Janos Pasztor

Practical Docker Operations

Stuff I do


ENTRECLOUD


The Ops Notepad
document your cloud


LET'S
CODE
.HU


Opsbears
We BEAR the load of IT operations


IntoTech


Refactor
ZONE


Refaktor
Magazin

Follow me!

https://pasztor.at

@janoszen

About this talk

1. Maintaining your Build Stack
2. Orchestrating your Cluster
3. Pitfalls and Recommendations

About this talk

1. Maintaining your Build Stack
2. Orchestrating your Cluster
3. Pitfalls and Recommendations

About this talk

1. Maintaining your Build Stack
2. Orchestrating your Cluster
3. Pitfalls and Recommendations

Maintaining your Build Stack

How Docker Images are Built

Maintaining your Build Stack

How Docker Images are Built

```
FROM ubuntu:16.04
```

Maintaining your Build Stack

How Docker Images are Built

```
FROM ubuntu:16.04
RUN apt-get install...
```

Maintaining your Build Stack

How Docker Images are Built

FROM ubuntu:16.04
RUN apt-get install...
COPY files/etc /etc

Maintaining your Build Stack

How Docker Images are Built

FROM ubuntu:16.04
RUN apt-get install...
COPY files/etc /etc
COPY init.sh /init.sh

Maintaining your Build Stack

How Docker Images are Built

FROM ubuntu:16.04
RUN apt-get install...
COPY files/etc /etc
COPY init.sh /init.sh
CMD /init.sh

Maintaining your Build Stack

How Docker Images are Built

FROM ubuntu:16.04	e87eea024487
RUN apt-get install...	c90c59c78830
COPY files/etc /etc	31c6577f6847
COPY init.sh /init.sh	54511612f1c4
CMD /init.sh	9e54da99b80c

Maintaining your Build Stack

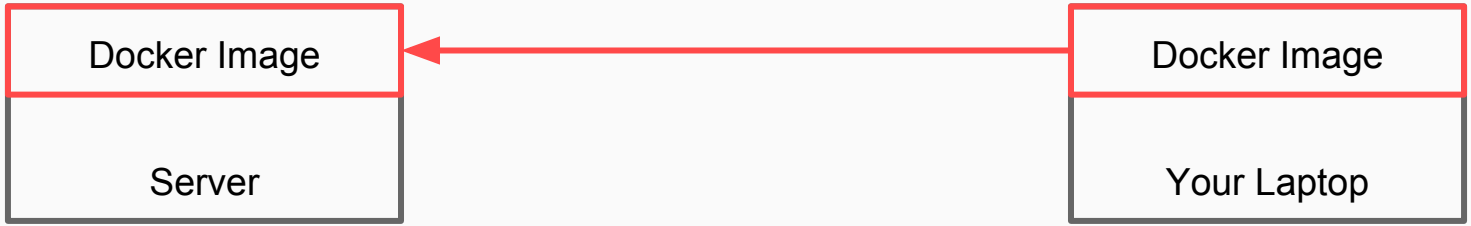
How Docker Images are Built

	FROM ubuntu:16.04	e87eea024487
	RUN apt-get install...	c90c59c78830
	COPY files/etc /etc	31c6577f6847
	COPY init.sh /init.sh	54511612f1c4
latest	CMD /init.sh	9e54da99b80c

