Introduction to Linux

Introduction to Linux World, Virtualization And Linux Console

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg

Have a Question?

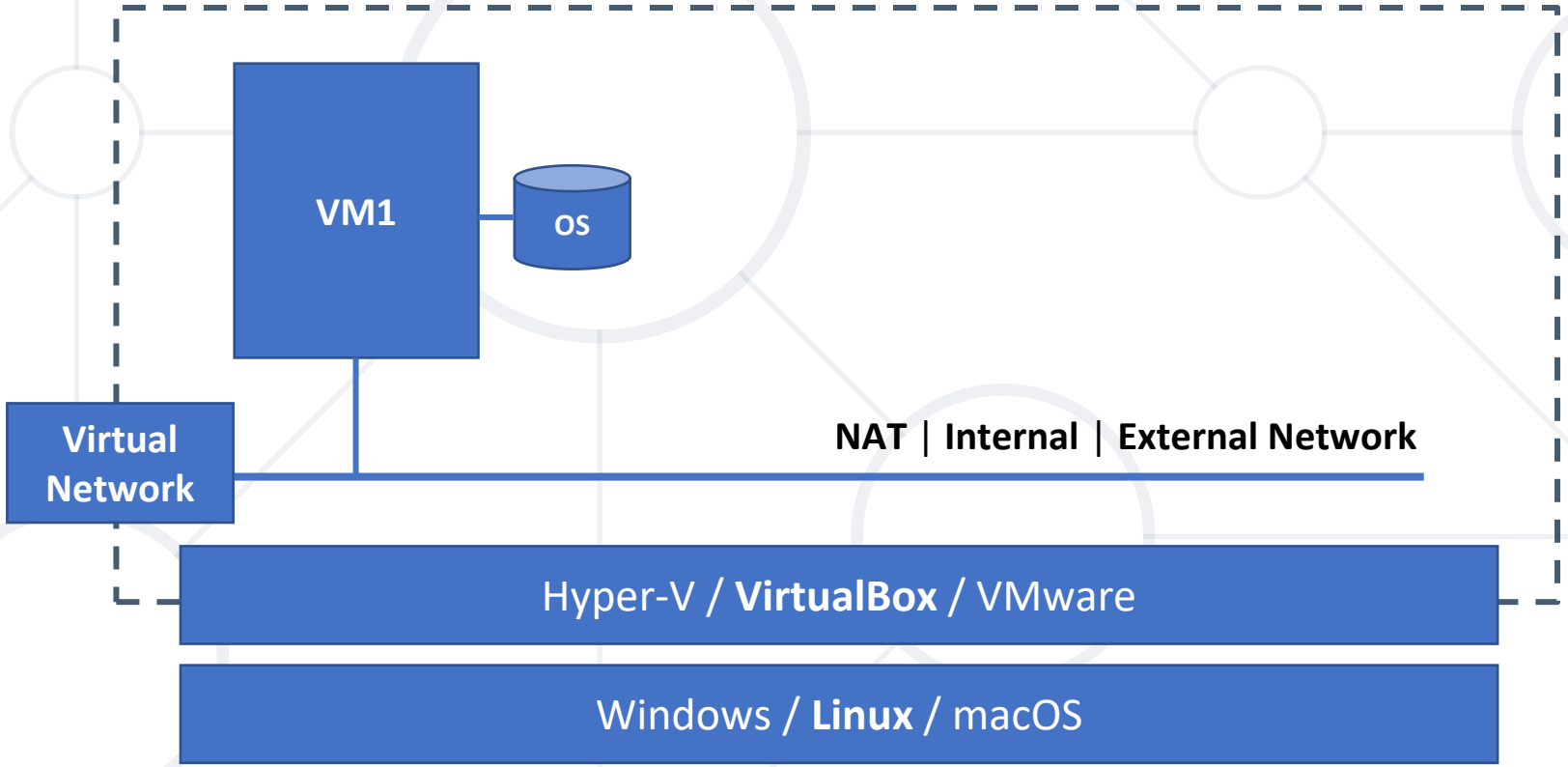
Software University

sli.do

#LSA

Table of Contents

Lab Infrastructure



Introduction to Linux

Why? Where? What is (Not)?

Why Linux?

- It is a **phenomenon**
  - Went all the way from a **student's hobby** to **world domination**
- Internet **runs** on Linux
  - Operating system for **over 95%** of the top one million domains
- It runs on **100%** of the **top 500** supercomputers*
- There is **huge demand** for Linux skills
- It is both **challenging** and **fun**

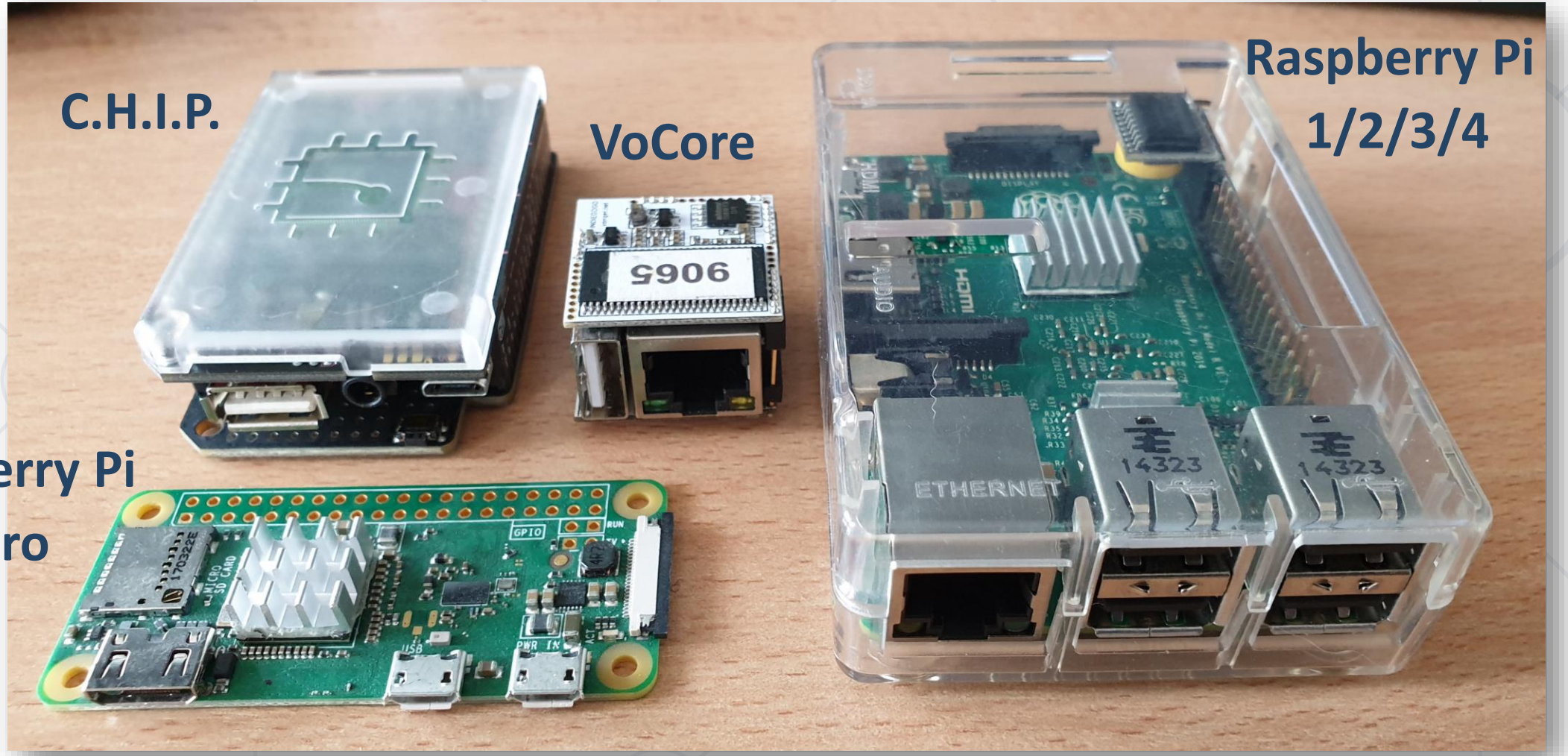
* *https://www.top500.org/statistics/details/osfam/1*

Where is Linux?

- **Every** aspect of our life
  - **Server rooms** and **data centers**
  - **Home** and **office**
  - **Business** and **government**
  - **Spacecraft** and **regular** transportation
  - **Hospitals**, **scientific** and **medical** laboratories
- Devices of **any** size
  - From **tiniest** devices, trough PCs, servers, and **supercomputers**

Tiny Linux Computers

Software University



What is (Not) Linux?

- It is **NOT** the complete OS, it is just the **Kernel**
- **Often** the term is used to refer to the whole OS
- A Linux-based system is a **modular Unix-like** operating system
- Linux kernel is a **monolithic** kernel

Linux System Architecture

Building Blocks and Definitions

Linux Definitions

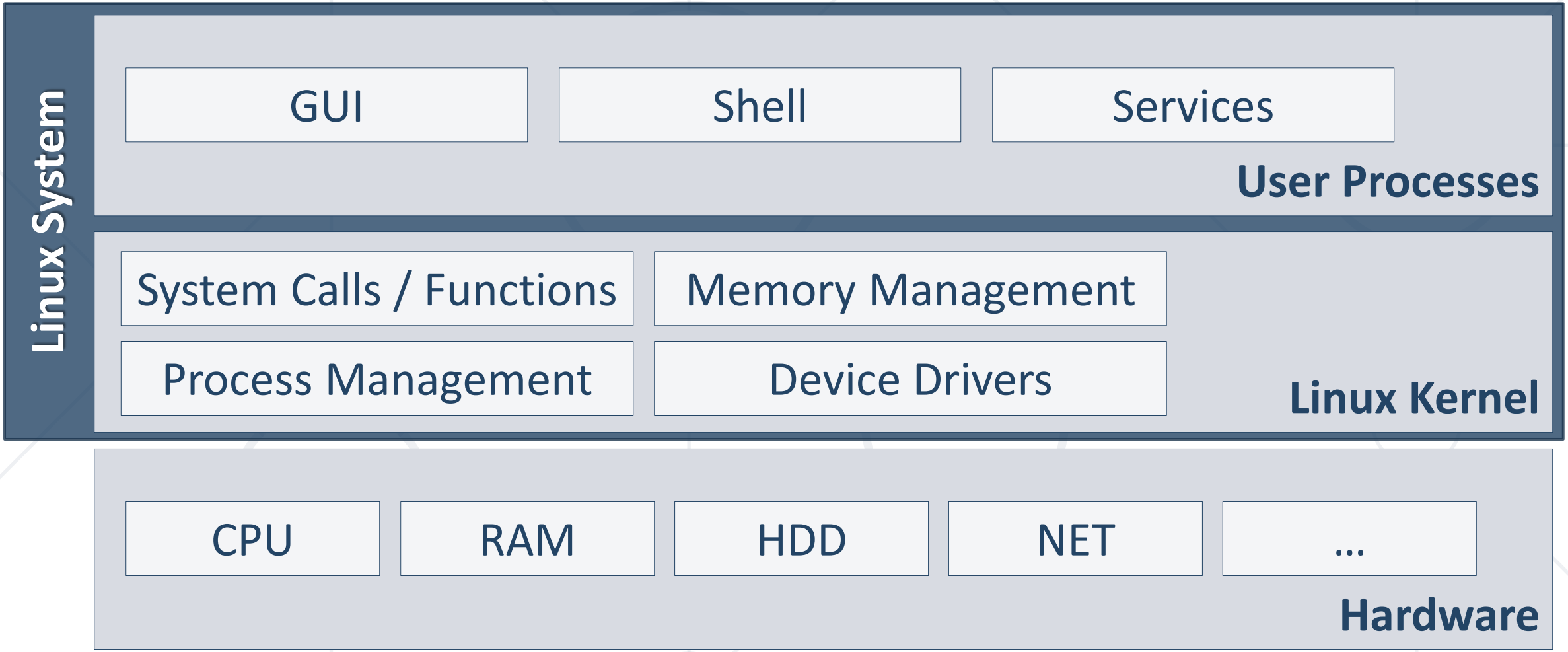
- Kernel
  - The **core** of the OS and **manages** the hardware
- Kernel Space
  - The **area of memory** that **only** the kernel can access
- Process
  - The **running programs** that the kernel manages
- User Space
  - The main **memory allocated** by the kernel for **user processes**

Linux Users and Groups

Software University

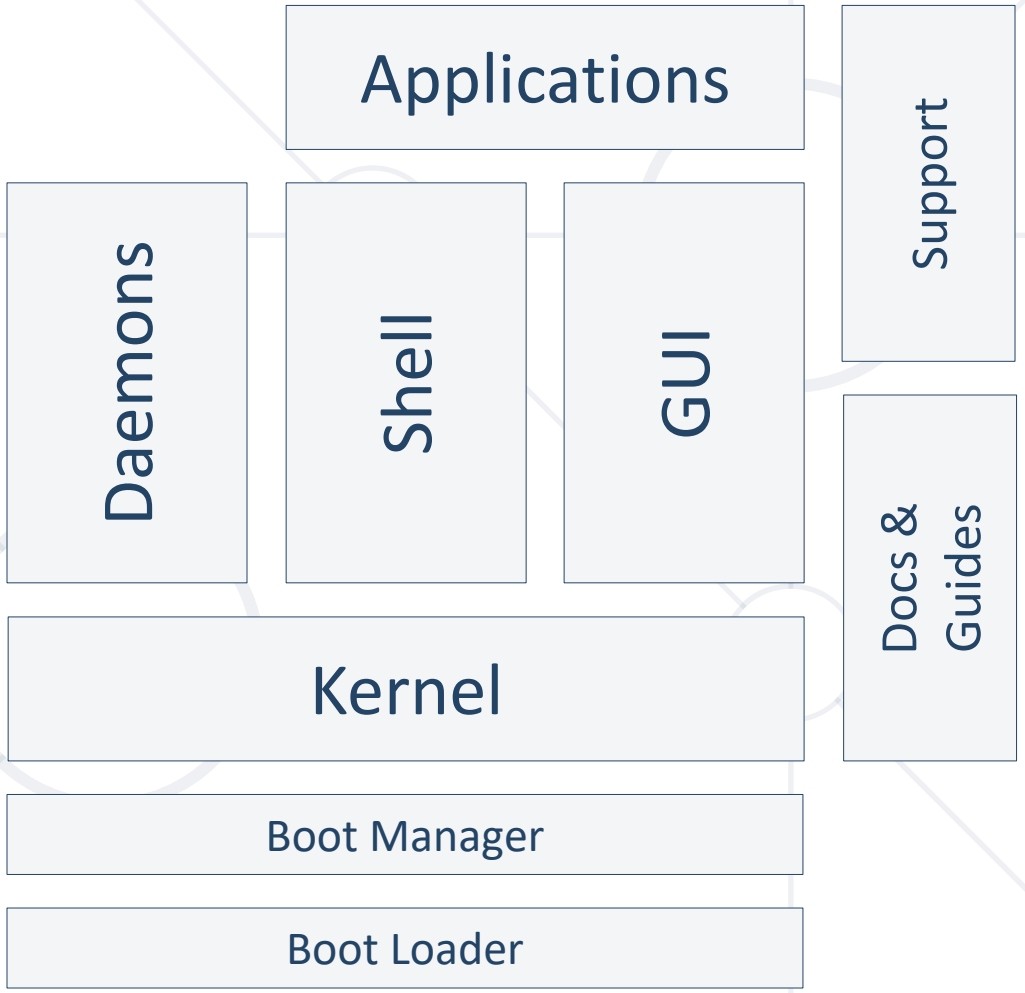
- Users
  - Entity that can **run processes** and **owns files**
  - Support **permissions** and **boundaries**
  - **Superuser** (**root**), **system** users, and **regular** users
- Groups
  - **Set** of users
  - Users in a group can **share** file access

