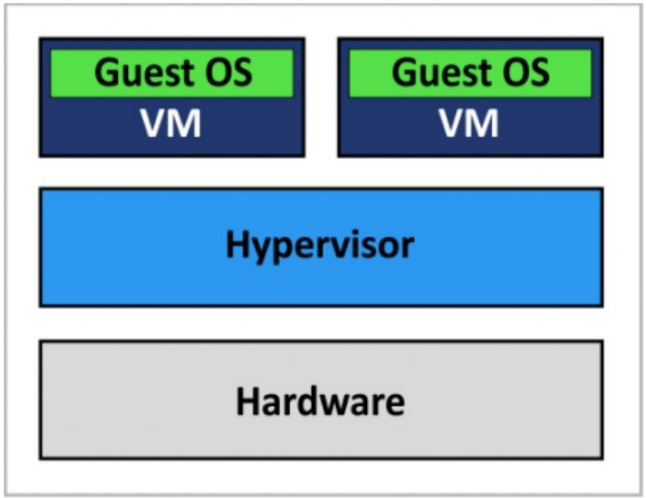
The **Hypervisor** or **Virtual Machine Manager** (VMM) level provides:

- **Hardware abstraction**
  - Virtual HW and virtual device drivers
- Standalone **VM instances** independent and isolated from the Host OS
- **Full Guest OS** (e.g. Linux) runs on top of the virtualized HW in each VM instance
- **Large disk images**

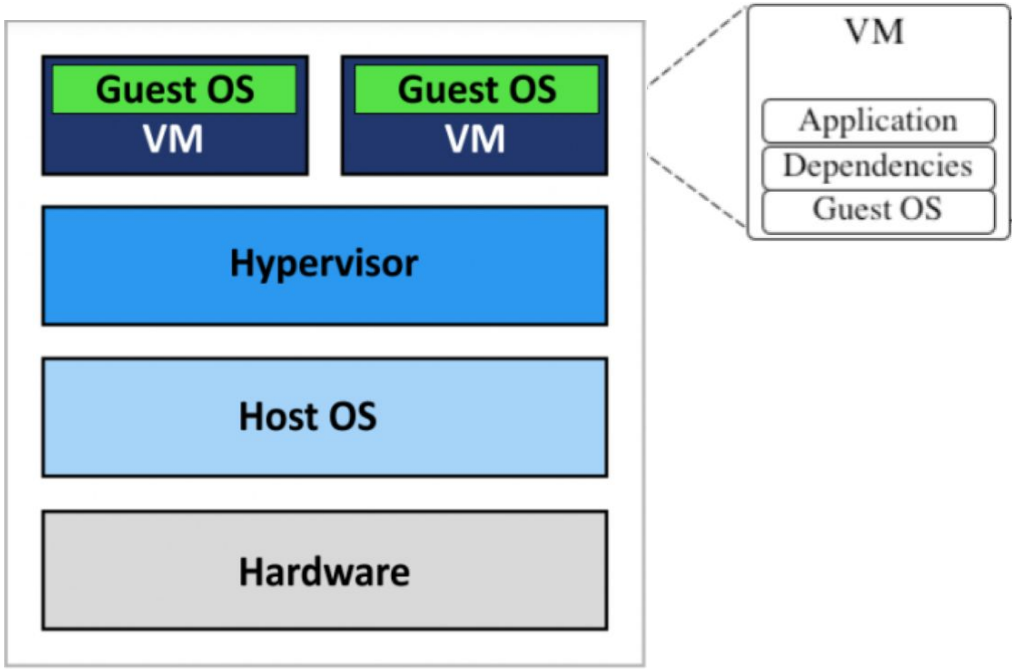
Oracle VM VirtualBox

Kernel-based Virtual Machine

Hypervisor-Based Virtualization Architecture



Type 1 Hypervisor
(Bare-Metal Architecture)



Type 2 Hypervisor
(Hosted Architecture)

Hypervisor-Based Virtualization Architecture

- **Type 1 Hypervisor (native or bare-metal)**
  - Operate on top of the Host's HW

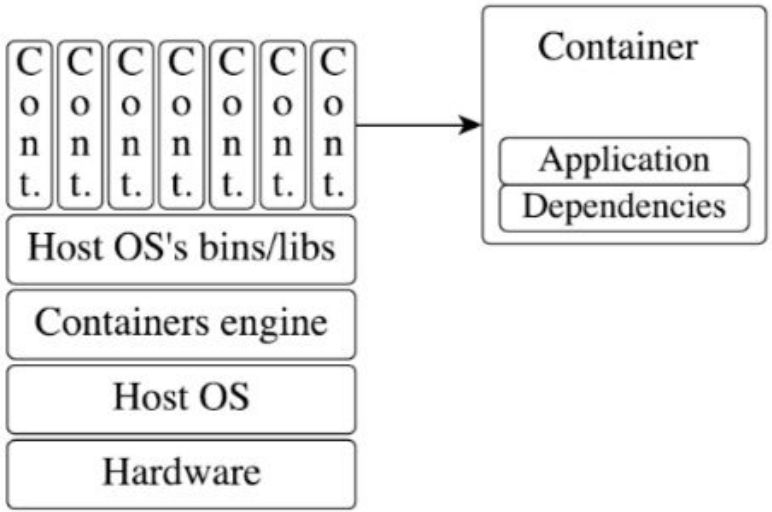
- **Type 2 Hypervisor (client or hosted)**
  - Operate on top of the Host's OS

Virtualization approaches

Container-based virtualization architecture

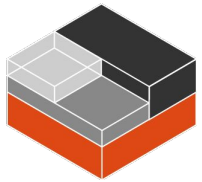
Container-Based Virtualization Architecture

- Isolation of processes at the **OS level**
  - Avoid overhead for HW virtualization
- Containers run on top of the same **shared Host OS kernel** and libraries
- One or more processes can be run within a container

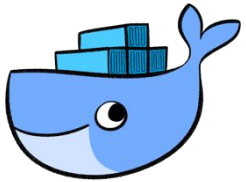


Container-Based Virtualization Architecture

- **High-density** of virtualized instances
- **Small disk images**
- Multi-tenant **security** issues
  - Host kernel exposed to all the containers
  - Worse resource isolation w.r.t. hypervisors
- Claim to offer **superior performance** with respect to hypervisor-based solutions
- **Very popular** virtualization solution!



