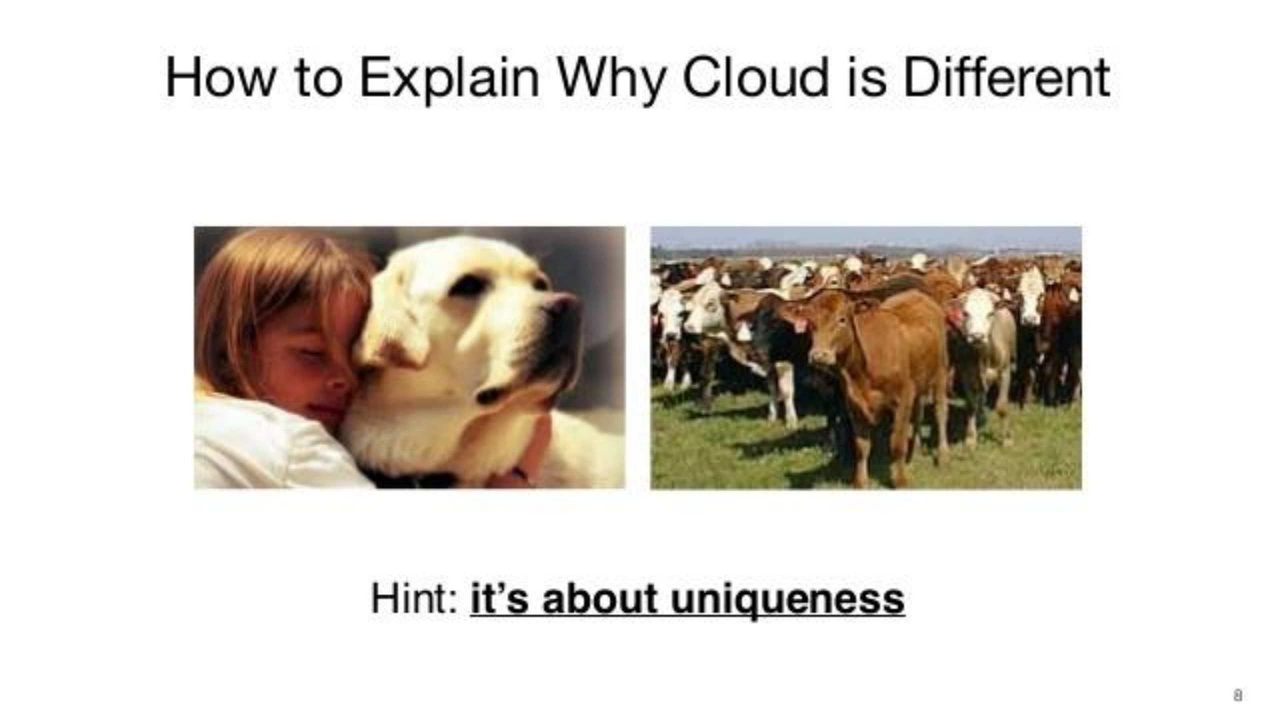
IK WIL

Cloud Native

vijfhart
IT-OPLEIDINGEN

Cloud Native

Introduction



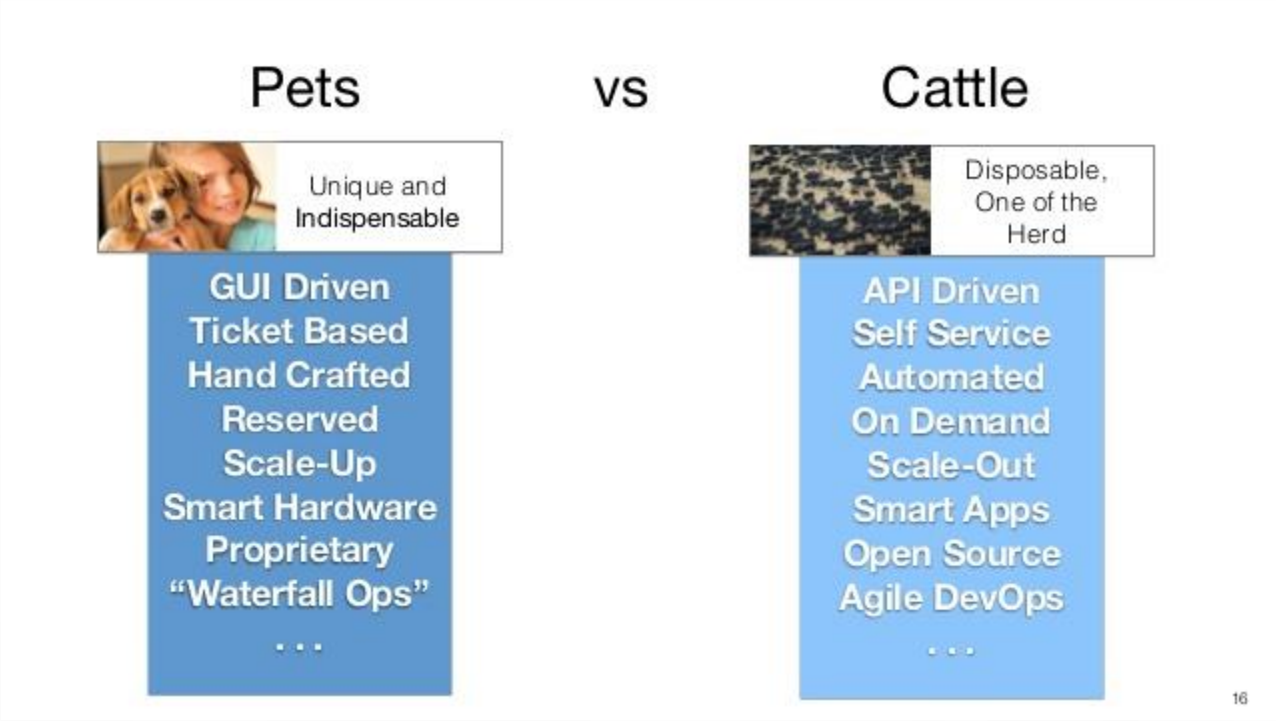
How to Explain Why Cloud is Different

Hint: **it's about uniqueness**

8

Cloud Native

Introduction



Pets	vs	Cattle
Unique and Indispensable		Disposable, One of the Herd
GUI Driven		API Driven
Ticket Based		Self Service