Maintaining your Build Stack

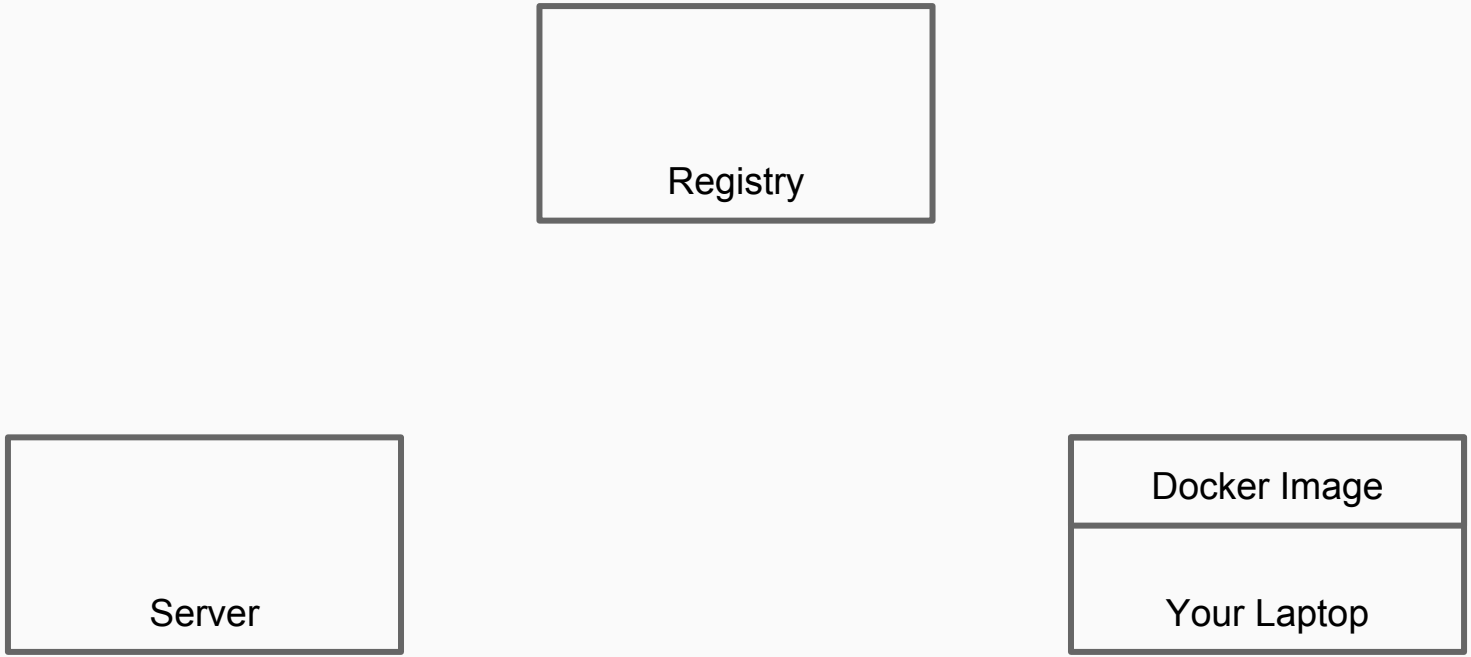
Maintaining your Build Stack

Maintaining your Build Stack

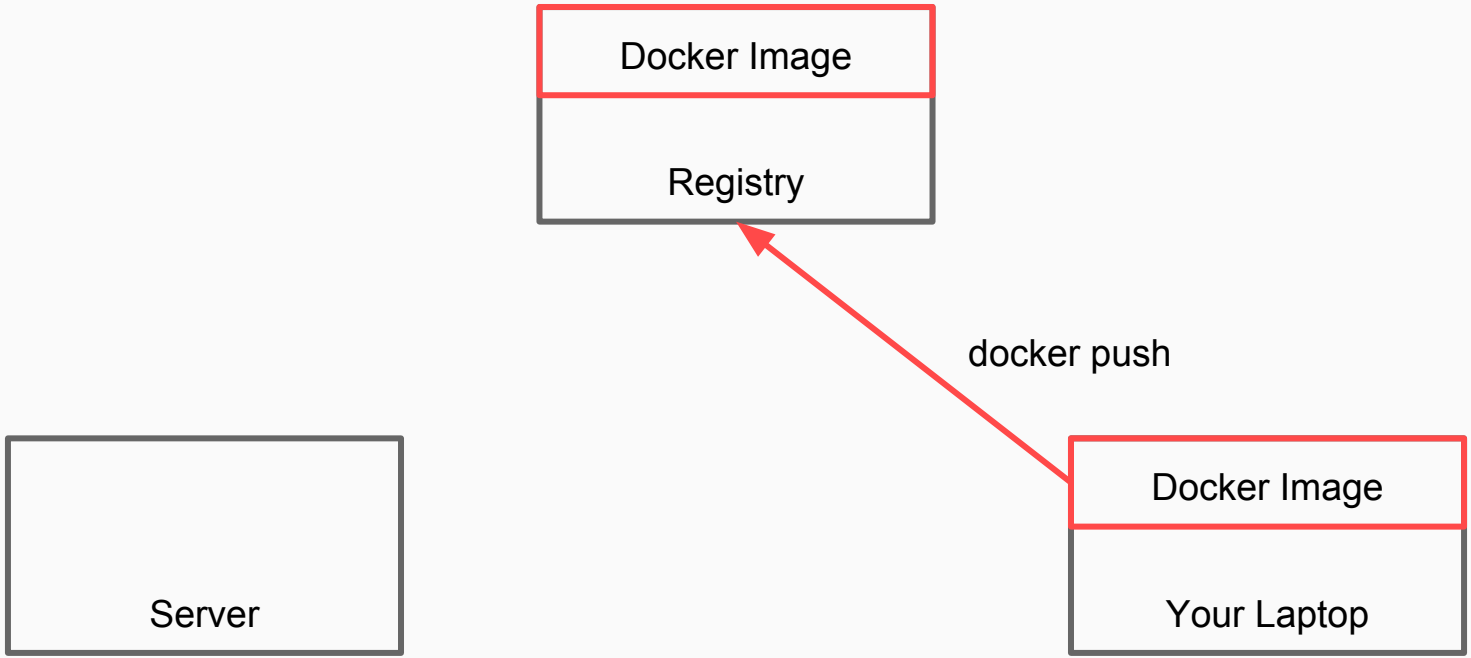


Maintaining your Build Stack

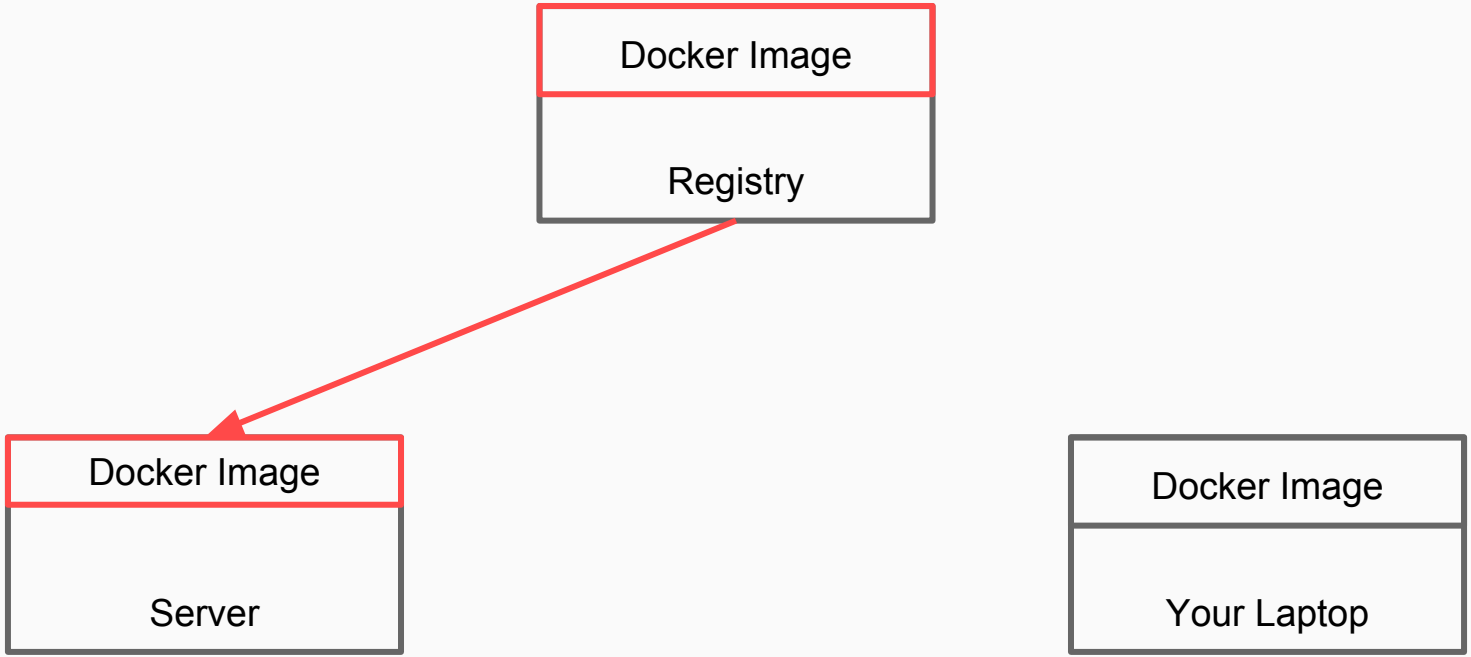
Maintaining your Build Stack



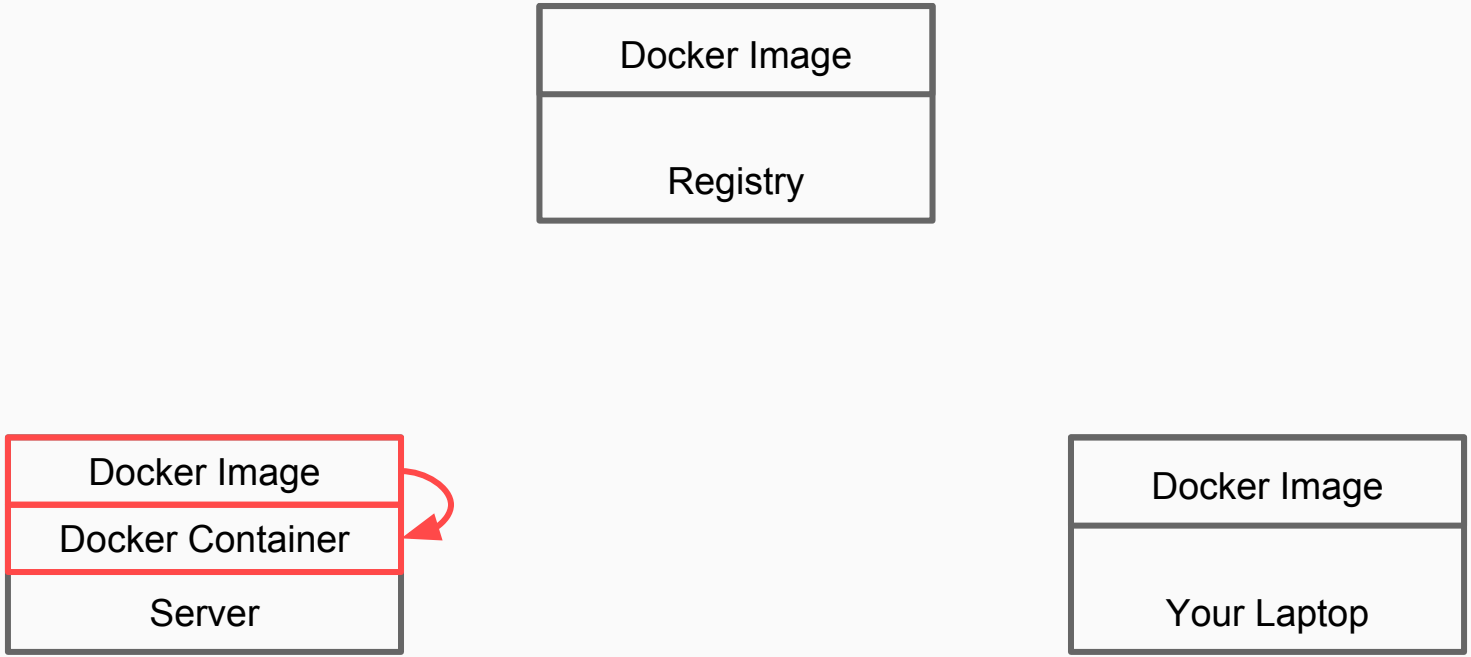
Maintaining your Build Stack



Maintaining your Build Stack



Maintaining your Build Stack



Follow me!

Problems:

Follow me!

