LPI 102.6 - Linux as a virtualization guest

Curs 2021 - 2022

ASIX M01-ISO Virtualization

Linux as a virtualization guest

Description

Key concepts:

- ❏ Understand the general concept of virtual machines and containers.
- ❏ Understand common elements virtual machines in an IaaS cloud, such as computing instances, block storage and networking.
- ❏ Understand unique properties of a Linux system which have to changed when a system is cloned or used as a template.
- ❏ Understand how system images are used to deploy virtual machines, cloud instances and containers.
- ❏ Understand Linux extensions which integrate Linux with a virtualization product.
- ❏ Awareness of cloud-init.

Commands and files:

- ❏ Virtual machine
- ❏ Linux container
- ❏ Application container
- ❏ Guest drivers
- ❏ SSH host keys
- ❏ D-Bus machine id

Virtualization

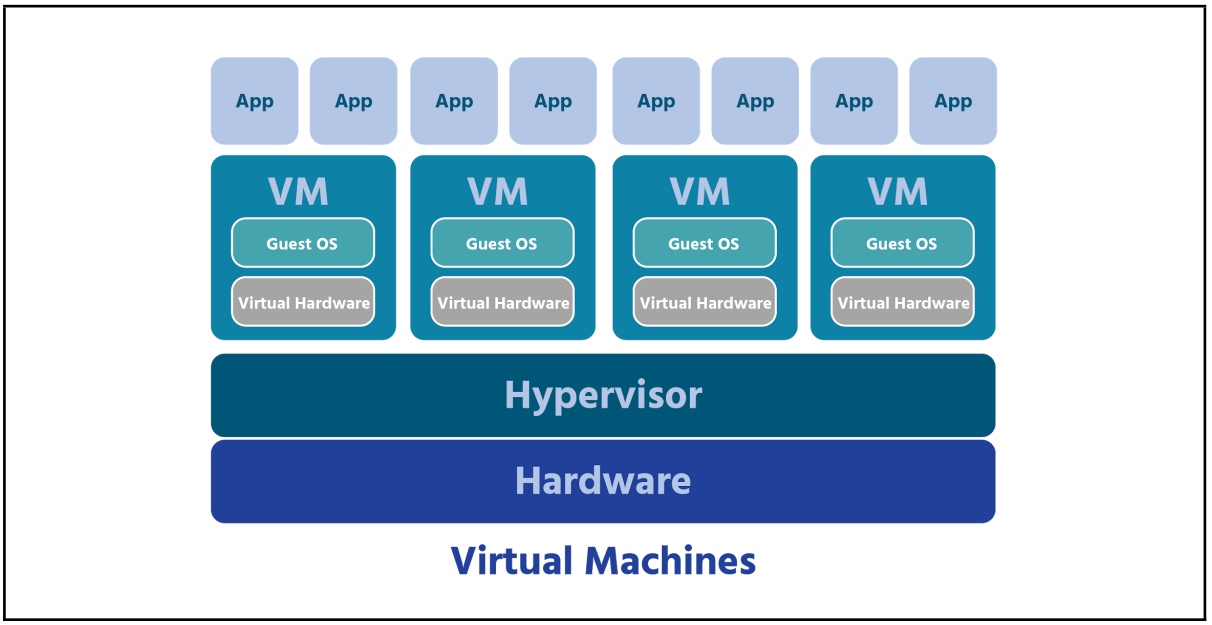
Virtual Machines

Virtualization refers to systems or programs that are running on software platforms rather than directly on computer hardware.

Virtualization is a technology that allows for a software platform, called a hypervisor, to run processes that contain a fully emulated computer system. The hypervisor is responsible for managing the physical hardware's resources that can be used by individual virtual machines. These virtual machines are called guests of the hypervisor. A virtual machine has many aspects of a physical computer emulated in software, such as a system's BIOS and hard drive disk controllers. A virtual machine will often use hard disk images that are stored as individual files, and will have access to the host machine's RAM and CPU through the hypervisor software.

