Devops labo 3

# Voor je begint

Voor je begint aan het opzetten van de servers ga je eerst pip en ansible moeten installeren. Pip installeer je met het commando:

$ sudo easy\_install pip

$ sudo pip install --upgrade pip

(Commando’s in dit verslag zijn gebasseerd op mac.)

Ansible installeer je met het commando:

$ sudo easy\_install pip

$ sudo pip install ansible

## Aanmaken subnets

Aan de loadbalancer en de database server gaan we eerst een specifiek ip adress geven.

**LB\_server = '192.168.50.2'**

**DB\_server = '192.168.200.2'**

Vervolgens geven we alle servers een subnet.

**Loadsbalancer\_subnet = '192.168.50.0/24'**

**FrontendServers\_subnet = '192.168.60.0/24'**

**BackendServers\_subnet = '192.168.150.0/24'**

**DatabaseServer\_subnet = '192.168.200.0/24'**

# Aanmaken front-end servers

Om te testen heb ik eerst 2 virtuele machines aangemaakt in een apart subnet en ervoor gezorgd dat deze machines naar elkaar kunnen pingen.

Vervolgens heb ik in de vagrantfile bij de front-end servers gezorgd voor een parallele werking. Op deze manier worden de machines sneller aangemaakt.

**(1..N\_fe).each do |machine\_id\_FE|**

**config.vm.define "frontend-#{machine\_id\_FE}" do |frontend|**

**frontend.vm.box = "bento/ubuntu-18.04"**

**frontend.vm.network "private\_network", ip: "192.168.60.#{1+machine\_id\_FE}"**

**#Parallel provisioning**

**if machine\_id\_FE == N\_fe**

**frontend.vm.provision "shell", path: "./create\_users"**

**frontend.vm.provision "shell", path: "./change\_ssh\_config"**

**frontend.vm.provision :ansible do |ansible\_fe|**

**ansible\_fe.limit = "all"**

**ansible\_fe.playbook = "playbooks/frontend.yml"**

**FrontendServers = []**

**#Build list of front end servers for setting groups**

**(1..N\_fe).each do |m\_id\_FE|**

**FrontendServers << "frontend-#{m\_id\_FE}"**

**end**

**ansible\_fe.groups = {**

**"feservers" => FrontendServers**

**}**

**ansible\_fe.extra\_vars = {**

**"lb\_server\_address"=> LB\_server #Voor Firewall rules**

**}**

**end**

**end**

**end**

**end**

Laatste stap is om nginx te laten runnen dit kan je checken via het commando:

**Sudo service nginx status**

Installatie nginx in paybook:

**- name: Install nginx on frontend machine**

**apt: name=nginx state=present**

Zorg er weer voor dat de poorten die wel toegelaten zijn open staan en degenen die niet toegelaten zijn gesloten zijn.

# Aanmaken Load balancer

Zorg er voor dat alle poorten die toegelaten zijn open staan en de poorten die niet toegestaan zijn gesloten zijn.

Vervolgens installeer, update en herstart HAproxy op de loadbalancer (dit moet je doen in de playbook van de loadbalancer):

**- name: Install HAProxy on loadbalancer**

**apt: name=haproxy state=present**

**- name: Update HAProxy config**

**template: src=../cfg/lb\_conf/haproxy.cfg**

**dest=/etc/haproxy/haproxy.cfg**

**backup=yes**

**- name: Restart service HAProxy on loadbalancer, in all cases**

**become: true**

**become\_user: root**

**service:**

**name: haproxy**

**state: restarted**

## Voorkomende problemen

Probleem => Load balancer kon niet verbinden met front end servers wegens de firewall.

Tijdelijke oplossing = open playbook front end en verwijder bij "Allow incoming www onlu from lb\_server\_address" de "src="{{lb\_server\_address}}""

Tijdelijke oplossing => src="{{lb\_server\_address}}" heb ik uit de playbook verwijdert.

# Aanmaken back-end server

Om te beginnen moet je eerst python installeren op uw backend vagrant machine. Deze was bij mij al geïnstalleerd.

Zorg er weer voor dat de poorten die wel toegelaten zijn open staan en degenen die niet toegelaten zijn gesloten zijn.

Installeer mysql in de playbook van de backend:

**- name: Install mysql**

**apt: name=mysql-server state=present**

## Voorkomende problemen

Vervolgens moet je Flask installeren. Hierbij kreeg ik de volgende foutmelding:

**TASK [install Flask] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**fatal: [lb]: FAILED! => {"changed": false, "msg": "No setuptools found in remote host, please install it first."}**

**fatal: [frontend-10]: FAILED! => {"changed": false, "msg": "No setuptools found in remote host, please install it first."}**

**fatal: [backend-10]: FAILED! => {"changed": false, "msg": "No setuptools found in remote host, please install it first."}**

Oplossing => Pip was niet geïnstalleerd.

# Bronnen

<https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html>

https://www.vagrantup.com/docs/provisioning/ansible.html