Problems:

High Bandwidth Usage

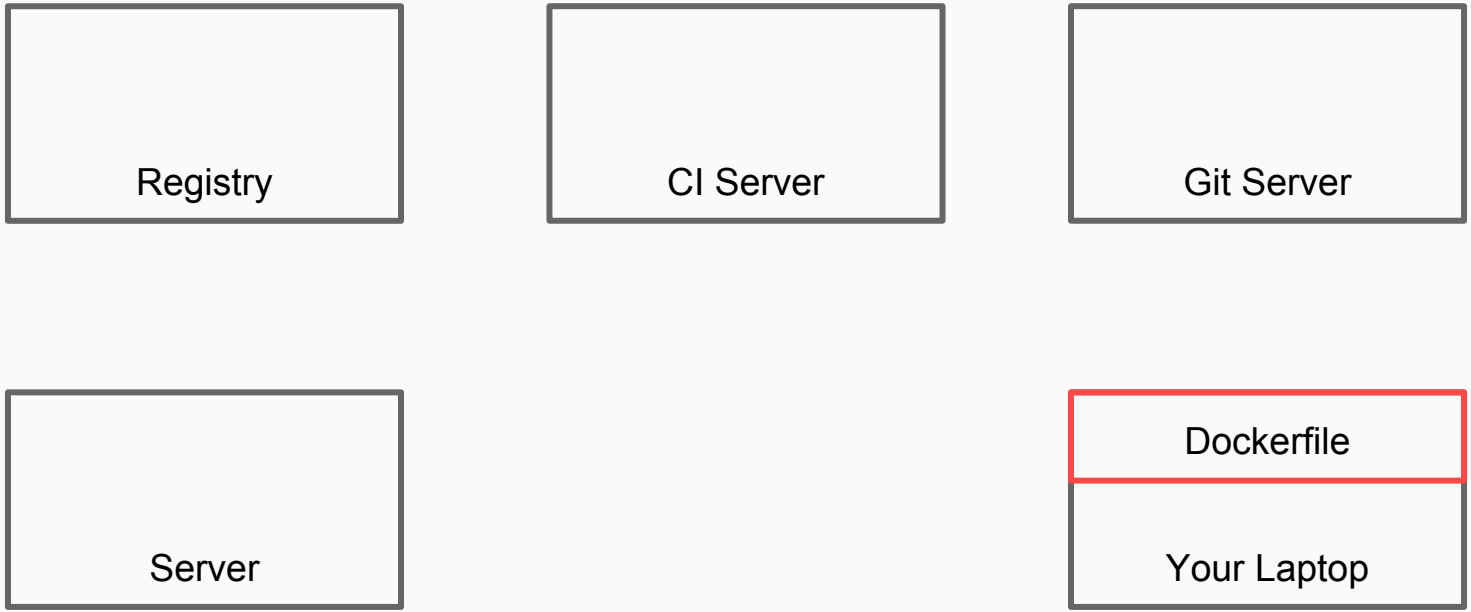
Follow me!

Problems:

