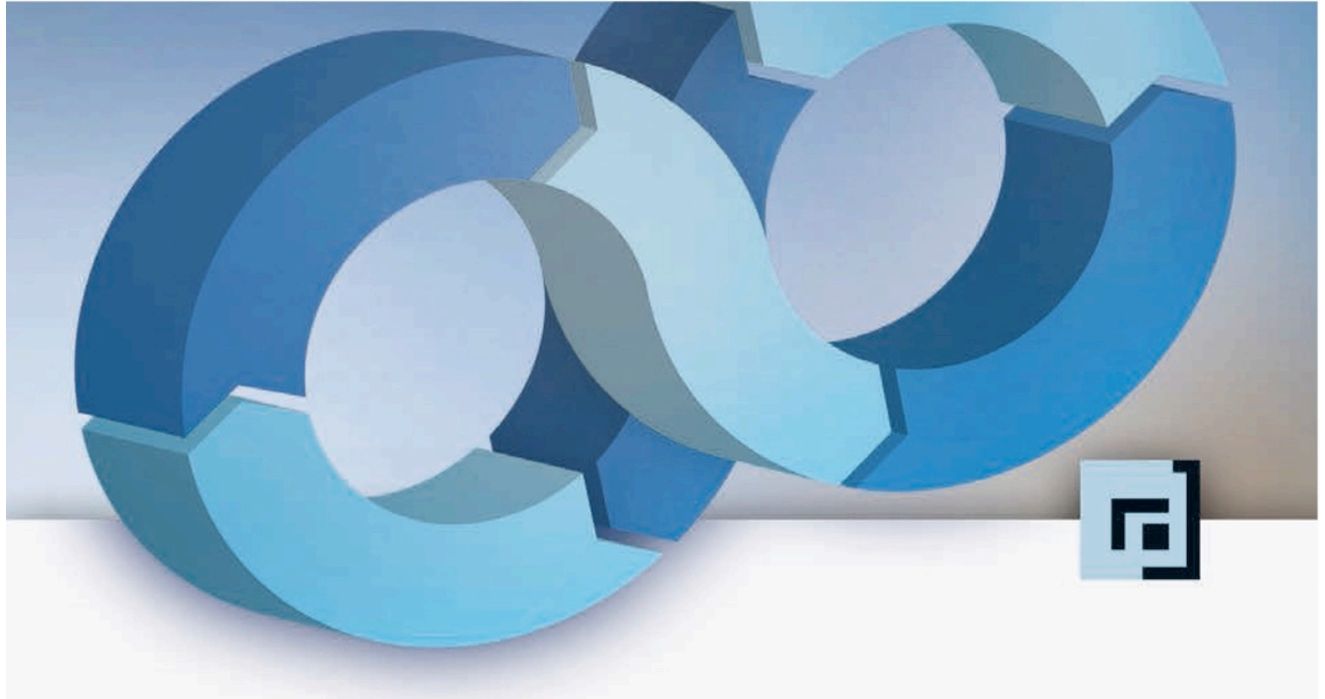
Infrastructure for Continuous Delivery & Microservices: PaaS or Docker?

Eberhard Wolff
Fellow
@ewolff



Eberhard Wolff

Continuous Delivery

Der pragmatische Einstieg

dpunkt.verlag

Eberhard Wolff

Microservices

Grundlagen flexibler Softwarearchitekturen

dpunkt.verlag

http://microservices-buch.de/

Microservices

Flexible Software Architectures

Eberhard Wolff

http://microservices-book.com/

Why This Talk?

- 2008: Google App Engine
- Google Infrastructure for the masses!