Linux System Architecture

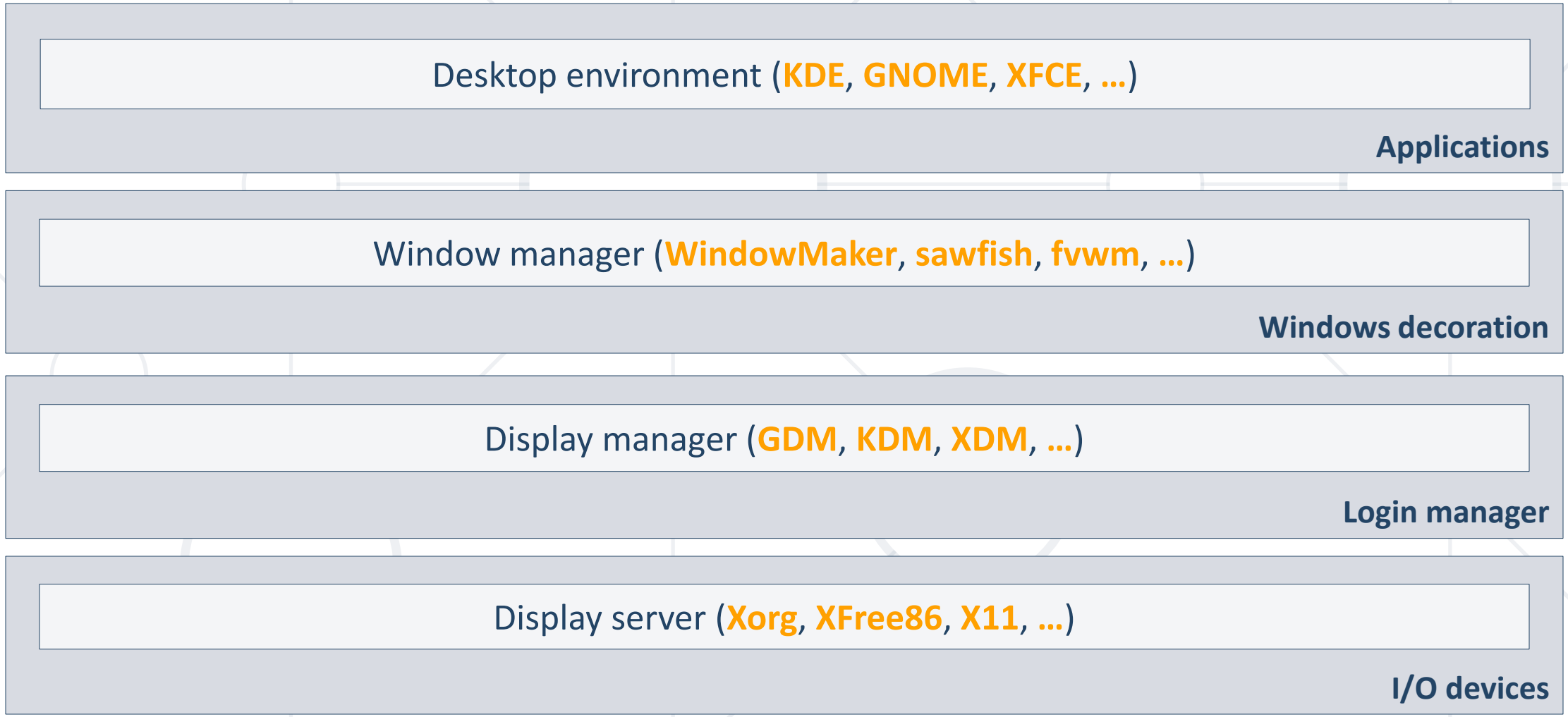


Linux Distribution

- System components
  - Boot Loader
  - Boot Manager
  - Kernel
- User components
  - Daemons
  - Shell
  - Graphical Environment
  - User Applications
- Documentation and Support



Linux Graphical Environment



Distribution Families

Criteria. Families. Our Choice

Family Criteria and Specifics

- Criteria *(to unite them as a family)*
  - Share the **same code** base
  - Same **package format** and **package management system**
  - Members are **derivatives**
- Specifics *(to distinguish them as members)*
  - Different **Release** / **Life** cycles
  - Different **purpose**
  - **Community** / **Commercial** support

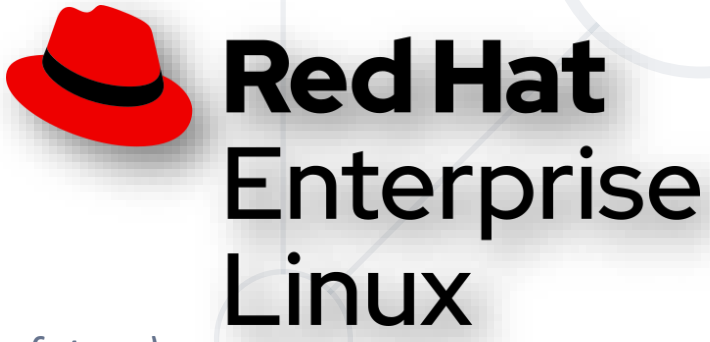
Distribution Families – Debian *

- **Debian** (*https://www.debian.org/*)
  - **Knoppix** (*http://www.knoppix.org/*)
  - **Kali Linux** (*https://www.kali.org/*)
  - **Deepin** (*https://www.deepin.org/en/*)
  - **SteamOS** (*http://store.steampowered.com/steamos/*)
  - **Ubuntu** (*https://www.ubuntu.com/*)
    - **KUbuntu, LUbuntu, XUbuntu**
    - **Linux Mint** (*https://linuxmint.com/*)
    - **ElementaryOS** (*https://elementary.io/*)

* Partial list

Distribution Families – Fedora/Red Hat *

- **Fedora** (*https://getfedora.org/*)
  - **CentOS Stream** (*https://www.centos.org/*)
    - **Red Hat Enterprise Linux** (*https://www.redhat.com/en*)
      - **AlmaLinux OS** (*https://almalinux.org/*)
      - **Oracle Linux** (*https://www.oracle.com/linux/*)
      - **Rocky Linux** (*https://rockylinux.org/*)



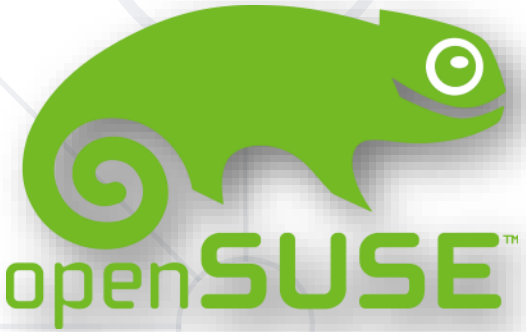
* Partial list *(and it will change in the near future)*

Distribution Families – openSUSE/SUSE

Software University

- **openSUSE Tumbleweed**
  - openSUSE Krypton*
  - openSUSE GNOME Next*
  - **openSUSE Leap (Micro)**
  - **openSUSE MicroOS**
  - openSUSE Argon*

https://www.opensuse.org/



- **Gecko Linux** (*https://geckolinux.github.io/*)
- **SUSE Linux Enterprise** (*https://www.suse.com/*)

* Distributions for enthusiasts and developers that want to test the latest possible software

Distribution Families – Others

- **Mandriva Linux** (ex-Mandrake, discontinued)
  - **OpenMandriva** (*https://www.openmandriva.org/*)
  - **Mageia** (*https://www.mageia.org/en/*)
  - **PCLinuxOS** (*https://www.pclinuxos.com/*)
  - **ROSA Linux** (*https://www.rosalinux.ru/*)
- **Arch Linux*** (*https://www.archlinux.org/*)
- **Gentoo*** (*https://www.gentoo.org/*)
- **Slackware*** (*http://www.slackware.com/*)

* They have their own families with derivative distributions

Our Distributions of Choice

- **Any of** (stick to one family and one distribution)
  - **AlmaLinux OS 9.x** (**CentOS Stream**, **Fedora**, **OracleLinux**, **Rocky Linux**)
  - **openSUSE Leap 15.x** (**openSUSE Tumbleweed**)
  - **Debian 12.x** (**Ubuntu Server 22.04** or **24.04**)
- Reasons
  - Part of an **enterprise** supported families
  - **Huge** install base
  - Often seen (at least their enterprise siblings) in **job announcements**
  - **Certification** tracks
- Alternatives – explore ***https://distrowatch.com/*** for more

How to Obtain Linux Distribution

Software University

- **Free** to **download** and **redistribute**
- Download options
  - Direct download using **HTTP** / **FTP** usually from the **closest mirror**
  - **Torrents** (are good)
- Download artefacts
  - **Live media** or **install media** (CD / DVD / USB)
  - Different **size** (net, boot, minimal, standard, everything, …)
  - Different **purpose** (server, desktop, workstation, …)
  - Different **architecture** (i386, x86_64 *(amd64)*, aarch64 *(arm64)*, …)

AMD64 ≠ ARM64

A Word On Versions and Release Cycles

Software University

- **Release cycles**
  - **Version-based distributions** have new release on a fixed term and there is a change in either the minor or major version number
  - **Rolling distributions** do not have versions at all and there is no schedule for new releases

A Word On Versions and Release Cycles

- **Versions**
  - Some have **single** version number, while others have **major** and **minor**

Fedora 41

Ubuntu 24.04

- **Support period**
  - **Long term support** (**LTS**) versions are supported for 3, 5 or more years
  - **Short term support** (**STS**) versions are supported usually until the next release or some time after that

Linux Ecosystem

Regulations and Organizations

(Some) Major Organizations in Linux World

Software University

- **Free Software Foundation (FSF)**
  - Mission: Promotes computer user **freedom**. Sponsors **GNU**
  - URL: ***http://www.fsf.org/***
- **The Linux Foundation**
  - Mission: Project incubator. **Performance** based certifications
  - URL: ***https://www.linuxfoundation.org/***
- **Linux Professional Institute (LPI)**
  - Mission: Distribution **agnostic** certifications
  - URL: ***http://lpi.org/***

Linux Regulations

- **Linux Standard Base (LSB)**
  - Purpose: Standardize the **software system structure**
  - Maintainer: **The Linux Foundation**
  - URL: ***https://refspecs.linuxfoundation.org/lsb.shtml***
  - Version: **5.0**
- **File System Hierarchy Standard (FHS)**
  - Purpose: Defines the directory **structure** and directory **contents**
  - Maintainer: **The Linux Foundation**
  - URL: ***https://refspecs.linuxfoundation.org/FHS_3.0/fhs/index.html***
  - Version: **3.0**

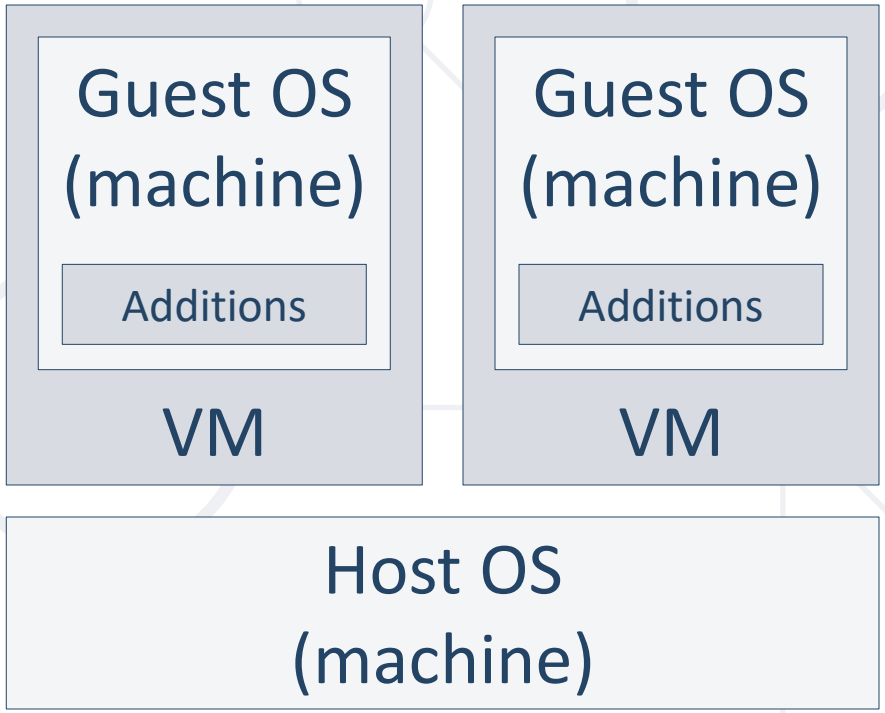
Time For a Break

Introduction to Virtualization

Fundamental Principles and Use Cases

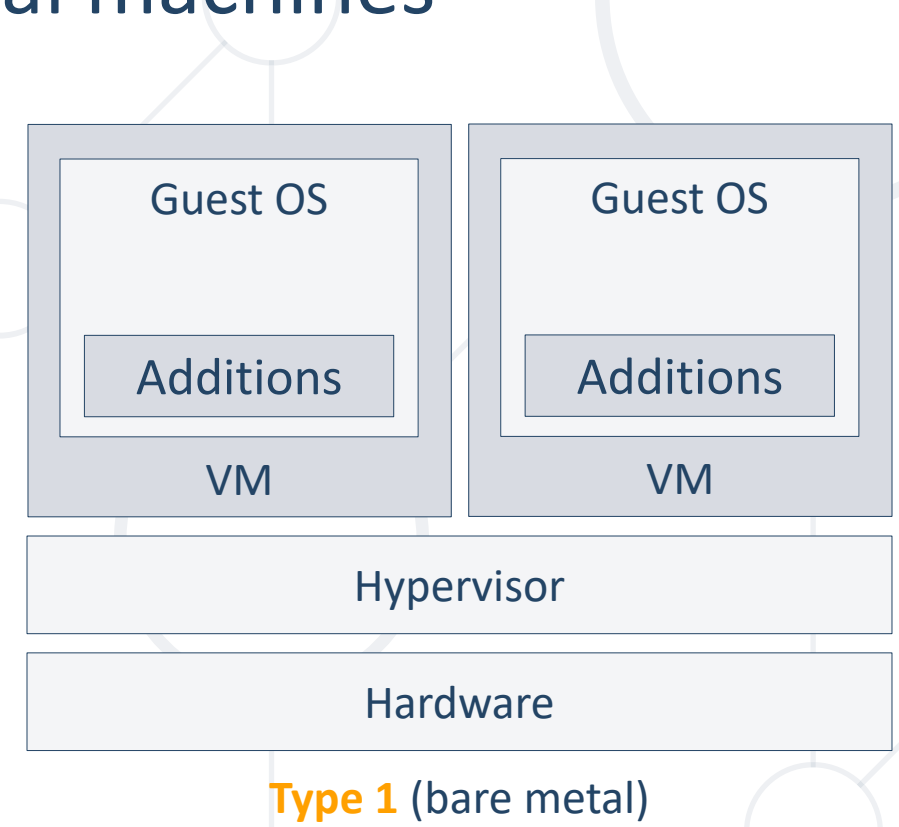
What is Virtualization?

