



OWASP  
AppSec EU  
**Belfast**  
8-12 May, 2017

# Creating an Appsec Pipeline with Containers in a week

## How we Failed and Succeeded

Jeroen Willemsen

# About me

Jeroen Willemsen  
@commjoenie  
jwillemsen@xebia.com



``Security architect``  
``Full-stack developer``  
``Mobile security``







# Agenda

- The challenge
- The solution
- Bumps on the road
- Recap



# The Challenge

What could possibly go wrong?



# The Challenge

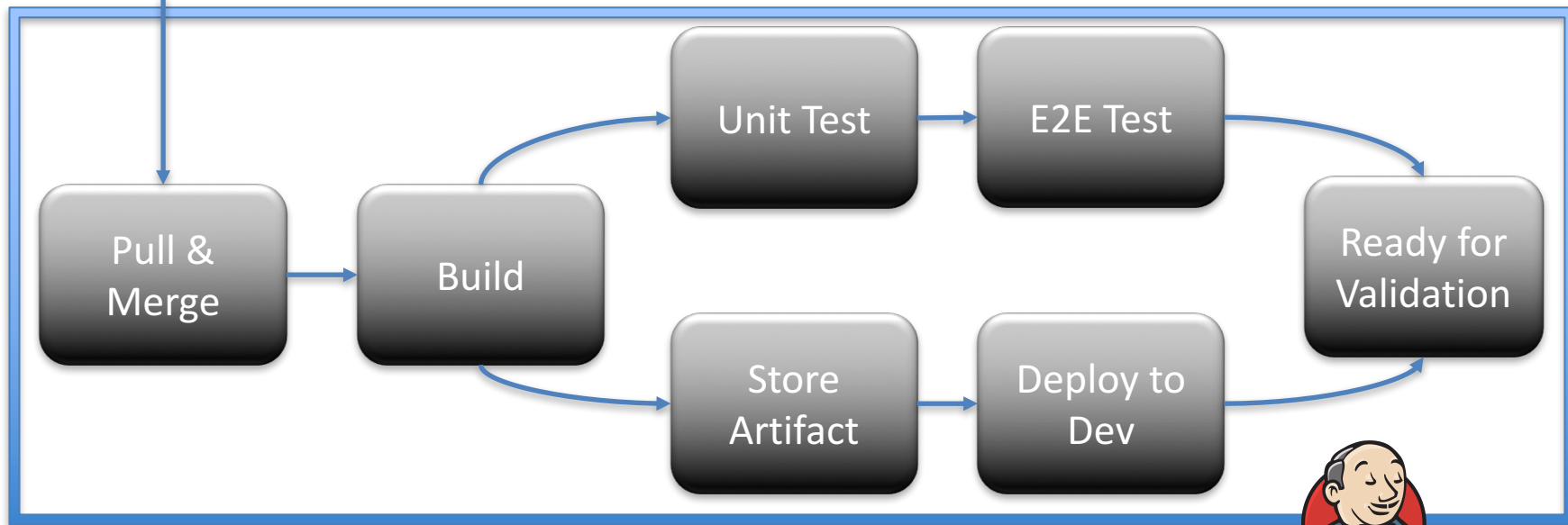
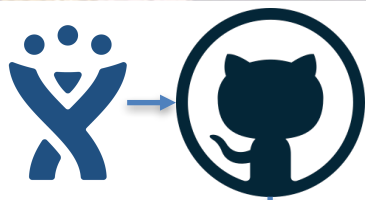


