



kubernetes

kubernetes

"It compiles, ship it!"

Oscar Buse

13 februari 2018

Linux User Group Nijmegen

Inleiding

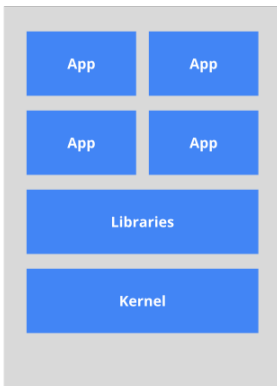
Dit praatje gaat over kubernetes.

De onderwerpen die aan bod komen:

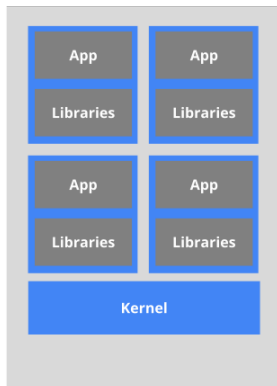
- Eerdere vormen van virtualisatie (VM's en Docker).
- Wat is kubernetes (definitie)?
- Overzicht kubernetes architectuur.
- Installatie van een kubernetes cluster.
- Basis concepten: pods, services, replica-sets.
- Meer concepten: labels, deployments, kubernetes networking.
- Werken met kubernetes: yaml manifests en kubectl. Theorie met voorbeelden.
- Waarom kubernetes?
- Referenties.

Eerdere vormen van virtualisatie

- VMs, bv. met VMware, KVM of Xen.
- Containers, bv. met lxc of Docker. Kubernetes gebruikt standaard Docker als containerplatform.
Containers zijn zeer compact vergeleken met VMs.

The old way: Applications on host

*Heavyweight, non-portable
Relies on OS package manager*

The new way: Deploy containers

*Small and fast, portable
Uses OS-level virtualization*

Wat is kubernetes?

Definitie: "open source platform voor het automatiseren van het uitrollen en beheren van applicatie containers."

Enkele concepten typisch voor een kubernetes cluster:

- "los" van het onderliggende OS en bijbehorende libraries.
- immutability (vergelijk bv. "apt-get update" vs "vervang container met een nieuwe").
- declaratieve configuratie (*ipv imperatief*). Je beschrijft een (eind)toestand in plaats van "de weg er naar toe".
- self-healing: om de (eind)toestand zo te houden.

Duidelijke scheiding tussen *application developer* en *orchestration engineer*:

developer

weinig zorgen mbt onderliggende hardware.

engineer

weinig zorgen (of kennis) over de applicaties die op zijn platform draaien.

Hoezo *devops* ;-)

De kubernetes architectuur

- Master node(s) en workers node(s).
- Nodes bevatten pods (kleinste *deployable* eenheid binnen k8s).
- Pods bevatten containers (k8s komt standaard met docker).
- kubernetes is zelf ook opgebouwd uit pods met containers (hoeft niet).

Master node(s) onderdelen:

- API: beheren/uitvragen van je kubernetes cluster.
- Scheduler: scheduled de verschillende pods .
- Controller: bewaakt de *state* van je cluster.
- Storage voor allerei config parameters en state: etcd (key value).
- kube-dns: eigen dns service (veel aan/uit situaties).
- kubelet: de *node agent*, manages containers.
- kube-router: managed het netwerk (CNI).

Workers node(s) onderdelen:

- kubelet: de *node agent*, manages containers.
- kube-router: managed het netwerk (CNI).

Installatie

Veel mogelijkheden: `https://kubernetes.io/docs/setup/pick-right-solution/`

Keuze tussen bv. Kaas of eigen hardware.

Om te oefenen op een single node: minikube.

Ter lering ende vermaak: er zijn ook voorgedraaide images om een k8s cluster uit te rollen op Raspberry Pi's.

De stappen (meer info Linux magazine december 2016):

- installeer HypriotOS op 3 (of meer) RPi's.
- connect de RPi's in hetzelfde netwerk.
- zorg dat, bv. via de master, internet connectiviteit is.
- `install docker-multinode.deb`
- `master# kube-config install`
- `master# export IP_ADDRESS=10.129.5.1`
- `master# kube-config enable-master`
- `workers# kube-config install`
- `master# kube-config enable-worker "master-ip"`

Een andere (independent) install kan met kubeadm op VM's.
De stappen (referentie urls aan het eind):

- maak een Vagrantfile voor de VM's (1 master, 2 workers).
- add k8s repo (voor CentOS7:
https://packages.cloud.google.com/yum/repos/kubernetes-el7x86_64).
- `yum install kubelet kubeadm kubectl`
- `systemctl enable kubelet`
- `master# kubeadm init`
- `master# export KUBECONFIG=/etc/kubernetes/admin.conf`
- `master# kubectl apply -f kubeadm-kuberouter.yaml`
- `workers# kubeadm join -token "token" "master IP":6443`

REST API

- k8s komt (uiteraard) met een REST API.
- Deze API kent json en yaml.
- Programmeren tegen de API, curl en het commando kubectl.
- Alle k8s "resources" (pods, services, replica-sets, deployments, labels, volumes, ...) zijn te configgen met json of yaml.
- Gebruik de "-o yaml" bij je get requests om te leren wat allemaal mogelijk is.

Doe bv.

```
kubelet get deployment nginx -o yaml >  
nginx-deployment.yaml
```

(straks meer).

- Met 1 enkel yamlfile bouw je je hele kubernetescluster op!
Straks zien we hier een uitgebreid voorbeeld van.

kubectl

De command line interface naar de API server.

- `kubectl get nodes`
- `kubectl get pods -o wide`
- `kubectl get services -o yaml`
- `kubectl endpoints`
- `kubectl get deployments`
- `kubectl describe ...`
- `kubectl run ..`
- `kubectl scale -replicas=n deployment/"name"`
- `kubectl set image deployment/nginx nginx=nginx:1.10 -all`
- `kubectl apply -f applicatie.yaml`

Troubleshoot

- standaard tooling, vooral netstat
- kubectl view/get/describe
- container logs , exec

Waarom kubernetes?

- de voordelen van containers:
 - Geen dependency issues: overall (OTAP) dezelfde environment.
 - Lichtgewicht
 - Development en deployment wordt een geautomatiseerd proces.
- Vereenvoudiging voor systembeheerders van het uitrollen en beheren van servers (en bijbehorende services).
- Vereenvoudiging voor developers van het uitrollen van software.
- Snelheidswinst voor beiden.
- Efficient gebruik van resources.
- Zeer portable.
- Horizontaal schaalbaar.

Referenties

- Website: <https://www.kubernetes.io>
- Cheatsheet: <https://kubernetes.io/docs/reference/kubectl/cheatsheet/>
- Stackoverflow: <https://stackoverflow.com/search?q=kubernetes>
- Kubeadm: <https://kubernetes.io/docs/setup/independent/create-cluster-kubeadm/>
- Setup with kubeadm:
<https://medium.com/@joatmon08/playing-with-kubeadm-in-vagrant-machines-3659>

Bier?