- Virtualization is the **act** of creating a **software-based** or virtual (rather than physical) version of something
- Main definitions
  - Host OS (machine)
  - Virtual machine
  - Guest OS (machine)
  - Guest additions

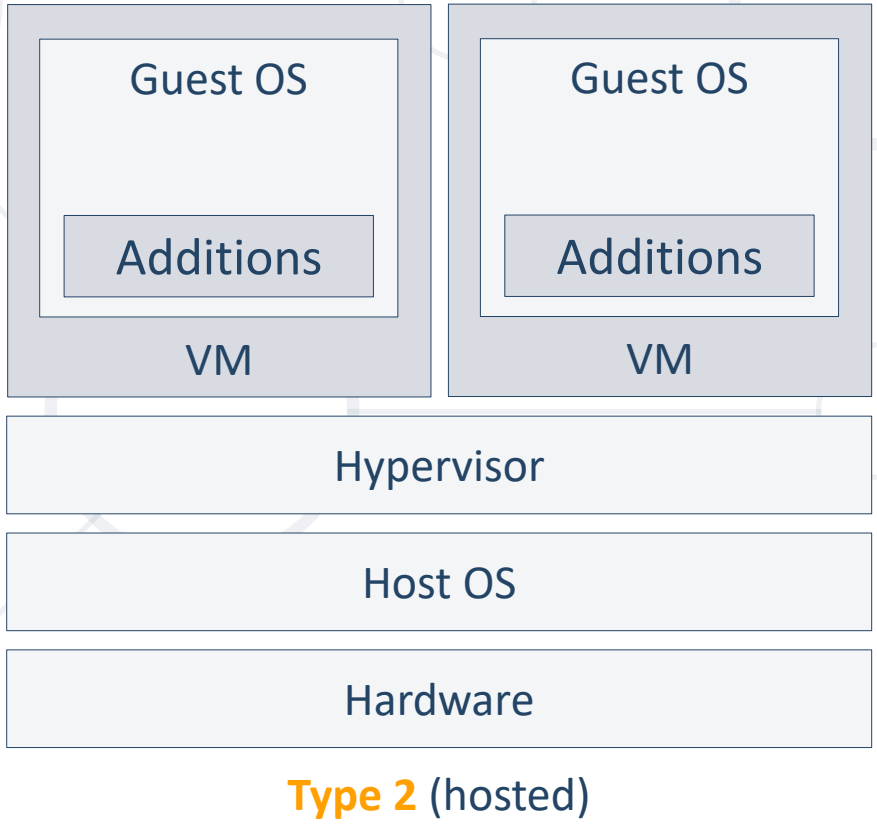


Hypervisors

- A **hypervisor** or **virtual machine monitor** (**VMM**) is computer software, firmware, or hardware, that creates and runs virtual machines



Type 1 (bare metal)



Type 2 (hosted)

Use Cases

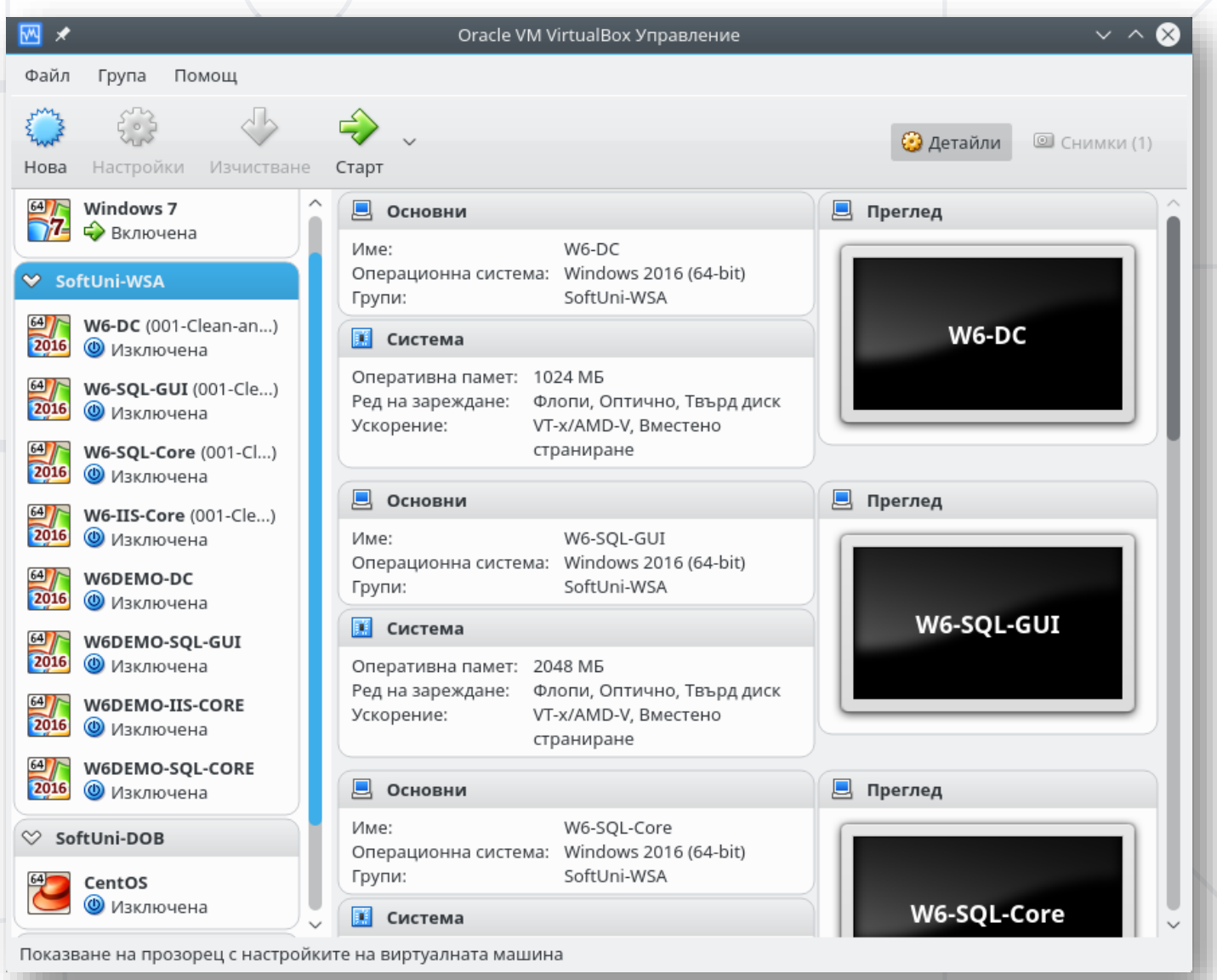
- Infrastructure **consolidation**
  - Better usage and utilization of the available hardware
- Maintain **separate environments**
  - For example – development, test, production
- **Testing** and **evaluation**
  - Test a newer software version or evaluate a product
- **High availability** and **disaster recovery**

Our Case

- We would like to
  - Install **multiple** machines on **limited** hardware resources
  - Manage their **isolation**
  - Manage their **state** – our own time-machine
  - **Move**, **export**, and **import** them
  - **Clone** them – create **multiple copies** out of one master
- The answer is **Virtualization**

Oracle VirtualBox

- Cross-platform
- Broad guest OS support
- Easy to install
- Simple GUI
- Automation options
- Free



A Note on Virtualization

Software University

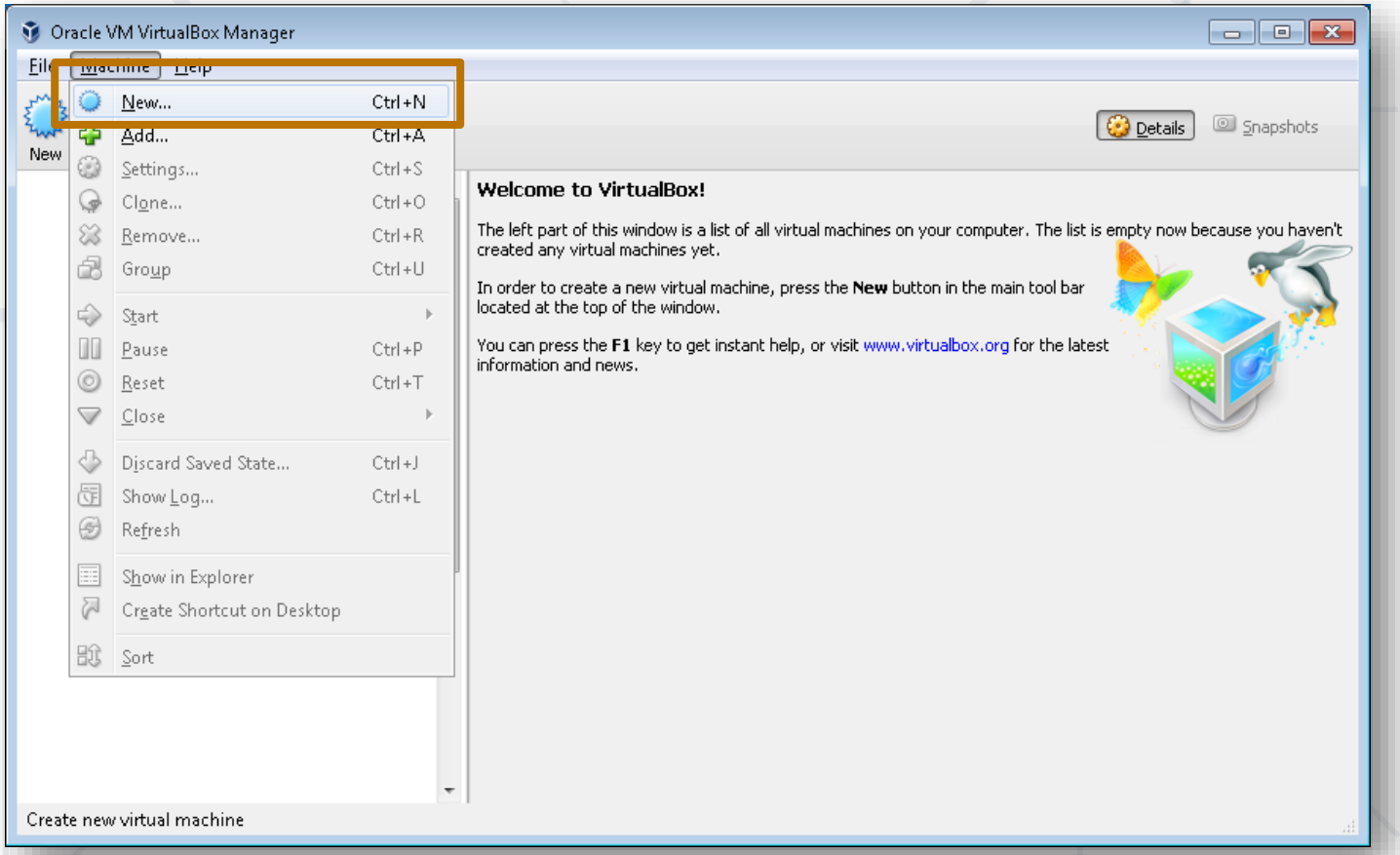
- Yes, there are many options. Some of them are:
  - **Oracle VirtualBox**
  - **VMware Workstation** (Player / Pro) / **VMware Fusion** (Player)
  - **Hyper-V**
  - **VMware ESXi**
- You can experiment with all of them and select the one you like
- Be sure that there is only one installed at any given moment
- Basic principles and actions are the same no matter the tool

Getting to Know VirtualBox

Main Functionalities, Usage Scenarios

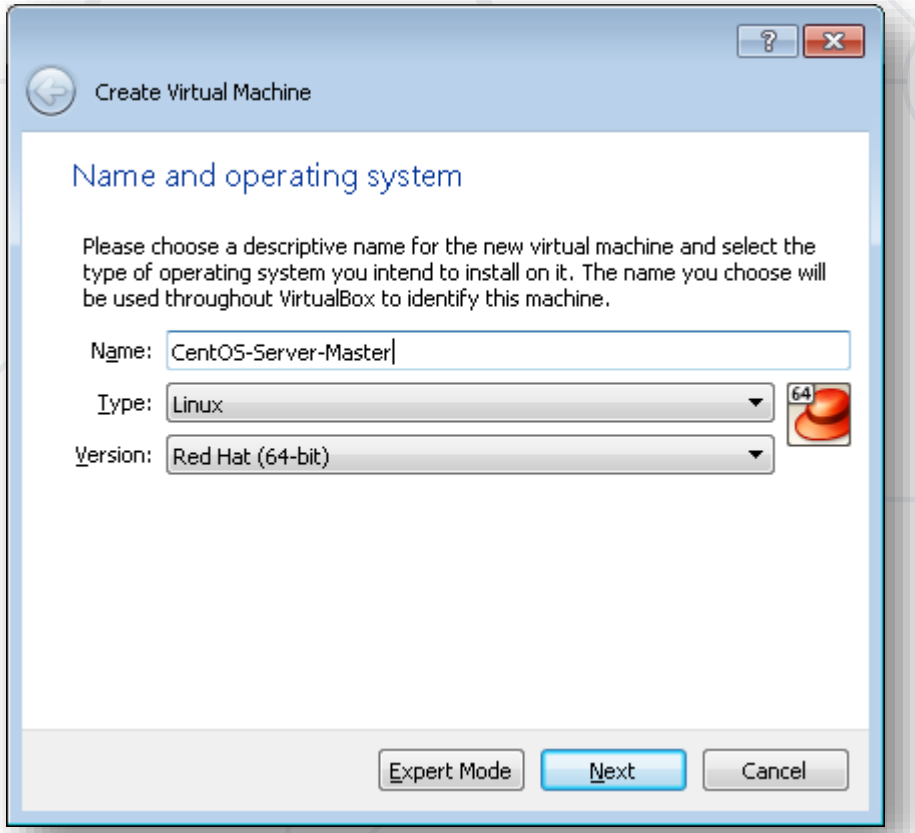
Problem: New Virtual Machine

- Create a new virtual machine
  - Set **correct** parameters
  - Avoid **overcommit**
  - **Provision** hardware