# The Challenge: The landscape








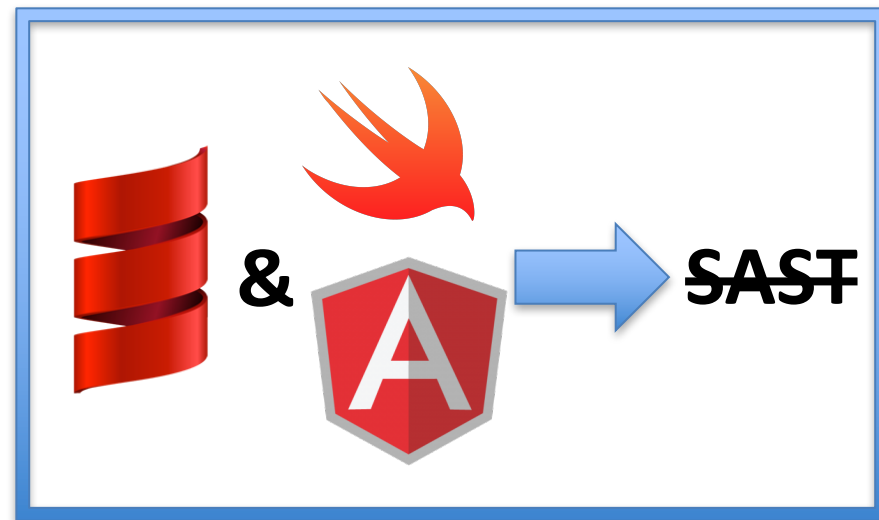
# The Challenge: Existing workflow





# The Challenge: New entries

- OWASP Dependency-Check
- License checkers
-  clair
-  & 
- Etc...







# The Solution

We got there kind off...



# The Solution: Extend build step

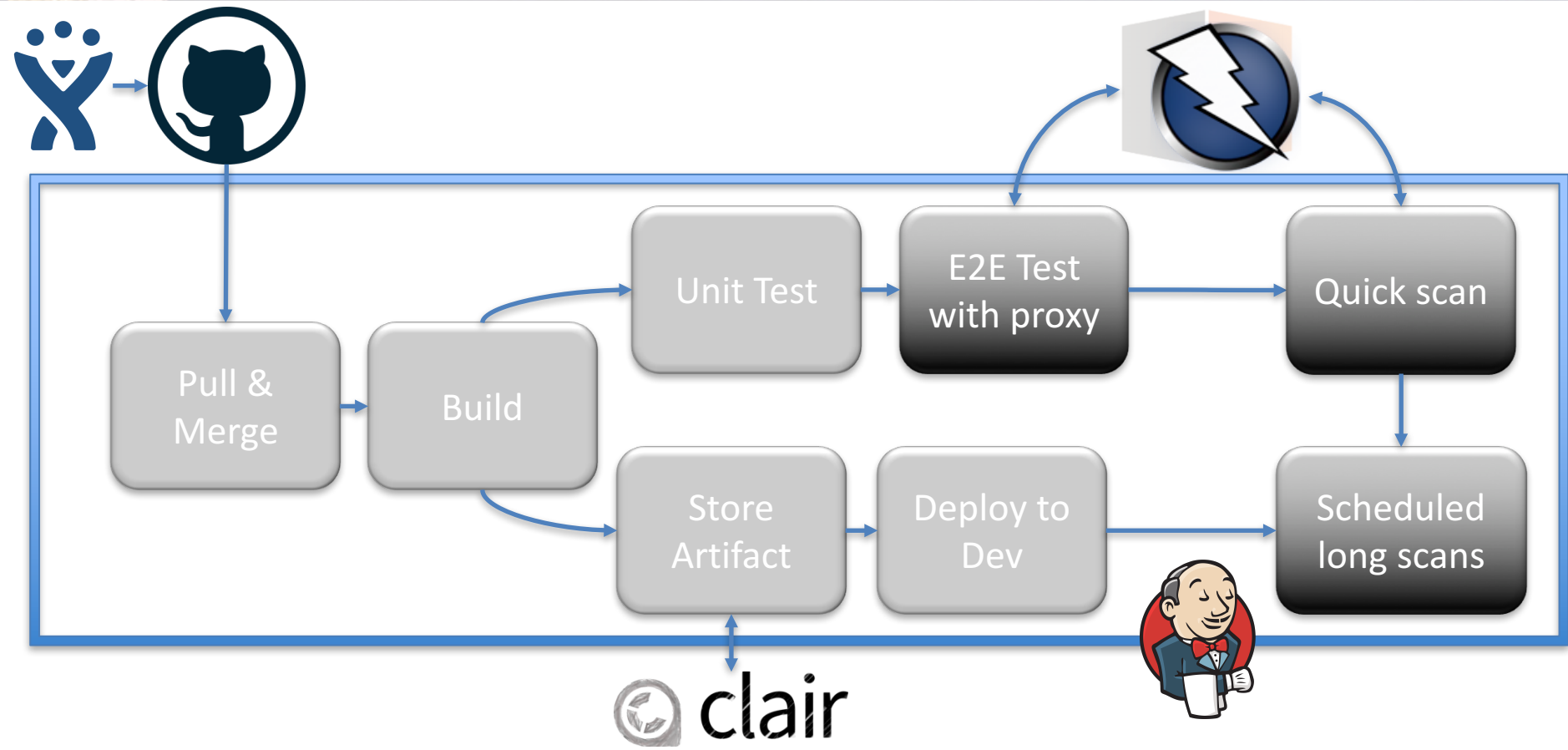
Add dependency & license checkers on top of  
quality tooling.