Linux Containers



Docker

Virtualization approaches

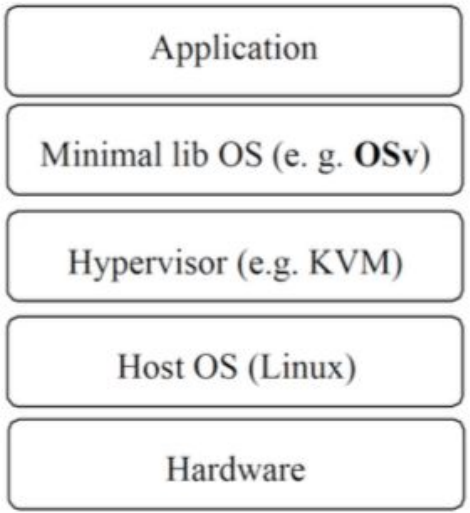
Alternative solutions: unikernels

Alternative Virtualization Architecture

- **Unikernels** are an emerging solution

- Library Operating System
  - Run single applications
  - Avoid the overhead and configuration of a complete Guest OS

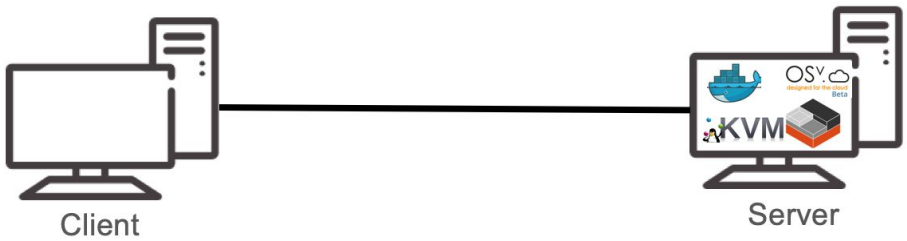
- Run on top of a hypervisor
  - Virtual hardware interface
  - Isolation benefits



Considerations on performance

A word on performance comparison

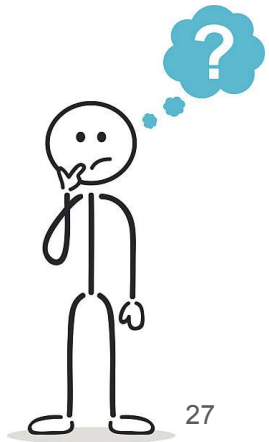
- Interesting metrics
  - **Instance startup time**
  - **Image size**
  - **Image footprint** (memory usage)
  - **Memory throughput** (Disk I/O)
  - **CPU** (e.g., execution time, FLOPs)
  - **Network latency** (Round Trip Time)
  - **Network I/O** (e.g., number of UDP/TCP transactions – request, response – between client and server)



Summary: how to choose the most suited technology for the intended purpose?

- Container-based solutions and emerging ones (like unikernels) are challenging the traditional hypervisor-based approach in cloud computing

- Light-weight technologies offer
  - Shorter instance startup time
  - Tiny image sizes
  - Smaller amount of memory required
  - Dense deployment of instances



Summary: how to choose the most suited technology for the intended purpose?

- Containers introduce almost negligible overhead
  - Tradeoff between versatility and ease of use and security aspects
- Hypervisors performance continuously improve
- Network efficiency is a delicate aspect for all the solutions

About **performance** and **overheads**:

- https://www.backblaze.com/blog/vm-vs-containers/
- https://www.stratoscale.com/blog/data-center/running-containers-on-bare-metal/
- https://www.brightcomputing.com/blog/containerization-vs.-virtualization-more-on-overhead

Hands-on with VirtualBox

Hands-on with VirtualBox: exercise

- Let's install Oracle VirtualBox 1
- We want to create a complete Ubuntu Linux VM 2
- Run the *Ubuntu Guest OS* on top of the *Host OS* through the *VirtualBox hypervisor* 3

Configuring a Linux environment will be useful for the next hands-on session with Docker!

Hands-on with VirtualBox

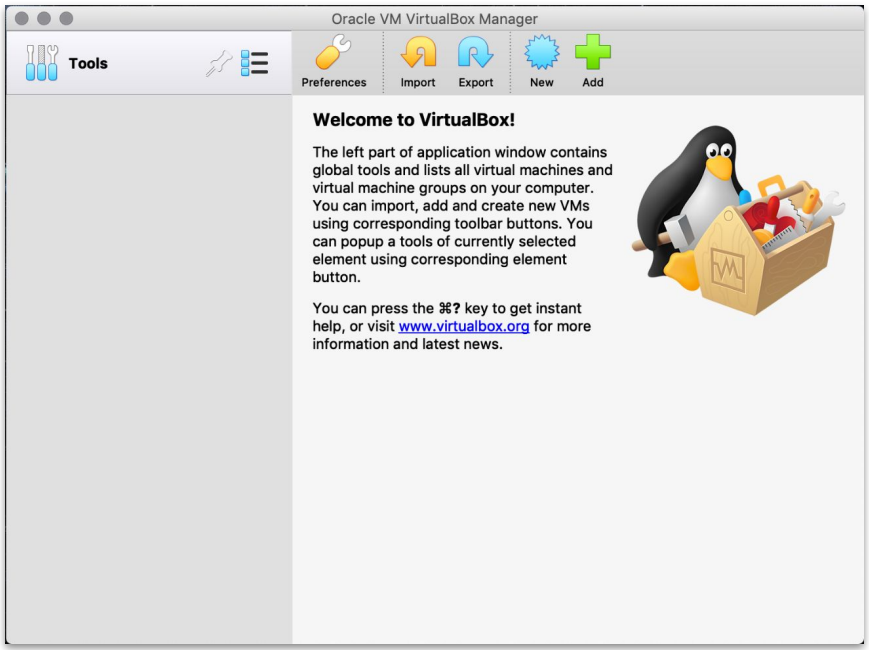
Installation

Hands-on with VirtualBox: installation

1

- Download Oracle VirtualBox from the official site
  - https://www.virtualbox.org/wiki/Downloads

  and install the package suitable for your platform



Installation is successful if you are able to display this window!

Hands-on with VirtualBox

Ubuntu image

Hands-on with VirtualBox: Ubuntu image

2

- Download **Ubuntu 18.04 LTS** from the official site
  - https://releases.ubuntu.com/18.04.5/

Docker guide assumes that you work on a Ubuntu OS!

Ubuntu 18.04.4 LTS

Download the latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years, until April 2023, of free security and maintenance updates, guaranteed.

Ubuntu 18.04 LTS release notes

Recommended system requirements:

- 2 GHz dual core processor or better
- 4 GB system memory
- 25 GB of free hard drive space
- Either a DVD drive or a USB port for the installer media
- Internet access is helpful

Download

For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors, and past releases see our alternative downloads.