A virtual machine, or VM. This is where a complete operating system is installed on virtual hardware using a hypervisor to manage processes. The hypervisor is software that is installed on top of hardware, such as a server, creating a virtualization layer and acting as a platform for VMs to be created on. The hypervisor creates virtual resources such as CPUs, memory, hard disks and I/O communication interfaces like network and serial ports.

The operating system running in a VM is called a guest operating system.



Virtual machines are typically deployed from an image which contains the guest OS, application programs, and whatever customizations the administrator has configured for system management.

Vendors:

VMWARE

One of the oldest and most widely deployed platforms is VMware, owned by Dell Technologies.

KVM

KVM, or Kernel-based Virtual Machine is an open source hypervisor that is built into Linux. As an open source program, it has the advantage of being non-proprietary and lacking licensing fees.

Xen

The Xen project is another open source hypervisor. It is considered a Type 1 or bare metal system because it runs directly on hardware without a host OS.

Hyper-V

Hyper-V is a hardware virtualization product offered by Microsoft.

VirtualBox

VirtualBox is a popular open source product from Oracle that is relatively easy to install and configure. It is a Type 2 hypervisor that runs on Windows, Macintosh, and Linux hosts.

libvirt / qemu / virt-manager

libvirtd is the standard virtual machine driver in linux.

Image formats:

raw

A raw or full disk type is a file that has all of its space pre-allocated.

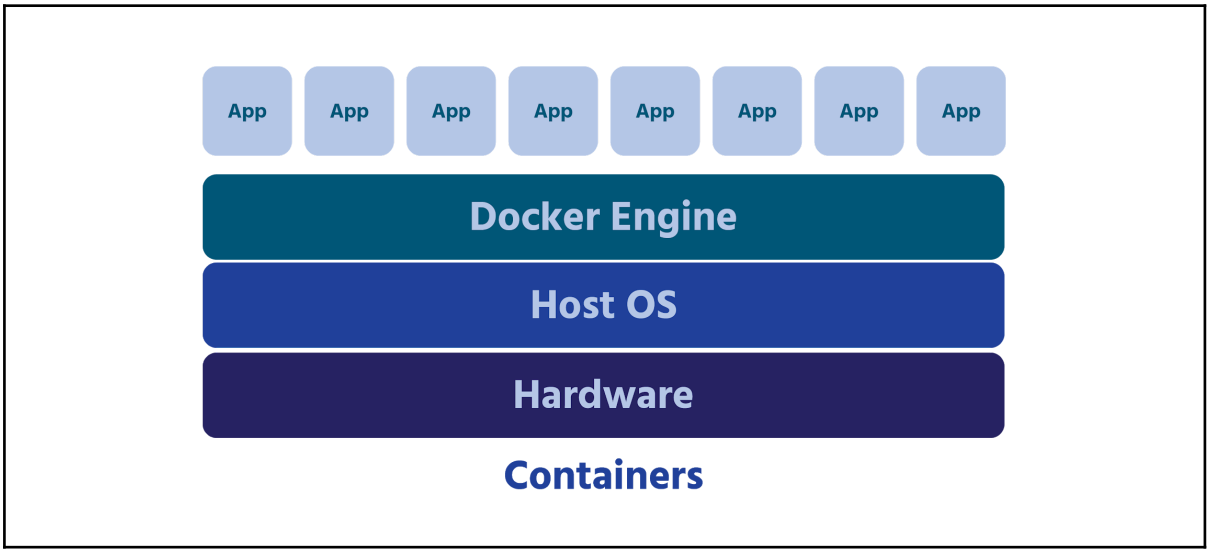
qcow2

Copy-on-write (also referred to as thin-provisioning or sparse images) is a method where a disk file is created with a pre-defined upper size limit. The disk image size only increases as new data is written to the disk.

Containers

Linux Containers allow system designers to bypass traditional operating systems and access computing resources differently. A containerized application relies on a container engine to communicate with the host OS without a hypervisor, or a guest VM.

Doing this provides several advantages, including requiring fewer resources, cutting release cycle times, and abstracting program elements from host operating systems for better portability.

