

# DevOps / CI with JavaScript

Delivering somewhat continuously

# Your friendly presenter (again)

Jeffrey Labonski  
jlabonski@chariotsolutions.com  
Consultant, Chariot Solutions

# Your friendly presenter (again)

Jeffrey Labonski  
jlabonski@chariotsolutions.com  
Consultant, Chariot Solutions  
*Hardcore devops believer*

# Devops is chronically misunderstood

So diluted it's almost homeopathic by now



# What it is

(My Understanding of Truth, by Jeff Labonski, Grade 9)

# What it is

(My Understanding of Truth, by Jeff Labonski, Grade 9)

- About People

# What it is

(My Understanding of Truth, by Jeff Labonski, Grade 9)

- About People
- About Processes



# What it is

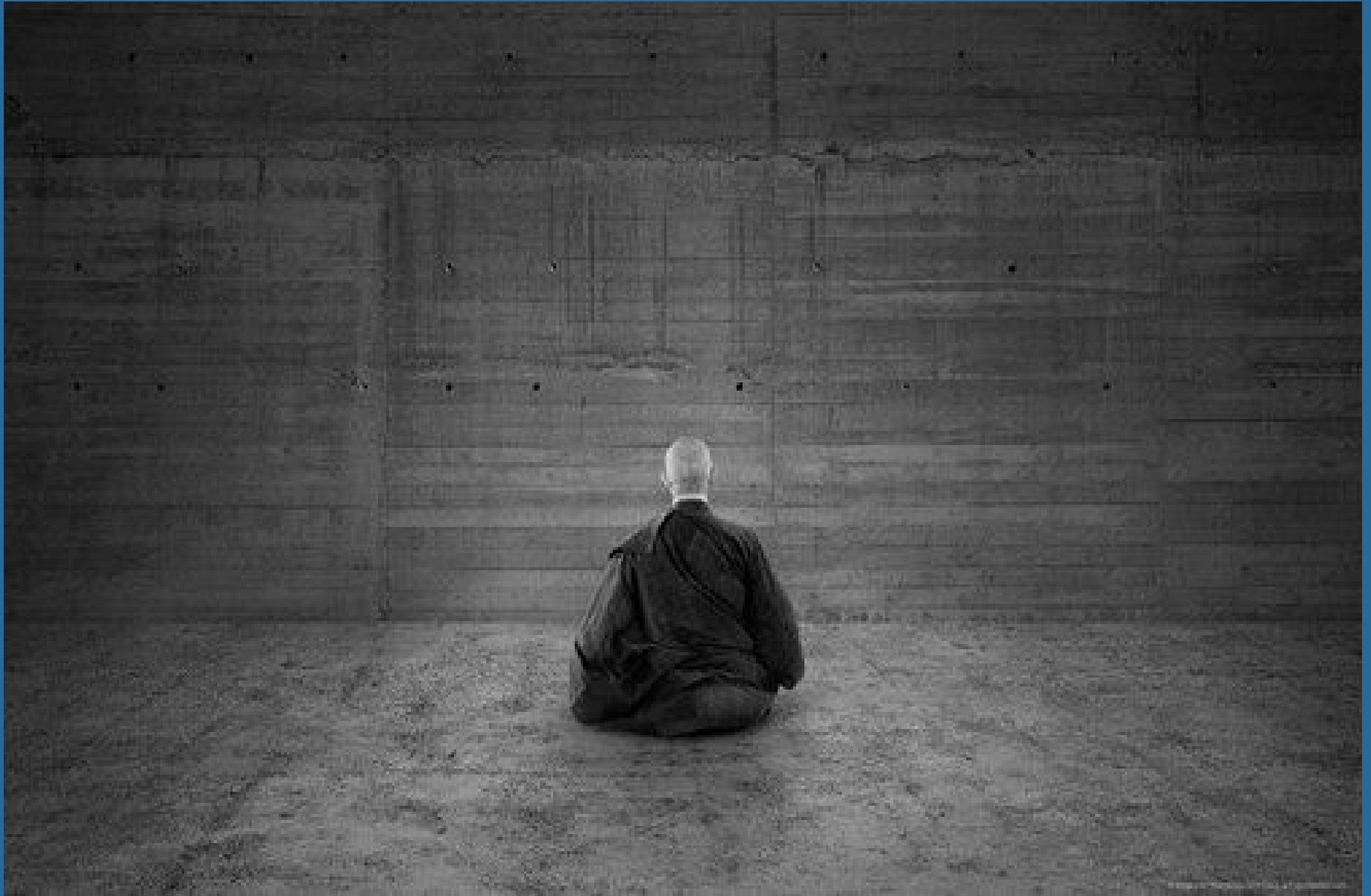
(My Understanding of Truth, by Jeff Labonski, Grade 9)