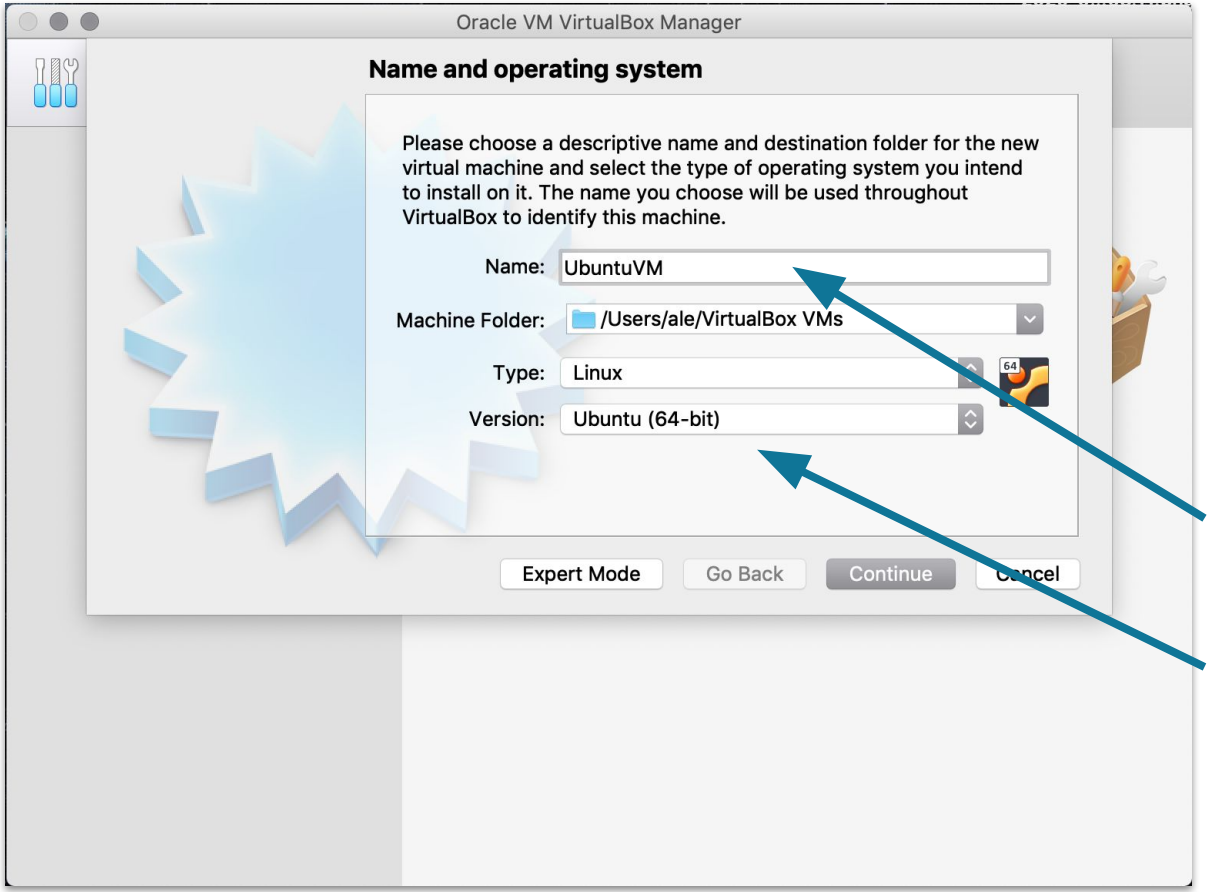
NOTE:
If you encounter difficulties in running this image on your Host machine, you can download and use more lightweight Ubuntu-based distributions (e.g. Lubuntu from https://lubuntu.me/downloads/)

Hands-on with VirtualBox

Configuring the hardware settings for the VM

Hands-on with VirtualBox: configuration

2



Choose system type

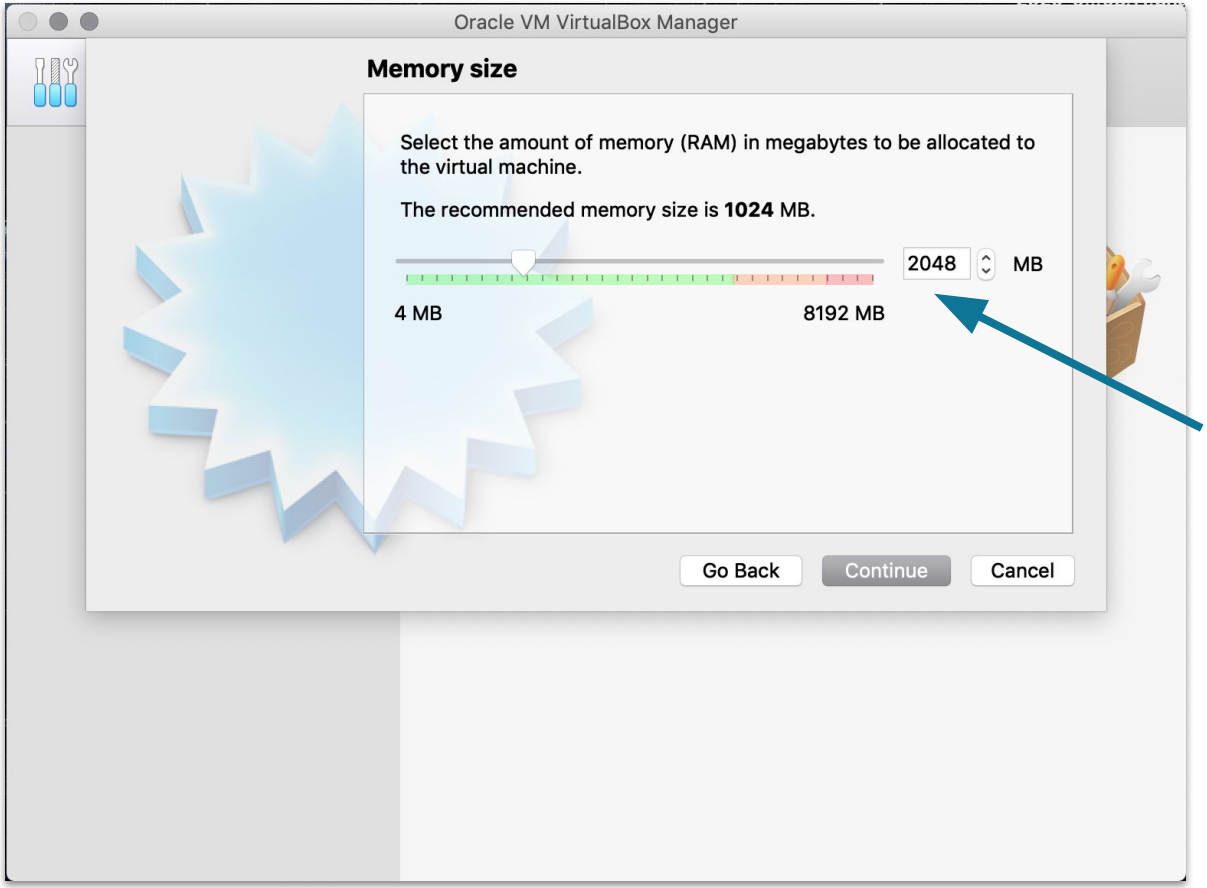
Click on New button on the Oracle VM VirtualBox Manager window