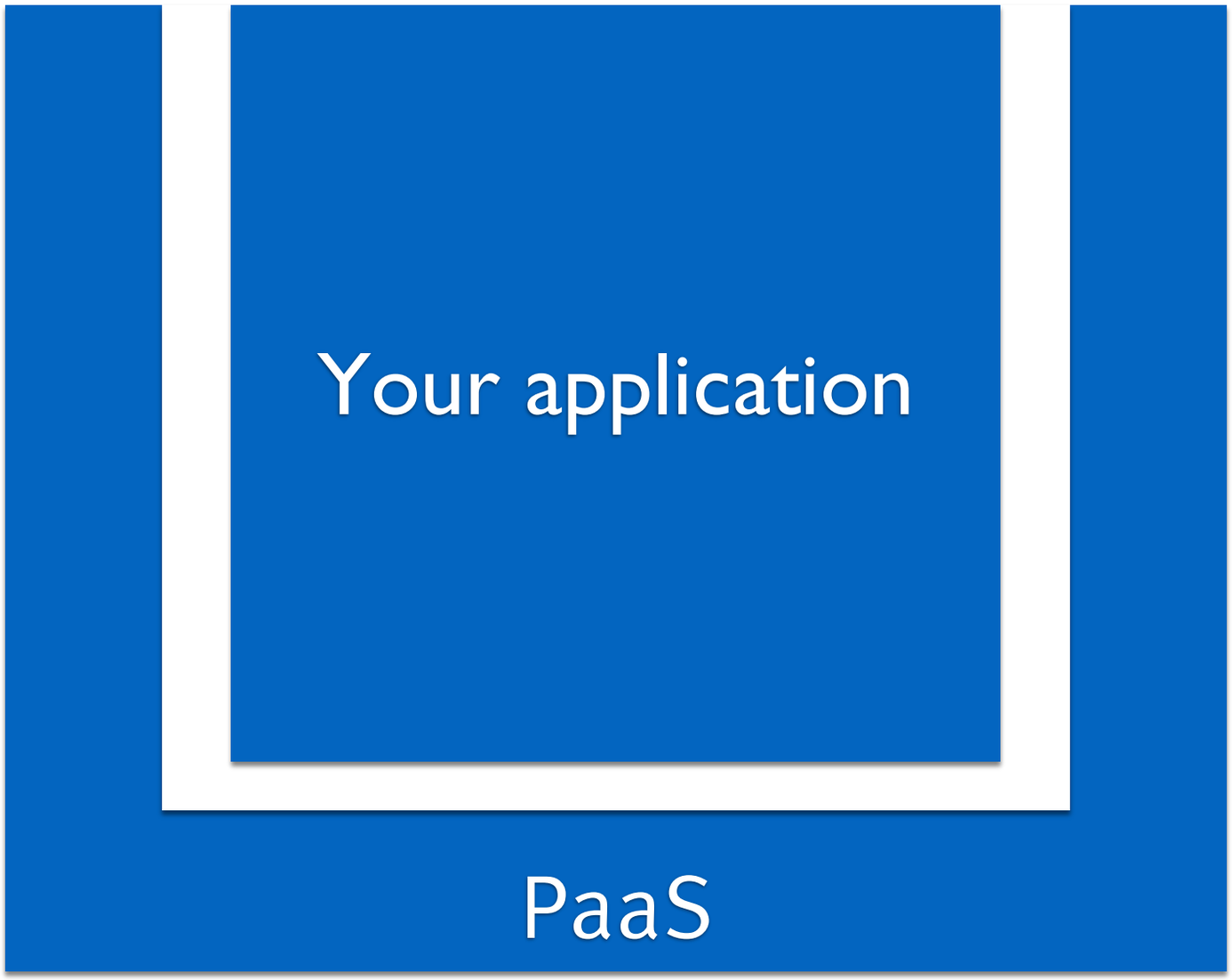
- 2010: Selling PaaS into Enterprise
- Future of application development!

- Didn‘t really take off

What is Cloud?

Cloud = Self Service

Infrastructure as a Service	Platform as a Service	Software as a Service
Virtual Servers	Virtual Application Server	Software or Service that you use
Manage Everything Yourself	Handles Scale-Out Mostly Managed by Provider	Components that you integrate into your app



Including database, scaling, monitoring, HTTP handling…

Cloud might be…

- Private / internal “on premise”
- Public “off premise”

Why PaaS?

