

Virtualisatie

Oscar Buse
14 februari 2017
NLUG

Inhoudsopgave

Inhoud

Inhoud

Overzicht onderwerpen

Inleiding

Inleiding

Terminologie

Historie

Definitie virtualisatie

KVM

KVM

KVM, QEMU en libvirt

Installatie

virsh en virt-manager

Virtueel netwerk

Storage pools en volumes

maak een VM

Concluderend

Conclusies

Hypervisor vs container

Kubernetes

Overzicht onderwerpen

- Terminologie
- Historie
- Wat is virtualisatie?
- KVM, QEMU en libvirt
- KVM - command line en grafisch
- Virtuele netwerken
- Storage pools en volumes
- Maken van een VM, installatie OS.

Terminologie

virtualisatie wikipedia: "virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources."

Terminologie

virtualisatie wikipedia: "virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources."

host het "host" systeem: het systeem wat guest systemen ondersteunt.

Terminologie

virtualisatie wikipedia: "virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources."

host het "host" systeem: het systeem wat guest systemen ondersteunt.

guest de virtuele systemen (VM).

Terminologie

virtualisatie wikipedia: "virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources."

host het "host" systeem: het systeem wat guest systemen ondersteunt.

guest de virtuele systemen (VM).

hypervisor software welke runt op het host systeem en de onderliggende hardware voor quest systemen virtualiseert (beschikbaar maakt middels *hardware emulatie*).

Terminologie

virtualisatie wikipedia: "virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources."

host het "host" systeem: het systeem wat guest systemen ondersteunt.

guest de virtuele systemen (VM).

hypervisor software welke runt op het host systeem en de onderliggende hardware voor quest systemen virtualiseert (beschikbaar maakt middels *hardware emulatie*).

hardware emulatie hardware wat door het host systeem (de hypervisor) als virtuele hardware aan de guests wordt gepresenteerd.

Terminologie

virtualisatie wikipedia: "virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources."

host het "host" systeem: het systeem wat guest systemen ondersteunt.

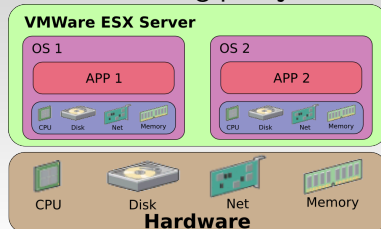
guest de virtuele systemen (VM).

hypervisor software welke runt op het host systeem en de onderliggende hardware voor quest systemen virtualiseert (beschikbaar maakt middels *hardware emulatie*).

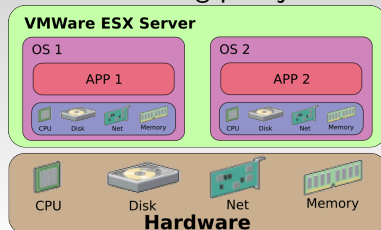
hardware emulatie hardware wat door het host systeem (de hypervisor) als virtuele hardware aan de guests wordt gepresenteerd.

container "lichtgewicht VM.." (later meer).

In een eenvoudig plaatje:



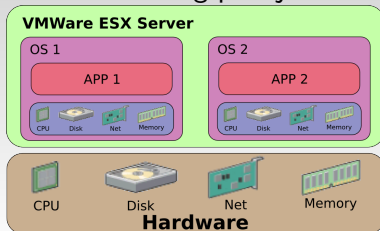
In een eenvoudig plaatje:



Nog wat meer termen:

DRS Distributed Resource Scheduler. aan/uitschakelen hosts, migreren guests.

In een eenvoudig plaatje:

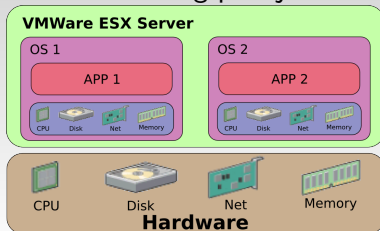


Nog wat meer termen:

DRS Distributed Resource Scheduler. aan/uitschakelen hosts, migreren guests.