Write your system name

Select the OS type and version

Hands-on with VirtualBox: configuration

2

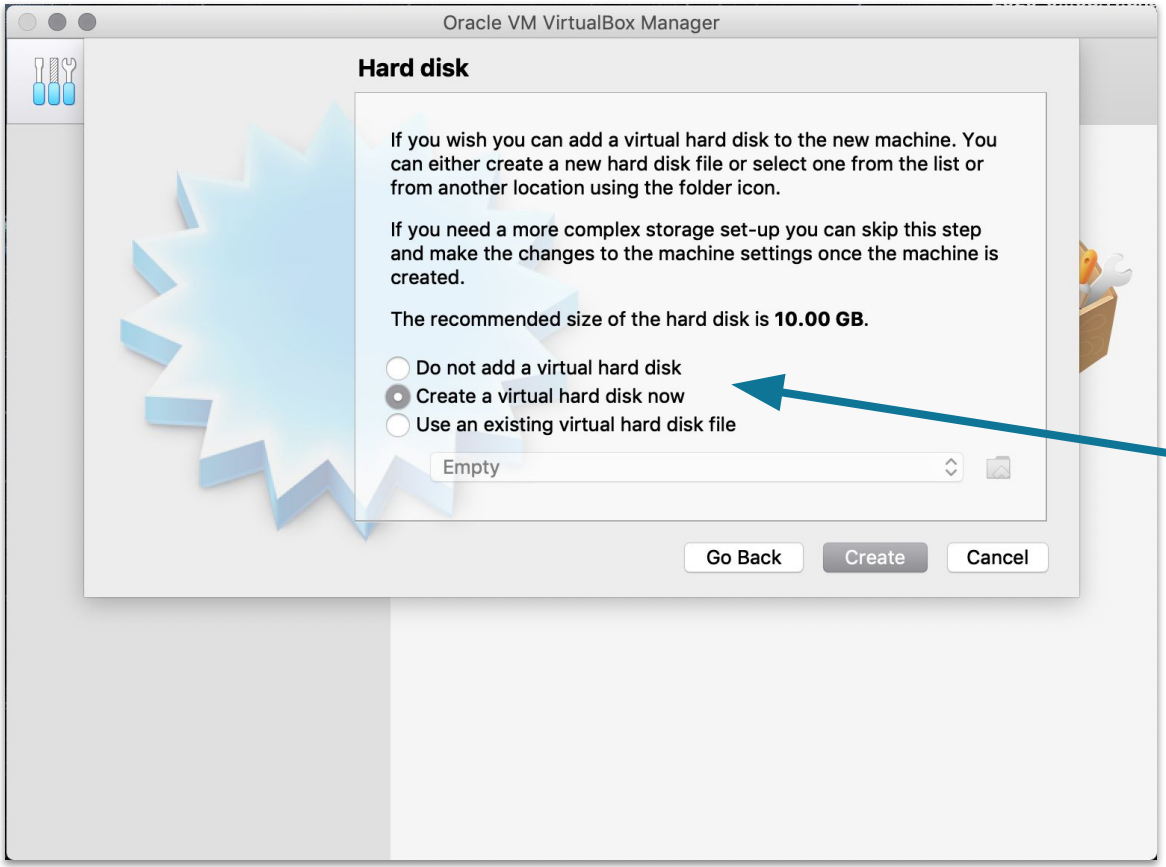


Select the amount of RAM

Here I set 2 GB of RAM (2048 MB)

