Virtualizácia a emaily

kriza (Richard Staňa)
kr1za.github.io/
UPJŠ – AOS

Ako si vybrať

- LXC, LXD
- Docker
- HyperV
- Qemu
- KVM
- Virtualbox
- XEN
- Parallers Desktop
- Vmware Workstation
- …

LXC vs QEMU/KVM vs Hyper-V

LXC stands for Linux Containers

Spoločný kernel s hostom

Rychlejšie

Ľahšie

KVM is an acronym for Kernel-Based Virtual Machine

Vlastný kernel

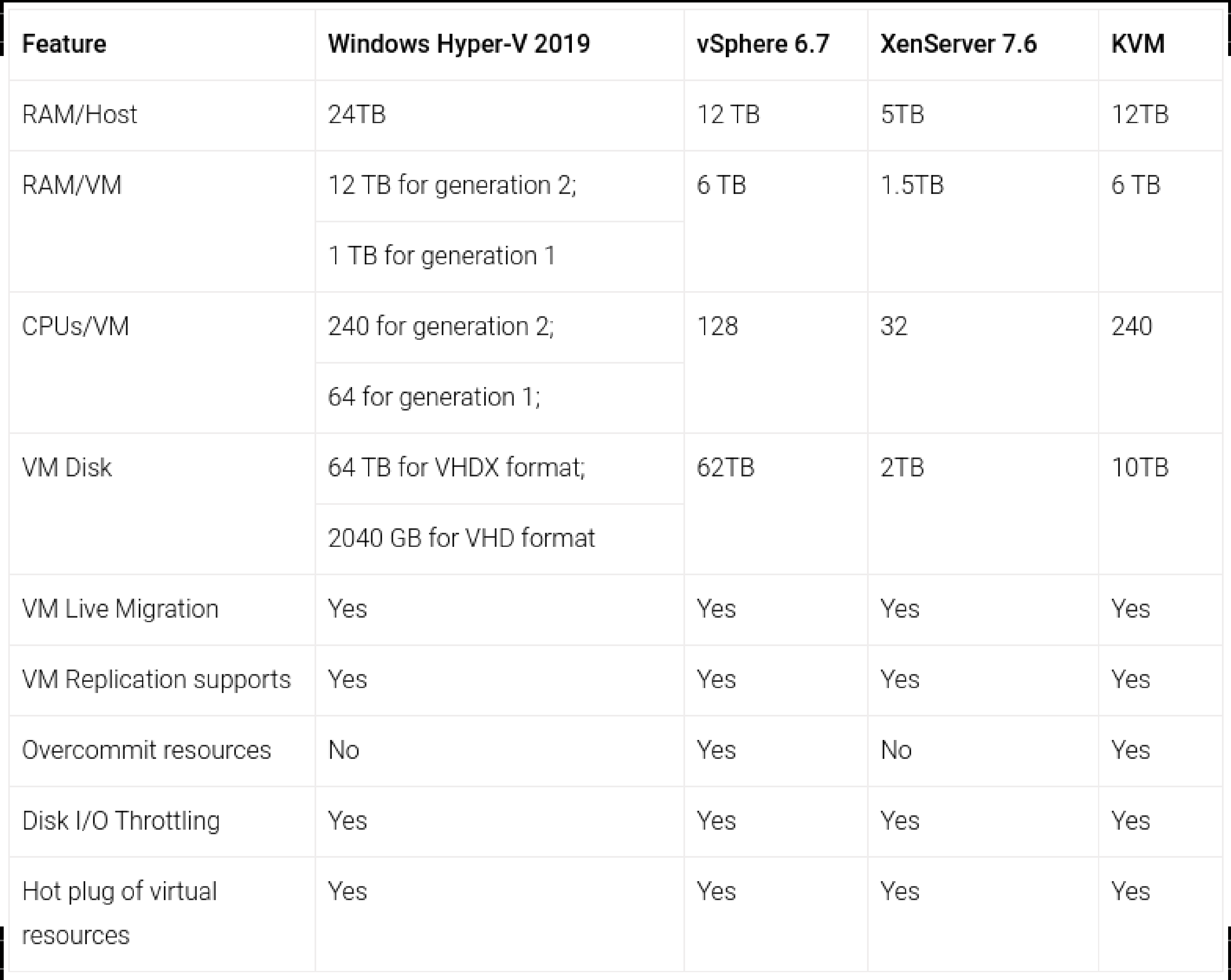
Virtualizácia na HW

Pomalšie

Ťažšie

Rôzny OS

Windows...