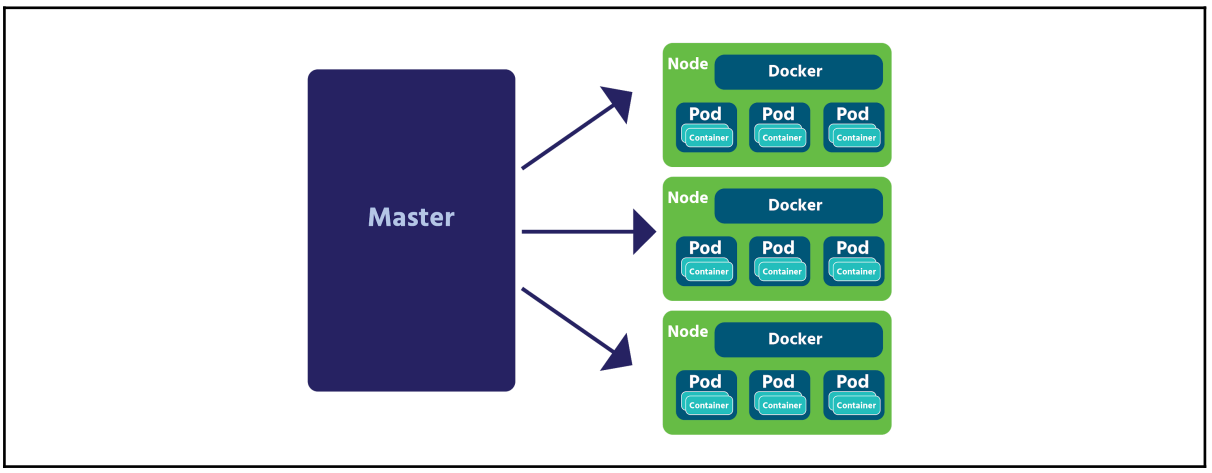


Docker

Docker is the container engine that allows programmers and system engineers to create containerized applications. These applications are stand-alone components that do not rely on any host OS to perform their functions.

Kubertenes

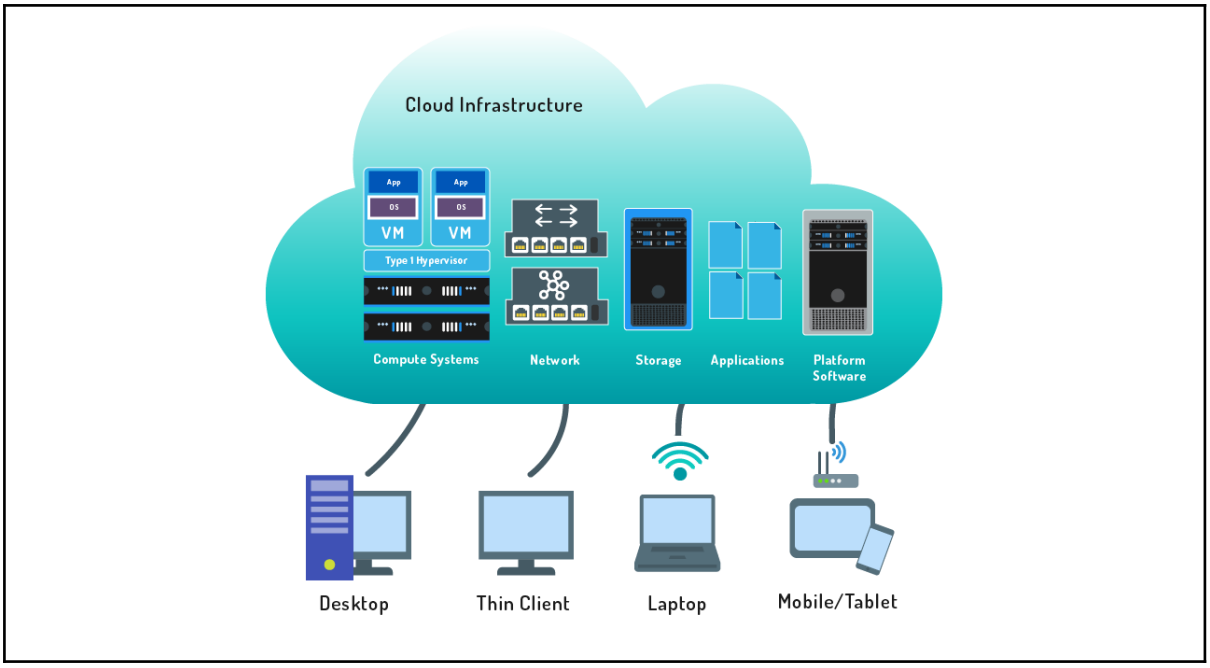
Kubernetes, originally developed by Google, was converted to an open source project in 2014. Kubernetes can be described as a platform that “provides a container-centric management environment” for containers, microservices, and cloud infrastructure.