Hand Crafted		Automated
Reserved		On Demand
Scale-Up		Scale-Out
Smart Hardware		Smart Apps
Proprietary		Open Source
“Waterfall Ops”		Agile DevOps
. . .		. . .

16

Cloud Native

Definition

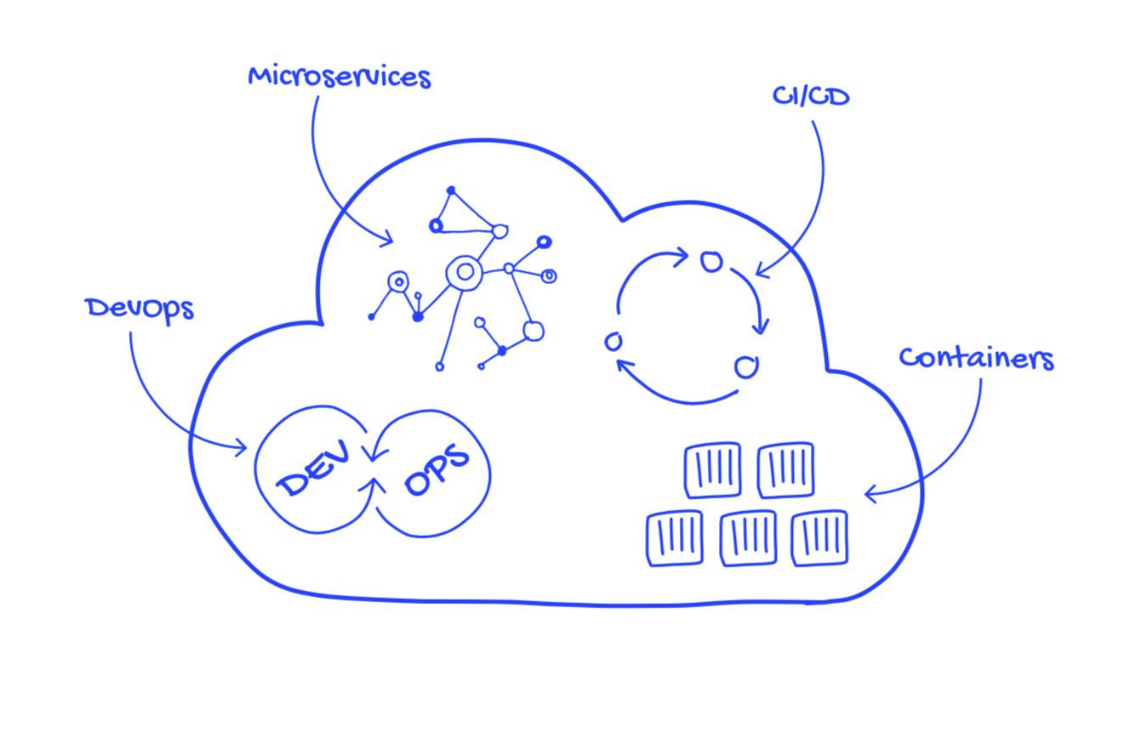
Cloud native computing is an approach in software development that utilizes cloud computing to "build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds".