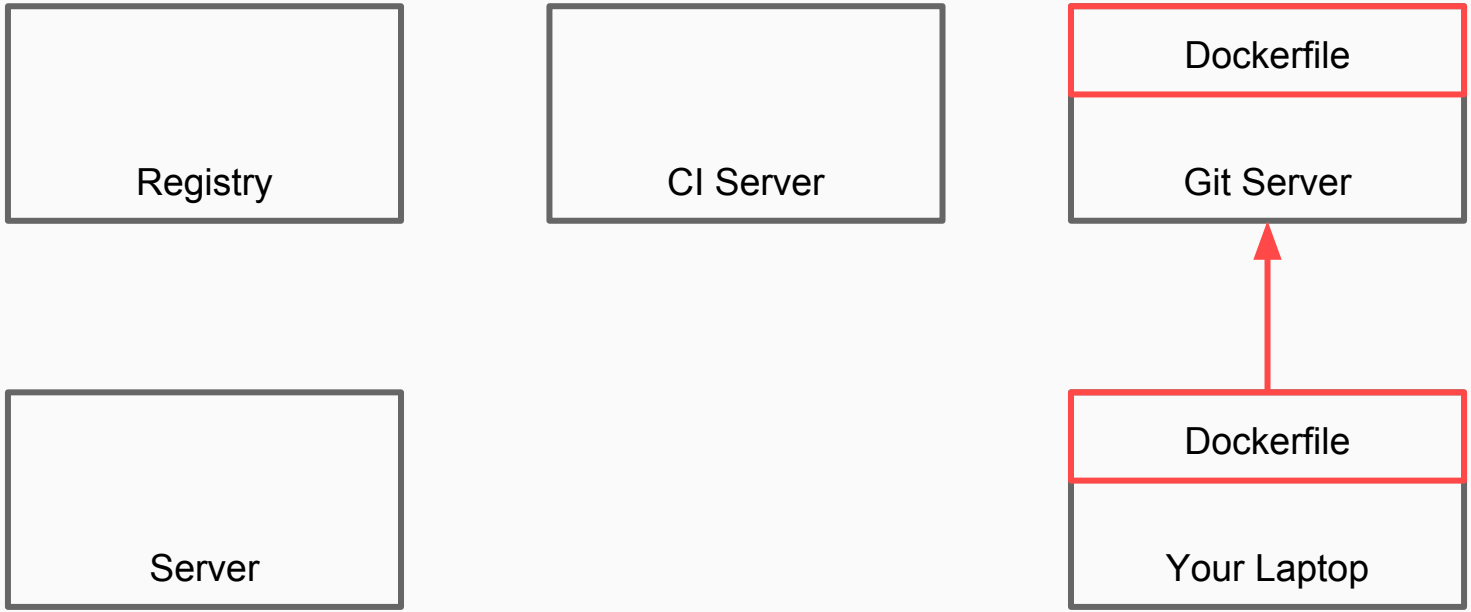
High Bandwidth Usage

Different Hashes on Different Machines

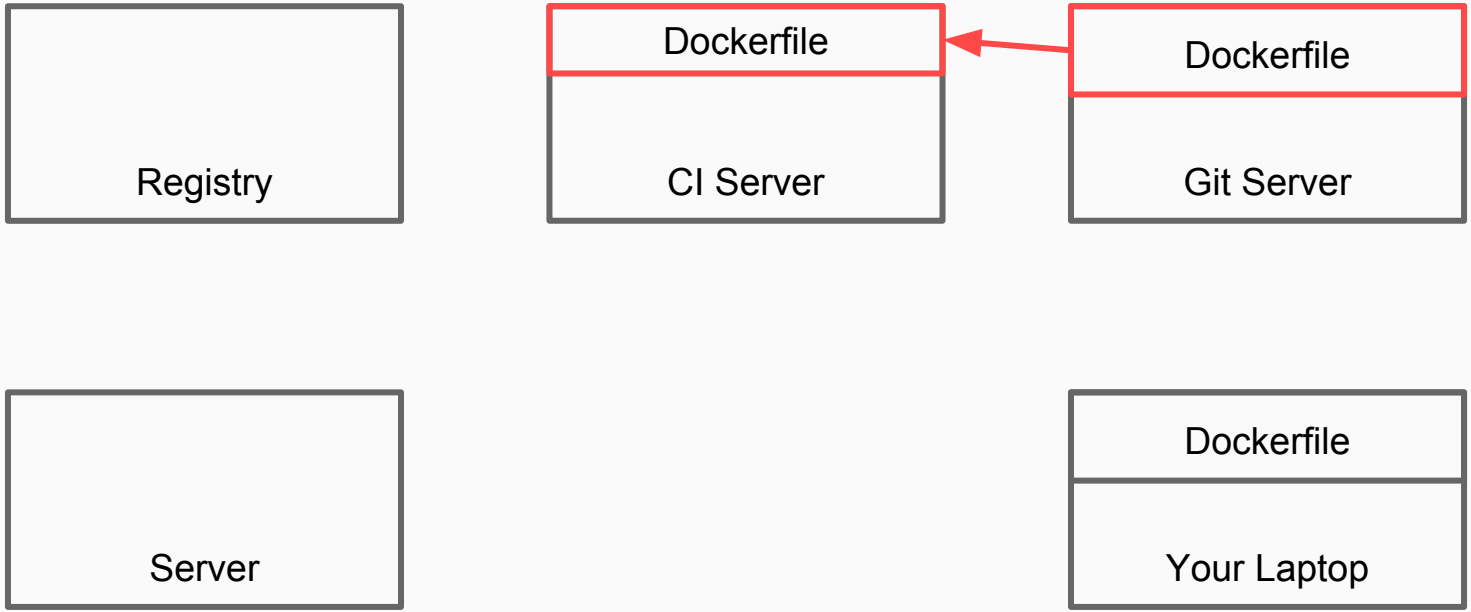
Maintaining your Build Stack



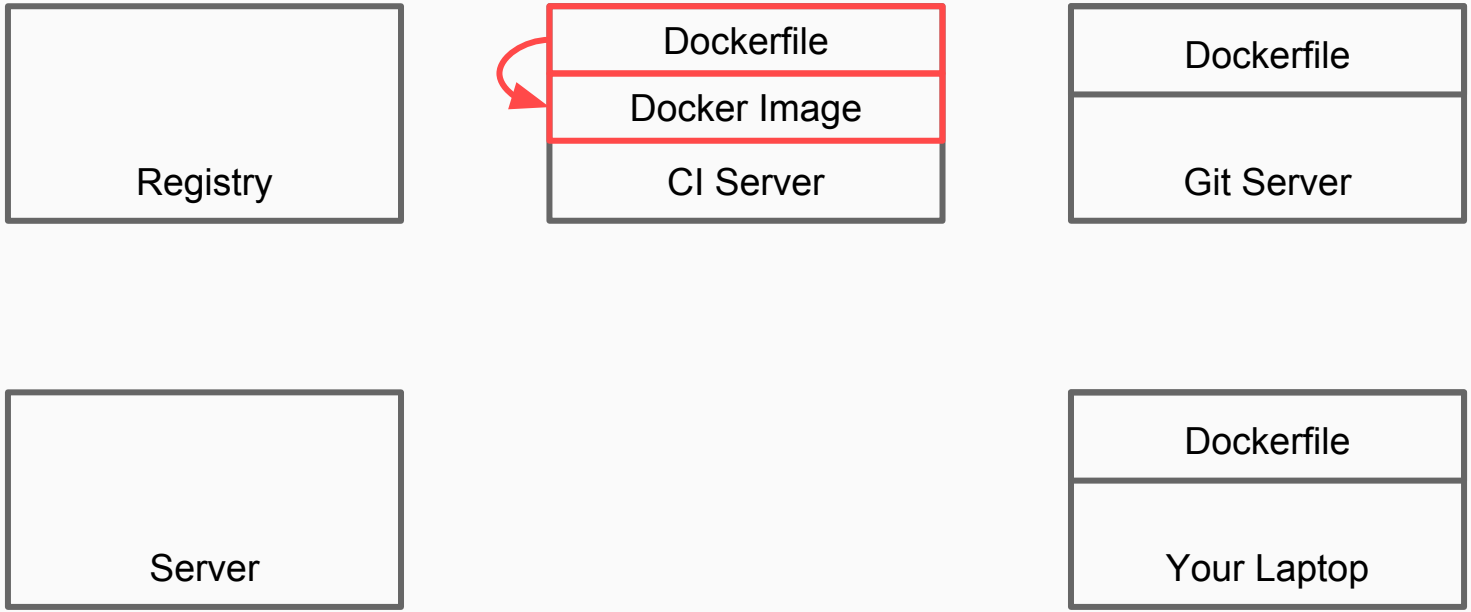
Maintaining your Build Stack



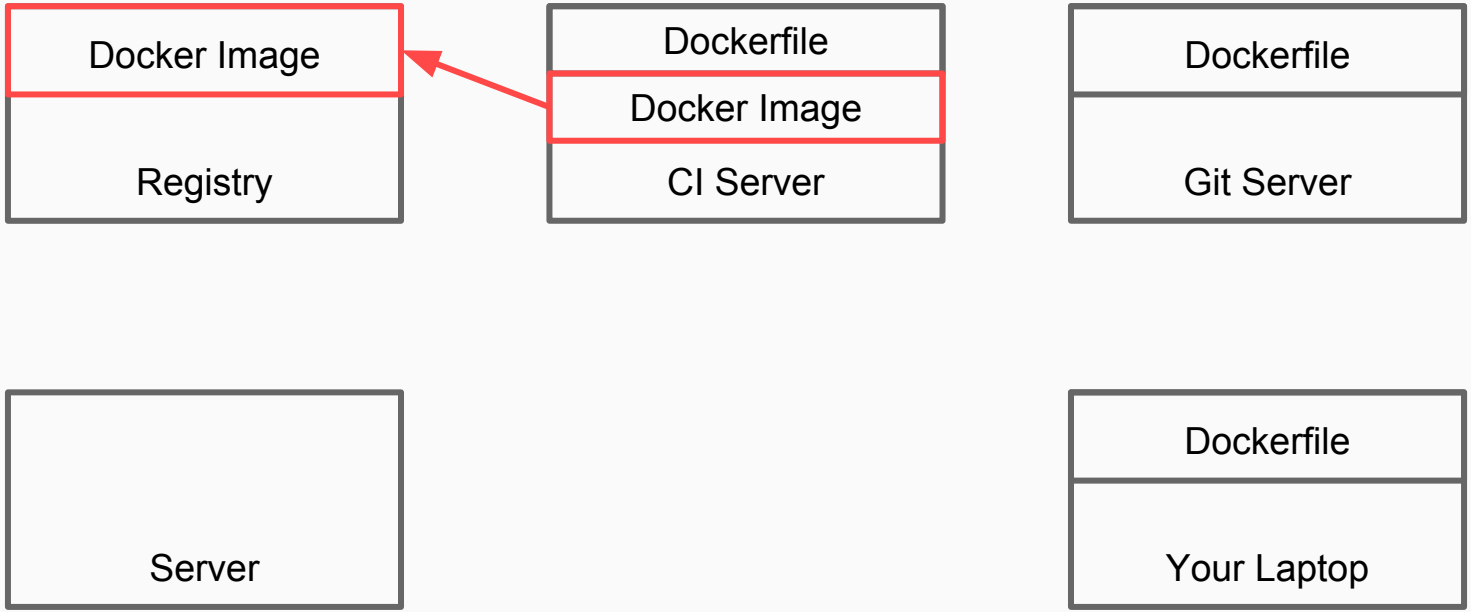
Maintaining your Build Stack



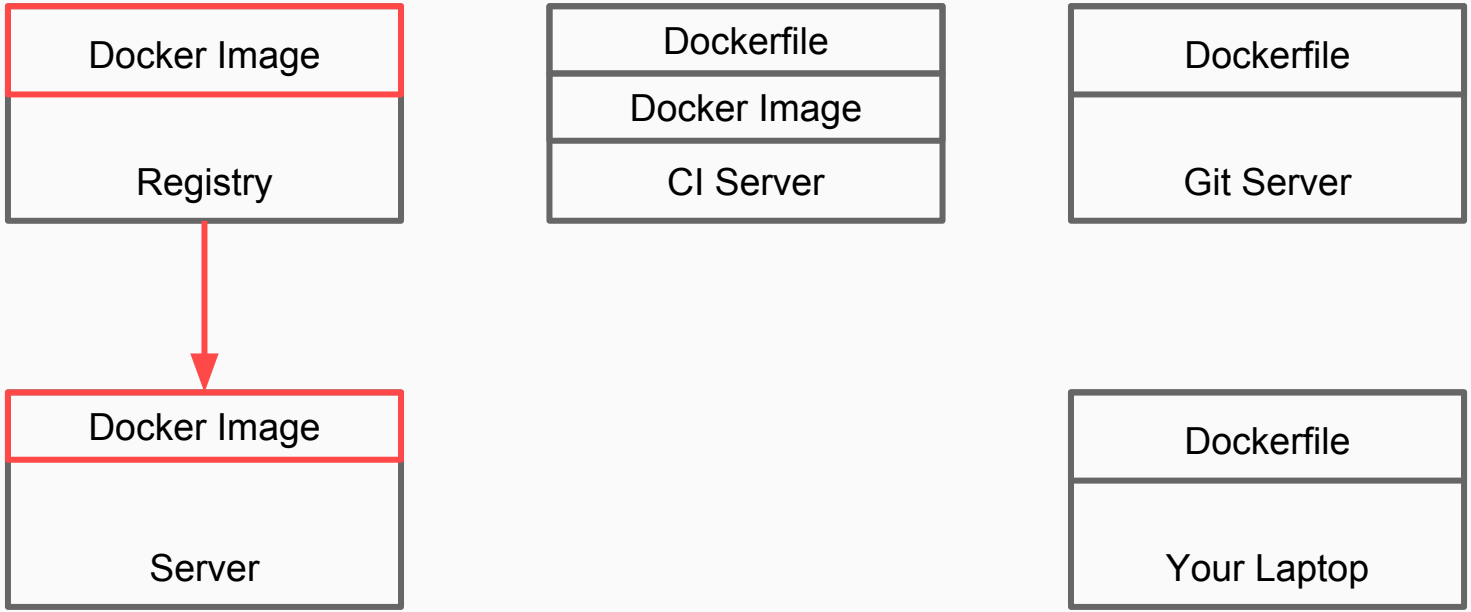
Maintaining your Build Stack



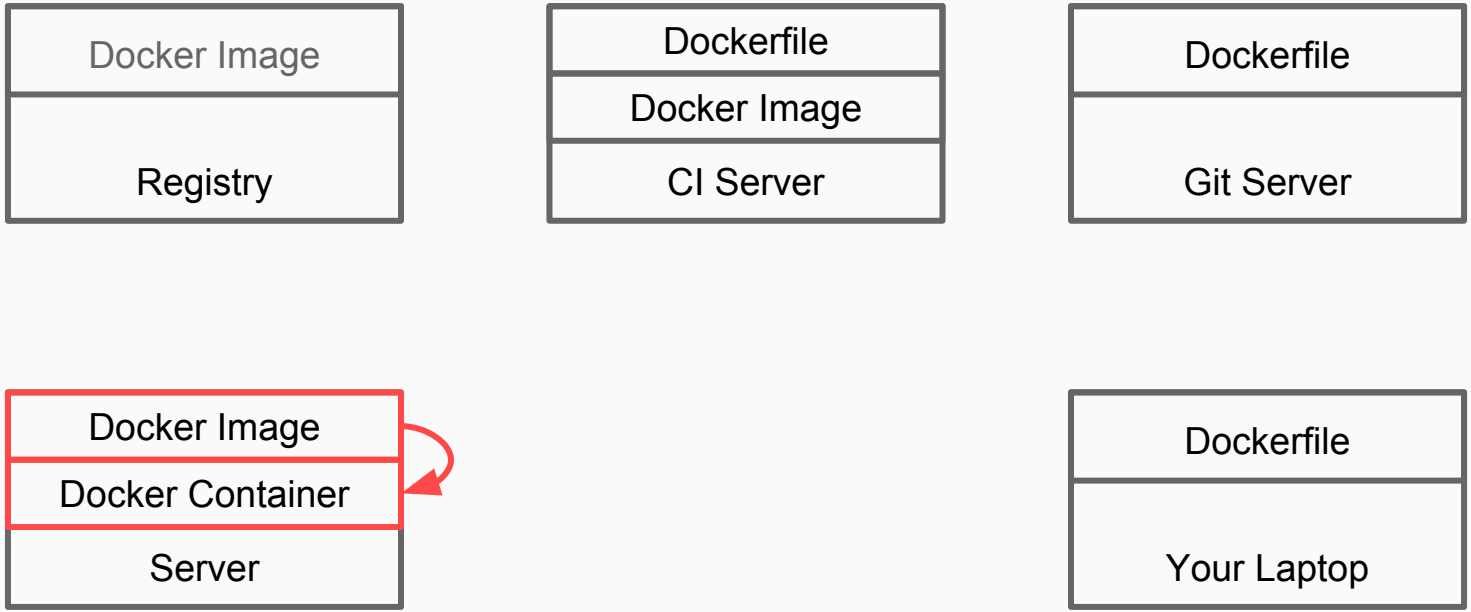
Maintaining your Build Stack



Maintaining your Build Stack



Maintaining your Build Stack



Maintaining your Build Stack

	docker	amazon web services	GitLab		circleci	Google Cloud Platform
CI server	YES		YES	YES	YES	YES
Docker Repository	YES	YES	YES			YES

Orchestrating your Cluster

How do you run a Docker image?

Orchestrating your Cluster

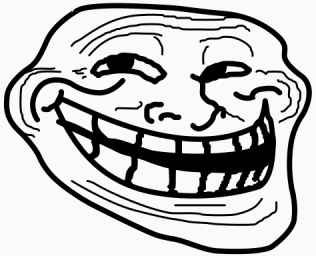
How do you run a Docker image?

```
ssh youruser@example.com "docker run yourcontainer"
```

Orchestrating your Cluster

How do you run a Docker image?

```
ssh youruser@example.com "docker run yourcontainer"
```



Orchestrating your Cluster

Orchestration tools

Orchestrating your Cluster

Orchestration tools

Where is my service?

Orchestrating your Cluster

Orchestration tools

Where is my service?

Rolling updates

Orchestrating your Cluster

Orchestration tools

Where is my service?

Rolling updates

Scaling

Orchestrating your Cluster

Orchestration tools

Where is my service?

Rolling updates

Scaling

Virtual networks

Orchestrating your Cluster

Orchestration tools

Where is my service?

Rolling updates

Scaling

Virtual networks

...

Orchestrating your Cluster

Orchestration tools

AWS EC2 Container Service

Docker Swarm

Kubernetes

...

Orchestrating your Cluster

EC2 Container Service

Orchestrating your Cluster

EC2 Container Service

+ Integrated with AWS

Orchestrating your Cluster

EC2 Container Service

+ Integrated with AWS
+ Autoscaling (limited)

Orchestrating your Cluster

EC2 Container Service

+ Integrated with AWS
+ Autoscaling (limited)
+ Integrated registry

Orchestrating your Cluster

EC2 Container Service

+ Integrated with AWS
+ Autoscaling (limited)
+ Integrated registry

- Using the ELB is a must

Orchestrating your Cluster

EC2 Container Service

+ Integrated with AWS
+ Autoscaling (limited)
+ Integrated registry

- Using the ELB is a must
- No built-in DNS server

Orchestrating your Cluster