HA High Availability. Detecteren host failure, migreren guests.

In een eenvoudig plaatje:



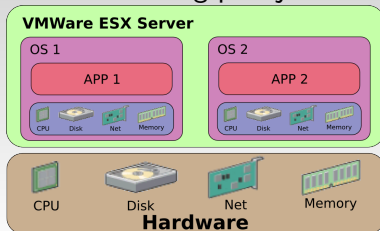
Nog wat meer termen:

DRS Distributed Resource Scheduler. aan/uitschakelen hosts, migreren guests.

HA High Availability. Detecteren host failure, migreren guests.

volledige (full) virtualisatie . Alles gevirtualiseerd: guest ziet functioneel echte hardware. Best lastig in de begindagen (VMware was goed bezig (straks meer)).

In een eenvoudig plaatje:



Nog wat meer termen:

DRS Distributed Resource Scheduler. aan/uitschakelen hosts, migreren guests.

HA High Availability. Detecteren host failure, migreren guests.

volledige (full) virtualisatie . Alles gevirtualiseerd: guest ziet functioneel echte hardware. Best lastig in de begindagen (VMware was goed bezig (straks meer)).

para virtualisatie : de guest heeft een aangepaste kernel (Xen).

Historie

- Vóór virtualisatie: veel "losse" servers:

Historie

- Vóór virtualisatie: veel "losse" servers:
 - Verspilling van CPU, RAM, disk, netwerk en (niet te vergeten) power resources (InterNLnet ging het stroomverbruik van het datacenter met meer dan 20% naar beneden (90% > 6x%)).

Historie

- Vóór virtualisatie: veel "losse" servers:
 - Verspilling van CPU, RAM, disk, netwerk en (niet te vergeten) power resources (InterNLnet ging het stroomverbruik van het datacenter met meer dan 20% naar beneden ($90\% > 6x\%$)).
- Pioniers: Xen en VMware:

Historie

- Vóór virtualisatie: veel "losse" servers:
 - Verspilling van CPU, RAM, disk, netwerk en (niet te vergeten) power resources (InterNLnet ging het stroomverbruik van het datacenter met meer dan 20% naar beneden ($90\% > 6x\%$)).
- Pioniers: Xen en VMware:
 - Xen: para virtualisatie: aangepaste kernel voor guest systeem nodig.

Historie

- Vóór virtualisatie: veel "losse" servers:
 - Verspilling van CPU, RAM, disk, netwerk en (niet te vergeten) power resources (InterNLnet ging het stroomverbruik van het datacenter met meer dan 20% naar beneden ($90\% > 6x\%$)).
- Pioniers: Xen en VMware:
 - Xen: para virtualisatie: aangepaste kernel voor guest systeem nodig.
 - VMware: hypervisor ESX. '*binary translation*: guest calls door de hypervisor vertaald (geen aangepast guest systeem nodig).

- HVM (Hardware Virtual Machine). Bv. Intel VT-x en AMD-V (vanaf ca. 2006).

- HVM (Hardware Virtual Machine). Bv. Intel VT-x en AMD-V (vanaf ca. 2006).
 - extra instructies op de CPU speciaal voor virtualisatie.

- HVM (Hardware Virtual Machine). Bv. Intel VT-x en AMD-V (vanaf ca. 2006).
 - extra instructies op de CPU speciaal voor virtualisatie.
 - Maakte volledige virtualisatie makkelijker (juist vanwege de complexe x86 architectuur was VT-x/AMD-V erg welkom).

- HVM (Hardware Virtual Machine). Bv. Intel VT-x en AMD-V (vanaf ca. 2006).
 - extra instructies op de CPU speciaal voor virtualisatie.
 - Maakte volledige virtualisatie makkelijker (juist vanwege de complexe x86 architectuur was VT-x/AMD-V erg welkom).
 - Tevens een enorme snelheidswinst.

