

DevOps



Infrastructuur trends

- Systemen gebruiken meerdere servers, voor schaalbaarheid en fout-bestendigheid
- O.a. door goed gebruik DTAP hebben developers vaker te maken met infrastructuur
- Ontstaan grote verzameling software-tools, die het configureren van infra automatiseren

DevOps

Combinatie van development en system administration

“a cross-disciplinary community of practice dedicated to the study of building, evolving and operating rapidly-changing resilient systems at scale.”

QA

continuous deployment

snelle recovery

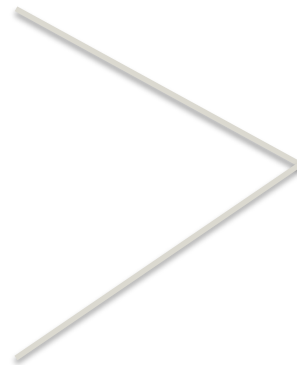
korte fix-tijden

D=P

Agile

Lean

12-Factor



verschillende tools

slim ingezet

DevOps tools

- **Configuration Management**

- Puppet
- Chef
- Cfengine
- SaltStack
- Ansible

- **Log management**

- rsyslog
- Logstash
- Graylog2

- **Deployment**

- Fabric
- Capistrano

- **Central Controller**

- Func
- Fabric

- **Continuous Integration**

- Jenkins

- **Create and Provision**

- Packer
- Terraform

- **Monitoring and Measurement**

- Graphite
- StatsD
- Collectd
- Nagios
- Ganglia
- Sensu

- **Virtual Environment or Dev Environment management**

- OpenStack
- Vagrant

- **Basic Infrastructure**

- Docker
- Apache Mesos
- Kubernetes
- CoreOS
- Serf
- Consul
- Flynn

- **Test**

- Server Spec



VAGRANT

Vagrant



**Lightweight, reproduceerbare,
portable ontwikkelomgevingen**

Scripted virtual machine
management

Vagrant



```
$ vagrant init hashicorp/precise32  
$ vagrant up
```

```
$ vagrant ssh
```

Vagrant



```
$ vagrant up
...
$ vagrant ssh
...
vagrant@precise32:~$ ls /vagrant
Vagrantfile
```


Vagrantfile



```
Vagrant.configure("2") do |config|  
  config.vm.box = "hashicorp/precise32"  
  config.vm.provision :shell, path: "bootstrap.sh"  
end
```

```
#!/usr/bin/env bash  
  
apt-get update  
apt-get install -y apache2  
if ! [ -L /var/www ]; then  
  rm -rf /var/www  
  ln -fs /vagrant /var/www  
fi
```

```
$ vagrant ssh  
...  
vagrant@precise32:~$ wget -qO- 127.0.0.  
1
```

Vagrant



- `vagrant init`
- `vagrant up`
- `vagrant ssh`
- `vagrant suspend`
- `vagrant destroy`
- `vagrant status`
- e.v.a.

Vagrant



- Vagrantfile
- Boxes
 - Linux images
 - LAMP, Puppet, e.a. pre-installed
- Providers
 - standaard: VirtualBox
 - Vmware, Hyper-V, Docker
- Provisioning
 - standaard: shell
 - Chef, Puppet, Ansible, Docker

Waarom Vagrant?

- Reproduceerbare ontwikkelomgevingen
- Low-impact: Vagrantfile bij code in repository

Waarom niet?

- Zwaarder dan een pure container-omgeving (zoals Docker)





CHEF™

Chef



IT automation

Automatisch en dynamisch
machines configureren

“High-level bash”

Chef – recipees



Infrastructure == code

```
1 | file 'motd' do
2 |   content 'hello world'
3 | end
```

```
$ chef-apply hello.rb
Recipe: (chef-apply cookbook)::(chef-apply recipe)
  * file[motd] action create
    - create new file motd
    - update content in file motd from none to de031d
    --- motd          2014-05-13 14:52:54.025253948 -0700
    +++ /tmp/.motd20140513-3474-5263n1 2014-05-13 14:52:54
    @@ -1 +1,2 @@
    +hello world
```