Technologies such as **containers**, **microservices**, **serverless functions** and **immutable infrastructure**, deployed via declarative code are common elements of this architectural style.

1. "CNCF Cloud Native Definition v1.0". GitHub(CNCF). 2018-06-11. Retrieved 2020-05-15.
2. "What is Cloud-Native? Is It Hype or the Future of Software Development?". Stackify. 2018-02-07. Retrieved 2019-08-29.
3. "What is cloud native computing? - Open-Source Insider". www.computerweekly.com. Retrieved 2019-08-29.

Cloud Native

Definition



https://info.container-solutions.com/introduction-to-cloud-native

Micro services

- SOA variant
- Fine grained
- Loosely coupled
- Lightweight (HTTP)

SOA Principles of Service Design

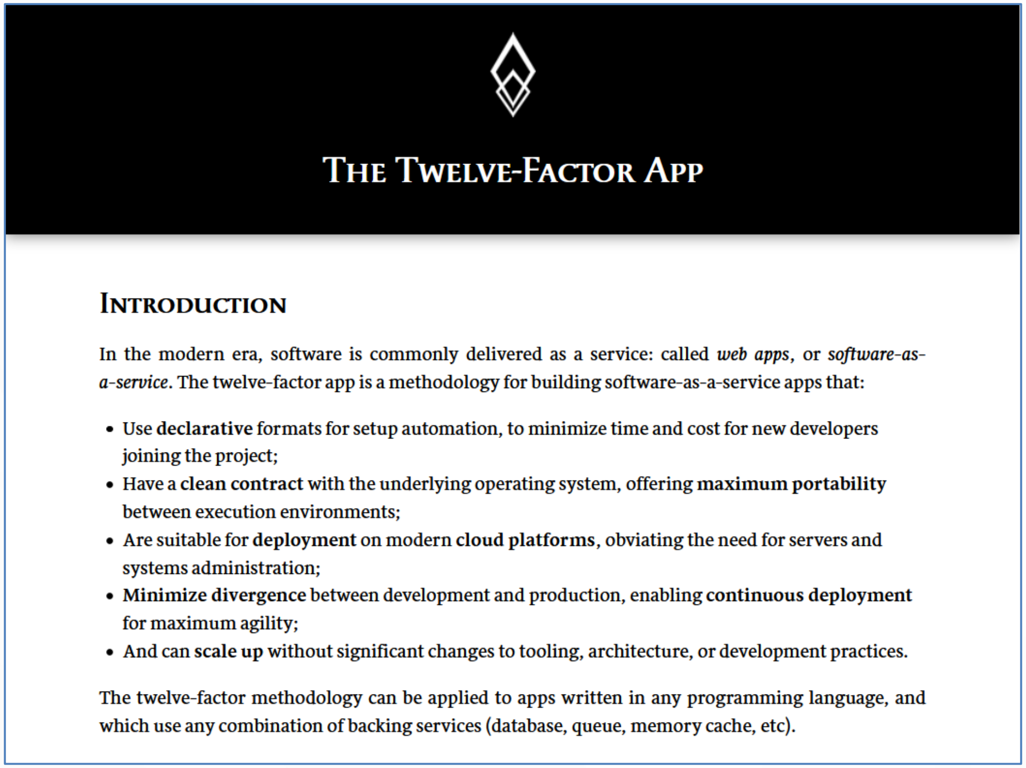
- Standardized Service Contract
- Loosely Coupled
- Service Reusability
- Service Autonomy
- Service Statelessness
- Service Discoverability
- Service Composability
- Service Interoperability

[SOA Principles of Service Design; Erl Thomas]

Good luck!

2200 pages

Twelve Factor App



THE TWELVE-FACTOR APP

INTRODUCTION

In the modern era, software is commonly delivered as a service: called *web apps*, or *software-as-a-service*. The twelve-factor app is a methodology for building software-as-a-service apps that:

- Use **declarative** formats for setup automation, to minimize time and cost for new developers joining the project;
- Have a **clean contract** with the underlying operating system, offering **maximum portability** between execution environments;
- Are suitable for **deployment** on modern **cloud platforms**, obviating the need for servers and systems administration;
- **Minimize divergence** between development and production, enabling **continuous deployment** for maximum agility;
- And can **scale up** without significant changes to tooling, architecture, or development practices.