- Scaling?
- Pay-as-you-go?
- Quickly and easily deploy applications

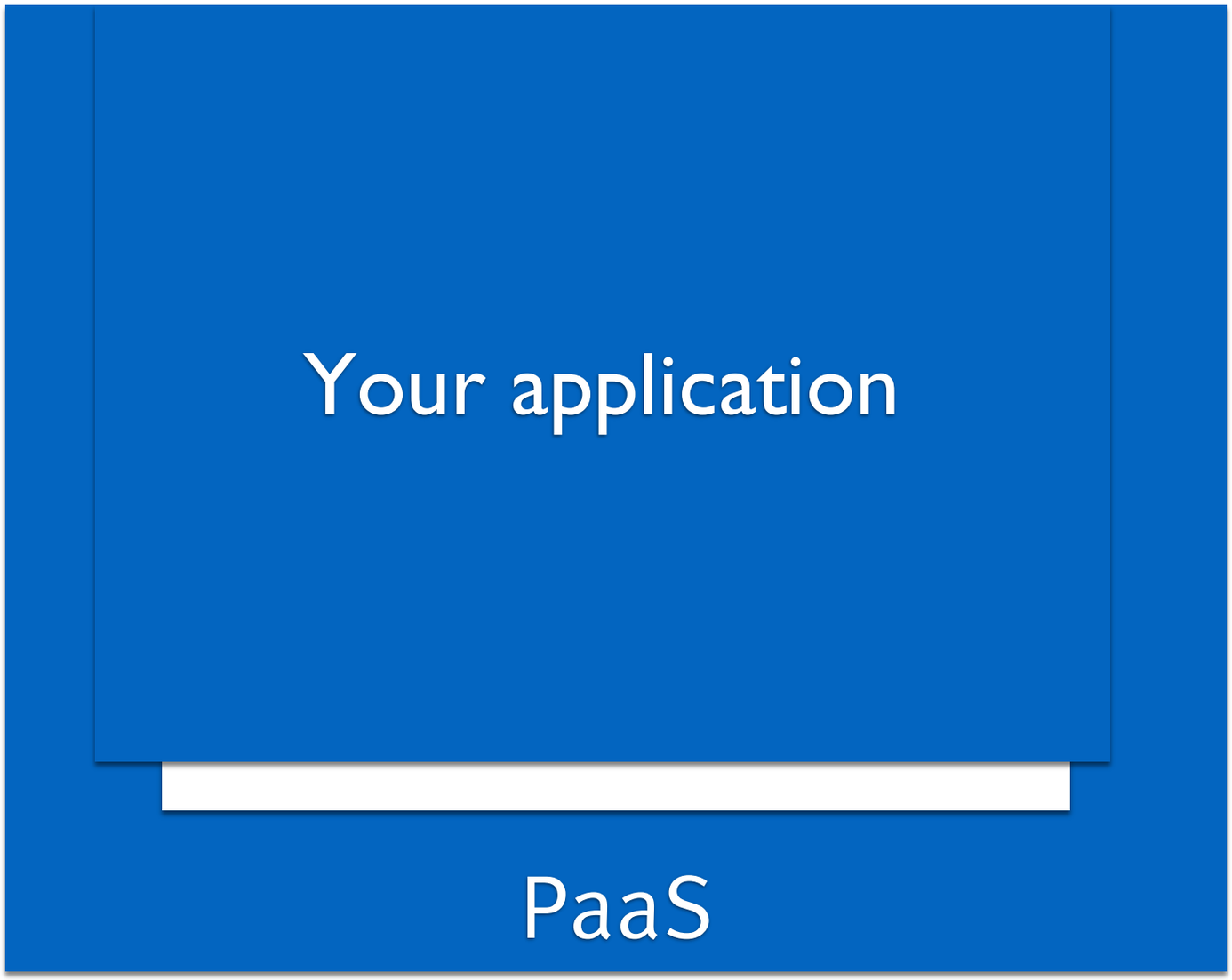
Continuous Delivery is the value propostion of PaaS