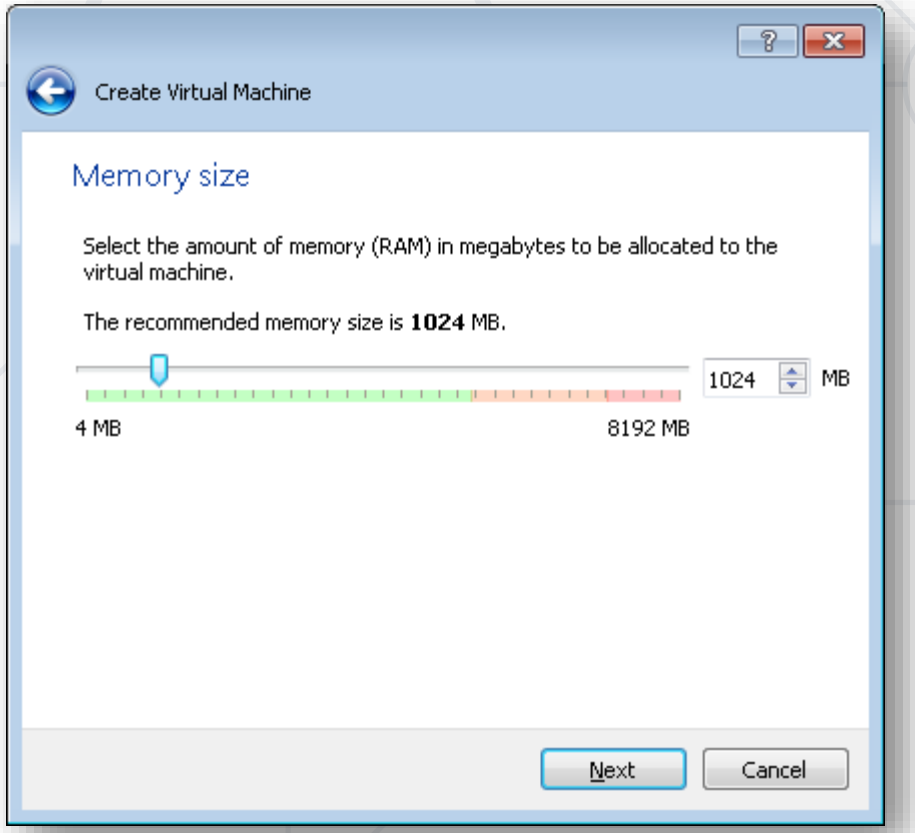
Solution: New Virtual Machine

- Enter **name**, **type**, and **version**



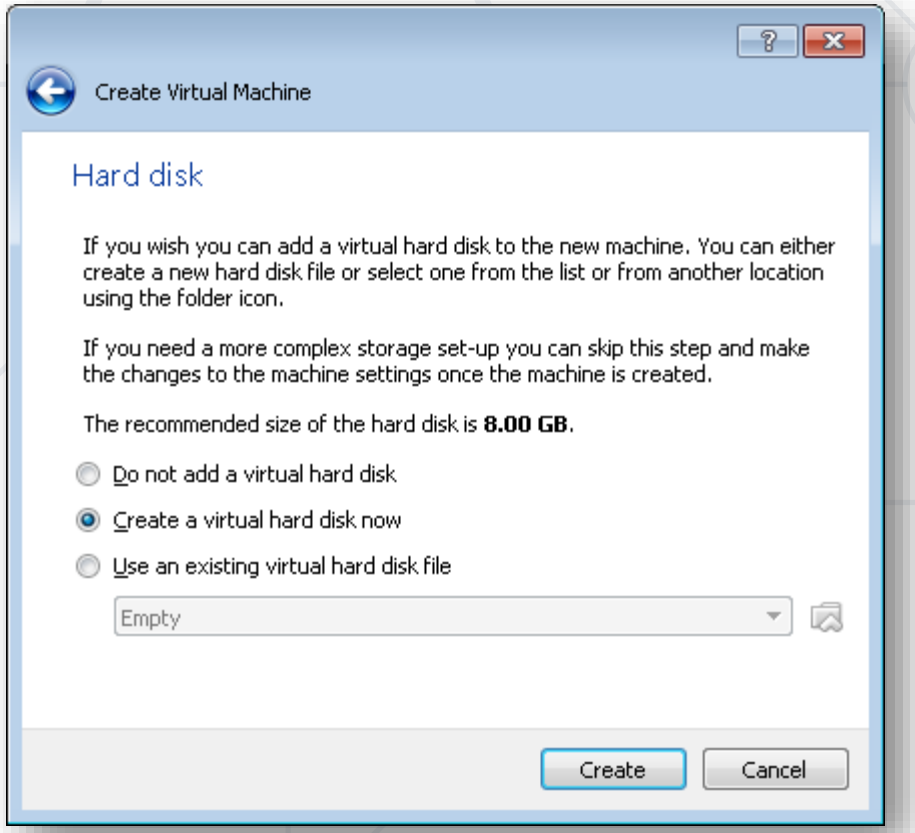
Solution: New Virtual Machine

- Select memory **size**



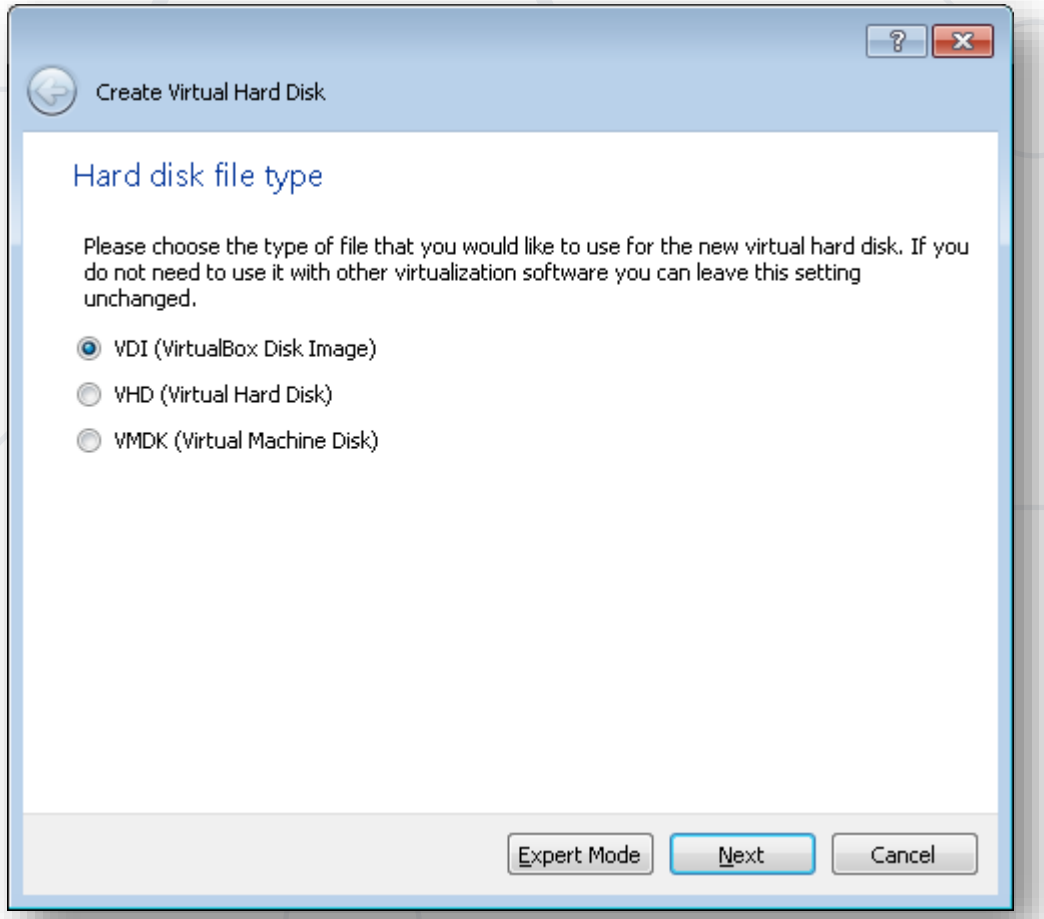
Solution: New Virtual Machine

- Select **existing** or create **new** disk



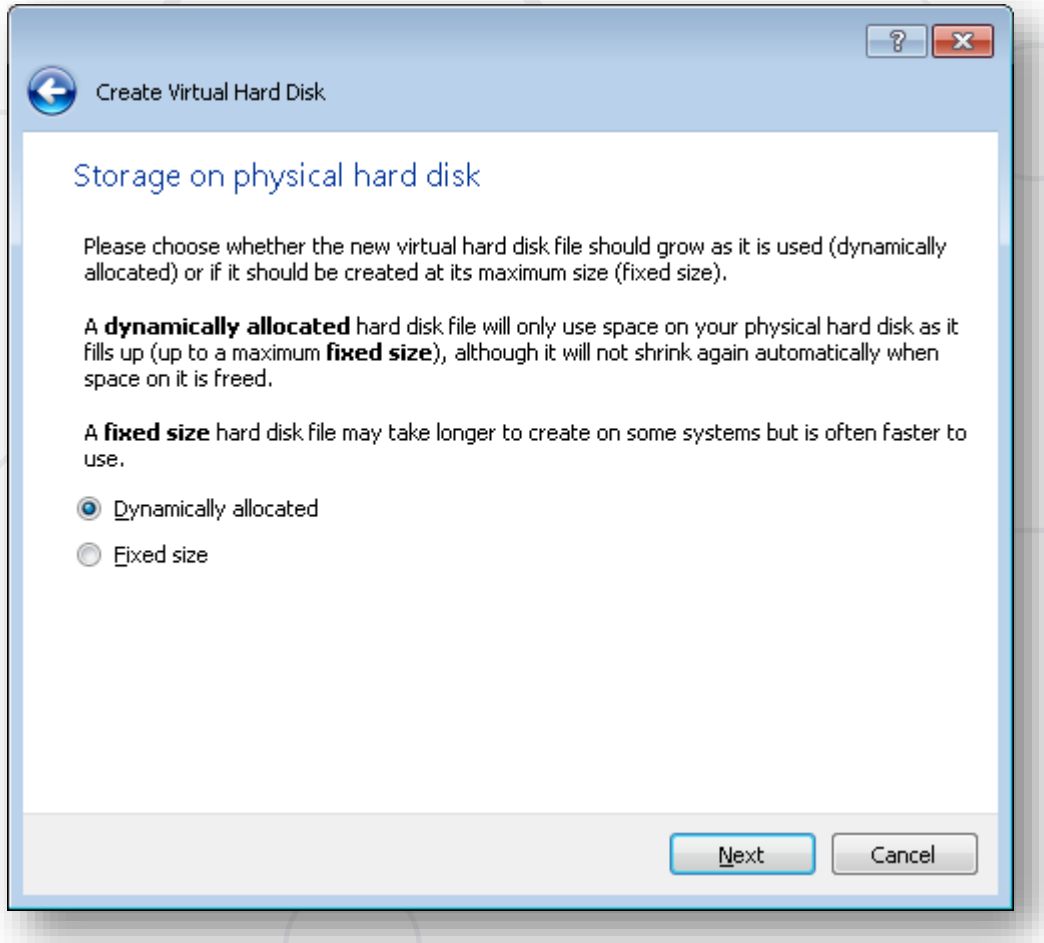
Solution: New Virtual Machine

- Select new **disk type**



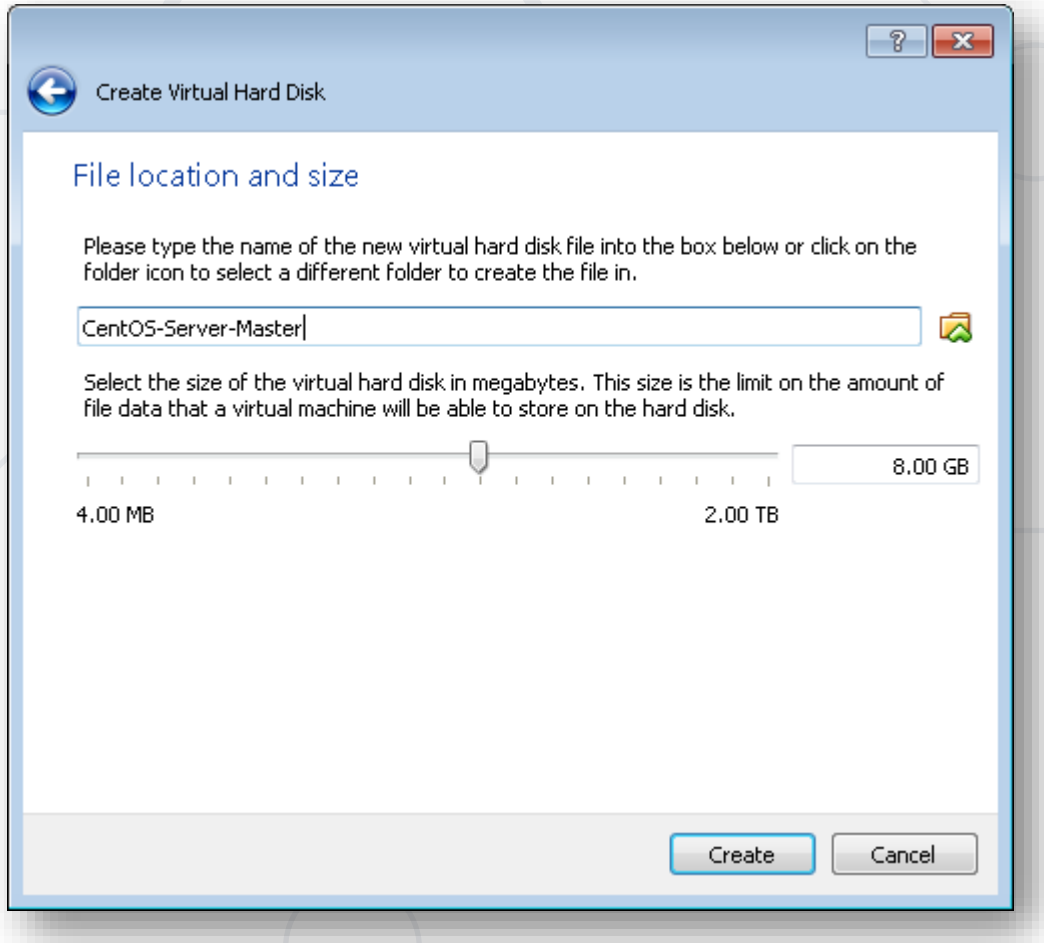