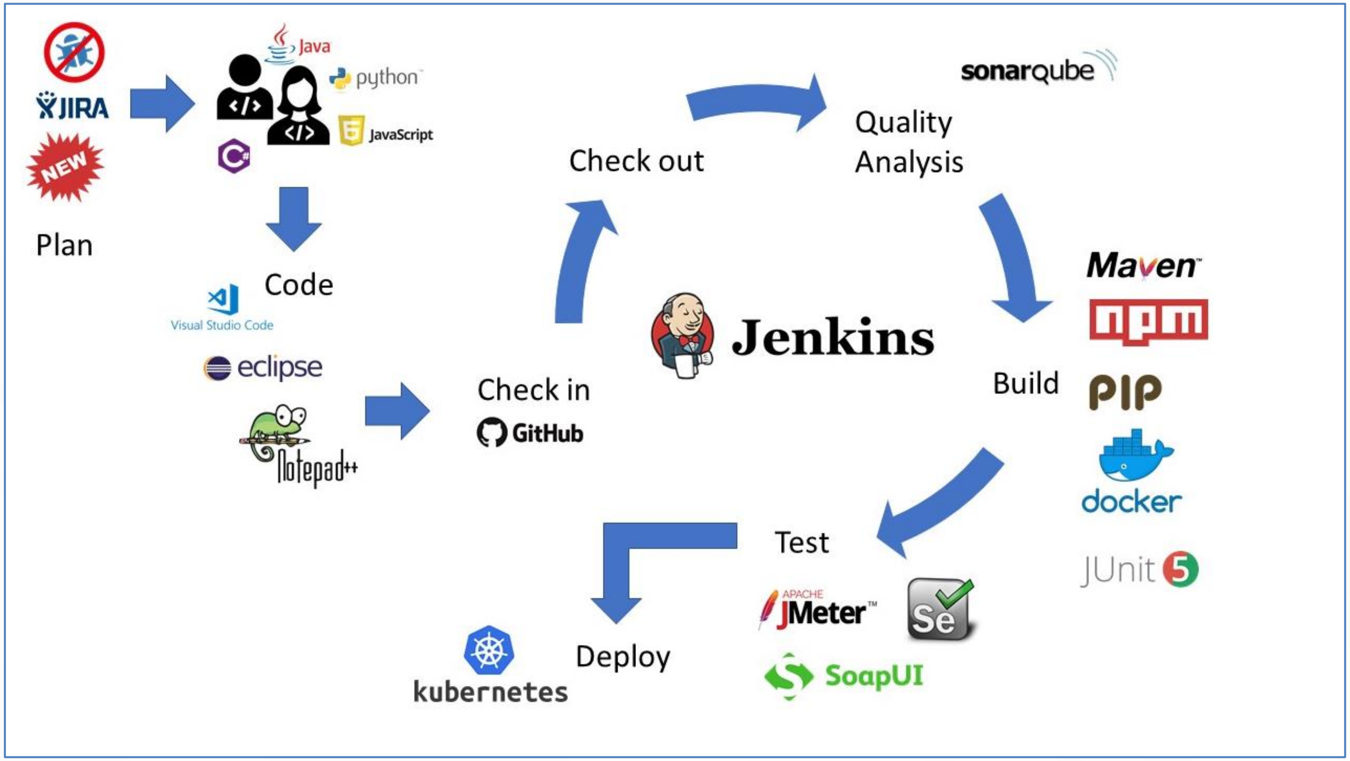
The twelve-factor methodology can be applied to apps written in any programming language, and which use any combination of backing services (database, queue, memory cache, etc).

https://12factor.net

Micro services

- Polyglot: Java, Python, Ruby, Go, C#, …
- Modular
- Parallel development
- Good fit with continuous integration and delivery

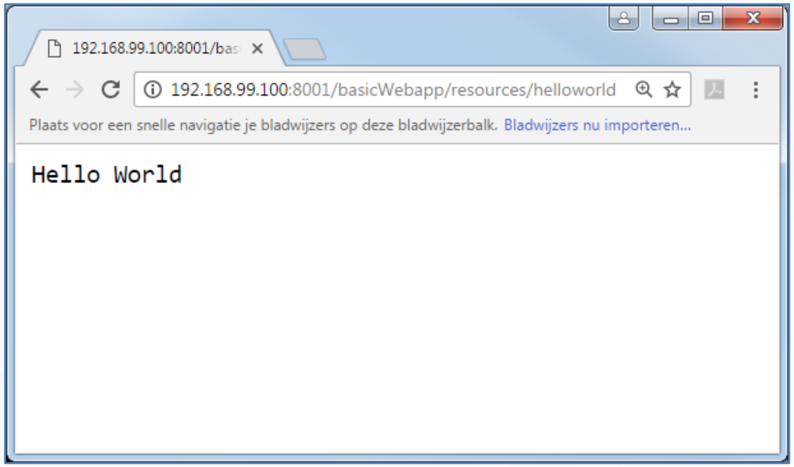
Continuous Integration/Delivery



Java micro service

```
import javax.ws.rs.*;
import javax.ws.rs.core.*;

@Path("/helloworld")
public class HelloWorldREST {
  @GET
  @Produces(MediaType.TEXT_PLAIN)
  public String sayHello() {
    return "Hello World";
  }
}
```