Hands-on with VirtualBox: configuration

2

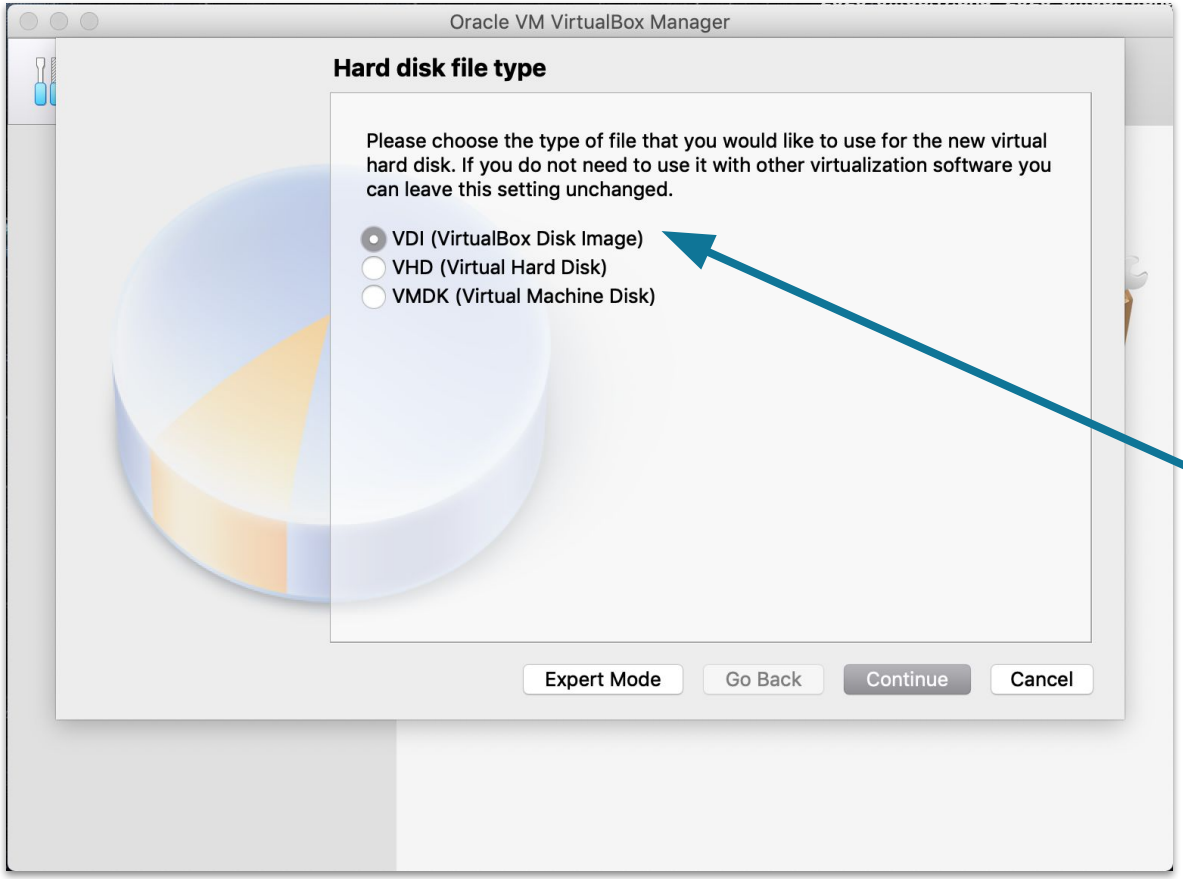


Hard-disk settings

Select "Create a virtual hard disk now" to make a virtual disk space

Hands-on with VirtualBox: configuration

2

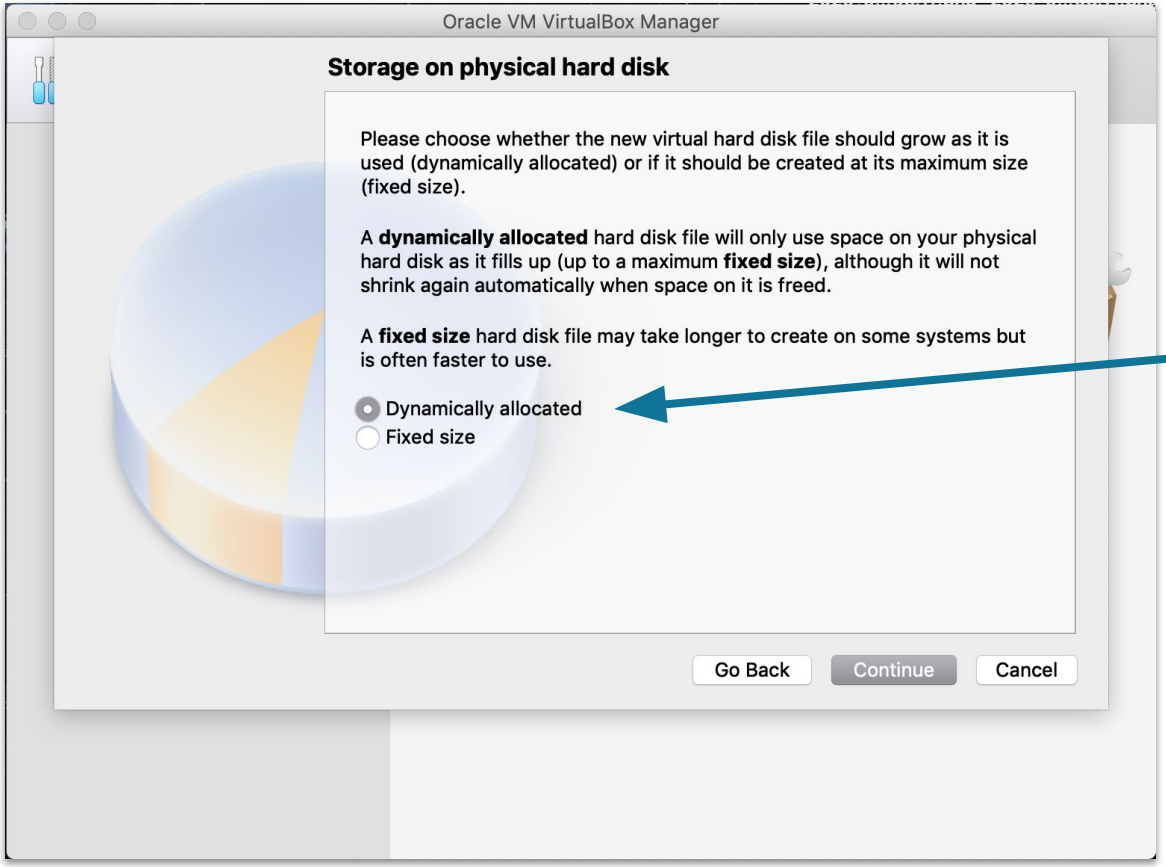


Hard-disk settings

Select "VDI" to use the VirtualBox standard disk image format

Hands-on with VirtualBox: configuration

2

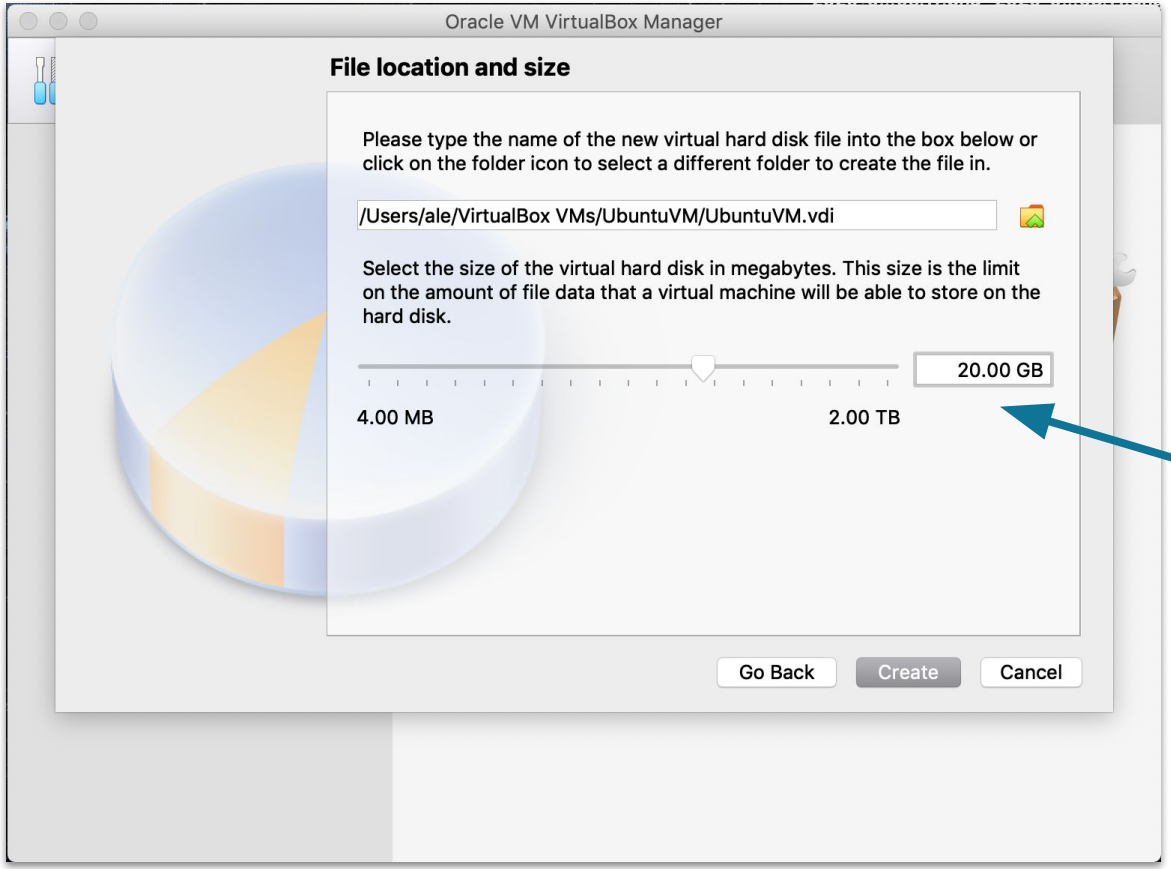


Hard-disk settings