- About People
- About Processes
- A Philosophy

# The Blameless Postmortem

# What it is not

- A nice title for SAs like "Site Reliability Engineer"
- Magic hidden inside TOGAF or other large cert organizations
- Purchasable
- Easy
- Implementable



**Of course, software is what enables it to  
occur**

**The journey takes a while**

# BUY SEVERAL DEVOPS TODAY! \*

\* This statement has not been evaluated by the Software and Servers Administration. This product is not intended to diagnose, treat, cure, or prevent any project failures.

**Let's actually do things!**



# Can't trust developers

# Can't trust developers

They are all lazy and write bad code

# Can't trust developers

They are all lazy and write bad code

Let's work to prevent that

# Leverage toolchains

- Tighten compilation
  - tsconfig.json
  - react proptypes
  - babel / minify

# Leverage toolchains more

- Lint, lint, lint
  - tslint
  - eslint

# Leverage toolchains more

- Lint, lint, lint
  - tslint
  - eslint
  - package-json-verifier

# Leverage toolchains more

- Lint, lint, lint
  - tslint
  - eslint
  - package-json-verifier
  - npm audit

# Leverage toolchains more

- Lint, lint, lint
  - tslint
  - eslint
  - package-json-verifier
  - npm audit
  - snyk



# Leverage toolchains more

- Lint, lint, lint
  - tslint
  - eslint
  - package-json-verifier
  - npm audit
  - snyk
  - markdownlint

# Leverage toolchains more

- Lint, lint, lint
  - tslint
  - eslint
  - package-json-verifier
  - npm audit
  - snyk
  - markdownlint
  - tsqLint

# Don't never not stop adding quality checks

They are better than you at this

```
scripts: {  
  "snyk:test": "snyk test",  
  "snyk:protect": "snyk protect",  
  "lint": "standardx --verbose",  
  "lint:fix": "standardx --verbose --fix",  
  "lint:sql": "tsqllint sql/*.sql",  
  "validate-package": "pjbv -w -r",  
  "check-node": "check-node-version --package",  
  "test": "cross-env NODE_ENV=unit nyc --reporter=lcov mocha",  
  "coverage:report": "nyc report",  
  "coverage:test": "nyc check-coverage",  
  ...  
}
```

```
scripts: {  
  "preflight": "npm run snyk:test && npm run lint && npm run
```

**Prevent engineering from doing bad things to  
the repository**

# Prevent engineering from doing bad things to the repository

Quality Gates



# Bitbucket Pipelines



```
image:
  name: node:stable

pipelines:
  default:
    - step:
        name: "Lint, build, test"
        caches:
          - node
        script:
          - 'npm install'
          - 'npm run validate-package'
          - 'npm run lint'
          - 'npm run lint:sql'
          - 'npm test'
```

Add a branch permission to specify write and merge access to a specific branch. [Learn more](#)

Branch	Access type	Users and groups	Custom settings
 develop	Write access	None	Deleting this branch is not allowed Rewriting branch history is not allowed
	Merge via pull request	 Developers	Check for at least 2 approvals Check for unresolved pull request tasks Check the last commit for 1 successful build and no failed builds

# Pipelines

[What's new](#)
[Schedules](#)
[Caches](#)
[Usage](#)

[All branches](#)

OTHER BRANCHES

[develop](#) MAIN BRANCH

[status\\_code\\_fix](#)
[webSocket\\_route](#)
[webSocket](#)
[nes](#)
[master](#)

## Pipeline

## Status

## Started

## Duration

#26		Flx linting, move chat server to /api/chat Jeffrey Labonski  469f3c3  develop	<b>Successful</b>	10 days ago	2 min 39 sec
#25		Merge remote-tracking branch 'origin/develop' into develop Matt Gilbride  a0d2750  develop	<b>Failed</b>	11 days ago	1 min 11 sec
#24		conditionally broadcasts disconnect message James Kent  278d824  develop	<b>Failed</b>	12 days ago	1 min 8 sec
#23		broadcasting disconnect message James Kent  2ff8288  develop	<b>Failed</b>	12 days ago	1 min 9 sec
#22		lint Matt Gilbride  fc3251d  develop	<b>Successful</b>	12 days ago	2 min 1 sec

chariot-spa-day / ... / Pipelines



#25

Rerun

✦ a0d2750 Merge remote-tracking branch  
'origin/develop' into develop

🔗 develop

🕒 1 min 11 sec 📅 11 days ago



## Pipeline



Develop: lint, build, test,...