Solution: New Virtual Machine

- Select disk **space allocation** mechanism



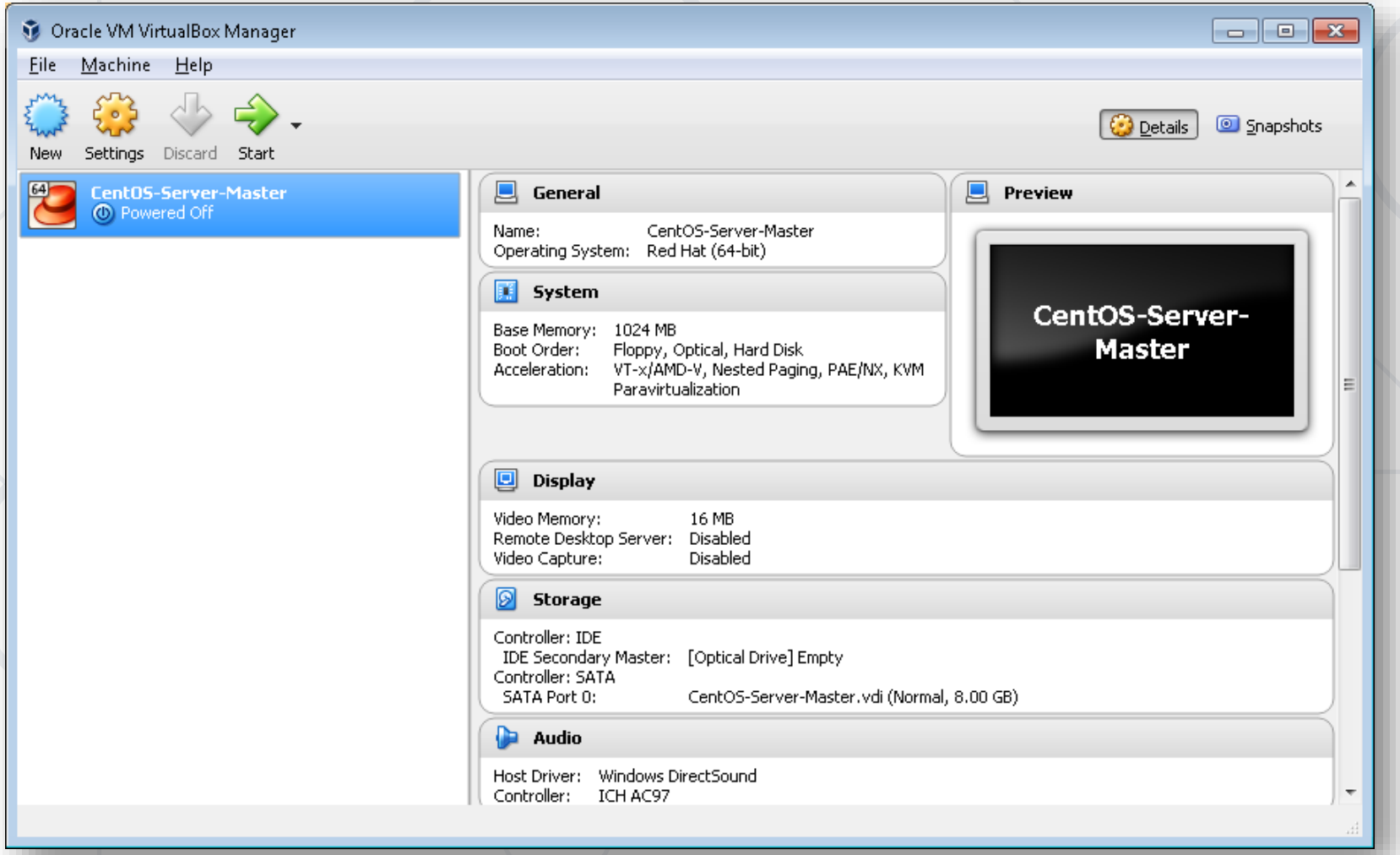
Solution: New Virtual Machine

- Select disk **location** and **size**



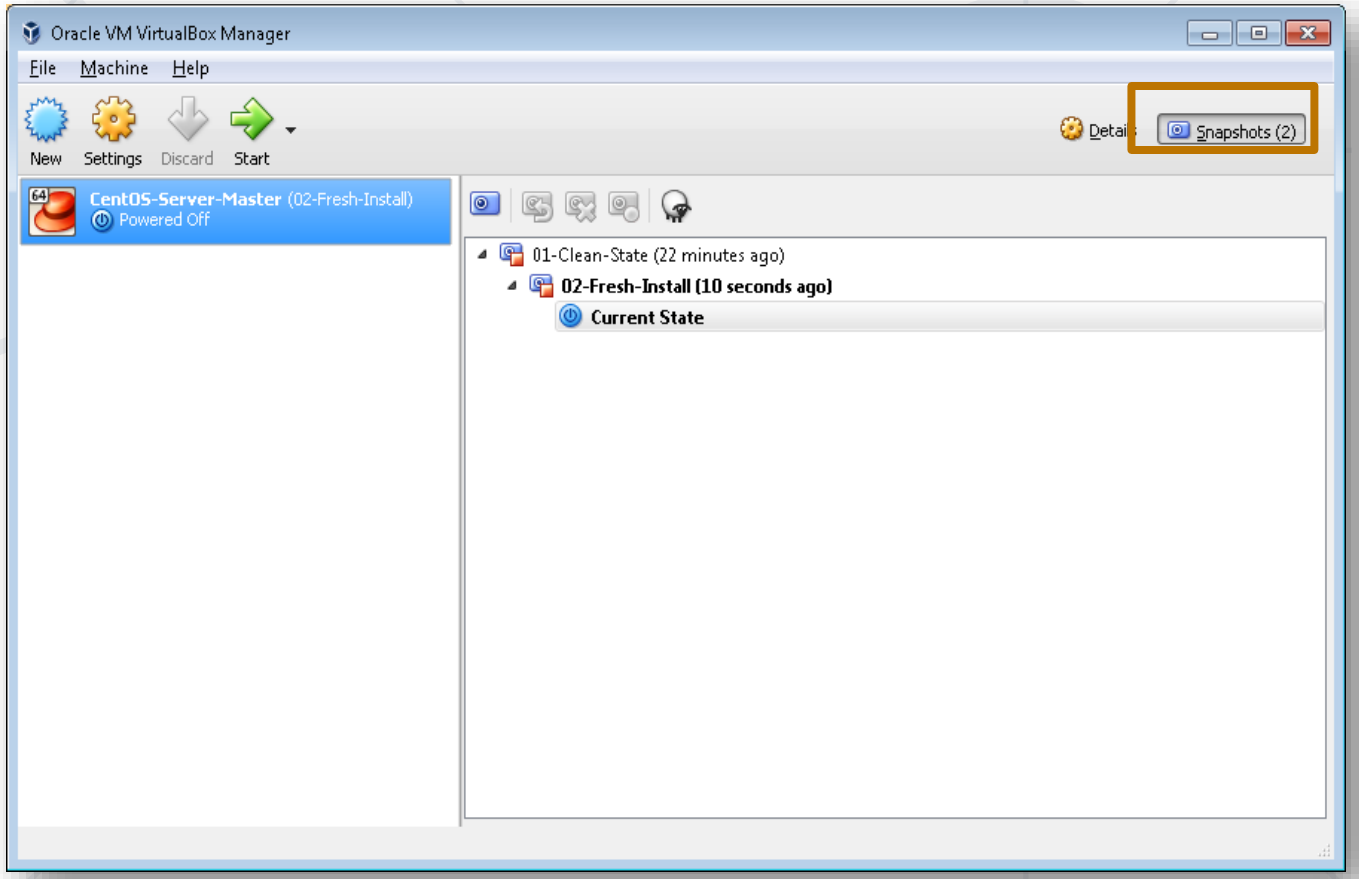
Solution: New Virtual Machine

- Our new virtual machine is created



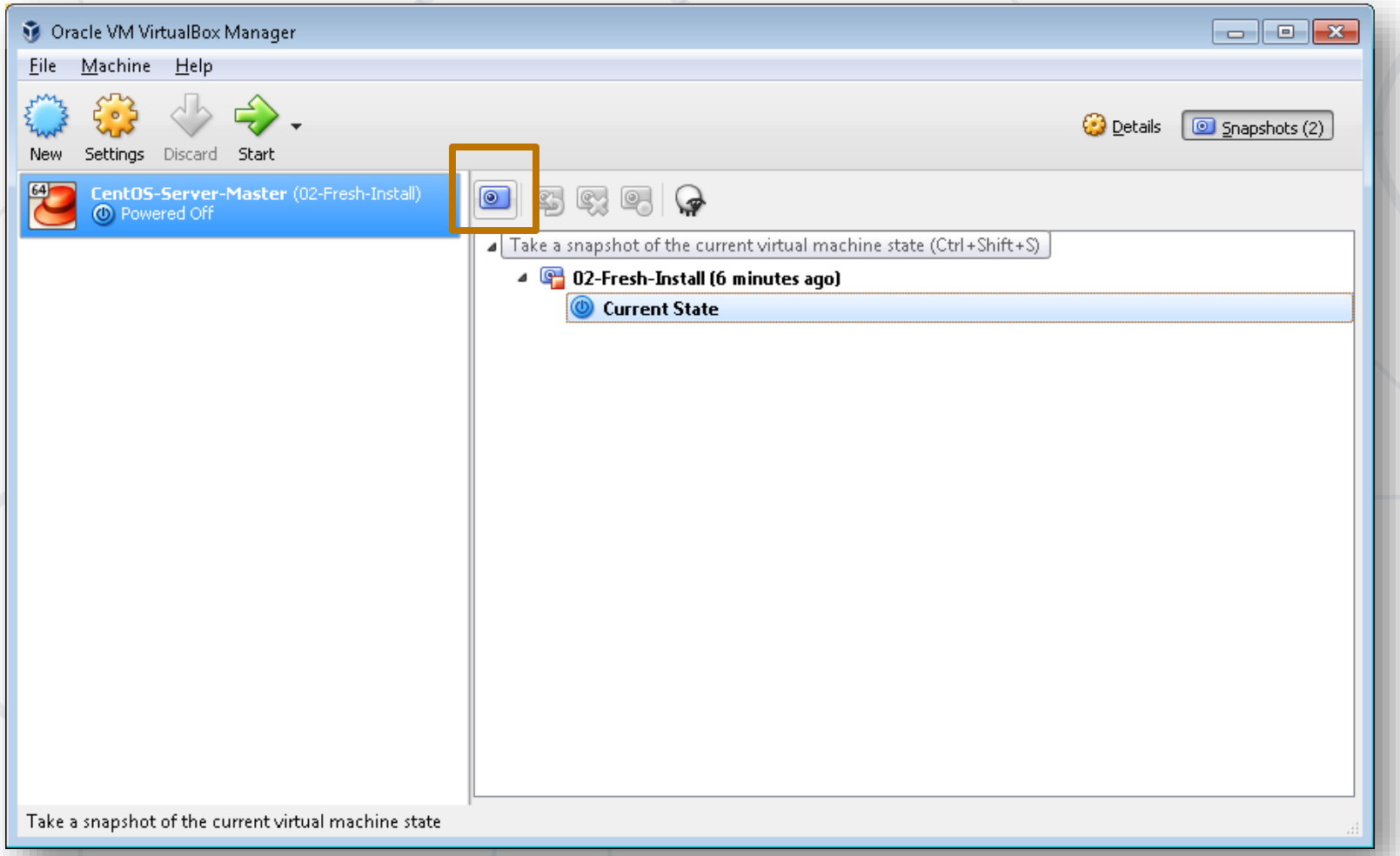
Problem: Track Changes in a VM

- **Move** through machine **states**
  - **Save** current state
  - **Return** to a previous state
  - **More** than one branch