Chef – recipees



```
$ chef-apply hello.rb
Recipe: (chef-apply cookbook)::(chef-apply recipe)
  * file[motd] action create (up to date)
```

```
$ echo 'hello robots' > motd
```

```
$ chef-apply hello.rb
Recipe: (chef-apply cookbook)::(chef-apply recipe)
  * file[motd] action create
    - update content in file motd from 9b0c18 to b1522f
    --- motd          2014-05-13 15:03:47.638770524 -0700
    +++ /tmp/.motd20140513-4170-130uqxh  2014-05-13 15:04:47.638770524 -0700
    @@ -1,2 +1,2 @@
    -hello robots
    +hello chef
```


Chef – recipees



```
1 | package 'apache2'
```

```
$ sudo chef-apply webserver.rb
Recipe: (chef-apply cookbook)::(chef-apply recipe)
  * package[apache2] action install
    - install version 2.4.6-2ubuntu2.2 of package apache2
```

```
$ sudo chef-apply webserver.rb
Recipe: (chef-apply cookbook)::(chef-apply recipe)
  * package[apache2] action install (up to date)
```

Chef – recipees



```
1  package 'apache2'
2
3  service 'apache2' do
4    action [:start, :enable]
5  end
6
7  file '/var/www/html/index.html' do
8    content '<html>
9      <body>
10        <h1>hello world</h1>
11      </body>
12    </html>'
13  end
```

Chef – cookbooks



Cookbook:

- recipees
- files
- templates
- attribute values
- etc.

Chef – server



Chef



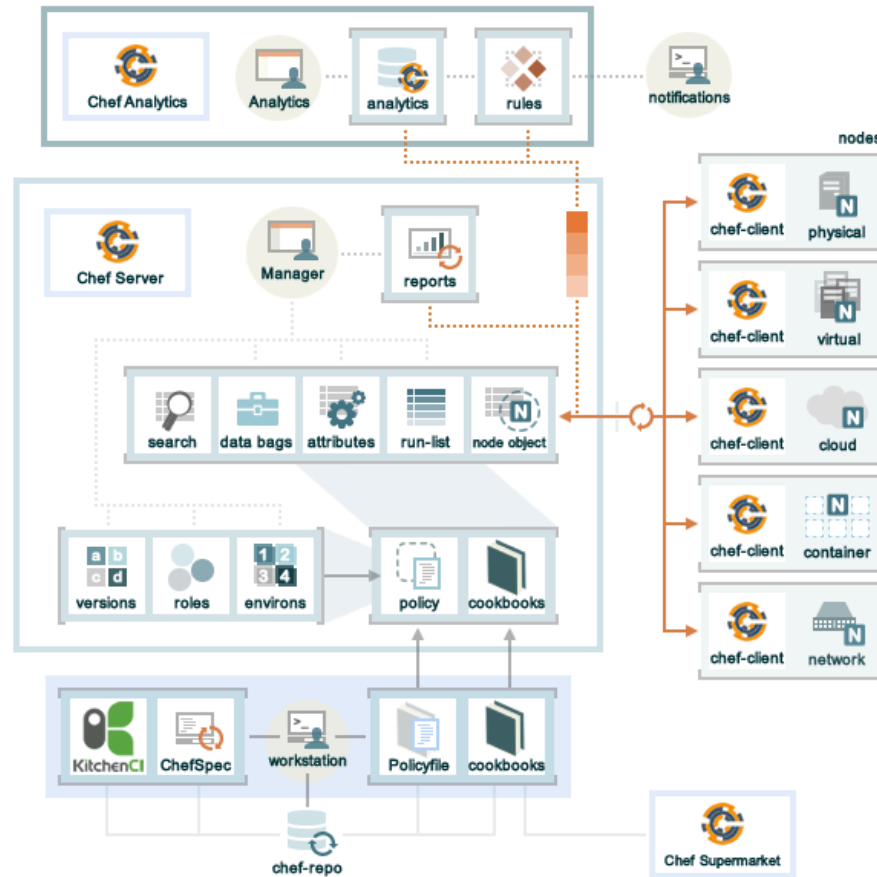
Onderdelen

- History
 - recipees, nodes
- Reporting
- Analytics
- Testing
- Rules
- enz.

Tools

- *Knife*: interact with server
- *Berkshelf*: cookbooks vanaf het web
- *Kitchen*: test cookbooks
- enz.