IAAS Infrastructure as a Service

IaaS is a form of cloud computing, which is the delivery of on-demand shared computing resources (software and/or data) to organizations and users through the internet.

By abstracting the machines being managed from physical hardware, it becomes possible and desirable to construct systems in software as much as possible. Orchestrating systems such as OpenStack, Apache Cloudstack, and OpenNebula has made cloud computing possible.



VM & Container configuration

When a new virtual machine or container is created, it needs to be able to connect to the outside world. Utilities such as *cloud-init* and kickstart scripts (for VMs) that can help simplify this process. These utilities can call other programs through an API, read metadata from a server, or configure instances.

SSH, or secure shell, is used to communicate with and manage both VMs and containers. Administrators can also generate key pairs manually with the *ssh-keygen* command.

Another factor that needs to be managed is the unique id of a system, or uuid. This uuid should be generated and added to the D-Bus machine ID configuration file during installation or when first booting the VM or container. The *dbus-uuidgen* command can be used to generate or read a universally unique ID for a system. This ID takes the form of a 128-bit number that is unique.

Other critical settings to verify to ensure virtualized products, such as VMs and containers, function properly are the *virtualization extensions*, which are effectively hardware support for virtualization that is built into the CPU, such as Intel's *VT-x* and AMD's AMD-V extensions. These settings can be verified on a host system in the /proc/cpuinfo file and in the system BIOS.

```
$ lscpu | grep -E "vmx|svm"
Flags:                          fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb
rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc
cpuid aperfmperf tsc_known_freq pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3
sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes
```

```
xsave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb invpcid_single ssbd ibrs
ibpb stibp ibrs_enhanced tpr_shadow vnmi flexpriority ept vpid ept_ad fsgsbase
tsc_adjust sgx bmi1 avx2 smep bmi2 erms invpcid avx512f avx512dq rdseed adx smap
avx512ifma clflushopt intel_pt avx512cd sha_ni avx512bw avx512vl xsaveopt xsavec xgetbv1
xsaves split_lock_detect dtherm ida arat pln pts hwp hwp_notify hwp_act_window hwp_epp
hwp_pkg_req avx512vbmi umip pku ospke avx512_vbmi2 gfni vaes vpclmulqdq avx512_vnni
avx512_bitalg avx512_vpopcntdq rdpid sgx_lc fsrm md_clear flush_l1d arch_capabilities

# lscpu

# cat /proc/cpuinfo
```

```
$ dbus-uuidgen --get
f8bff953853f49b4963ed3aa06461c80
```

```
# ssh-keygen

# ssh-copy-id -i <public_key> user@cloud_server
```

Example Exercises

1. Check for the cpu virtualization extensions.
2. Show the system UUID.

3. Install virt-manager (libvirt).
4. Enable libvirtd at system boot.
5. Start virtual machine using an standard Cloud image qcow2.

6. Install docker.
7. Start docker-hello container.
8. Start an interactive debian container.

9. Realitza els exercicis indicats a: 102.6 Linux as a virtualization guest

10. Realitza els exercicis del Question-Topics 102.6.