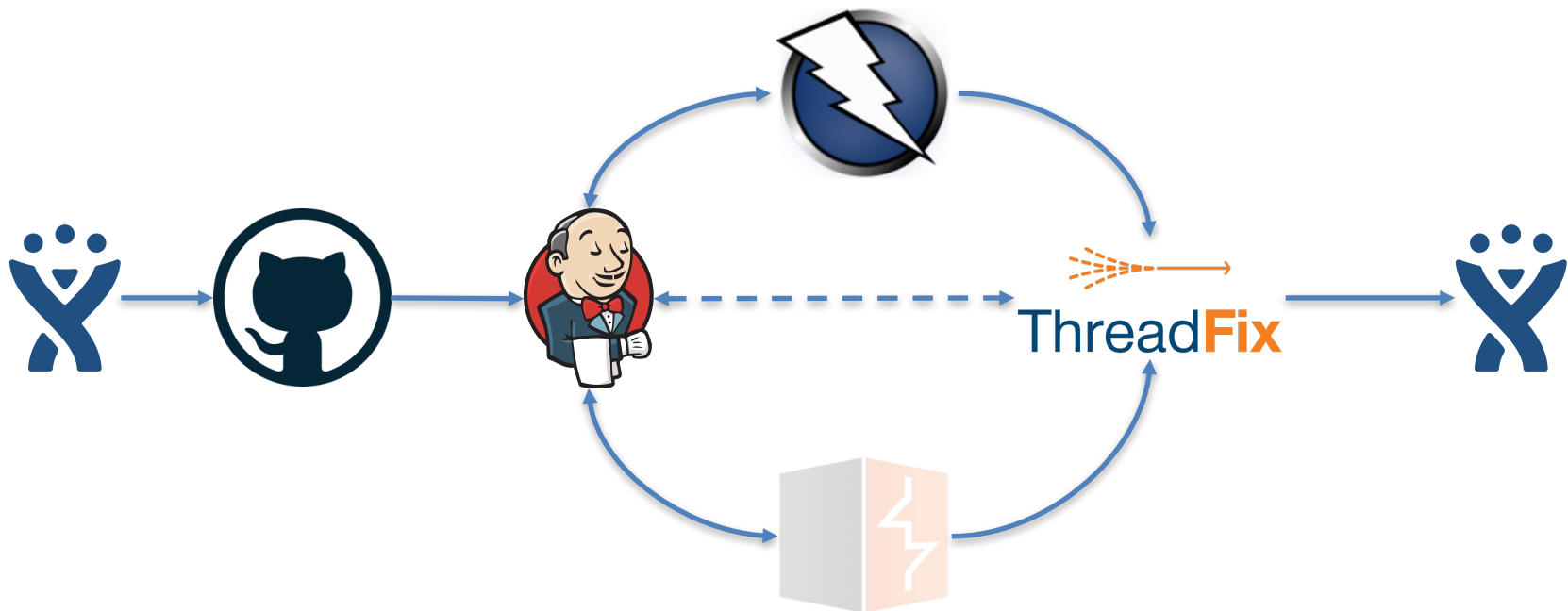
Get feedback FAST!