- HVM (Hardware Virtual Machine). Bv. Intel VT-x en AMD-V (vanaf ca. 2006).
 - extra instructies op de CPU speciaal voor virtualisatie.
 - Maakte volledige virtualisatie makkelijker (juist vanwege de complexe x86 architectuur was VT-x/AMD-V erg welkom).
 - Tevens een enorme snelheidswinst.
 - Xen, VMWare, KVM, ...

- HVM (Hardware Virtual Machine). Bv. Intel VT-x en AMD-V (vanaf ca. 2006).
 - extra instructies op de CPU speciaal voor virtualisatie.
 - Maakte volledige virtualisatie makkelijker (juist vanwege de complexe x86 architectuur was VT-x/AMD-V erg welkom).
 - Tevens een enorme snelheidswinst.
 - Xen, VMWare, KVM, ...
 - Alleen voor cpu/memory. Access van netwerk en disk nog steeds via emulatie (door de hypervisor). Wel vaak via speciale drivers voor zowel guest als hypervisor (PV's: Para virtualized Drivers).

Dus wat is virtualisatie?

Dus wat is virtualisatie?

Er is niet 1 definitie: erg afhankelijk van implementatie:

Dus wat is virtualisatie?

Er is niet 1 definitie: erg afhankelijk van implementatie:

- para?
- volledig?
- containers?

Dus wat is virtualisatie?

Er is niet 1 definitie: erg afhankelijk van implementatie:

- para?
- volledig?
- containers?

Wel belangrijk dat je de terminologie en mogelijkheden kent.

KVM, QEMU en libvirt

- KVM:
 - Onderdeel van de kernel van het host systeem.
 - Geen complete hypervisor.
 - Maakt gebruik van de hardware virtualisatie (VT-x/AMD-V). Verzorgt de mapping tussen fysieke en virtuele CPU's.
- QEMU (Quick EMUlator):
 - Software op zichzelf. Emuleert hardware (ook disken, netwerk, PCI, USB, ...)
 - Werkt samen met KVM maar kan ook geheel zelfstandig als hypervisor dienen.
- libvirt: software voor het managen van de hypervisor en guests.
 - daemon libvirtd
 - cli -en grafische tools

Hypervisor = combinatie van KVM en QEMU.

Installatie

Bijvoorbeeld voor CentOS:

```
yum -y install qemu-kvm libvirt virt-install bridge-utils  
systemctl start libvirtd  
systemctl enable libvirtd
```

virsh

Keuze om VM's te maken/deleten etc.. uit cli (`virsh`) en grafische tool (`virt-manager`).

Virtueel netwerk

VM's kunnen we een eigen virtueel netwerk geven. Dit kan makkelijk met virt-manager. Maar kan ook met cli en een template:

Virtueel netwerk

VM's kunnen we een eigen virtueel netwerk geven. Dit kan makkelijk met virt-manager. Maar kan ook met cli en een template:

```
virsh# net-list
virsh# net-list --all
virsh# net-define /root/netwerk1.xml
virsh# net-autostart netwerk1
virsh# net-start netwerk1
virsh# net-info netwerk1
```

Storage pools en volumes

Met libvirt kun je storage pools maken.

- de default storage pool is in /var/lib/libvirt/images
- een storage pool bevat storage volumes (de disken)

Maak een nieuwe pool:

```
mkdir /var/lib/libvirt/pool1  
virsh pool-define-as pool1 dir - - - - /var/lib/libvirt/pool1  
virsh pool-autostart pool1  
virsh pool-start pool1  
virsh vol-create-as pool1 volume1.img 10G
```

maak een VM

```
virt-install -r 1024 --vcpus=1 -n debian8 \  
-f /var/lib/libvirt/pool1/debian8.img \  
--cdrom /home/user1/Downloads/debian-8.5.0-amd64-netinst.iso
```

Hypervisor vs container

De functie van de hypervisor is wat bleekjes tegenwoordig.
Enkele nadelen:

Hypervisor vs container

De functie van de hypervisor is wat bleekjes tegenwoordig.