Select "Dynamically allocated"

Hands-on with VirtualBox: configuration

2

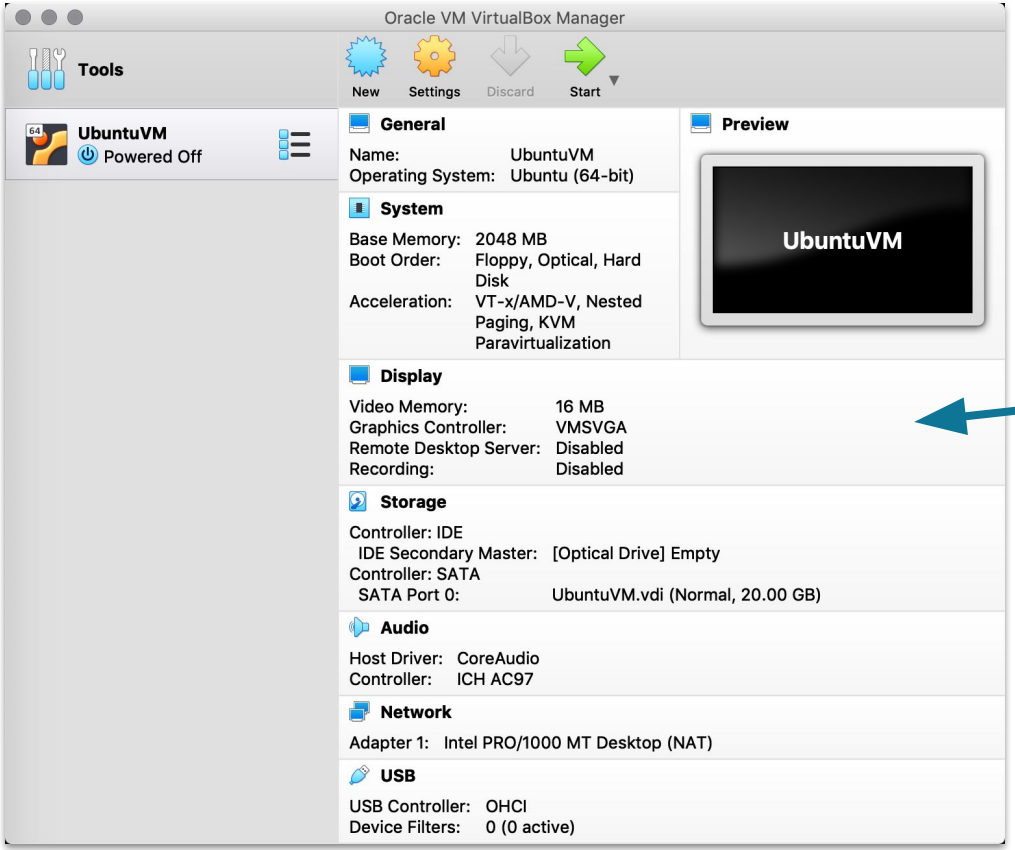


Hard-disk settings

Set the maximum hard drive size

Hands-on with VirtualBox: configuration

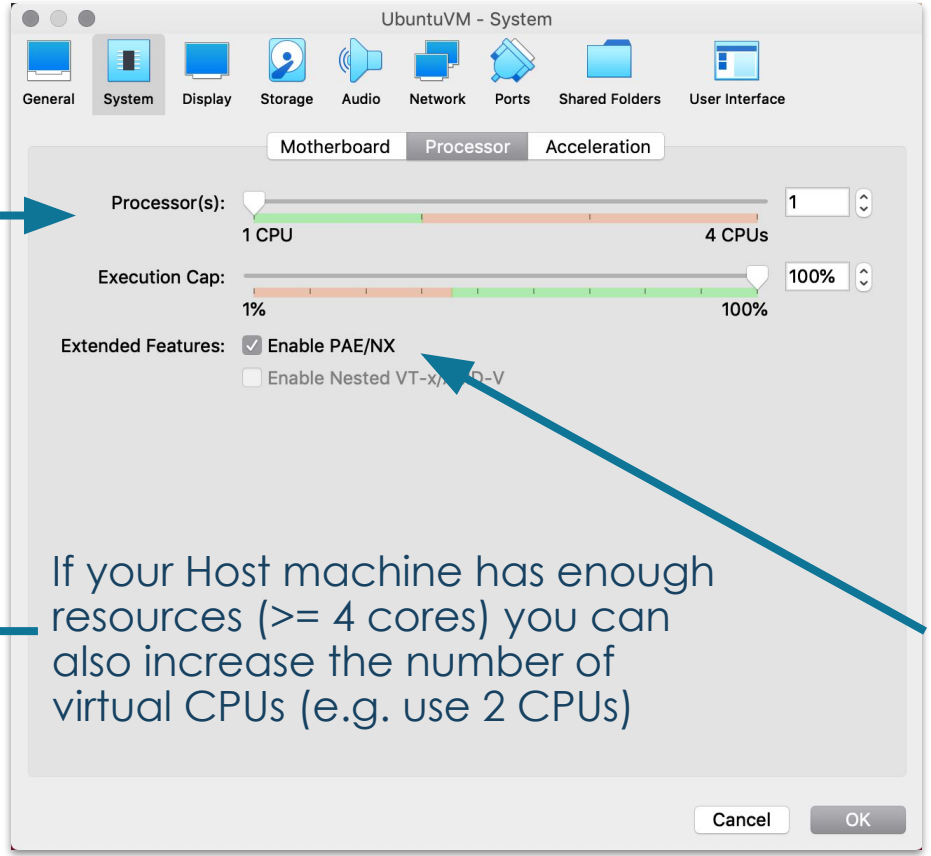
2



The configuration of the hardware settings is now completed

Hands-on with VirtualBox: optional

2



If your Host machine has enough resources (>= 4 cores) you can also increase the number of virtual CPUs (e.g. use 2 CPUs)