# The Solution: Feeding ZAP & BURP



# The Solution: DAST & reporting







# The Solution: Clair

- Run Clair on the created containers.
- *Todo: run Clair regularly on the registry, add whitelists & integrate with Threadfix.*



# The solution: Containerize!

- Our tools embedded in containers:
  - + Less additional platform complexities
  - + Can run anywhere (locally / deployed)
  - + Easy to scale
  - Still need to manage the data!
  - More assets that might contain vulnerabilities
- Not perfect: Still have to harden our assets.



# The solution: a starting point



```
./clair-scanner app/threadfix example-whitelist.yaml http://10.200.98.63:6060  
10.200.98.63
```

```
2017-05-12 10:50:19.712897 I | Analyzing  
014fdc7e45e4e7c5967856fc65d7bb5ff0b324fe4ef1ac8ce448843ab310416a  
And 9 other layers...
```

Giving:

```
2017-05-12 10:50:19.854789 I | Image contains unapproved vulnerabilities: [CVE-2017-6508]
```

- A vulnerability in wget...
  - Used when creating the container
  - Not used during runtime





# The Solution: Did it work?

# YES!

Not all components are in, but feedback is  
already of great value

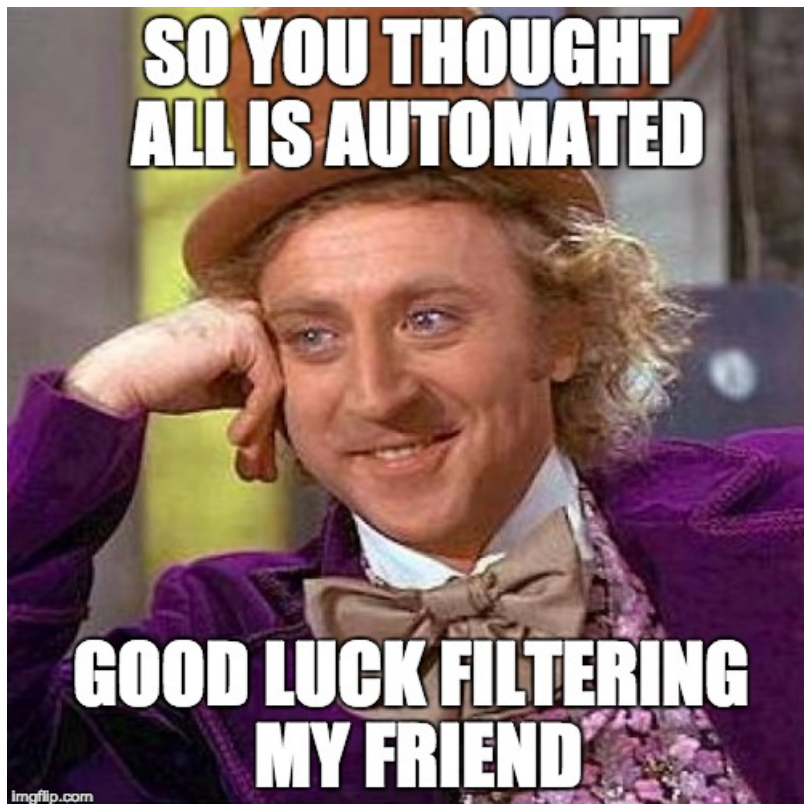


# The bumps on the road

And their countermeasures



# Bump 1: False positives





# Bump 1: False positives

- Use settings/plugins in app → no scaling.