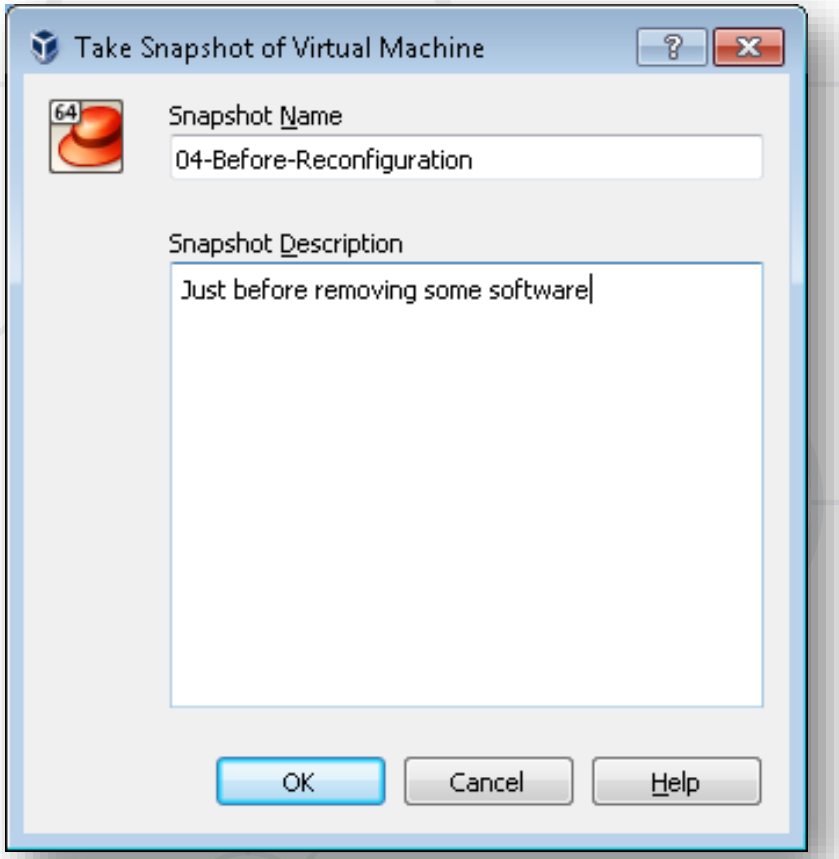
Solution: Track Changes in a VM

- Create snapshot



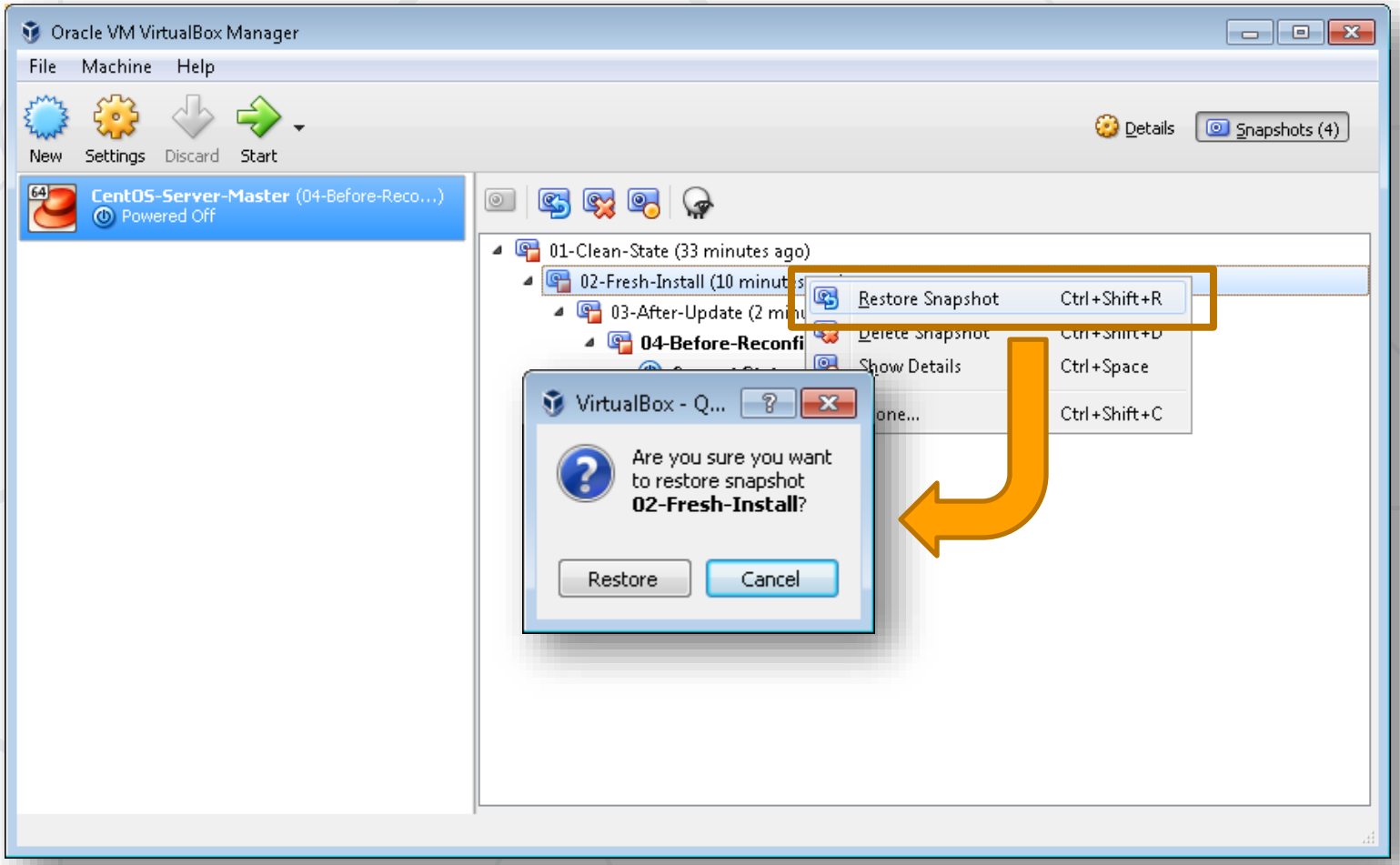
Solution: Track Changes in a VM

- Enter snapshot details



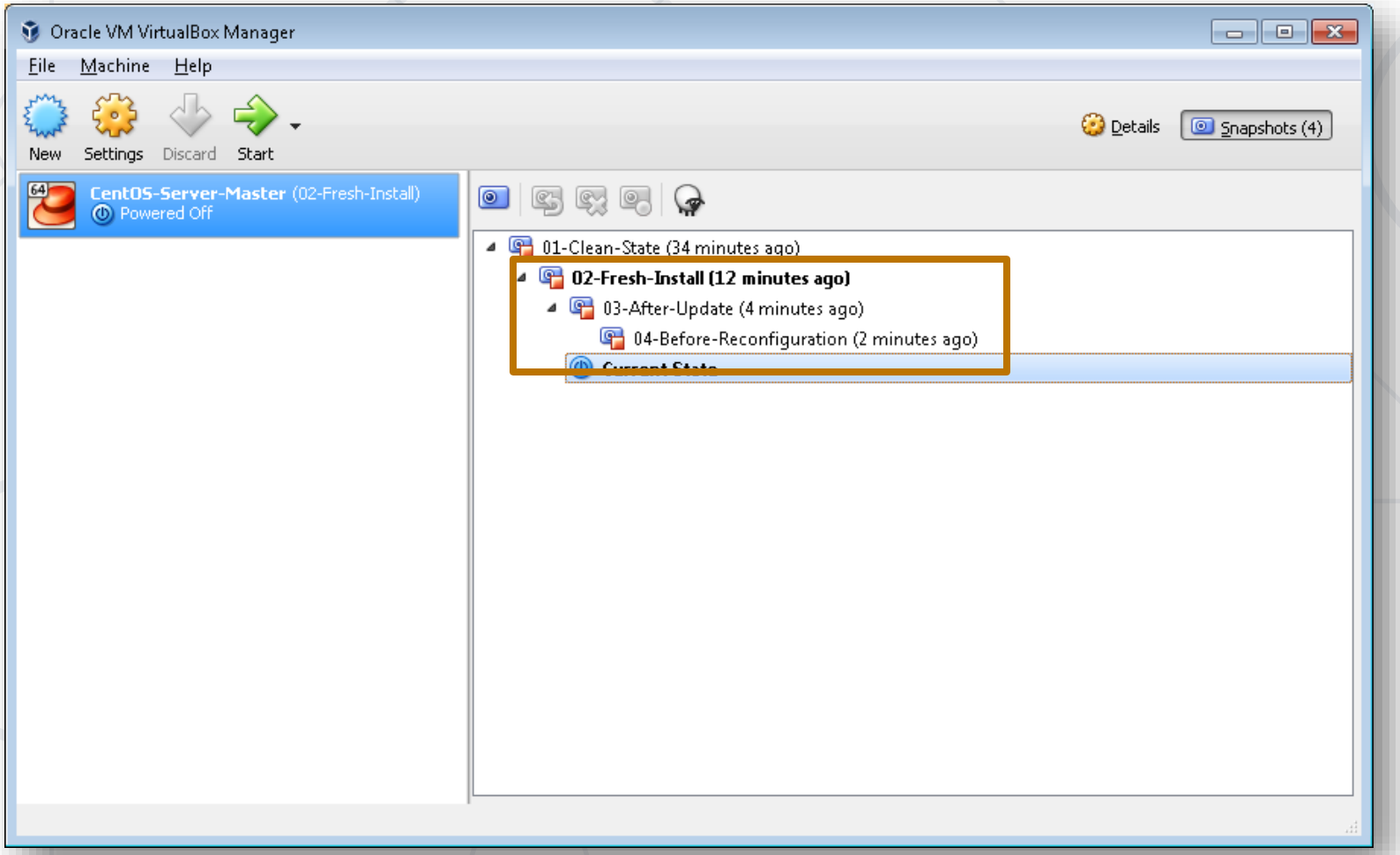
Solution: Track Changes in a VM

- Restore from a snapshot



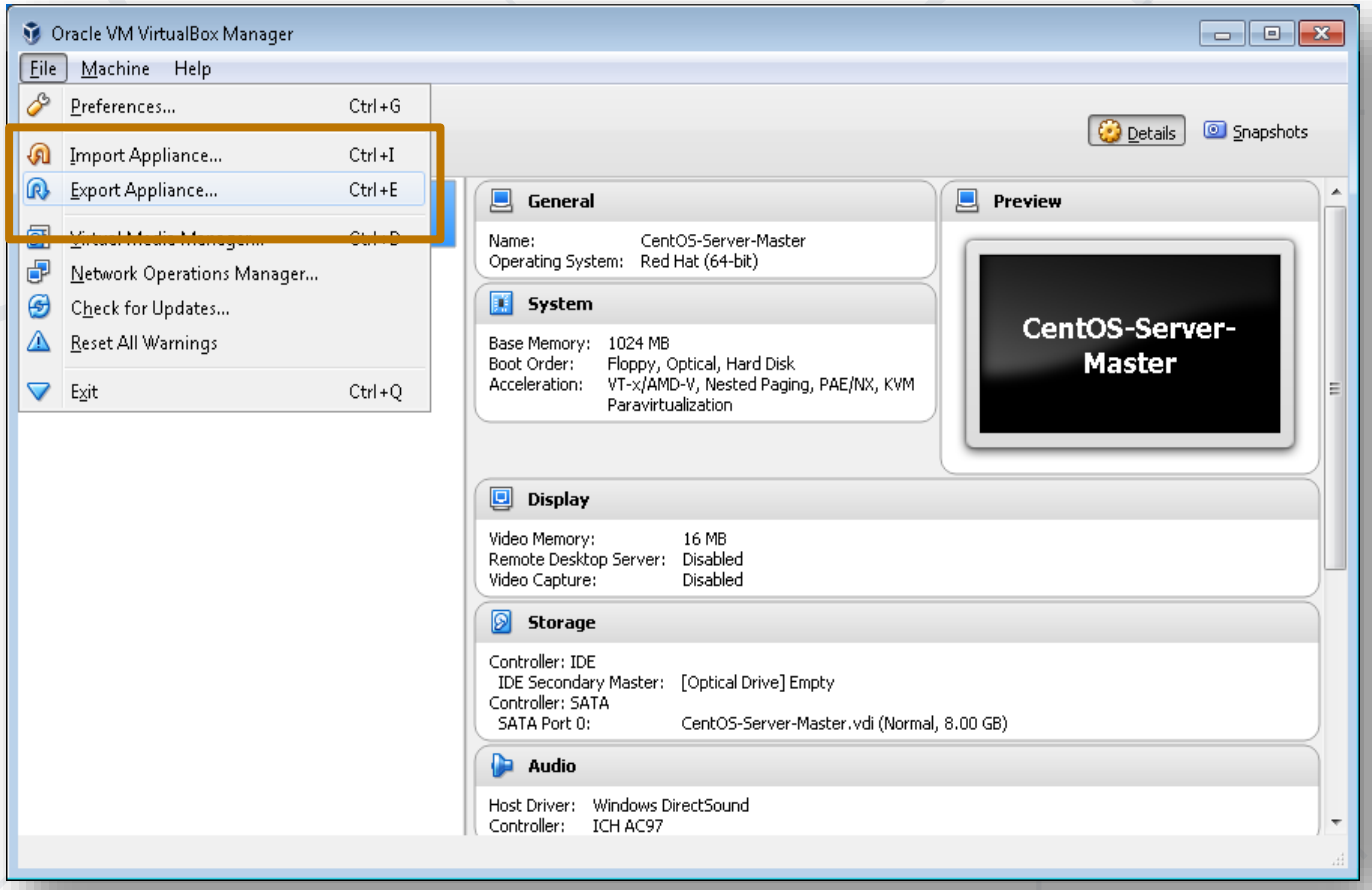
Solution: Track Changes in a VM

- Current state has been switched



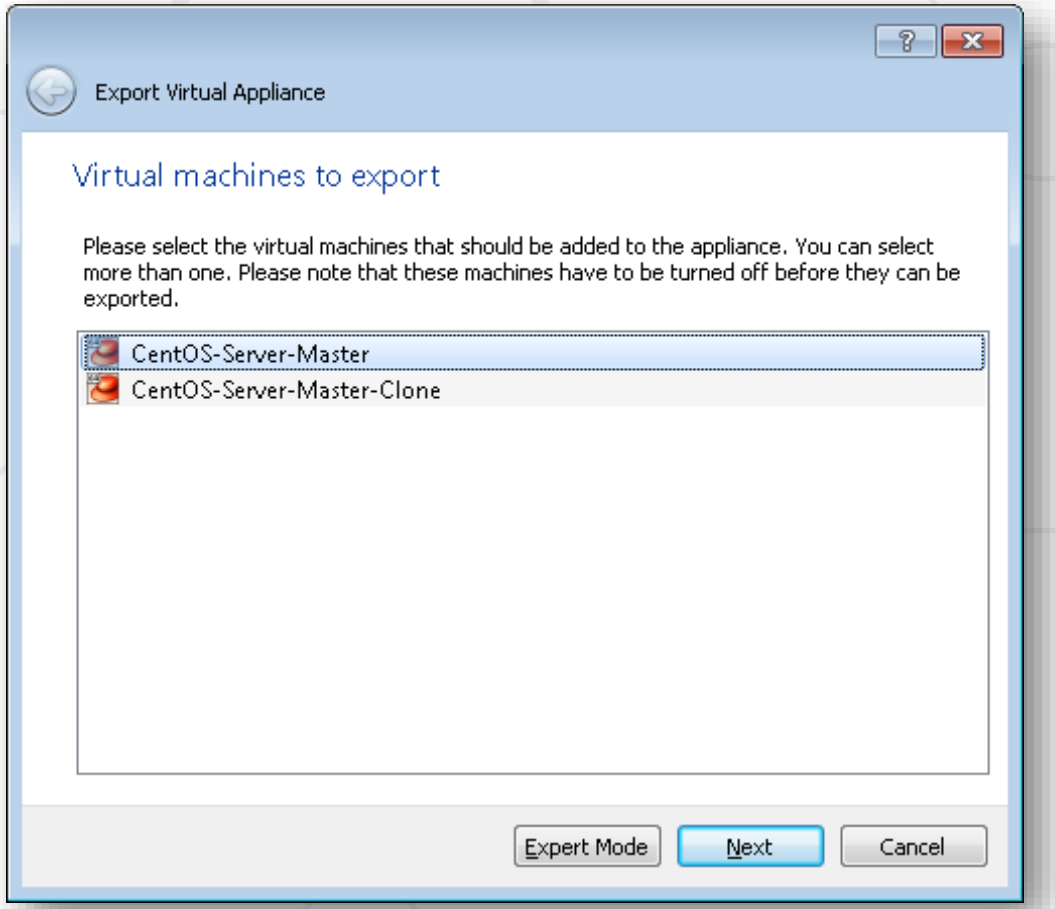
Problem: Import / Export Machines

- Import / export or attach existing machine
  - **Import** machine
  - **Export** machine
  - **Register** existing machine



