



**FUGA CLOUD**



# Dé Nederlandse Cloud

- **Hosted in Nederland**
- **Nederlandse en Europese wet- en regelgeving**
- **Gemakkelijk bereikbaar**
- **SLA 99,99% uptime, Flash only platform**
- **AVG ready, ISO27001, ISO20000, NEN7510**



- **Open community**
- **Open Source**
- **Open Standaarden**
- **Open communicatie**
- **OpenStack**



# Infra-as-a-Service



# Infra-as-Code

- **Bespaard tijd**
- **Voorkomt human error**
- **Reproduceerbaar platform/applicatie**
- **Verbeteren van beveiligingsstandaard**
- **Code is uitwisselbaar**



# Infrastructure-as-Code

bij Fuga Cloud

# Cloud Terminologie

**Cloud** : verzameling van virtuele diensten

**Instance** : server in de cloud

**Volume** : virtuele schijf

**Image** : volume met een besturingssysteem

**Flavor** : specifieke uitvoeren van een instance (x RAM, x CPU, x SSD)

**Security Group** : virtuele firewall

**Bootstrap** : initialiseren van een instance

**OpenStack** : vrij beschikbare cloud platform software



# De tools

SSH, Terraform, cloud-init en OpenStack Client





# Secure Shell

ssh, Chrome Secure Shell App or putty.exe



# Terraform

Uitrollen en beheren van infrastructuur met code



# CLI

## Command Line Interface

```
$ terraform version  
Terraform v0.12.12
```



# Cloud Infrastructuur

Instances, Volumes, Networks, Security Groups, DNS records



# Providers

OpenStack, AWS, Linode, OVH, PowerDNS, Docker

```
$ cat -n webserver-1.tf
1 resource "openstack_compute_instance_v2" "webserver_1" {
2
3     name          = "webserver-1"
4
5     flavor_name   = "c2.medium"
6
7     image_name    = "Ubuntu 18.04 LTS"
8
9     network {
10         name = "public"
11     }
12
13     security_groups = ["default"]
14
15     user_data = "${data.template_cloudinit_config.webserver.rendered}"
16
17 }
```



# State

Concept van gewenste staat (infra in code) tegen de live infrastructuur

```
$ terraform plan
Refreshing Terraform state in-memory prior to plan...

Terraform will perform the following actions: ...

Plan: 1 to add, 0 to change, 0 to destroy.
```



# Deterministisch

“Een deterministisch proces is een opeenvolging van uitkomsten die een causaal verband hebben, toeval speelt geen rol. **Als de beginvoorwaarden gelijk zijn, dan zal de uitkomst ook altijd gelijk zijn.**”





# Waarom Terraform?

Platform onafhankelijk en ondersteund meer dan 200 providers



# cloud-init

Automatische configuratie van cloud instances



# First Boot

cloud-init service in de instance polt het cloud platform voor user-data



# User-data

scripts en cloud-config

```
% cat -n user_data.yaml
1 #cloud-config
2
3 packages:
4   - nginx
5
6 write_files:
7   - path: /var/www/html/index.html
8     content: |
9       <html>
10         <head>
11           <title>IT=Alkmaar</title>
12         </head>
13         <body>
14           <h1>Welkom bij de Infra is Code workshop van Fuga Cloud!</h1>
15         </body>
16       </html>
```



# Operating System support

Meeste Linux-distributies, Windows, macOS en sommige UNIX/BSDs



# Waarom cloud-init?

cloud onafhankelijk, eenvoudig en first-boot



# OpenStack Client

Uitrollen en beheren van infrastructuur op OpenStack





# CLI

## Command Line Interface

```
$ openstack --version  
openstack 3.19.0
```

# Cloud Terminologie

**Cloud:** verzameling van virtuele diensten

**Instance:** server in de cloud

**Volume:** virtuele schijf

**Image:** volume met een besturingssysteem

**Security Group:** virtuele firewall

**Flavor:** specifieke uitvoeren van een instance (x RAM, x CPU, x SSD)

**Bootstrap:** initialiseren van een instance

**OpenStack:** vrij beschikbaar cloud platform



# Aan de slag!

Open een terminal

## Stap 1: login met je Fuga Cloud workshop account

```
$ ssh fugaXX@workshop.dev.fuga.cloud
#
# Fuga Workshop Server
#
Last login:
fugaXX@workshop:~$
```

## Stap 2: bekijk de code

```
$ cd ~/workshop

$ cat -n main.tf
 1 variable image_name {
 2     default = "Ubuntu 18.04 LTS"
 3 }
 4
 5 variable flavor_name {
 6     default = "t2.tiny"
 7 }
...

$ cat -n user_data.yaml
 1 #cloud-config
 2 runcmd:
 3     - [ sh, -c, 'echo ubuntu:it=alkmaar | chpasswd' ]
...
```

## Stap 3: cloud instance uitrollen

```
$ cd ~/workshop
$ terraform init
$ terraform plan
$ terraform apply --auto-approve
...
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

cloud_instance_floating_ip = 185.78.x.x
cloud_instance_id = <uuid>
cloud_instance_name = fugaXX-bionic
```