Issues With PaaS

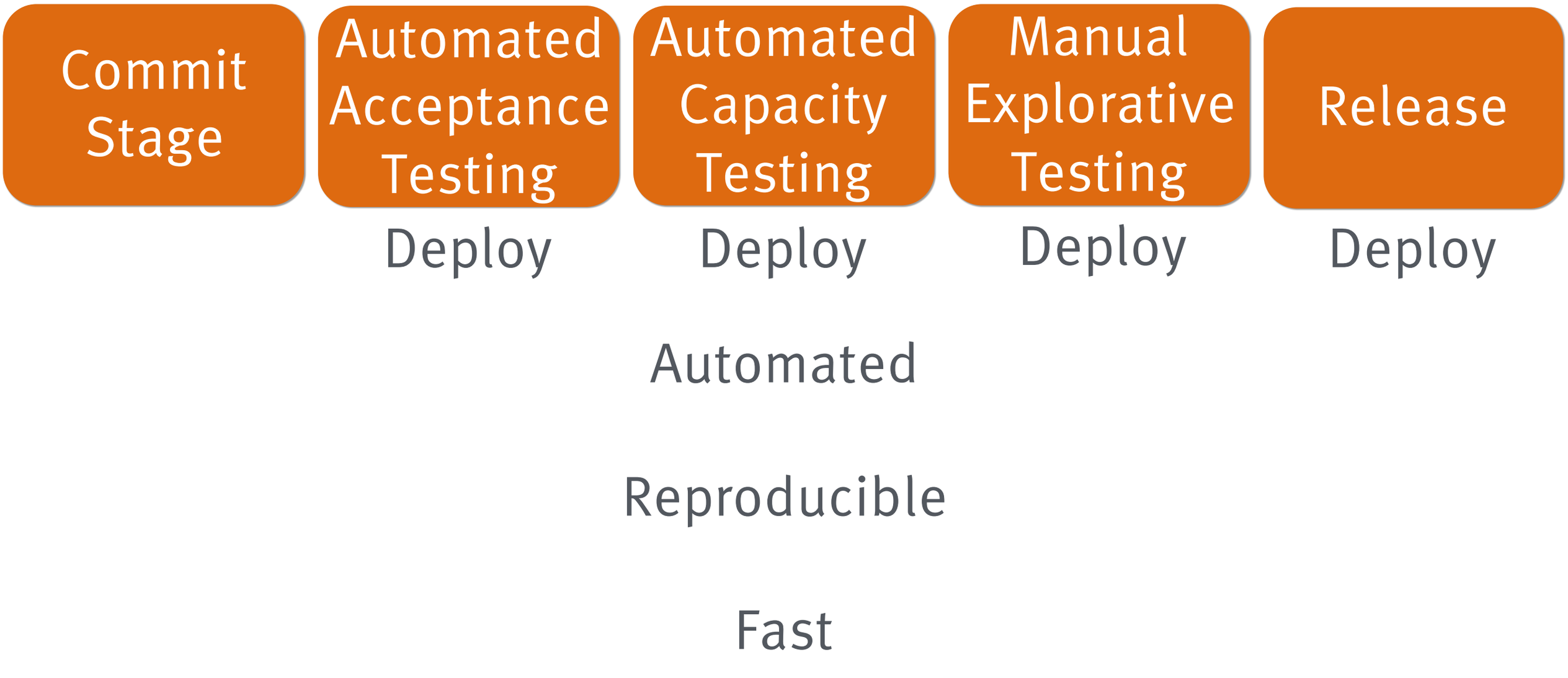
- Standardized infrastructure
- Not flexible
- Hard to migrate existing applications

- Installing PaaS on-premise hard
- Enterprise=On-Premise

- Huge success for Internet apps


Your application
PaaS

Continuous Delivery: Build Pipeline



Continuous Delivery: State of the Art

- Roll your own deployment automation
- Chef, Puppet, Ansible
- Use some kind of virtualization
- ...or Docker