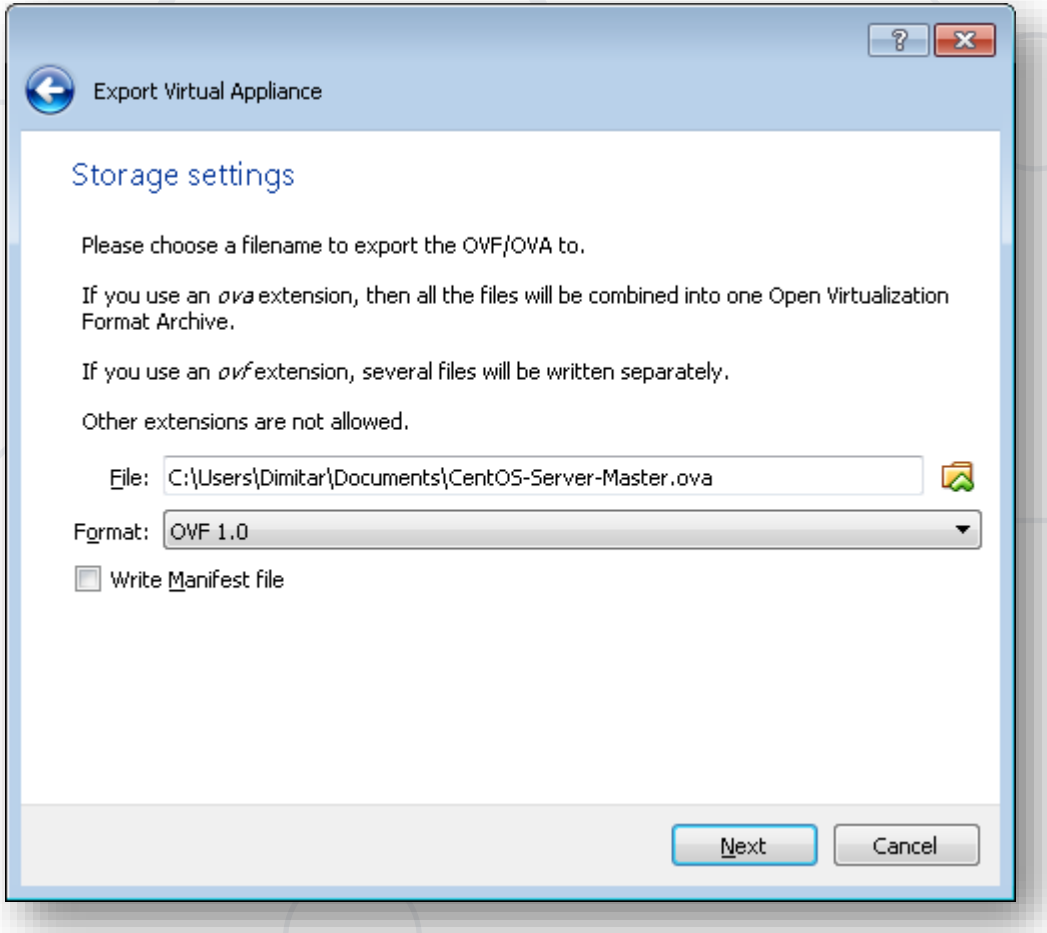
Solution: Import / Export Machines

- Export a machine



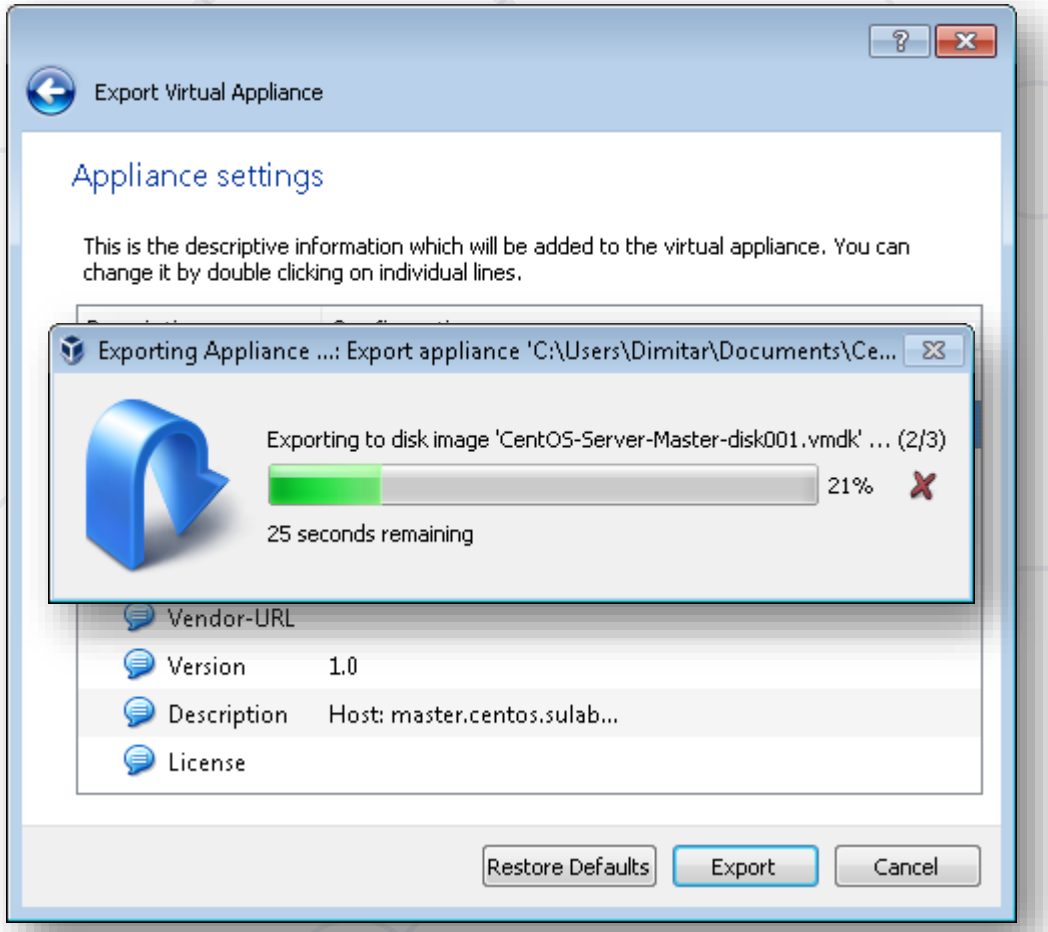
Solution: Import / Export Machines

- Export type and place



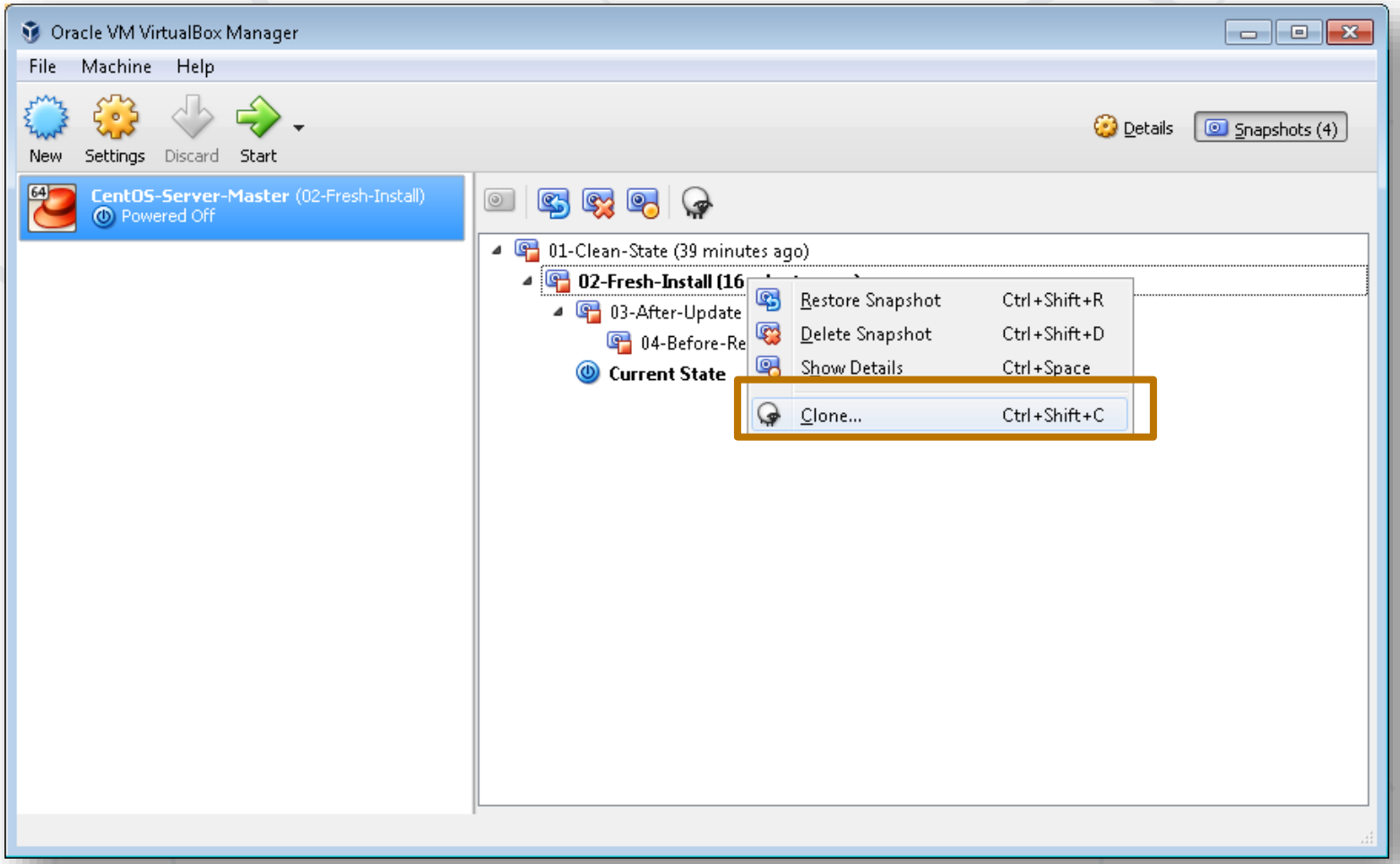
Solution: Import / Export Machines

- Final settings



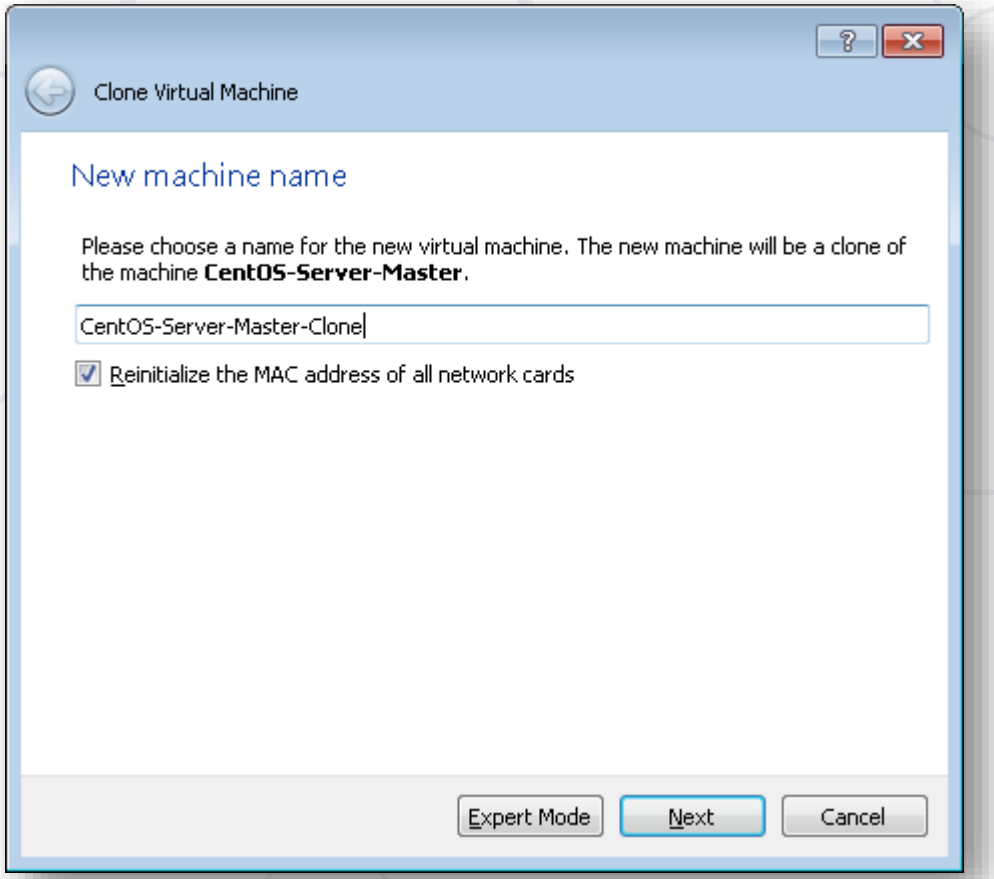
Problem: One Source – Multiple Targets

- Create **copies** of a virtual machine
  - **Stand-alone** copies
  - **Linked** copies