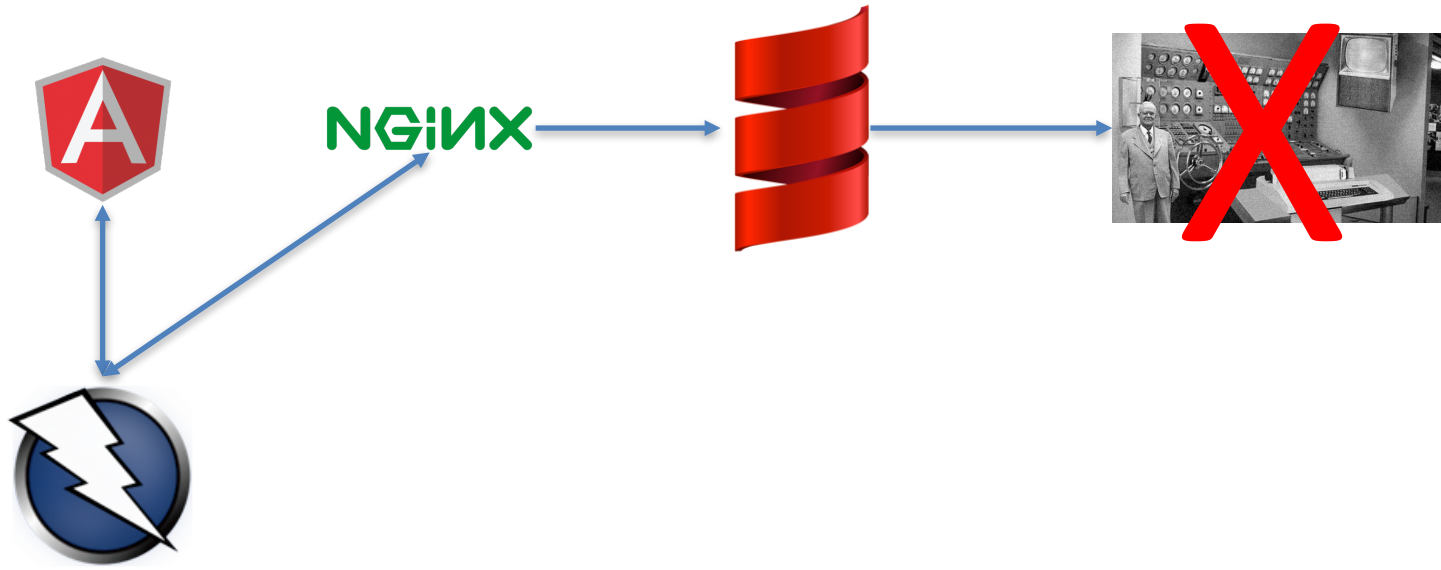
- Use a DB with a framework:



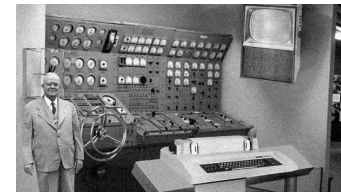
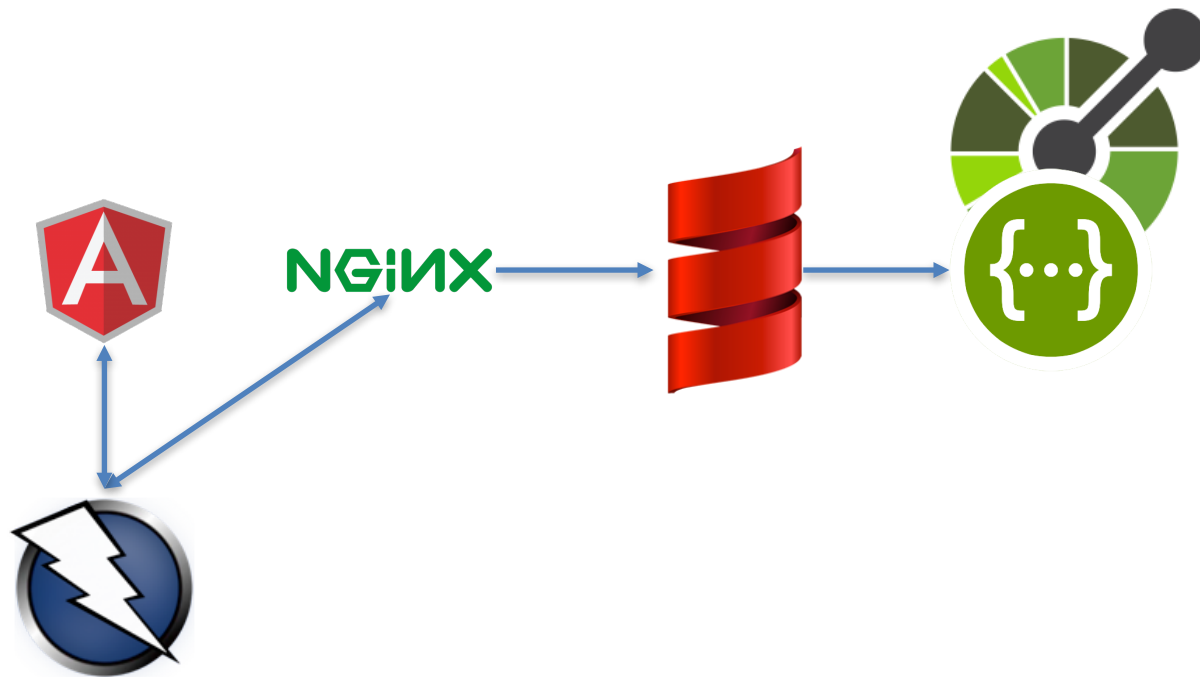
**BDD-Security**