1m 11s

Build Artifacts docker +



Build setup

34s >

npm install

5s >

npm run validate-package

<1s >

npm run lint

2s ▾

+ npm run lint

> spaday-server@0.1.0 lint /opt/atlassian/pipelines/agent/build

> standardx --verbose

standardx: Use JavaScript Standard Style (tweaked by standardx) (<https://github.com/standard/standardx>)  
standardx: Run `standardx --fix` to automatically fix some problems.

/opt/atlassian/pipelines/agent/build/src/websockets/ws.js:35:25: Missing space before =>. (arrow-spa

npm ERR! code ELIFECYCLE

npm ERR! errno 1

npm ERR! spaday-server@0.1.0 lint: `standardx --verbose`

npm ERR! Exit status 1

npm ERR!

npm ERR! Failed at the spaday-server@0.1.0 lint script.

npm ERR! This is probably not a problem with npm. There is likely additional logging output above.

npm ERR! A complete log of this run can be found in:

npm ERR! /root/.npm/\_logs/2018-11-15T13\_56\_25\_248Z-debug.log

Build teardown

<1s >

[Bitbucket] Pipeline for develop failed on dc4ee1a (chariotspaday/server) ➤

Inbox x

Failure x



**Bitbucket** <notifications-noreply@bitbucket.org>

to me ▾

Tue, Nov 13, 1:44 PM (13 days ago)



chariotspaday / server

❗ **Failed** | ⌚ 1 min 33 sec

🔗 develop



**Jeffrey Labonski** committed **dc4ee1a**  
NES work in progress

[View this pipeline](#)

[Configure email notifications](#)

Bitbucket

↩ Reply

➡ Forward

# Can't trust ops

# Can't trust ops

They are all lazy and can't deploy software

**Build artifacts from code, automatically**



```
npm run build:prod && tar -cvzf angular-dist.tar.gz dist
scp angular-dist.tar.gz qaserver:/angular

ssh qaserver
cd /angular
rm -rf dist && tar -xvzf angular-dist.tar.gz
```

```
pipelines:
  default:
    - step:
        name: "Lint, build, test"
        caches:
          - node
        artifacts:
          - '*.tar.gz'
        script:
          - 'npm install'
          - 'npm run lint'
          - 'npm test'
          - 'npm run build:prod'
          - 'tar -cvzf angular-dist.tar.gz --totals dist'
```

# Actions across branches

```
branches:
  develop:
    - step:
        name: "Develop: lint, build, test, publish snapshot"
        caches:
          - node
        artifacts:
          - '*.tar.gz'
        script:
          - 'npm install'
          - 'npm run validate-package'
          - 'npm run lint'
          - 'npm test'
          - 'npm pack'
          - 'npm publish'
```

## SSH key



### SSH keys for Pipelines

Generate SSH keys or add your own keys into Pipelines. If you already have a key in your Docker container you don't need to do anything here – we'll just use that.

Generate keys

Use my own keys

## Known hosts

Add a host address below, then click **Fetch** to see the host's fingerprint. Read about [known hosts for Pipelines](#).

Host address

Click 'Fetch' to see the host's fingerprint

Fetch

Host address

Fingerprint

You haven't added a host yet

```
artifacts:
```

- '\*.tar.gz'

```
script:
```

- 'npm install'
- 'npm build:prod'
- 'tar -cvzf angular-dist.tar.gz --totals dist'
- 'scp \*.tar.gz "\${REMOTE\_SSH}"'
- 'ssh angular@gaserver /usr/local/bin/update\_package.sh'



Builds - test-spring

[Distribute](#)

[Back to all builds](#)

Build ID

638
628
616
614
604
601
585
584
578
576
573
571
570
559
557
554

Agent	Build Agent	Started	Duration	Principal	Artifactory Principal
Gradle/4.9	Gradle/4.9	26 November, 2018 17:...	1.2 minutes	root	test-bitbucket