Chef – systeem



Chef



Voorbeeld: Tie Kinetix

- Twee vagrant machines
 - front-end (Ubuntu, Node.js, AngularJS, Nginx)
 - back-end (Ubuntu, Java Play, MongoDB, ElasticSearch)
- Provisioning: Chef
 - 2x2 recipees voor front/back x dev/server
- Developer:

<pre>mkdir backend && cd backend</pre>	<pre>mkdir frontend && cd frontend</pre>
<pre>git clone <backend-repo></pre>	<pre>git clone <frontend-repo></pre>
<pre>vagrant up</pre>	<pre>vagrant up</pre>

Waarom Chef?

- Krachtige, complete configuratie-automatisering

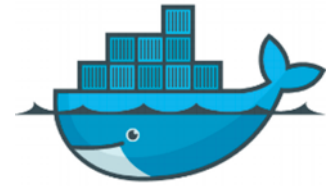
Waarom niet?

- Complex
 - Bash
- Web recipees veranderen





Docker

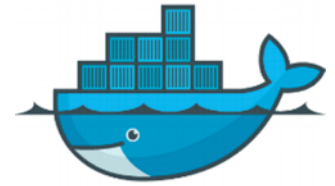


Een open platform voor developer en sysadmins om gedistribueerde applicaties te bouwen, versturen en runnen.

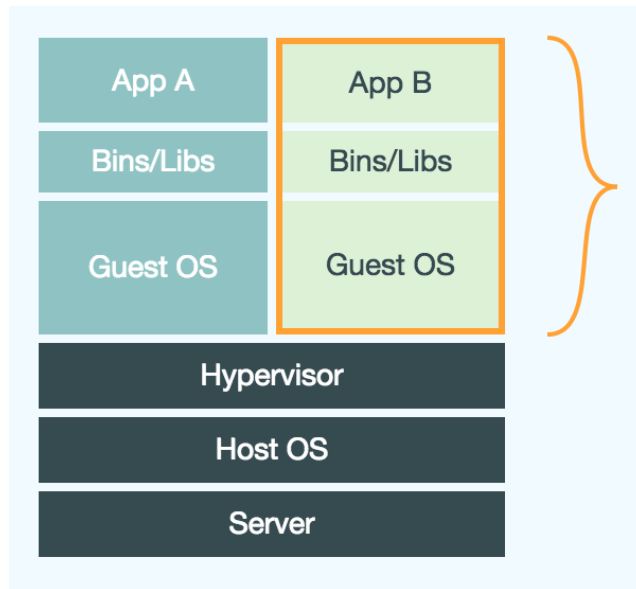
Application packager

Een lightweight per-process VM

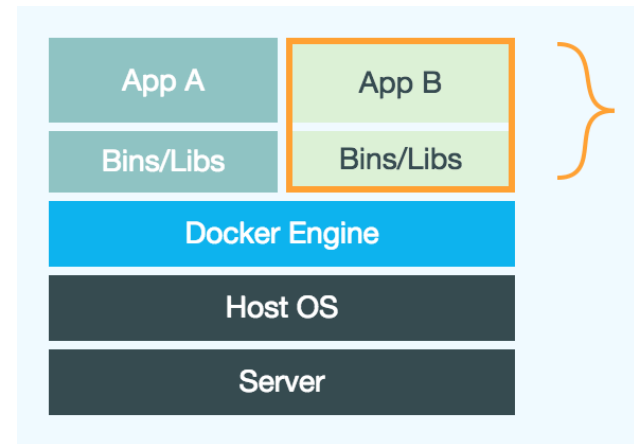
Docker



Een docker image is een VM image dat één applicatie runt, maar dan zonder het OS

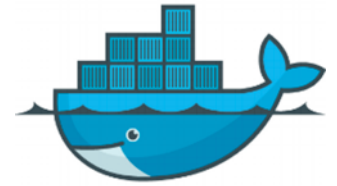


VM



Docker app

Docker

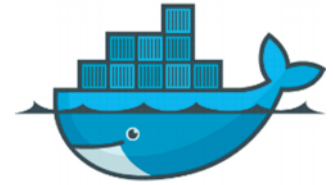


“Dockerized” applicatie:

- volledig self-contained
- platform onafhankelijk
- portable
- “works on my machine”
- deployment triviaal

```
apt-get install docker.io  
docker run ourrepo/ourapp
```