- Use apps like  ThreadFix &  DEFECTdojo

# Bump 2: Legacy APIs



# Bump 2: Legacy APIs



Test legacy APIs separately ☹️





# Bump 3: Not frustrate developers

- Give feedback fast!
- Automate all the things!
- Be part of the team
- Filter & suppress false positives ASAP
- Use known tooling



# Bump 4: Integrating Burpproxy

- Integration with Burp is not completed
  - Custom builds for containers
  - At time of testing: Additional extensions necessary to have a proper REST API

# Bump 5: False negatives....

Security automation does not mean: no manual pentesting.



Even when you add more tools (which we have to...).



## Bump 6: Platform team availability







OWASP  
AppSec EU  
**Belfast**  
8-12 May, 2017

# Recap



# Recap

- Automate all the things: get feedback FAST.
- Containerize
- Filter false positives
- Stub legacy APIs
- HELP developers, DO NOT frustrate!
- Still a need for manual pentesting & reviewing.
- Get platform-team support!
- Every part of the pipeline is a blessing!





OWASP  
AppSec EU  
**Belfast**  
8-12 May, 2017

# QUESTIONS?



OWASP  
AppSec EU  
**Belfast**  
8-12 May, 2017

# Thank you!

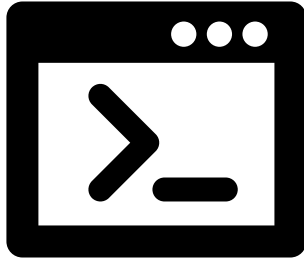




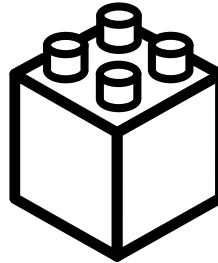
OWASP  
AppSec EU  
**Belfast**  
8-12 May, 2017

# Appendices

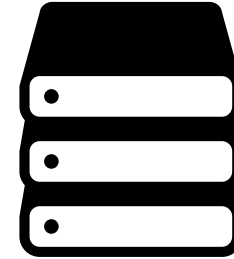
# App.1: hot-swappable platform



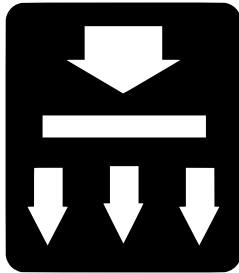
Infrastructure as Code



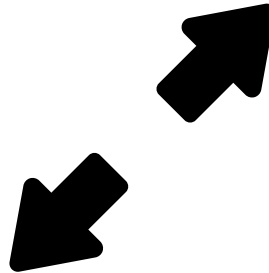
Static Host OS



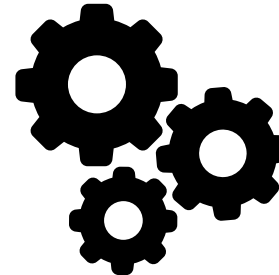
Externalize Data



High Availability  
By Default

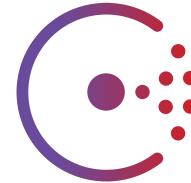
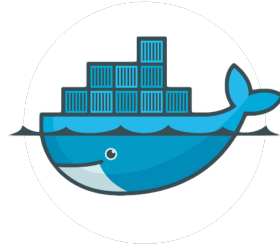


Use Autoscaling



Automated Repeatable  
Bootstrapping

# App.2: Actual deployment



NGINX

Render  
Fleet Unit  
File

Submit  
Fleet Unit

Start  
Containers

Register  
Service

Configure  
proxy