EC2 Container Service

+ Integrated with AWS
+ Autoscaling (limited)
+ Integrated registry

- Using the ELB is a must
- No built-in DNS server
- Slow rollout

Orchestrating your Cluster

EC2 Container Service

+ Integrated with AWS
+ Autoscaling (limited)
+ Integrated registry

- Using the ELB is a must
- No built-in DNS server
- Slow rollout
- No overlay network

Orchestrating your Cluster

EC2 Container Service

+ Integrated with AWS
+ Autoscaling (limited)
+ Integrated registry

- Using the ELB is a must
- No built-in DNS server
- Slow rollout
- No overlay network
- Outdated

Orchestrating your Cluster

Docker Swarm

Orchestrating your Cluster

Docker Swarm

+ (Very) Simple

Orchestrating your Cluster

Docker Swarm

+ (Very) Simple
+ Overlay Networks

Orchestrating your Cluster

Docker Swarm

+ (Very) Simple
+ Overlay Networks
+ Built-in DNS server and LB

Orchestrating your Cluster

Docker Swarm

+ (Very) Simple
+ Overlay Networks
+ Built-in DNS server and LB
+ Fast rollout

Orchestrating your Cluster

Docker Swarm

+ (Very) Simple
+ Overlay Networks
+ Built-in DNS server and LB
+ Fast rollout

- Very young (YMMV)

Orchestrating your Cluster

Docker Swarm

+ (Very) Simple
+ Overlay Networks
+ Built-in DNS server and LB
+ Fast rollout

- Very young (YMMV)
- No Autoscaling

Orchestrating your Cluster

Docker Swarm

+ (Very) Simple
+ Overlay Networks
+ Built-in DNS server and LB
+ Fast rollout

- Very young (YMMV)
- No Autoscaling
- Monolithic (no plugins)

Orchestrating your Cluster

Kubernetes

+ Overlay networks

Orchestrating your Cluster

Kubernetes

+ Overlay networks
+ Autoscaling

Orchestrating your Cluster

Kubernetes

+ Overlay networks
+ Autoscaling
+ Rack awareness

Orchestrating your Cluster

Kubernetes

+ Overlay networks
+ Autoscaling
+ Rack awareness
+ Plugins, everywhere

Orchestrating your Cluster

Kubernetes

+ Overlay networks
+ Autoscaling
+ Rack awareness
+ Plugins, everywhere
+ Makes you coffee

Orchestrating your Cluster

Kubernetes

+ Overlay networks
+ Autoscaling
+ Rack awareness
+ Plugins, everywhere
+ Makes you coffee

- Complex to set up

Orchestrating your Cluster

Kubernetes

+ Overlay networks
+ Autoscaling
+ Rack awareness
+ Plugins, everywhere
+ Makes you coffee

- Complex to set up
- Setup scripts and docs are beta / unstable / outdated

Pitfalls and Recommendations

Oops...