Docker

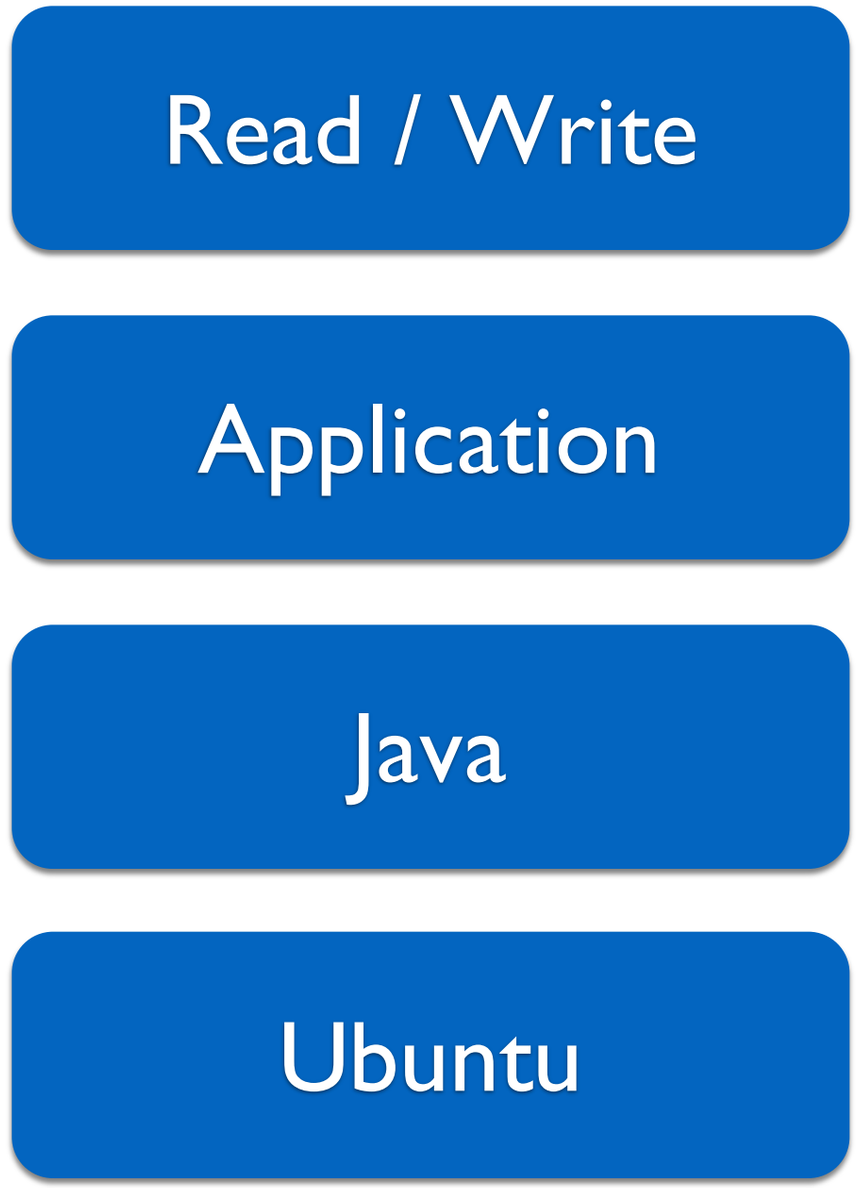
- No true virtualization
- Linux Containers (lxc)
- Shared kernel
- Separate file systems
- Separated network interfaces

Docker File Systems

- Storage backends
- Devicemapper (block devices)

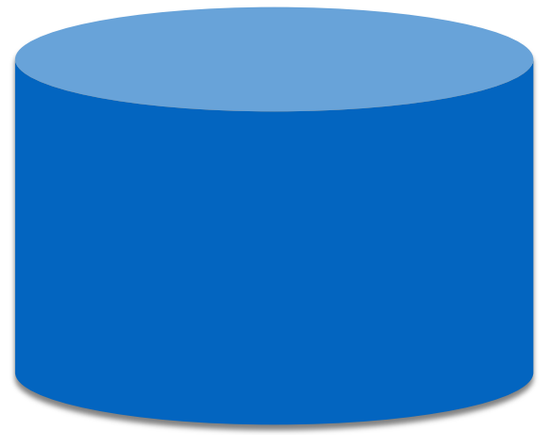
- Read only base images
- + Read/write image