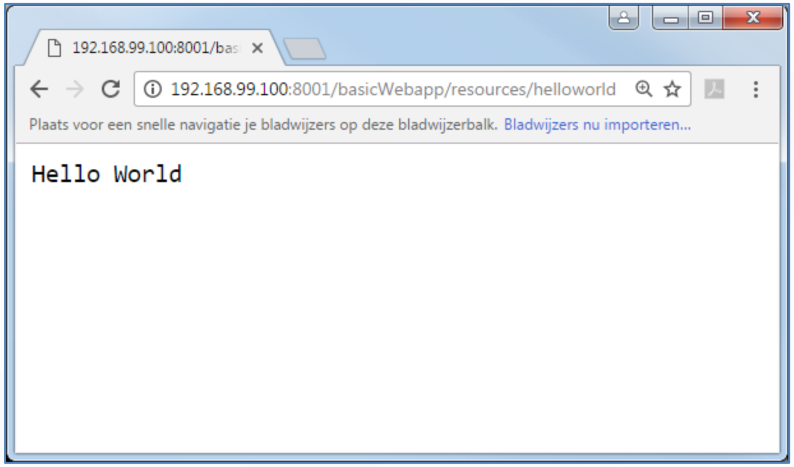

192.168.99.100:8001/bas
192.168.99.100:8001/basicWebapp/resources/helloworld
Plaats voor een snelle navigatie je bladwijzers op deze bladwijzerbalk. Bladwijzers nu importeren...

```
Hello World
```

Spring Boot micro service

```
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.bind.annotation.RequestMapping;

@RestController
public class HelloWorldSpring {
  @RequestMapping("/helloworld")
  public String sayHello() {
    return "Hello World";
  }
}
```



192.168.99.100:8001/bas
192.168.99.100:8001/basicWebapp/resources/helloworld
Plaats voor een snelle navigatie je bladwijzers op deze bladwijzerbalk. Bladwijzers nu importeren...

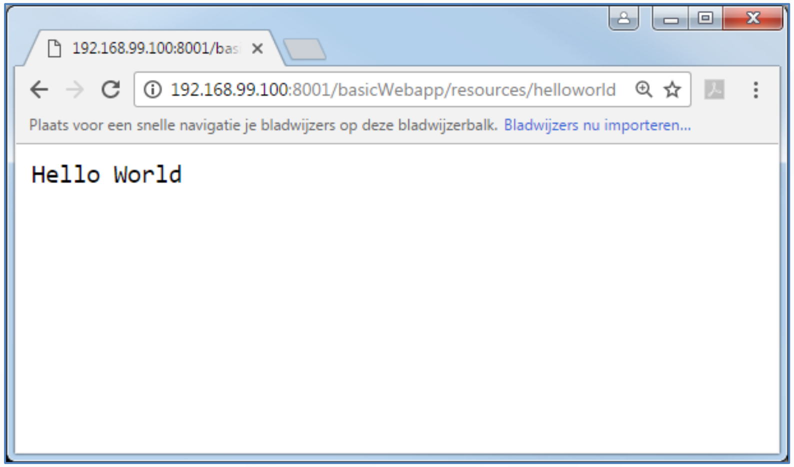
```
Hello World
```

Python Microservice

```
from flask import Flask

app = Flask(__name__)

@app.route('/helloworld')
def sayhello():
  return 'Hello World!'
```