## Stap 4: inloggen op console

```
$ terraform output cloud_instance_name  
<name>
```

```
$ openstack server list
```

ID	Name	Status
<uuid>	<name>	ACTIVE

```
$ openstack console url show <name>
```

Field	Value
type	novnc
url	<a href="https://novncproxy.api.ams.fuga.cloud/vnc_auto.html?token=&lt;token&gt;">https://novncproxy.api.ams.fuga.cloud/vnc_auto.html?token=&lt;token&gt;</a>

Ubuntu 18.04 LTS ubuntu tty1

ubuntu login: \_



## Stap 4: inloggen op console

```
$ cat -n ~/workshop/user_data.yaml
1  #cloud-config
2  runcmd:
3    - [ sh, -c, 'echo ubuntu:it=alkmaar | chpasswd' ]
...
```

## Stap 5: ping de cloud instance

```
$ terraform output cloud_instance_floating_ip
185.78.x.x

$ ping -c 1 -w 2 185.78.x.x
PING 185.78.x.x (185.78.x.x) 56(84) bytes of data.

--- 185.78.x.x ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1022ms
```



# Security Groups

Het zijn net firewalls en standaard houden ze alles tegen

## Stap 6: security group rules toevoegen

```
$ cd ~/workshop

$ vim main.tf
# "${openstack_compute_secgroup_v2.secgroup_1.name}",

#resource "openstack_compute_secgroup_v2" "secgroup_1" {
#  name      = "${var.prefix}secgroup"
#  description = "from all allow SSH access"
#
#  rule {
#    from_port = -1
#    to_port   = -1
#  }
#
#  ...
#}

$ terraform plan
...
Plan: 1 to add, 1 to change, 0 to destroy.

$ terraform apply --auto-approve
...
Apply complete! Resources: 1 added, 1 changed, 0 destroyed.
```

Verwijder alle “#” karakters!

## Stap 6: security group rules toevoegen

```
$ terraform output cloud_instance_floating_ip  
185.78.x.x
```

```
$ ping -c 1 -w 2 185.78.x.x  
PING 185.78.196.22 (185.78.196.22) 56(84) bytes of data.  
64 bytes from 185.78.196.22: icmp_seq=1 ttl=63 time=1.20 ms
```

```
--- 185.78.196.22 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 1.208/1.208/1.208/0.000 ms
```

## Stap 7: inloggen met SSH


```
$ ssh ubuntu@185.78.x.x  
Warning: Permanently added '185.78.197.51' (ECDSA) to the list of known hosts.  
ubuntu@185.78.197.51: Permission denied (publickey).
```



# SSH public key

Veiliger dan wachtwoorden en meerdere keys per account

## Stap 7: inloggen met SSH



```
$ cd ~/workshop

$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAABC...

$ vim user_data.yaml
#cloud-config
runcmd:
  - [ sh, -c, 'echo ubuntu:it=alkmaar | chpasswd' ]

ssh_authorized_keys:
  - ssh-rsa AAAABC...
...

$ terraform plan
...
Plan: 1 to add, 0 to change, 1 to destroy.

$ terraform apply --auto-approve
```

Kopieer en plak "ssh-rsa ..."!



## Stap 7: inloggen met SSH

```
$ terraform output myfirst_floating_ipv4
```

```
185.78.x.x
```

```
$ ssh utuntu@185.78.x.x
```

```
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-45-generic x86_64)
```

```
ubuntu@fugaXX-bionic:~/ $ curl http://ifconfig.me
```

```
185.78.x.x
```

```
ubuntu@fugaXX-bionic:~/ $ exit
```



# Software Installeren

Met code, niet handmatig, want deterministisch

## Stap 8: web-server installeren

```
$ cd ~/workshop

$ vim user_data.yaml
...
#packages:
# - nginx-light
#write_files:
# - path: /var/www/html/index.html
#   content: |
#     <html>
#       <head>
#         <title>IT=Alkmaar</title>

$ terraform plan
...
Plan: 1 to add, 0 to change, 1 to destroy.

$ terraform apply --auto-approve
...
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
```

Verwijder alle “#” karakters!

## Stap 9: wachten op de cloud-init service

```
$ terraform output cloud_instance_floating_ip
185.78.x.x

$ ssh ubuntu@185.78.x.x

ubuntu@fugaXX-bionic:~/$ sudo cloud-init status --wait
.....
status: done

ubuntu@fugaXX-bionic:~/$ exit

$ curl -i http://185.78.x.x
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)

<html>
  <head>
    <title>IT=Alkmaar</title>
```

## Stap 10: infrastructuur verwijderen

```
$ cd ~/workshop
```

```
$ terraform destroy --auto-approve
```

```
Destroy complete! Resources: 2 destroyed.
```

```
$ openstack server list -f json
```

```
[]
```



# Gratis Uitproberen?

Registreren - [my.fuga.cloud](https://my.fuga.cloud)

Workshop - [fuga.cloud/infra-as-code](https://fuga.cloud/infra-as-code)