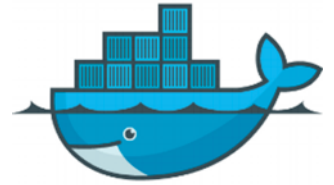
Docker



Docker is:

- git-achtig
 - `docker pull`
 - `docker commit`
- extreem snel
 - Linux containers
- ideaal voor schalen
- geschreven in Go
- Docker Hub
 - Zeer veel images beschikbaar
- eigen repo

Docker

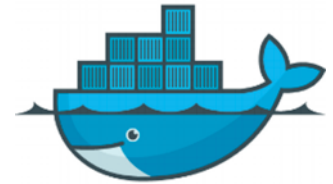


```
$ sudo docker run ubuntu:14.04 /bin/echo 'Hello world'  
Hello world
```

image \approx image, snapshot

container \approx instance, machine

Docker



```
$ sudo docker run -d ubuntu:14.04 /bin/sh -c "while true; do echo hello world; sleep 1; done"
1e5535038e285177d5214659a068137486f96ee5c2e85a4ac52dc83f2ebe4147
```

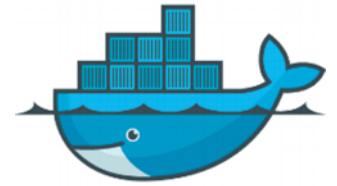
```
$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
1e5535038e28	ubuntu:14.04	/bin/sh -c 'while tr	2 minutes ago	Up 1 minute		insane_ba

```
$ sudo docker logs insane_babbage
hello world
hello world
hello world
. . .
```

```
$ sudo docker stop insane_babbage
insane_babbage
```

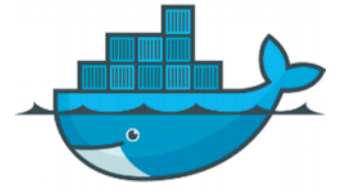
Docker



```
$ sudo docker run -d -p 5000:5000 training/webapp python app.py
```

```
$ sudo docker logs -f nostalgic_morse
* Running on http://0.0.0.0:5000/
10.0.2.2 - - [23/May/2014 20:16:31] "GET / HTTP/1.1" 200 -
10.0.2.2 - - [23/May/2014 20:16:31] "GET /favicon.ico HTTP/1.1" 404 -
```


Docker



Docker image aanpassen:

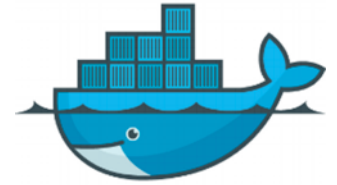
1. start
2. shell-login
3. je ding doen
4. exit
5. commit
6. push

Beter: DockerFile

- declaratief

```
# This is a comment
FROM ubuntu:14.04
MAINTAINER Kate Smith <ksmith@example.com>
RUN apt-get update && apt-get install -y ruby ruby-dev
RUN gem install sinatra
```

Docker – Tooling



In ontwikkeling...

Machine – installeer Docker lokaal, in cloud of in data center en connect

Swarm – clustering

Compose – start en stop multi-container applicaties



Voorbeeld TIE Kinetix



- Twee servers (minimaal) in productie
 - front-end (Ubuntu, Node.js, AngularJS, Nginx)
 - back-end (Ubuntu, Java Play, MongoDB, ElasticSearch)
- centrale T, A en P omgeving
- 5 developers met eigen D omgeving
- 1 update per twee weken (volgens scrum)
- Verschillende malen problemen met verschillen tussen de omgevingen

Docker

Voorbeeld: Tie Kinetix



- Docker image: Nginx
- Docker image: frontend (AngularJS, Node.js, grunt)
- Docker image: service layer (Java Play, Scala)
- Docker image: MongoDB (OTS)
- Docker image: ElasticSearch (OTS)
- Eigen repo
- `docker-compose up...?`
- Dev-container vs. prod-container...?
- Klein MongoDB/OS X dingetje

Waarom Docker?

- Proces-isolatie
- Lightweight
- P=D
- Twelve-Factor App ten voeten uit

Waarom niet?

- Docker nog jong
- Sommige applicaties misschien niet docker-compatible...?
- Rocket?