Pitfalls and Recommendations

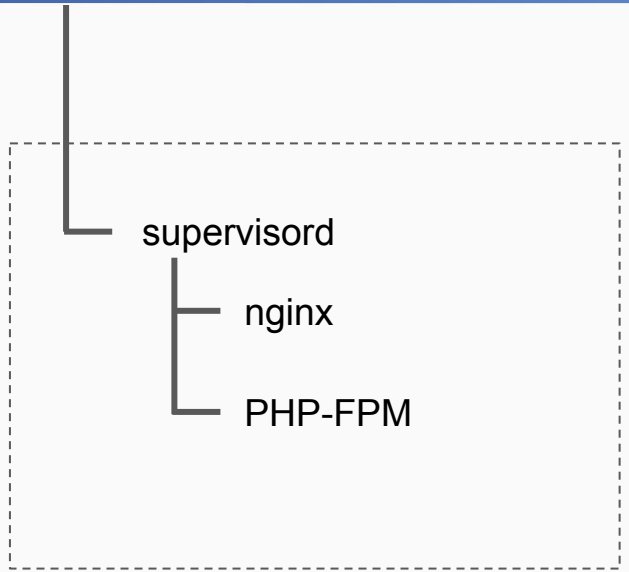
Multiple services in one container?

Pitfalls and Recommendations

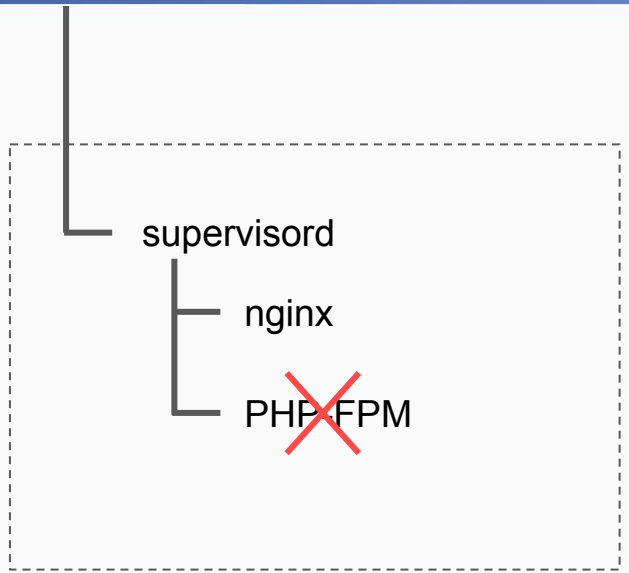
Multiple services in one container?

DON’T!

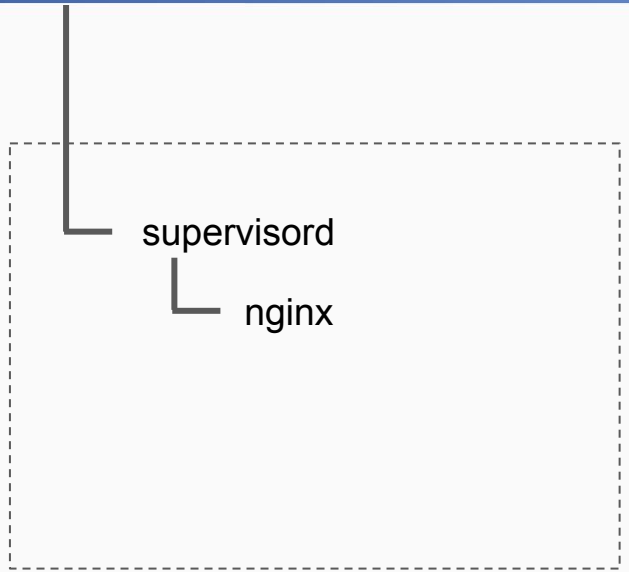
Pitfalls and Recommendations



Pitfalls and Recommendations



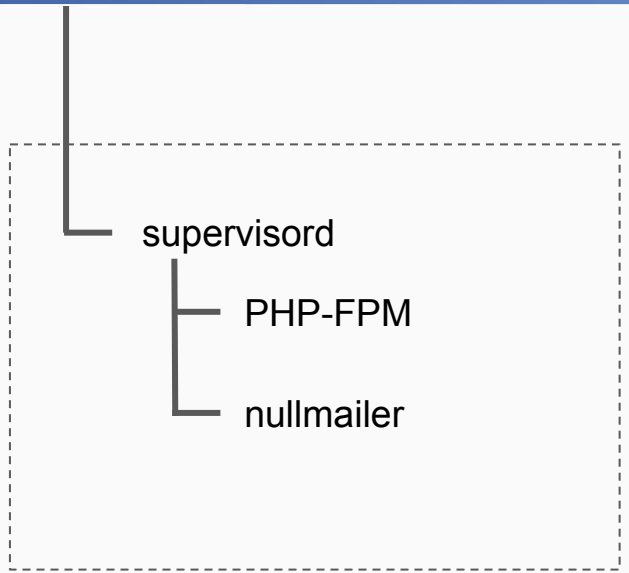
Pitfalls and Recommendations



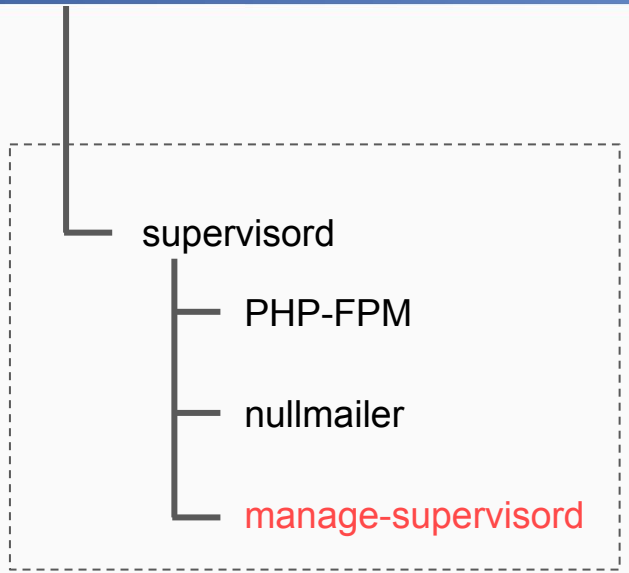
Pitfalls and Recommendations

Sidecar services

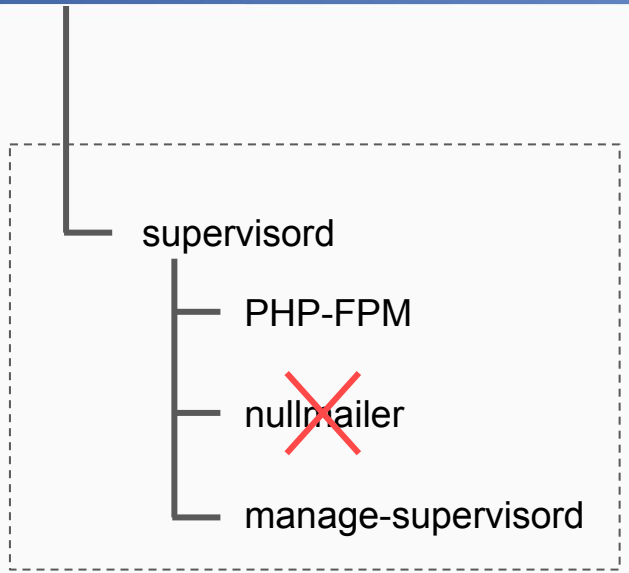
Pitfalls and Recommendations



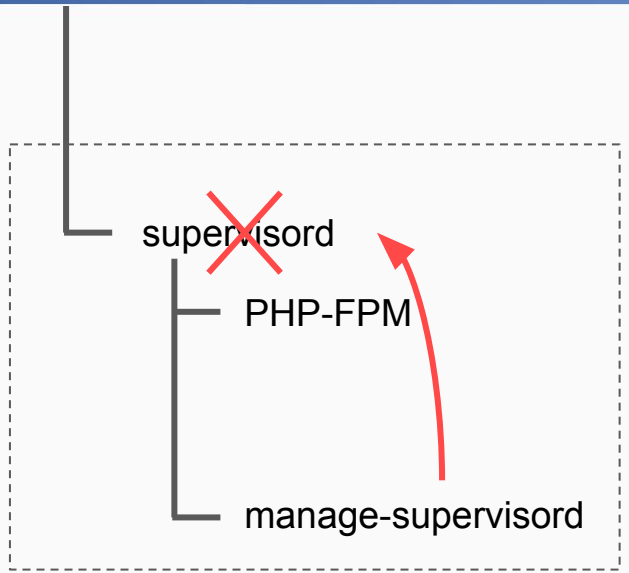
Pitfalls and Recommendations



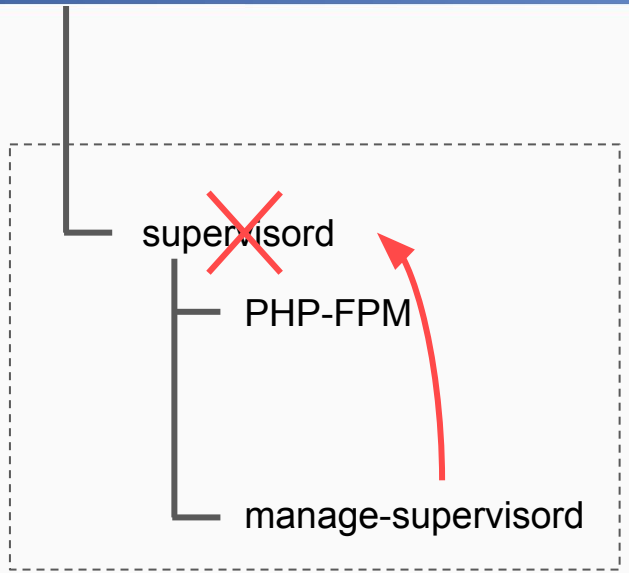
Pitfalls and Recommendations



Pitfalls and Recommendations



Pitfalls and Recommendations



https://github.com/opsbears/docker-supervisord

Pitfalls and Recommendations

Shell script in CMD?