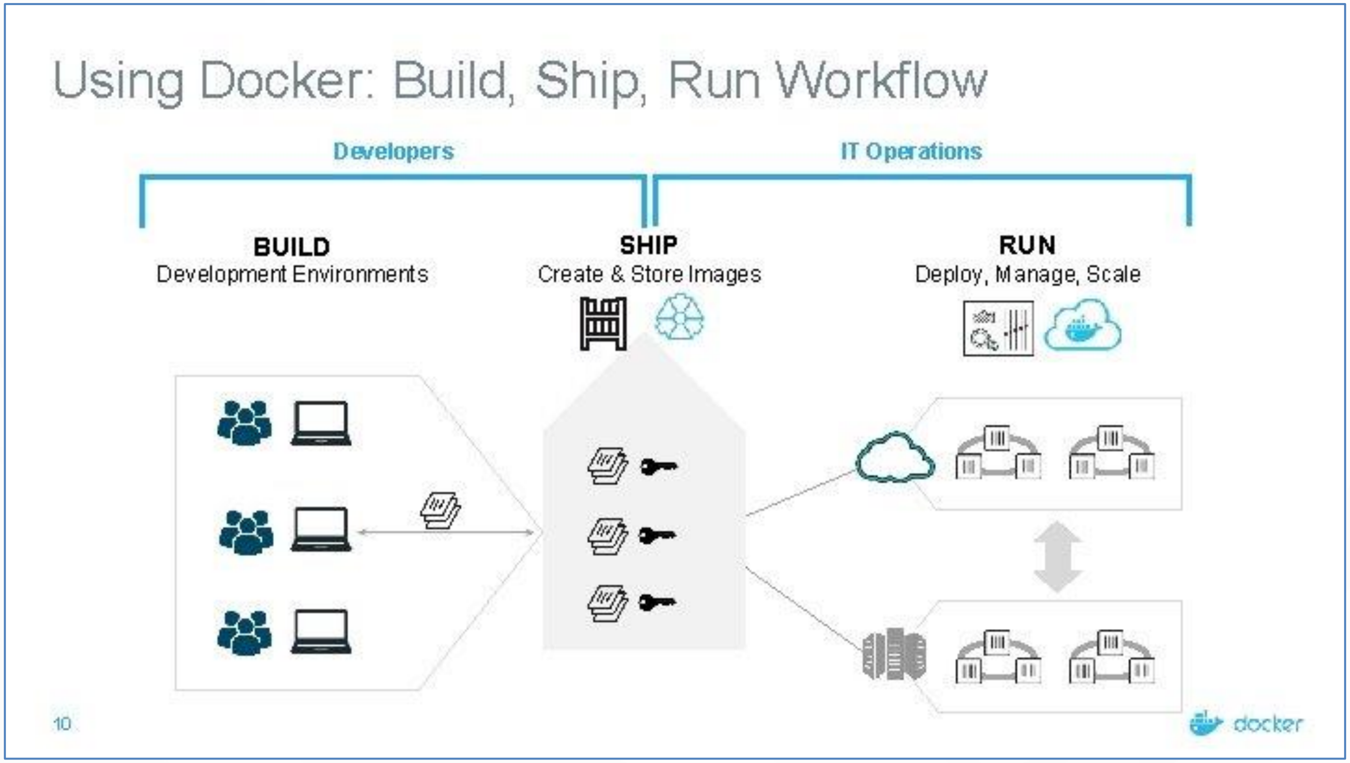
192.168.99.100:8001/basicWebapp/resources/helloworld

Plaats voor een snelle navigatie je bladwijzers op deze bladwijzerbalk. Bladwijzers nu importeren...

```
Hello World
```