Enkele nadelen:

- Netwerk/disk access via "paravirtualized drivers" (PV drivers). Verschillende PV-drivers nodig per hypervisor (VMware's ESX, Xen, KVM) én per guest OS (windows 7, windows 10, redhat, ubuntu etc..) > veel code!

Hypervisor vs container

De functie van de hypervisor is wat bleekjes tegenwoordig.

Enkele nadelen:

- Network/disk access via "paravirtualized drivers" (PV drivers). Verschillende PV-drivers nodig per hypervisor (VMware's ESX, Xen, KVM) én per guest OS (windows 7, windows 10, redhat, ubuntu etc..) > veel code!
- OS-patching in je VM's!

Hypervisor vs container

De functie van de hypervisor is wat bleekjes tegenwoordig.

Enkele nadelen:

- Netwerk/disk access via "paravirtualized drivers" (PV drivers). Verschillende PV-drivers nodig per hypervisor (VMware's ESX, Xen, KVM) én per guest OS (windows 7, windows 10, redhat, ubuntu etc..) > veel code!
- OS-patching in je VM's!
- Relatief veel resources nodig per VM (vergeleken met containers).

Hypervisor vs container

De functie van de hypervisor is wat bleekjes tegenwoordig.

Enkele nadelen:

- Netwerk/disk access via "paravirtualized drivers" (PV drivers). Verschillende PV-drivers nodig per hypervisor (VMware's ESX, Xen, KVM) én per guest OS (windows 7, windows 10, redhat, ubuntu etc..) > veel code!
- OS-patching in je VM's!
- Relatief veel resources nodig per VM (vergeleken met containers).
- DRS en HA niet meer zo'n USP's: er komen steeds meer "container management" tools zoals bv. "Kubernetes".

Hypervisor vs container

De functie van de hypervisor is wat bleekjes tegenwoordig.

Enkele nadelen:

- Netwerk/disk access via "paravirtualized drivers" (PV drivers). Verschillende PV-drivers nodig per hypervisor (VMware's ESX, Xen, KVM) én per guest OS (windows 7, windows 10, redhat, ubuntu etc..) > veel code!
- OS-patching in je VM's!
- Relatief veel resources nodig per VM (vergeleken met containers).
- DRS en HA niet meer zo'n USP's: er komen steeds meer "container management" tools zoals bv. "Kubernetes".
- Security is ook sterk verbeterd mbt containers. Zeker sinds de extra instructies op de CPU (VT-x/AMD-V).

Hypervisor vs container

De functie van de hypervisor is wat bleekjes tegenwoordig.

Enkele nadelen:

- Netwerk/disk access via "paravirtualized drivers" (PV drivers). Verschillende PV-drivers nodig per hypervisor (VMware's ESX, Xen, KVM) én per guest OS (windows 7, windows 10, redhat, ubuntu etc..) > veel code!
- OS-patching in je VM's!
- Relatief veel resources nodig per VM (vergeleken met containers).
- DRS en HA niet meer zo'n USP's: er komen steeds meer "container management" tools zoals bv. "Kubernetes".
- Security is ook sterk verbeterd mbt containers. Zeker sinds de extra instructies op de CPU (VT-x/AMD-V).
- Een hypervisor is goed in het ondersteunen van een hybride omgeving (verschillende OSsen). Maar wat is het voordeel daarvan?

We gaan dan ook niet alleen kijken naar de "gewone"
(hypervisor) virtualisatie (met KVM) maar ook naar containers:

We gaan dan ook niet alleen kijken naar de "gewone" (hypervisor) virtualisatie (met KVM) maar ook naar containers:

- Vandaag: KVM (+ QEMU en libvirt)

We gaan dan ook niet alleen kijken naar de "gewone" (hypervisor) virtualisatie (met KVM) maar ook naar containers:

- Vandaag: KVM (+ QEMU en libvirt)
- Volgende keer: containers (docker)

We gaan dan ook niet alleen kijken naar de "gewone" (hypervisor) virtualisatie (met KVM) maar ook naar containers:

- Vandaag: KVM (+ QEMU en libvirt)
- Volgende keer: containers (docker)
- Daarna: Kubernetes (..)