Pitfalls and Recommendations

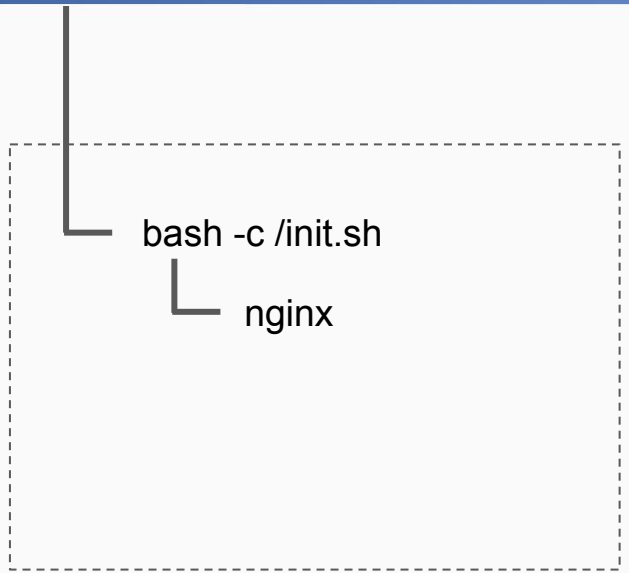
Shell script in CMD?

BE CAREFUL!

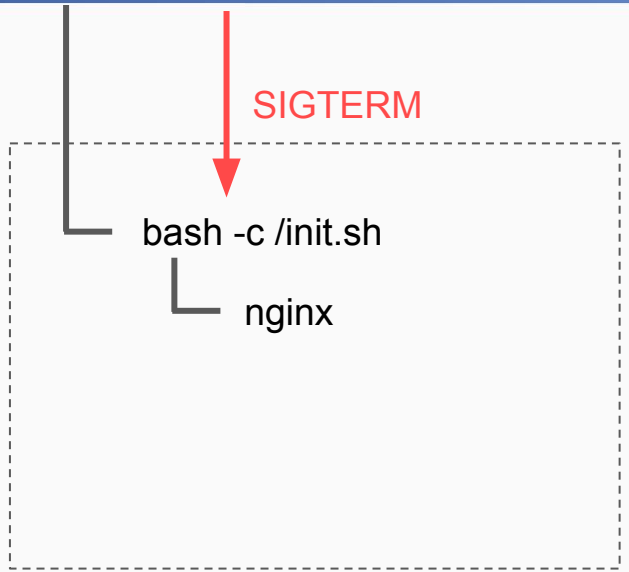
Pitfalls and Recommendations

```
#!/bin/bash

# Other stuff here

/usr/sbin/nginx -g "daemon off;"
exit $?
```

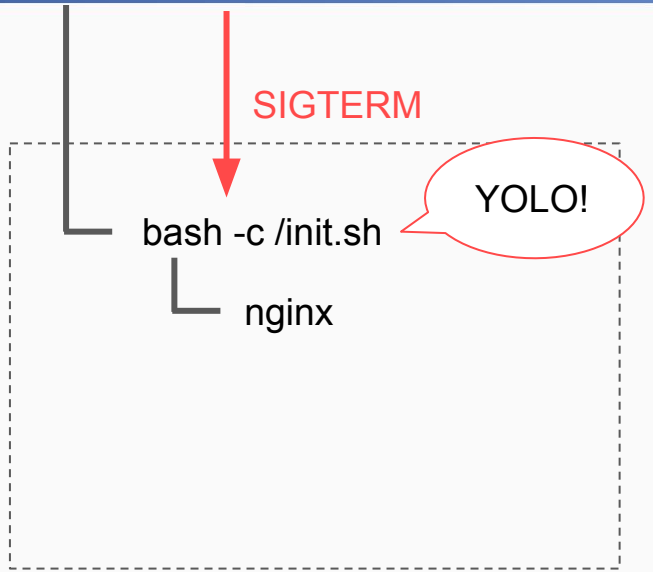
Pitfalls and Recommendations



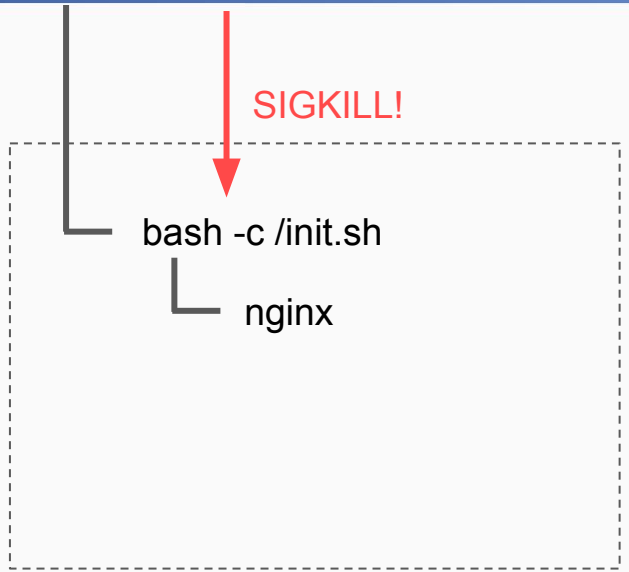
Pitfalls and Recommendations



Pitfalls and Recommendations



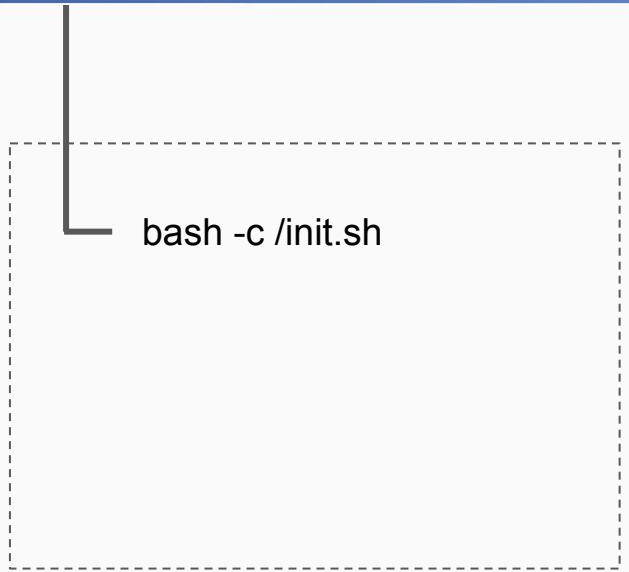
Pitfalls and Recommendations



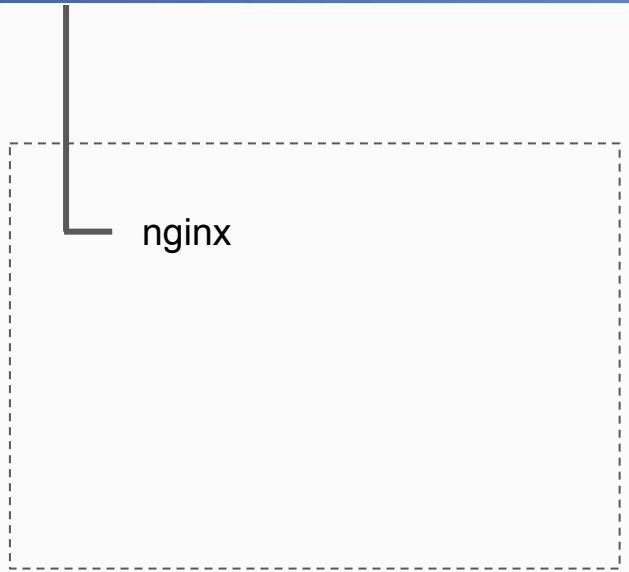
Pitfalls and Recommendations

```
#!/bin/bash

# Other stuff here

exec /usr/sbin/nginx -g "daemon off;"
```