- Can be stacked

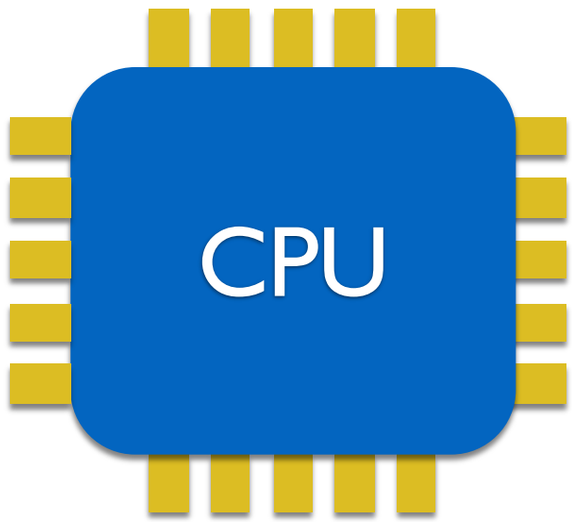


Docker = Lightweight

- One base image
- Other images are just a diff
- Little storage



- Kernel etc. shared
- Almost no overhead over a process



Docker =
Simple Deployment

29 lines (24 sloc) | 1.72 KB — Raw | Blame | History

```
FROM ubuntu:12.04
RUN apt-get update -qq
RUN apt-get dist-upgrade -y -qq --force-yes
RUN apt-get install -qq --force-yes -y wget python3 python3-minimal python-cairo-dev python-django python-ldap python-memcache pyt
RUN easy_install zope.interface
RUN easy_install twisted
RUN easy_install txamqp
RUN easy_install django-tagging==0.3.1
RUN easy_install daemonize

RUN wget -nv --no-check-certificate http://launchpad.net/graphite/0.9/0.9.9/+download/carbon-0.9.9.tar.gz ; tar -zxf carbon-0.9.9.
RUN cd carbon ; python setup.py -q install
RUN cd /opt/graphite/conf ; cp carbon.conf.example carbon.conf ; cp storage-schemas.conf.example storage-schemas.conf
RUN sed -i 's/^from twisted.scripts._twistd_unix //' /opt/graphite/lib/carbon/util.py

RUN wget -nv --no-check-certificate http://launchpad.net/graphite/0.9/0.9.9/+download/whisper-0.9.9.tar.gz ; tar -zxf whisper-0.9.
RUN cd whisper ; python setup.py -q install

RUN wget -nv --no-check-certificate http://launchpad.net/graphite/0.9/0.9.9/+download/graphite-web-0.9.9.tar.gz ; tar -zxf graphit
RUN cd graphite ; python check-dependencies.py ; python setup.py -q install
RUN cp graphite/conf/graphite.wsgi.example /opt/graphite/conf/graphite.wsgi
ADD local_settings.py /opt/graphite/webapp/graphite/
ADD settings.py /opt/graphite/webapp/graphite/settings.py
RUN cd /opt/graphite/webapp/graphite ; python manage.py syncdb --noinput ;  chown -R www-data:www-data /opt/graphite/storage/

ADD run.sh run.sh
CMD /bin/bash run.sh
EXPOSE 80 2003
```

RUN COPY

Docker = Simple Deployment

- Dockerfile
- Just a shell script

29 lines (24 sloc) | 1.72 KB — Raw | Blame | History

```
FROM ubuntu:12.04
RUN apt-get update -qq
RUN apt-get dist-upgrade -y -qq --force-yes
RUN apt-get install -qq --force-yes -y wget python3 python3-minimal python-cairo-dev python-django python-ldap python-memcache pyt
RUN easy_install zope.interface
RUN easy_install twisted
RUN easy_install txamqp
RUN easy_install django-tagging==0.3.1
RUN easy_install daemonize

RUN wget -nv --no-check-certificate http://launchpad.net/graphite/0.9/0.9.9/+download/carbon-0.9.9.tar.gz ; tar -zxf carbon-0.9.9.
RUN cd carbon ; python setup.py -q install
RUN cd /opt/graphite/conf ; cp carbon.conf.example carbon.conf ; cp storage-schemas.conf.example storage-schemas.conf
RUN sed -i 's/^from twisted.scripts._twistd_unix //' /opt/graphite/lib/carbon/util.py

RUN wget -nv --no-check-certificate http://launchpad.net/graphite/0.9/0.9.9/+download/whisper-0.9.9.tar.gz ; tar -zxf whisper-0.9.
RUN cd whisper ; python setup.py -q install

RUN wget -nv --no-check-certificate http://launchpad.net/graphite/0.9/0.9.9/+download/graphite-web-0.9.9.tar.gz ; tar -zxf graphit
RUN cd graphite ; python check-dependencies.py ; python setup.py -q install
RUN cp graphite/conf/graphite.wsgi.example /opt/graphite/conf/graphite.wsgi
ADD local_settings.py /opt/graphite/webapp/graphite/
ADD settings.py /opt/graphite/webapp/graphite/settings.py
RUN cd /opt/graphite/webapp/graphite ; python manage.py syncdb --noinput ;  chown -R www-data:www-data /opt/graphite/storage/

ADD run.sh run.sh
CMD /bin/bash run.sh
EXPOSE 80 2003
```