- Published Modules
- Environment
- Issues
- Licenses
- Diff
- Release History
- »

[Back To All Modules](#)

Module Details: com.test:test-spring:develop-SNAPSHOT

☐ Compare With Previous Build

4 Artifacts

Filter by Artifact Name

out of 1 < >

Artifact Name <input type="checkbox"/>	Type	Repo Path
test-spring-develop-SNAPSHOT-javadoc.jar	javadoc-jar	test-gradle-local/com/test/test-spring/develop-SNAPSHOT/test-spring-develop-...
test-spring-develop-SNAPSHOT-sources.jar	sources-jar	test-gradle-local/com/test/test-spring/develop-SNAPSHOT/test-spring-develop-...
test-spring-develop-SNAPSHOT.jar	jar	test-gradle-local/com/test/test-spring/develop-SNAPSHOT/test-spring-develop-...

# Down the rabbit hole: Docker



# Benefits

# Benefits

- Single, atomic deployment unit.

# Benefits

- Single, atomic deployment unit.
- No dependencies

# Benefits

- Single, atomic deployment unit.
- No dependencies
- Blazingly fast

# Benefits

- Single, atomic deployment unit.
- No dependencies
- Blazingly fast
- Run prod from home

# Benefits

- Single, atomic deployment unit.
- No dependencies
- Blazingly fast
- Run prod from home
- It's very cool

```
FROM node:10

RUN mkdir /node
RUN chown node:node /node
USER node
COPY --chown=node:node . /node
WORKDIR /node

# Optional- careful!
RUN npm rebuild --update-binary

ENTRYPOINT [ "node", "--use_strict" ]
```

```
FROM nginx:stable

COPY docker/nginx.conf /etc/nginx/nginx.conf
RUN mkdir /static
WORKDIR /static

COPY dist /static

CMD [ '/usr/sbin/nginx' ]
```



```
image:
  name: node:10

options:
  docker: true

pipelines:
  default:
    - step:
        - 'npm run build:prod'
        - 'docker login -u "${DOCKER_HUB_USER}" -p "${DOCKER_HUB_PASSWORD}"'
        - 'docker build --pull -t chariotspa/server:develop .'
        - 'docker push chariotspa/server:develop'
```

chariot-spa-day / ... / Pipelines



#26

Rerun

469f3c3 Flx linting, move chat server to /api/chat

develop

2 min 39 sec 10 days ago



## Pipeline



Develop: lint, build, test,...

2m 39s

Build Artifacts docker +



Build setup

46s >

npm install

6s >

npm run validate-package

<1s >

npm run lint

4s >

npm audit

1s >

npm test

2s >

tar -cvzf ../server-src-develop.tar.gz --exclude-vcs --exclude=node\_module...

<1s >

mv ../server-src-develop.tar.gz .

<1s >

docker login -u "\${DOCKER\_HUB\_USER}" -p "\${DOCKER\_HUB\_PASSWORD}"

<1s >

docker build --pull -t chariotspa/server:develop .

49s >

docker push chariotspa/server:develop

13s >

Build teardown

2s >

# Takeaway (whew!)

- Quality gates
- Automate your builds & deployment
- Configuration as code

# Takeaway (whew!)

- Quality gates
- Automate your builds & deployment
- Configuration as code
  - ECR / EBS / EC2 / ELB / RDS / ETC

# Takeaway (whew!)

- Quality gates
- Automate your builds & deployment
- Configuration as code
  - ECR / EBS / EC2 / ELB / RDS / ETC
  - Cloudformation / Terraform / GDM

# Takeaway (whew!)

- Quality gates
- Automate your builds & deployment
- Configuration as code
  - ECR / EBS / EC2 / ELB / RDS / ETC
  - Cloudformation / Terraform / GDM
  - Kubernetes

# Takeaway (whew!)

- Quality gates
- Automate your builds & deployment
- Configuration as code
  - ECR / EBS / EC2 / ELB / RDS / ETC
  - Cloudformation / Terraform / GDM
  - Kubernetes
  - ServiceNow / vmWare vRa & vRo

# Takeaway (whew!)

- Quality gates
- Automate your builds & deployment
- Configuration as code
  - ECR / EBS / EC2 / ELB / RDS / ETC
  - Cloudformation / Terraform / GDM
  - Kubernetes
  - ServiceNow / vmWare vRa & vRo
  - Wherever the cloud takes you



# Never stop improving