Feature	Windows Hyper-V 2019	vSphere 6.7	XenServer 7.6	KVM
RAM/Host	24TB	12 TB	5TB	12TB
RAM/VM	12 TB for generation 2; 1 TB for generation 1	6 TB	1.5TB	6 TB

CPUs/VM	240 for generation 2;
	64 for generation 1;
		128	32	240

VM Disk	64 TB for VHDX format; 2040 GB for VHD format	62TB	2TB	10TB
VM Live Migration	Yes	Yes	Yes	Yes
VM Replication supports	Yes	Yes	Yes	Yes
Overcommit resources	No	Yes	No	Yes
Disk I/O Throttling	Yes	Yes	Yes	Yes
Hot plug of virtual resources	Yes	Yes	Yes	Yes

Kontajner???

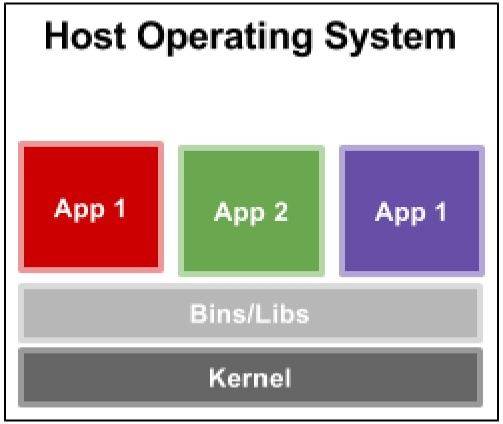
https://www.linuxjournal.com/content/everything-you-need-know-about-linux-containers-part-ii-working-linux-containers-lxc
https://www.mevako.sk/Kontajner-ziarovo-zinkovany-1100-l-d3812.htm?tab=description
https://www.majstore.sk/p/1372/kontajner-na-odpad-pozink-110l

LXC

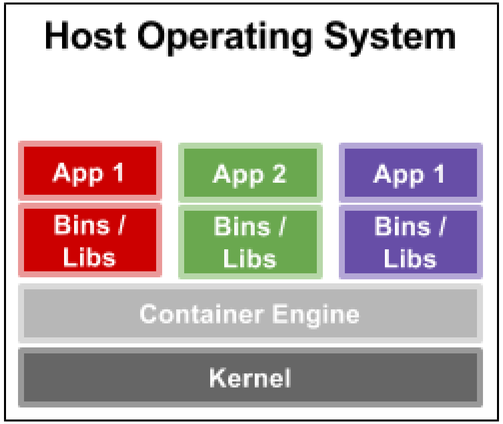
Unprivileged containers are the safest containers. Those use a map of uid and gid to allocate a range of uids and gids to a container. That means that uid 0 (root) in the container is actually something like uid 100000 outside the container. So should something go very wrong and an attacker manages to escape the container, they'll find themselves with about as many rights as a nobody user.

LXC

- sudo apt-get install lxc
- apt install libvirt-bin ?



Native Running Applications



Containerized Applications

https://www.linuxjournal.com/content/everything-you-need-know-about-linux-containers-part-ii-working-linux-containers-lxc

LXC

- /etc/default/lxc-net
  - USE_LXC_BRIDGE="true"
- /etc/lxc/default.conf
  - lxc.net.0.type = veth
  - lxc.net.0.link = virbr0
  - lxc.net.0.flags = up
  - lxc.apparmor.profile = generated
  - lxc.apparmor.allow_nesting = 1
- sudo apt-get install -qy libvirt-clients libvirt-daemon-system iptables ebtables dnsmasq-base
- sudo virsh net-start default
- sudo virsh net-autostart default

https://www.linuxjournal.com/content/everything-you-need-know-about-linux-containers-part-ii-working-linux-containers-lxc

LXC

- lxc-create -t download -n my-container
- lxc-start -n my-container -d
- lxc-info -n my-container
- lxc-ls -f
- lxc-attach -n my-container
- lxc-stop -n my-container
- lxc-destroy -n my-container