- Behind the scenes: Optimization
- Every Dockerfile line = filesystem snapshot
- Reuse snapshots for all other Dockerfiles

Docker‘s History

- Public PaaS must separate tenants
- Ideally multiple tenants per VM

- Docker started as foundation for dotCloud PaaS
- Docker and PaaS related

Why Docker Over PaaS?

- Still simple deployment
- Still simple installation
- Unlimited flexibility

Load Balancer

Log Parser

Your application

Monitoring

Database

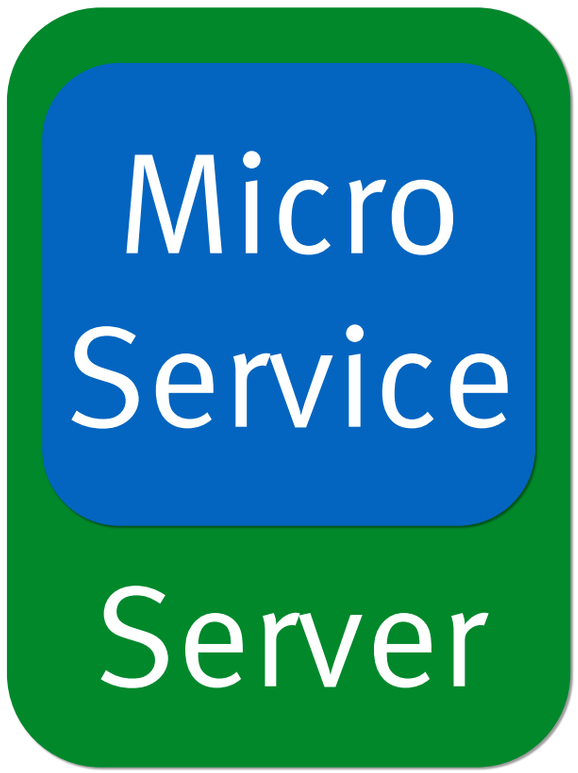
Cache

Microservices

Definition Microservice

- Independent deployment unit
- Separate data handling & storage
- Should include UI

- Process
- VM
- Docker container
- Any technology



Operations

- Operating 50-100 Microservices?

- Huge challenge

- Only option: Standardize

Standardize

Load Balancer ✓

Log Parser ✓

Your application

Monitoring ✓

Database ?

Cache ?

Standardized Environment?

Congrats on Building Your Own PaaS!

Demo: Elastic Beanstalk

Amazon Elastic Beanstalk

- Application:
  Contains all versions & environments
- Version:
  Deployabel artifact
- Environment:
  Runs a version of the application

Elastic Beanstalk Features

- Supports multiple environments
- Blue/green deployment support
- Scalable infrastructure
- Log files stored in S3
- Monitoring through Cloud Watch
- Also via CLI, API or Cloud Formation

But just a WAR

is not enough

Elastic Beanstalk

Additional Services

You own EC2 Virtual Machine

Turn Key Components

Databases (RDS)

ElastiCache

...

Paas Flexibility

Elastic Beanstalk Java, Go, Python...