Containers

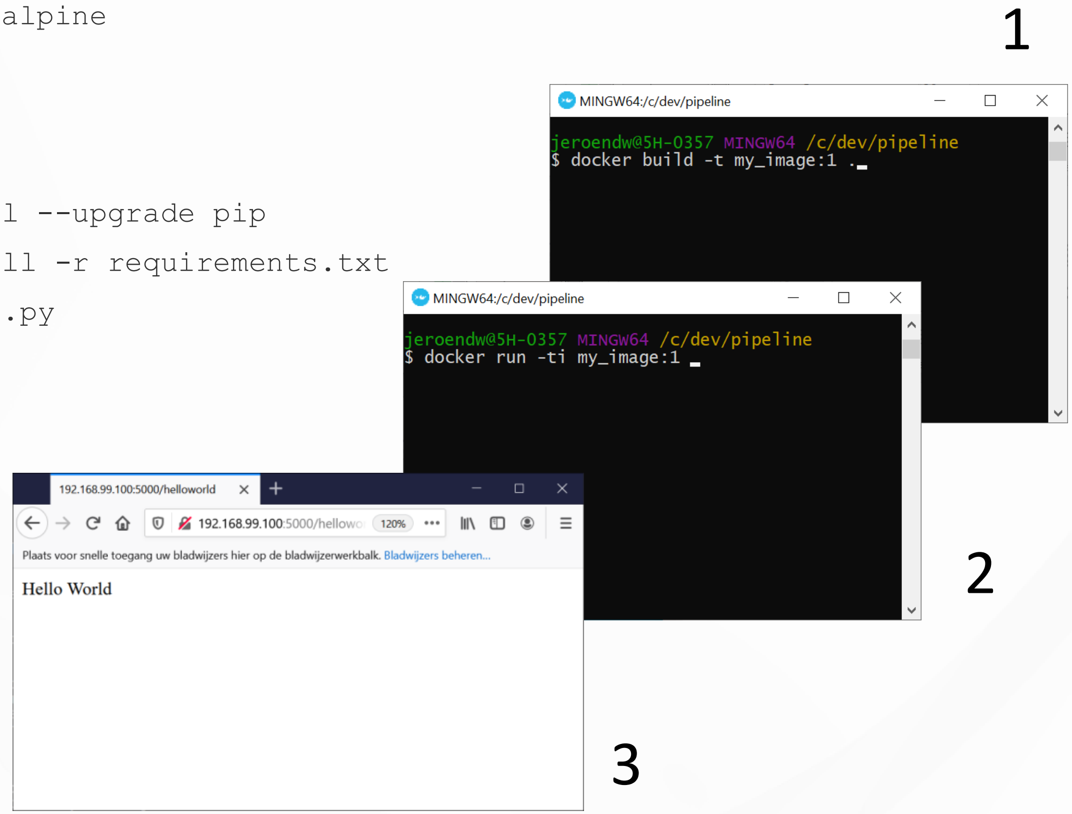
Build it, Ship it , Run it



Using Docker: Build, Ship, Run Workflow

Docker

```
FROM python:3-alpine
EXPOSE 5000
ADD . /code
WORKDIR /code
RUN pip install --upgrade pip
RUN pip3 install -r requirements.txt
CMD python app.py
```



1

```
jeroendw@5H-0357 MINGW64 /c/dev/pipeline
$ docker build -t my_image:1 .
```

2

```
jeroendw@5H-0357 MINGW64 /c/dev/pipeline
$ docker run -ti my_image:1
```

3

Plaats voor snelle toegang uw bladwijzers hier op de bladwijzerwerkbalk. Bladwijzers beheren...

Hello World

Container Orchestration

- Kubernetes
- Docker Swarm
- Apache Mesos

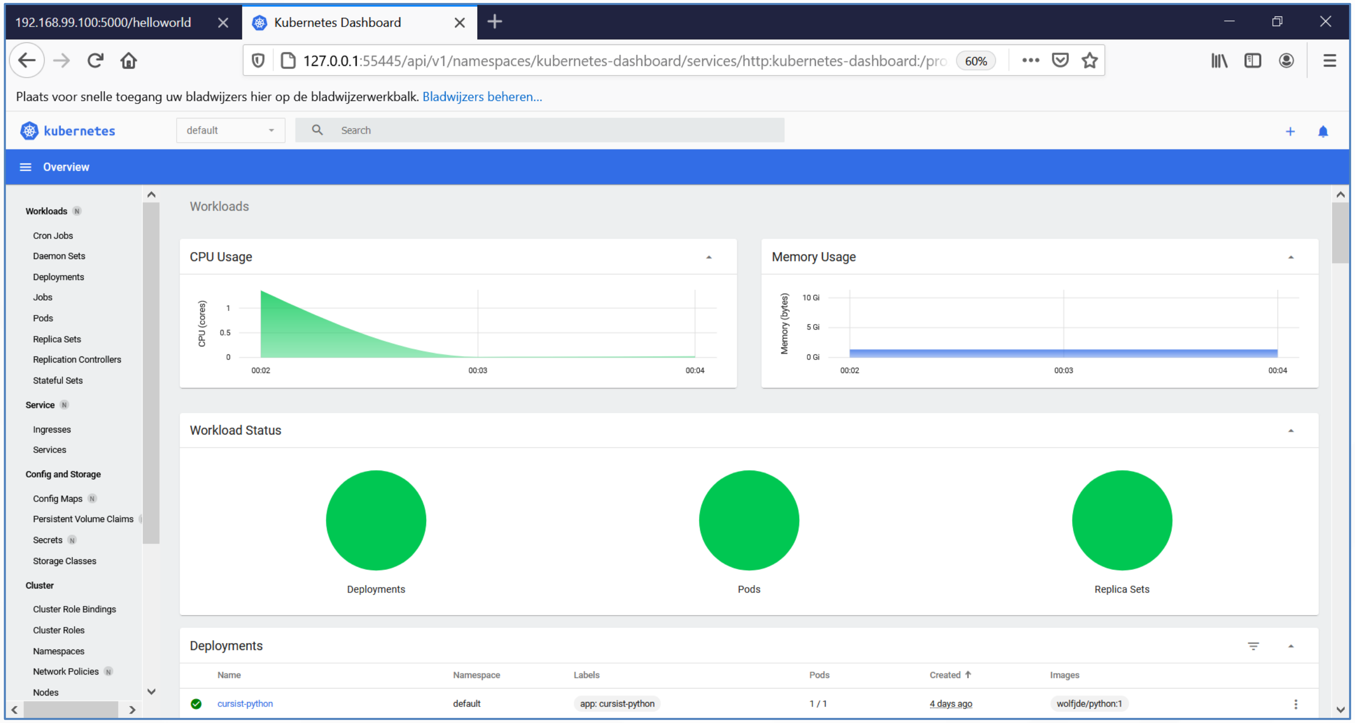
Solved all your problems. You're welcome.