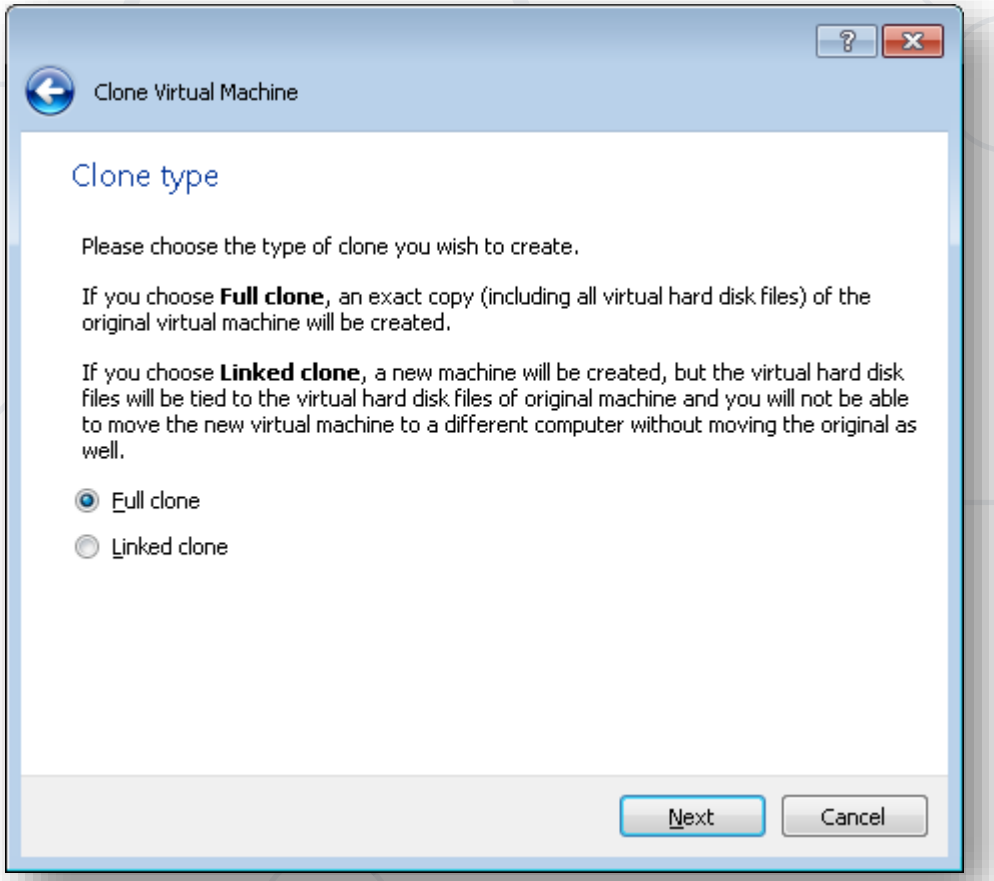
Solution: One Source – Multiple Targets

- Set a name for the new machine



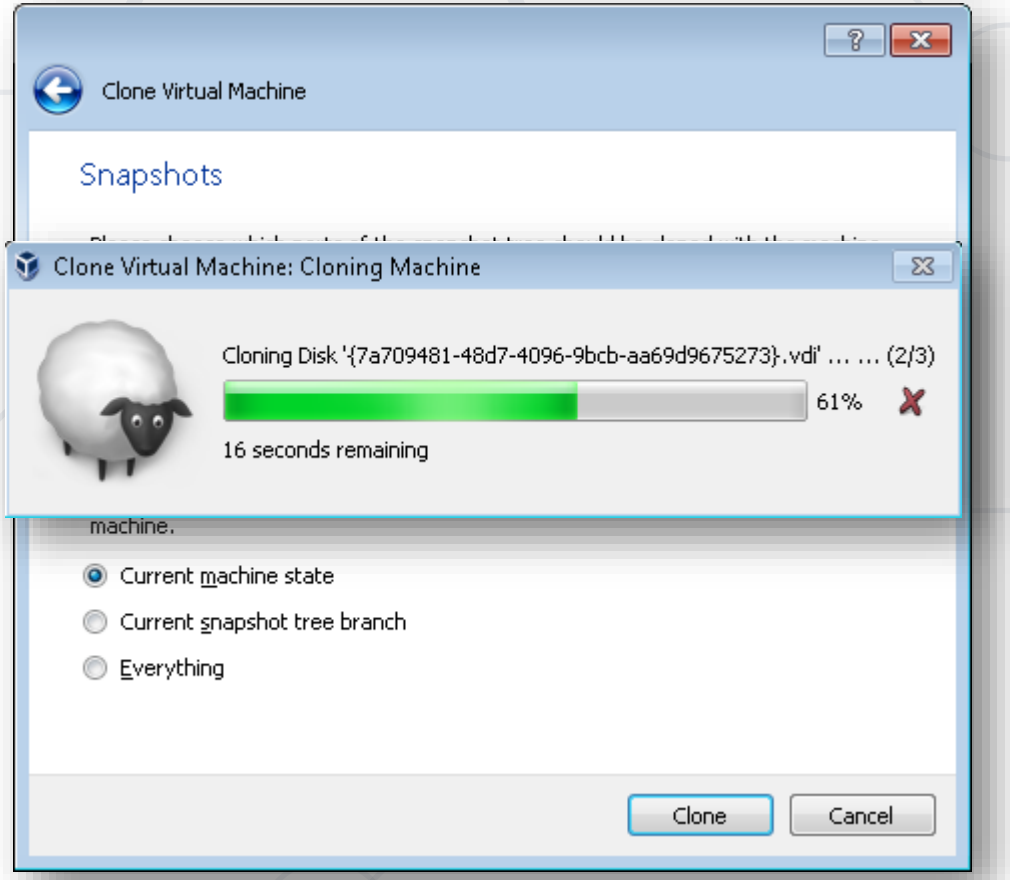
Solution: One Source – Multiple Targets

- Chose a clone type



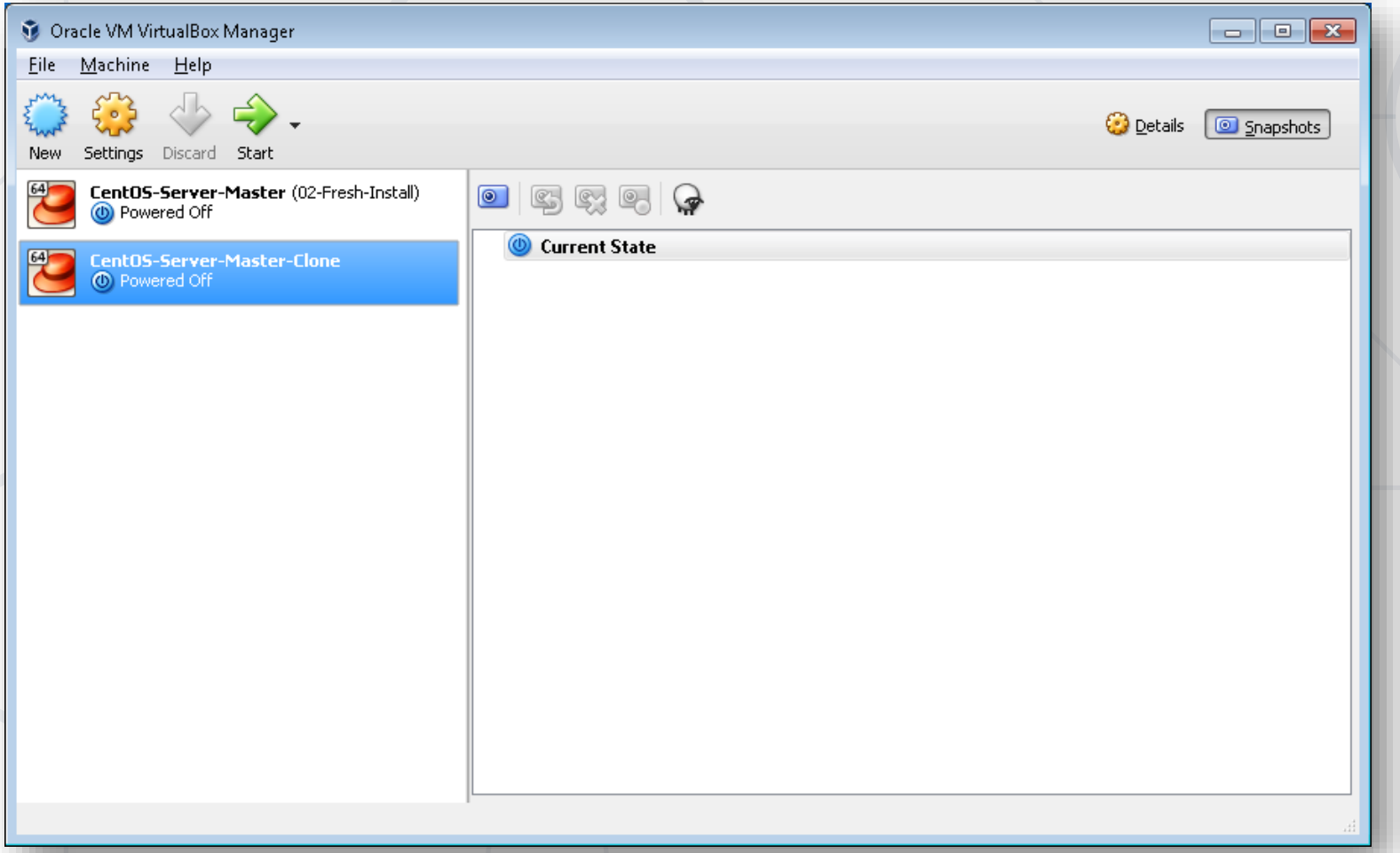
Solution: One Source – Multiple Targets

- Chose which parts to clone

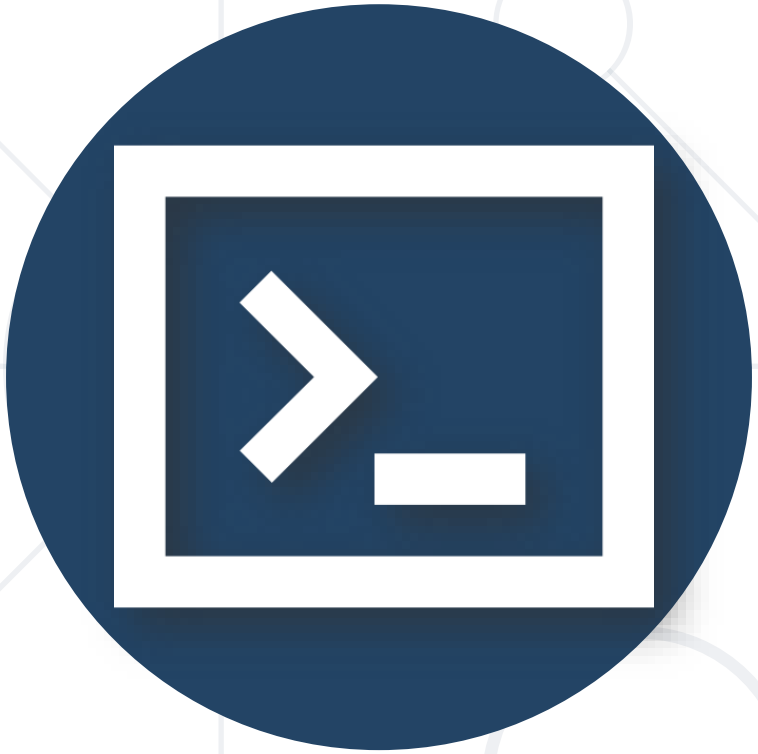


Solution: One Source – Multiple Targets

- Result



Practice

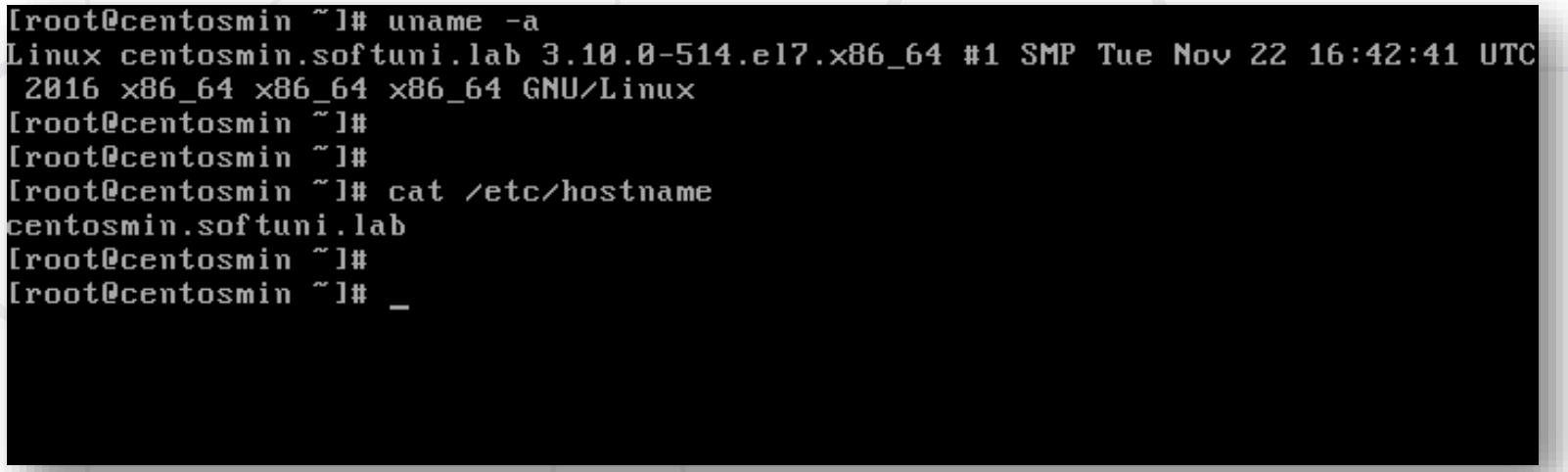


Introduction to Linux Console

Definition and Types

Shell Definition

- Shell = **Command line** interface
- It is a **software** that takes commands and passes them to the operating system
- When in GUI, we use **terminal emulators** to interact with the shell



Shell Types

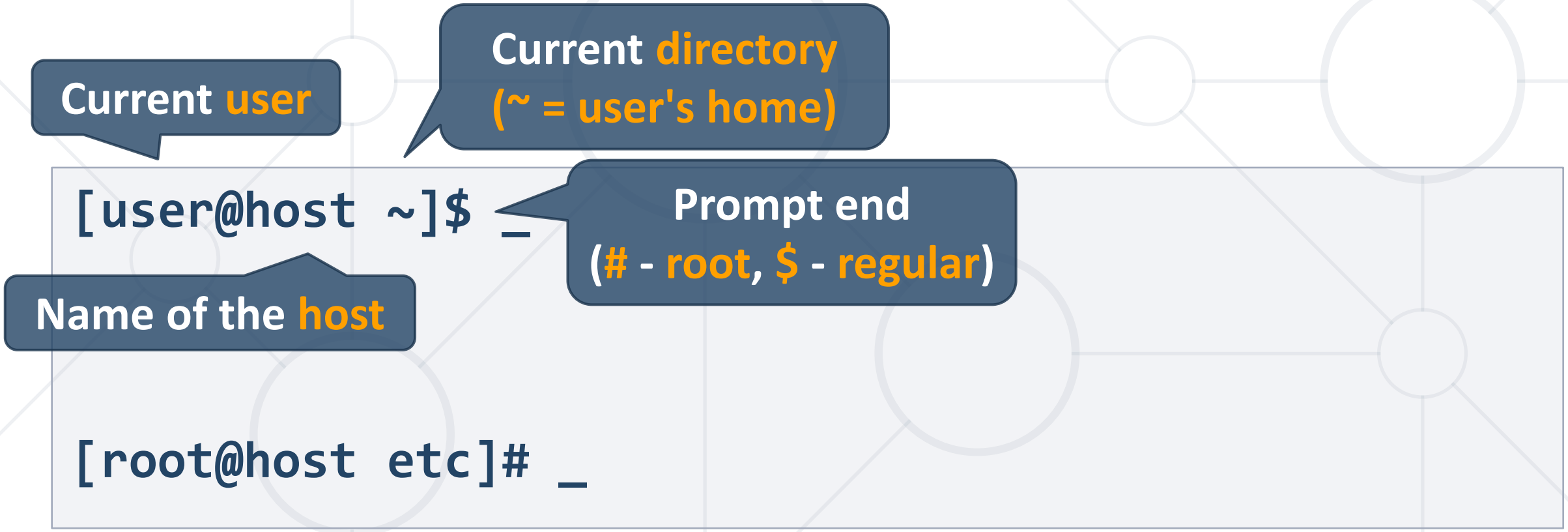
- Plenty of shells
  - **sh** (Bourne Shell) and **bash** (Bourne Again Shell)
  - **csh** (C Shell) and **tcsh** (Enhanced C Shell)
  - **ksh** (Korn Shell)
  - **zsh** based on bash, ksh, and tcsh
- Considering the invocation and interaction
  - **Interactive** and **non-interactive**
  - **Login** and **non-login**

Bash

- Used by **default** on most Linux distributions
- Offers **strong scripting** capabilities
- **Extensive** knowledge base
- Existing ready to use **snippets** and **solutions**

Bash Prompt

- Prompt parts



Environment Variables

Easily Access System Information

Environment Variables

- An environment variable is a **named object** that contains **data** used by **one** or **more applications**
- **Display** a variable's value

```
[user@host ~]$ echo $SHELL
/bin/bash
```

- **List** environment variables

```
[user@host ~]$ printenv
HOSTNAME=host.softuni.lab
SHELL=/bin/bash
...
```

Environment Variables

Variable	Meaning
HOSTNAME	Name of the host
USER	Current user
HOME	Path to the home of the current user
PWD	Path to the current working directory
OLDPWD	Previous value of PWD
SHELL	Path to the user's preferred shell
PATH	List of directories to search for executable files
HISTFILE	Path to the history file
HISTSIZE	Size of the command history

Keyboard Shortcuts

Increase Your Productivity

Moving the Cursor

Software University

Key Combination	Action
<Ctrl>+<A> or <Home>	Move to the beginning of the line
<Ctrl>+<E> or <End>	Move to the end of the line
<Ctrl>+<B> or <Left>	Move left (backward) one character
<Alt>+<B>	Move left (backward) one word
<Ctrl>+<F> or <Right>	Move right (forward) one character
<Alt>+<F>	Move right (forward) one word

Control the Terminal

Key Combination	Action
<Ctrl>+<L>	Clears the screen
<Ctrl>+<C>	Interrupt a foreground running process
<Ctrl>+<Z>	Stop the execution of a process
<Ctrl>+<D>	Closes the terminal
<Tab>	Completes a command or path

Command Line

Key Combination	Action
<Ctrl>+<P> or <Up>	Previous command in the history
<Ctrl>+<N> or <Down>	Next command in the history
<Ctrl>+<D> or <Del>	Deletes the symbol under the cursor
<Alt>+<D>	Deletes the rest of the word
<Ctrl>+<H> or <Backsp>	Deletes the symbol before the cursor
<Ctrl>+<U>	Erase from cursor to beginning of line
<Ctrl>+<K>	Erase from cursor to end of line

Basic Commands

Getting to Know Your System

pwd

- Purpose
  - Print name of the current/working directory
- Syntax

```
pwd [-LP]
```

- Examples

```
# Print current (home) directory of the logged user
[user@host ~]$ pwd
```

ls

- Purpose
  - List directory contents
- Syntax

```
ls [options] [files]
```

- Examples

```
# List current directory including hidden files
[user@host ~]$ ls -a
# Show long listing for /etc directory
[user@host ~]$ ls -l /etc
```

cd

- Purpose
  - Change the current directory
- Syntax

```
cd [options] [directory]
```

- Examples

```
# Go one level up
[user@host ~]$ cd ..
# Go to a folder using absolute path
[user@host ~]$ cd /etc
```

cat

- Purpose
  - Concatenate file(s) and print on the standard output
- Syntax

```
cat [options] [files]
```

- Examples

```
# Show contents of bash history file
[user@host ~]$ cat .bash_history
# Show and number contents of a file
[user@host ~]$ cat -n /etc/os-release
```

date

- Purpose
  - Print or set the system date and time
- Syntax

```
date [[options] [+format]] | [[options] [date]]
```

- Examples

```
# Show current date
[user@host ~]$ date
# Show current date with specific format applied
[user@host ~]$ date +%Y-%m-%d
```

cal

- Purpose
  - Display a calendar
- Syntax

```
cal [options]
```

- Examples

```
# Display current month
[user@host ~]$ cal
# Display previous, current, and next month
[user@host ~]$ cal -3
```

hostname

- Purpose
  - Get or set host's name
- Syntax

```
hostname [options]
```

- Examples

```
# Display host name
[user@host ~]$ hostname
# Display short host name
[user@host ~]$ hostname --short
```

hostnamectl

Software University

- Purpose
  - Control the system hostname
- Syntax

```
hostnamectl [options]
```

- Examples

```
# Display host name
[user@host ~]$ hostnamectl
# Set host name*
[root@host ~]# hostnamectl hostname h1.home.lab
```

** There is an alternative syntax:* ***hostnamectl set-hostname h1.home.lab***

uname

- Purpose
  - Print system information
- Syntax

```
uname [options]
```

- Examples

```
# Print kernel name
[user@host ~]$ uname
# Print full information about the system
[user@host ~]$ uname --all
```

uptime

- Purpose
  - Tell how long the system has been running
- Syntax

```
uptime [options]
```

- Examples

```
# Print system uptime and average load
[user@host ~]$ uptime
# Print system uptime in pretty format*
[user@host ~]$ uptime --pretty
```

history

- Purpose
  - Display or manipulate the history list
- Syntax

```
history [options]
```

- Examples

```
# Print history of executed commands
[user@host ~]$ history
# Clear history buffer and empty the history file
[user@host ~]$ history -cw
```

exit

- Purpose
  - Exit the shell
- Syntax

```
exit [status]
```

- Examples

```
# Exit the shell with the status of last command
[user@host ~]$ exit
# Exit the shell with status of 100
[user@host ~]$ exit 100
```

logout

- Purpose
  - Exit a login shell
- Syntax

```
logout
```

- Examples

```
# Exit a login shell
[user@host ~]$ logout
```

reboot*

- Purpose
  - Reboot the system
- Syntax

```
reboot [options] [arguments]
```

- Examples

```
# Reboot the system as a regular user
[user@host ~]$ sudo reboot
# Reboot the system as root
[root@host ~]# reboot
```

* On some distributions or distribution versions a **sudo** prefix is required for regular users

poweroff*

- Purpose
  - Power off the system
- Syntax

```
poweroff [options]
```

- Examples

```
# Power off the system as a regular user
[user@host ~]$ sudo poweroff
# Power off the system as root
[root@host ~]# poweroff
```

* On some distributions or distribution versions a **sudo** prefix is required for regular users

halt*

- Purpose
  - Halt the machine
- Syntax

```
halt [options]
```

- Examples

```
# Halt the system
[root@host ~]# halt
# Power off the system
[root@host ~]# halt -p
```

* On some distributions or distribution versions a **sudo** prefix is required for regular users

shutdown*

- Purpose
  - Halt, power-off or reboot the machine
- Syntax

```
shutdown [options] [time] [wall]
```

- Examples

```
# Turn off the machine in 10 minutes
[root@host ~]# shutdown -P +10
# Cancel pending shutdown
[root@host ~]# shutdown -c
```

* On some distributions or distribution versions a **sudo** prefix is required for regular users

wall

- Purpose
  - Send message to everybody's terminal
- Syntax

```
wall [-n] [message]
```

- Examples

```
# Default format
[user@host ~]$ wall 'Please logout of the system'
# Print message without the default banner
[user@host ~]$ wall -n 'Logout now!'
```

Establish a Connection

Ways to Connect to the Console

Connection Options

- **Local (VM Console) connection**
  - No network connection required
  - No additional software required
- **Remote connection**
  - Requires network connection
  - Requires additional software on the guest (**SSH service***)
  - Requires software on the host (**PuTTY** or other **SSH client**)
  - Allows us to copy and paste text and transfer files

** We will elaborate more on the topic in the coming modules*

Practice

Summary

- Linux is Everywhere
- Linux is not an OS, but a Kernel
- Kernel + Utilities = Distribution
- There are Three Main Distribution Families
- There are many types of Shell, but we will use bash
- Basic Commands
  - pwd, ls, cd, etc.
- Environment Variables
  - PATH, PWD, HOSTNAME, and etc.

Resources

- AlmaLinux OS (*https://mirrors.almalinux.org/isos.html*)
- CentOS Stream (*https://www.centos.org/centos-stream/*)
- Debian (*https://www.debian.org/distrib/*)
- Fedora Server (*https://getfedora.org/en/server/*)
- openSUSE Leap (*https://software.opensuse.org/distributions/leap*)
- Oracle Linux (*https://yum.oracle.com/oracle-linux-isos.html*)
- Rocky Linux (*https://rockylinux.org/download*)
- Ubuntu Server (*https://www.ubuntu.com/download/server*)

Questions?

SoftUni

Software University

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

СофтУни БУДИТЕЛ

Finance Academy

SoftUni Diamond Partners

Software University

SmartIT

Coca-Cola HBC Bulgaria

SUPER HOSTING .BG

createX

Postbank

top EMPLOYER България Bulgaria 2024 FOR A BETTER WORLD OF WORK

DRAFT KINGS

THE CROWN IS YOURS

VIVACOM

INDEAVR

Serving the high achievers

AMBITIONED

PHAR VISION

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg/
- © Software University – https://softuni.bg