Go to Settings from the main VirtualBox screen

Set some advanced configuration properties (complete guide at this link https://www.virtualbox.org/manual/ch03.html)

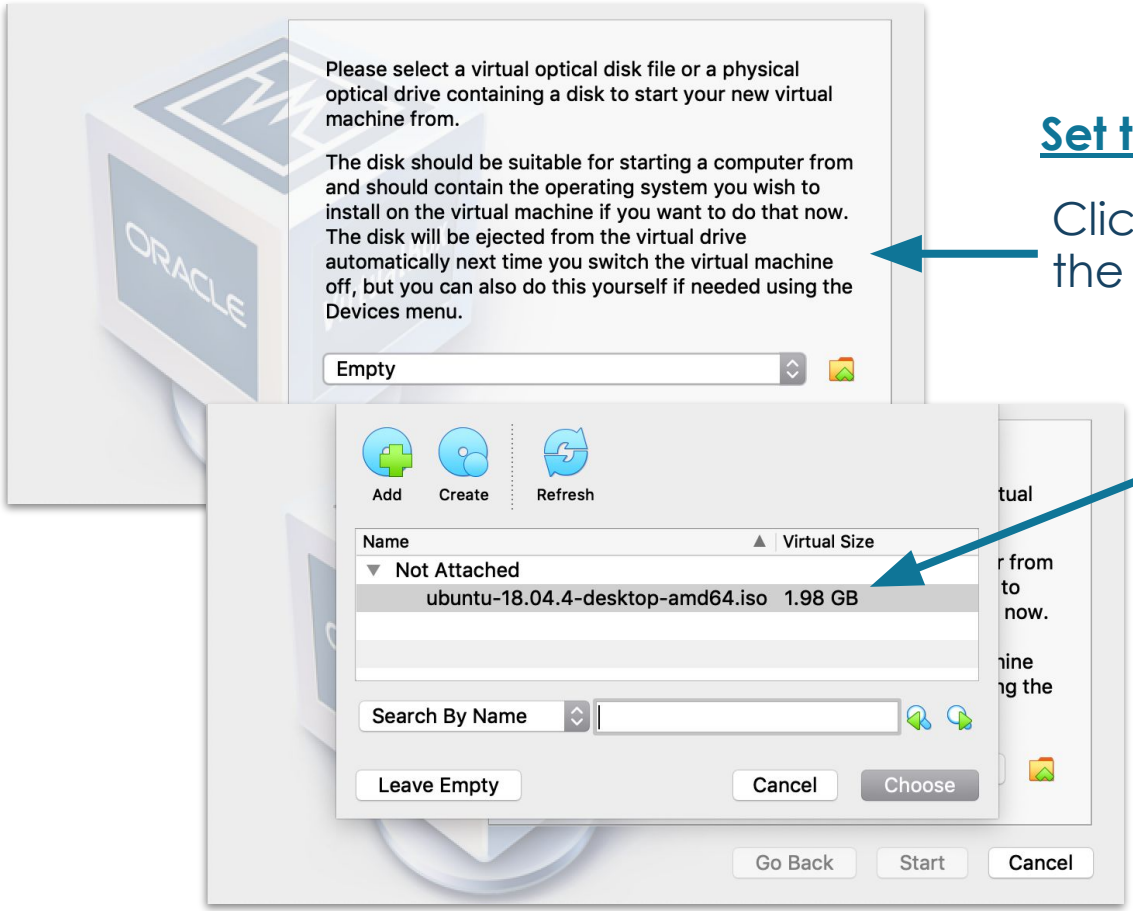
If PAE/NX is enabled, the Physical Address Extension capability of the Host CPU will be exposed to the VM

Hands-on with VirtualBox

Installing the Guest OS from the ISO image and launching the VM

Hands-on with VirtualBox: launch the VM

3



Set the Ubuntu image

Click Start and launch the system

Select the ISO file of the system (the one downloaded from the Ubuntu official site)

Hands-on with VirtualBox: launch the VM

3

Set the Ubuntu image

Now press Start and wait for the final phase: Ubuntu Linux installation

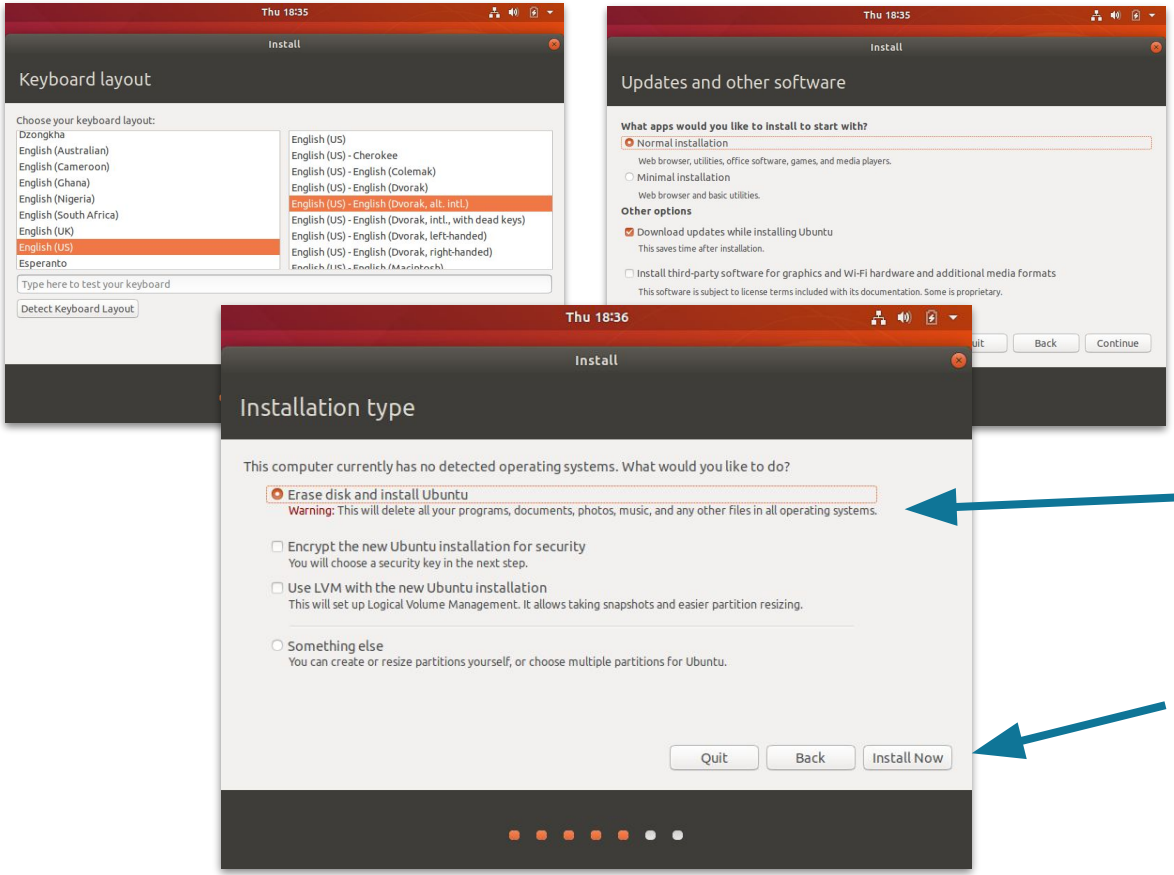
Hands-on with VirtualBox: install Ubuntu