I NEED TO KNOW WHY MOVING OUR APP TO THE CLOUD DIDN'T AUTOMATICALLY SOLVE ALL OUR PROBLEMS.
YOU WOULDN'T LET ME RE-ARCHITECT THE APP TO BE CLOUD-NATIVE.
JUST PUT IT IN CONTAINERS.
YOU CAN'T SOLVE A PROBLEM JUST BY SAYING TECHY THINGS.
KUBERNETES.

Dilbert.com @ScottAdamsSays

11-08-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel

Kubernetes



192.168.99.100:5000/helloworld

Kubernetes Dashboard

127.0.0.1:55445/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/pro

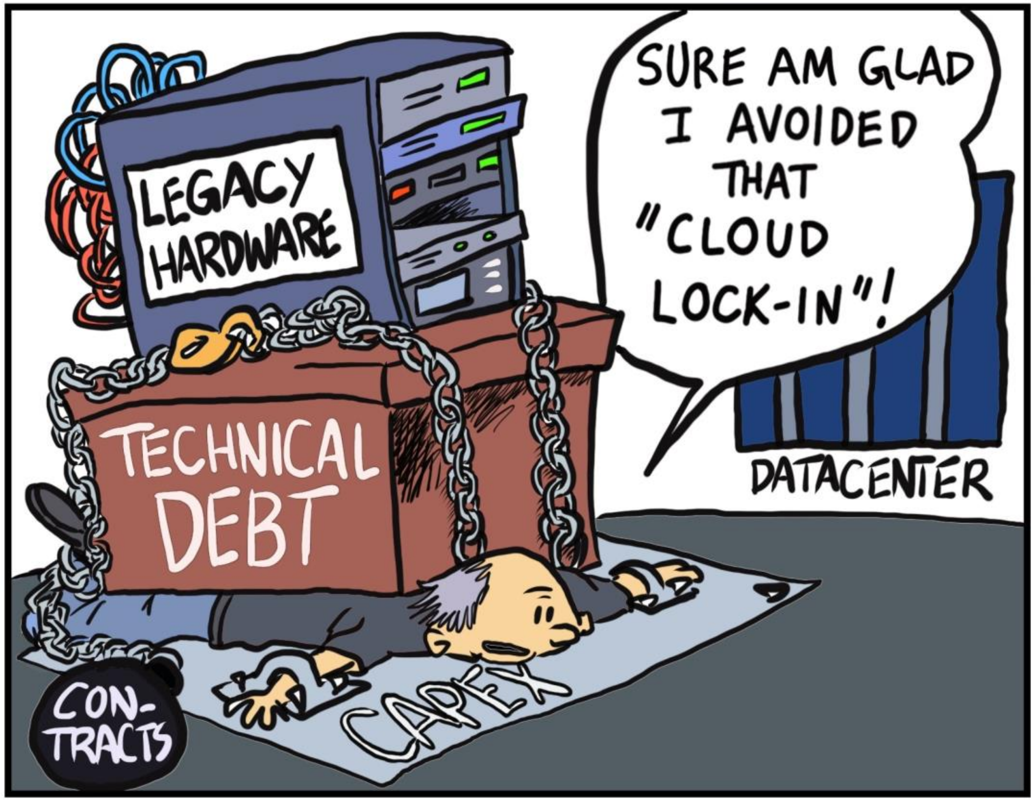
Plaats voor snelle toegang uw bladwijzers hier op de bladwijzerwerkbalk. Bladwijzers beheren…

Deployments

	Name	Namespace	Labels	Pods	Created ↑	Images
✓	cursist-python	default	app: cursist-python	1 / 1	4 days ago	wolfjde/python:1

Serverless

FaaS and Furious by Forrest Brazeal A CLOUD GURU



Vijfhart

Rokus Janssen, adviseur en accountmanager bij Vijfhart voor KPN

Heb je vragen of opmerkingen, neem dan vooral contact op met mij voor passend advies. Je kan mij bereiken via:

E: r.janssen@5hart.nl

T: 088 542 78 88

Graag tot snel!

IK WIL

Thank you!