Modify Beanstalk Image

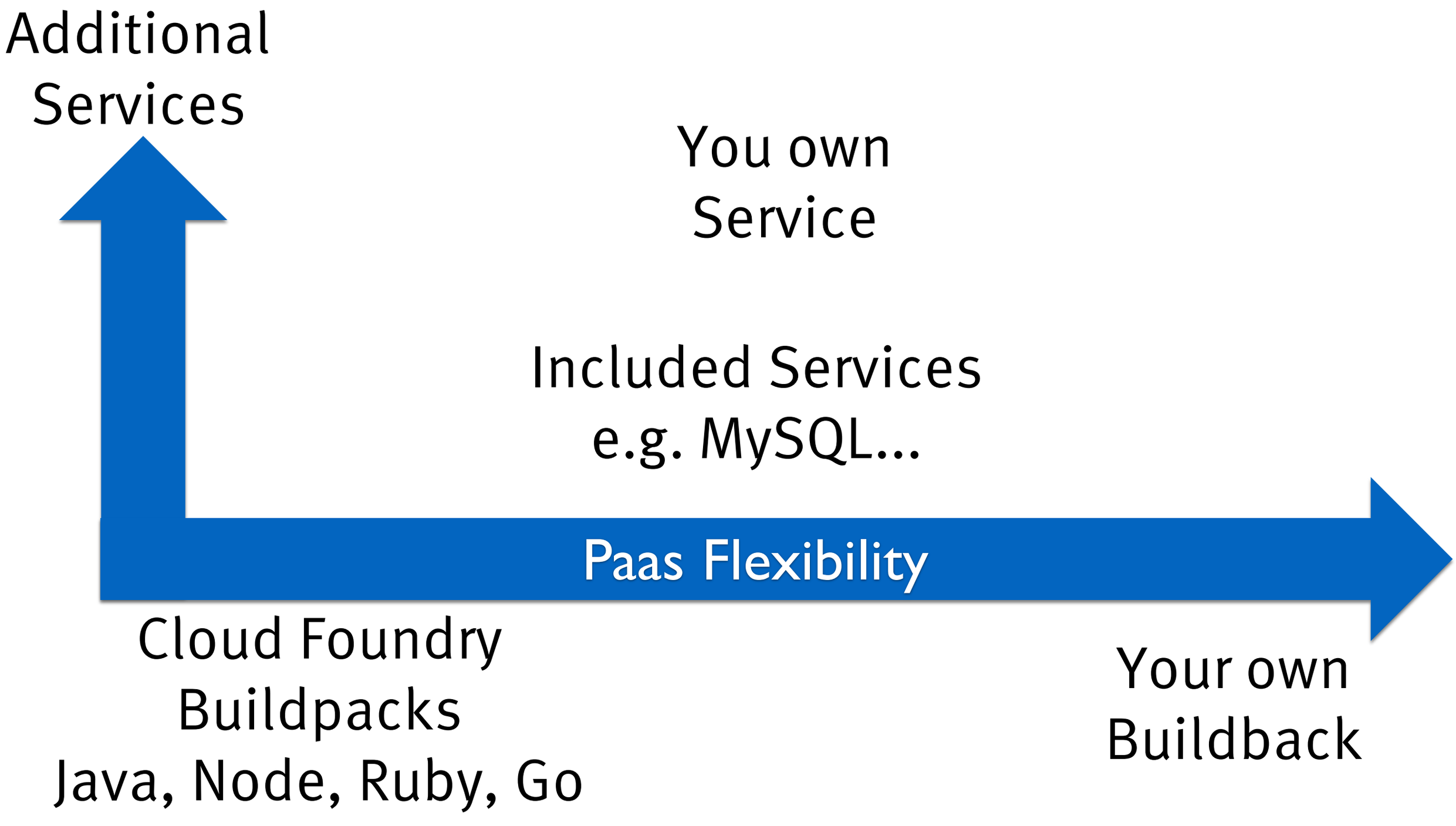
Beanstalk +Docker

But I want to run in my datacenter!

Cloud Foundry

- Open Source PaaS
- Foundation for IBM Bluemix, Pivotal CF ...

Cloud Foundry



Demo: Cloud Foundry

Conclusion

Conclusion

- Continuous Delivery = PaaS Value Proposition
- PaaS lacked flexibility
- Docker orignates from PaaS
- Microservices = Standardization
- Standardization = Your Own PaaS
- Buy instead of build?
- Modern PaaS provide the needed flexibility

Thank You!

@ewolff