3

Proceed with the installation

Hands-on with VirtualBox: install Ubuntu

3

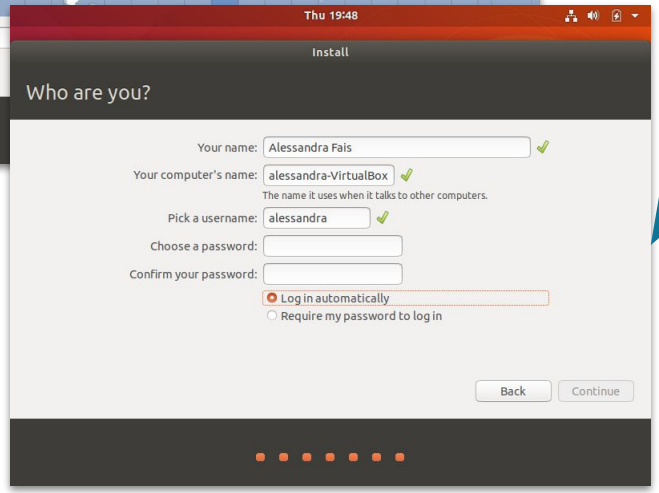


Select "Erase disk and install Ubuntu"

Then press "Install now" and "Continue" in the following message

Hands-on with VirtualBox: install Ubuntu

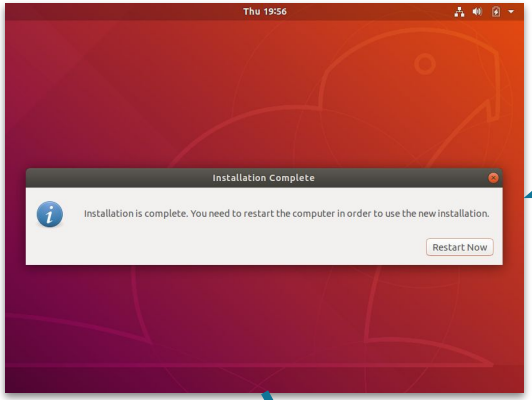
3



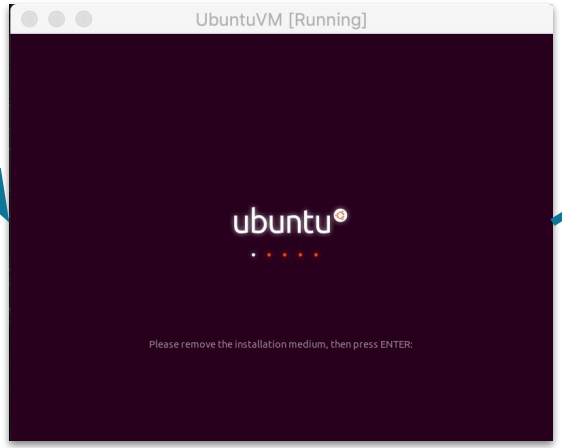
Set up your account

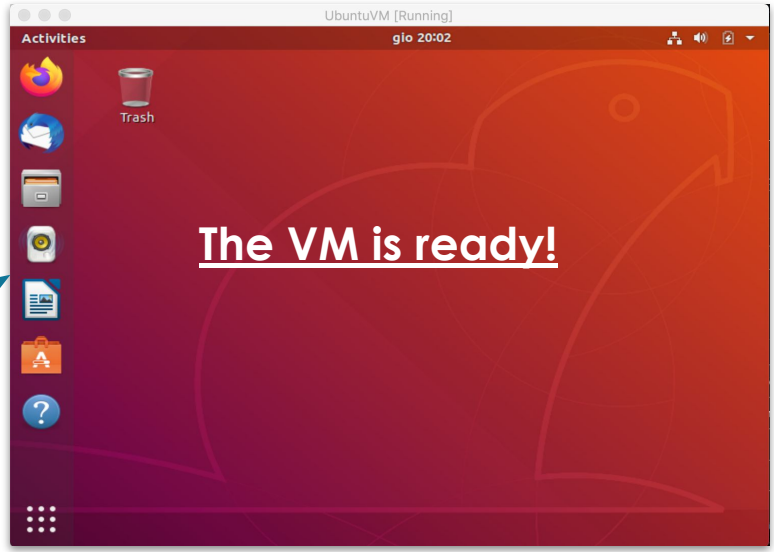
Ubuntu is installing...

Hands-on with VirtualBox: final result



After restarting you will have your brand new Ubuntu 18.04 LTS VM ready to use! Enjoy ;)





Useful references

Useful references

- Hypervisors vs Lightweight Virtualization: a Performance Comparison
  - http://faculty.washington.edu/wlloyd/courses/tcss562/research_papers/T3_Hypervisors_vs_Lightweight_Virtualization_A_Performance_Comparison.pdf
- A Performance Survey of Lightweight Virtualization Techniques
  - https://www.researchgate.net/publication/319407909_A_Performance_Survey_of_Lightweight_Virtualization_Techniques
- Kernel-based Virtual Machine (KVM)
  - https://www.redhat.com/en/topics/virtualization/what-is-KVM
- Oracle VM VirtualBox
  - https://download.virtualbox.org/virtualbox/6.1.4/UserManual.pdf
- Linux Containers
  - https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux_atomic_host/7/html/overview_of_containers_in_red_hat_systems/introduction_to_linux_containers
- Docker
  - https://www.docker.com/get-started
- vmware glossary
  - https://www.vmware.com/topics/glossary/