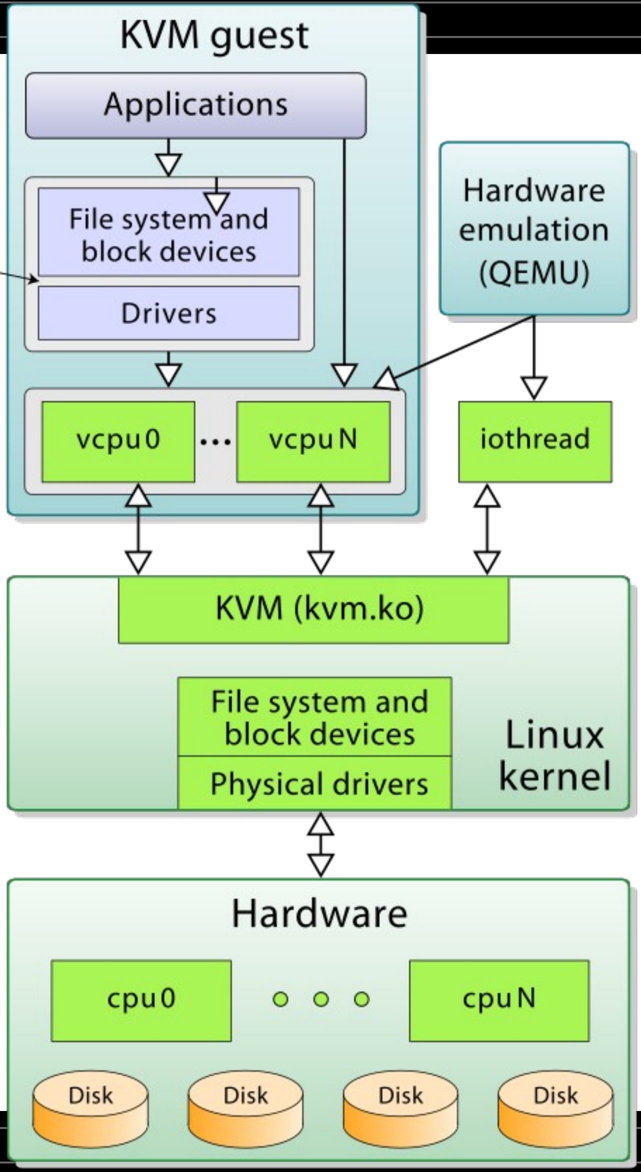
KVM VS QEMU

QEMU is a userland type 2 (i.e runs upon a host OS) hypervisor for performing **hardware virtualization** (not to be confused with hardware-assisted virtualization), such as disk, network, VGA, PCI, USB, serial/parallel ports, etc. It is flexible in that it can emulate CPUs via dynamic binary translation (DBT) allowing code written for a given processor to be executed on another (i.e ARM on x86, or PPC on ARM). Though QEMU can run on its own and emulate all of the virtual machines resources, as all the emulation is performed in software it is extremely slow.

KVM VS QEMU

KVM is a **Linux kernel module**. It is a type 1 hypervisor that is a **full virtualization** solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V)[1]. But what is full virtualization, you may ask? When a CPU is emulated (vCPU) by the hypervisor, the hypervisor has to translate the instructions meant for the vCPU to the physical CPU. As you can imagine this has a massive performance impact. To overcome this, modern processors support virtualization extensions, such as Intel VT-x and AMD-V. These technologies provide the ability for a **slice of the physical CPU to be directly mapped to the vCPU**. Therefor the the instructions meant for the vCPU can be directly executed on the physical CPU slice[2].

KVM VS QEMU



KVM guest's kernel

Only one thread can run QEMU code at any time (qemu_mutex)

Generates I/O requests to the host on guest's behalf and handles events

https://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine#/media/File:Kernel-based_Virtual_Machine.svg

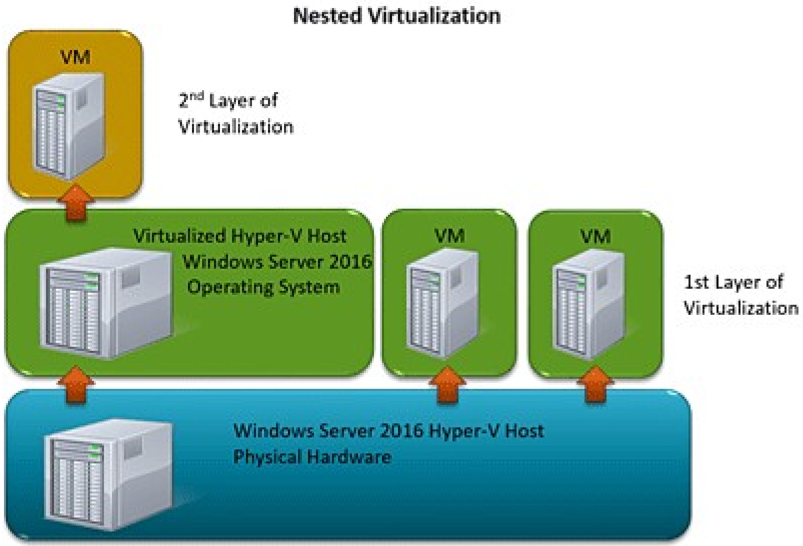
Inštalácia u mňa

- sudo pacman -S qemu
- sudo pacman -S libvirt
- sudo pacman -S dnsmasq vde2 bridge-utils openbsd-netcat iptables ebtables ?
- sudo usermod -a -G libvirt $(whoami)
- sudo systemctl start libvirtd.service

Hyper-V

- Windows…
- https://www.nakivo.com/blog/run-linux-hyper-v/

Nested virtualization



https://www.webopedia.com/TERM/N/nested-virtualization.html

Email

- Mailutils, postfix
- /etc/postfix/main.cf
  - inet_interfaces = loopback-only
  - mydestination = $myhostname, localhost.$your_domain, $your_domain
- mail -s "This is the subject" somebody@example.com <<< 'This is the message'

Sorry