Pitfalls and Recommendations



Pitfalls and Recommendations



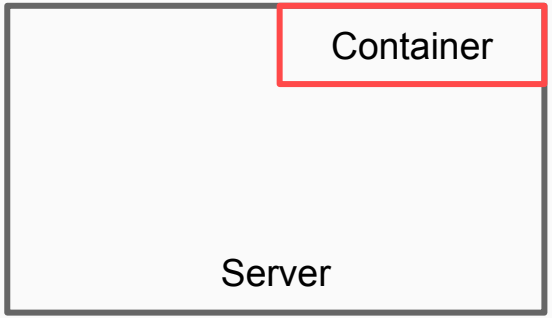
Pitfalls and Recommendations

Don’t update your containers!

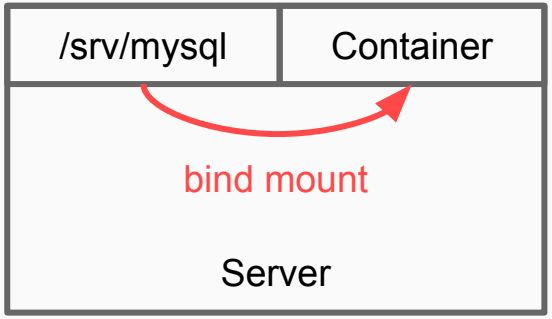
Pitfalls and Recommendations

Shared data?

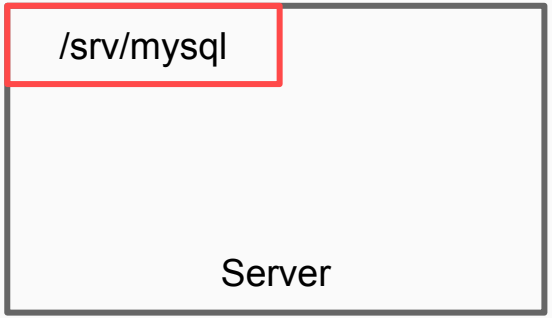
Pitfalls and Recommendations

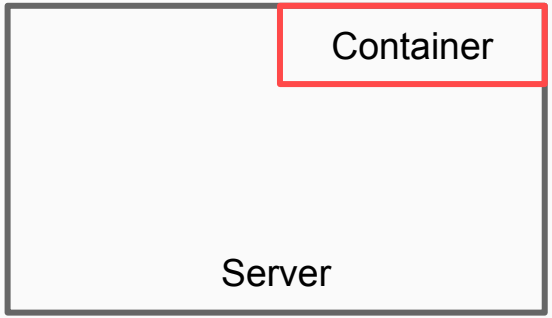


Pitfalls and Recommendations



Pitfalls and Recommendations





Pitfalls and Recommendations

Healthchecks?

Pitfalls and Recommendations

Dockerfile:

```
HEALTHCHECK \
    --interval=10s \
    --timeout=3s \
    CMD /usr/local/bin/healthcheck
```

healthcheck:

```
#!/bin/bash

test $(SCRIPT_NAME=/status
SCRIPT_FILENAME=/status
REQUEST_METHOD=GET cgi-fcgi -bind
-connect 127.0.0.1:9000 | grep
pool | cut -d: -f2 | sed 's/ //g')
== www || exit 1
```

Pitfalls and Recommendations

healthcheck:

```
#!/bin/bash

test $(SCRIPT_NAME=/status SCRIPT_FILENAME=/status REQUEST_METHOD=GET
cgi-fcgi -bind -connect 127.0.0.1:9000 | grep pool | cut -d: -f2 | sed 's/
//g') == www || exit 1
```

Pitfalls and Recommendations

Docker Swarm:

```
HEALTHCHECK \
    --interval=10s \
    --timeout=3s \
    CMD /usr/local/bin/healthcheck
```

Pitfalls and Recommendations

Kubernetes:

```
spec:
  containers:
  - name: yourpod
    livenessProbe:
      exec:
        command:
        - /usr/local/bin/healthcheck
      initialDelaySeconds: 5
      periodSeconds: 5
    readinessProbe:
      ....
```

Pitfalls and Recommendations

TEST YO’ CONTAINERS!

Pitfalls and Recommendations

docker-compose.test.yml

```
version: '3.2'
services:
  mysql:
    container_name: mysql
    build: .
    ...
 sut:
    build: ./test
    ...
```

Pitfalls and Recommendations

DO NOT HARD-CODE CREDENTIALS!

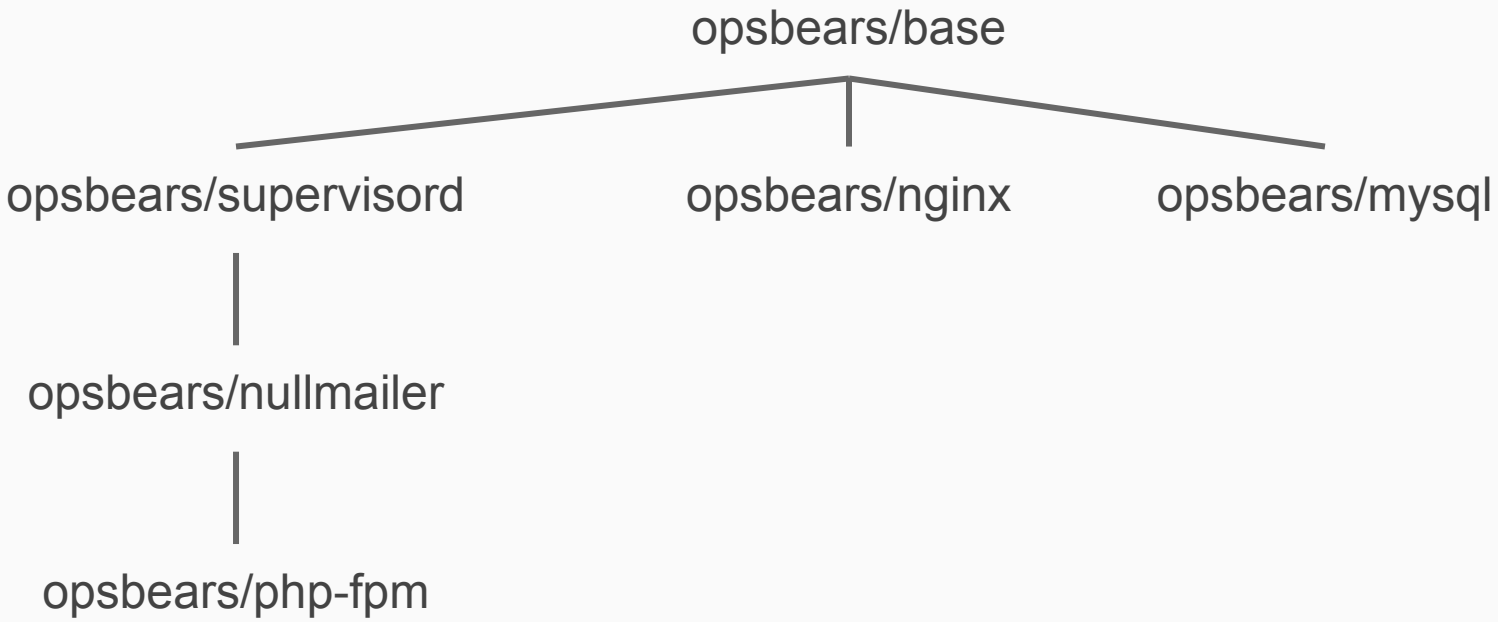
Pitfalls and Recommendations

VERSION YOUR IMAGES!

Pitfalls and Recommendations

CAREFUL WITH 3RD PARTY IMAGES!

Pitfalls and Recommendations



Pitfalls and Recommendations

REMOVE DEV STUFF!

That’s all!

Questions?

Many thanks to

Bence Sántha, Gábor Vereb, Dávid Papp

for their inspiration and feedback.