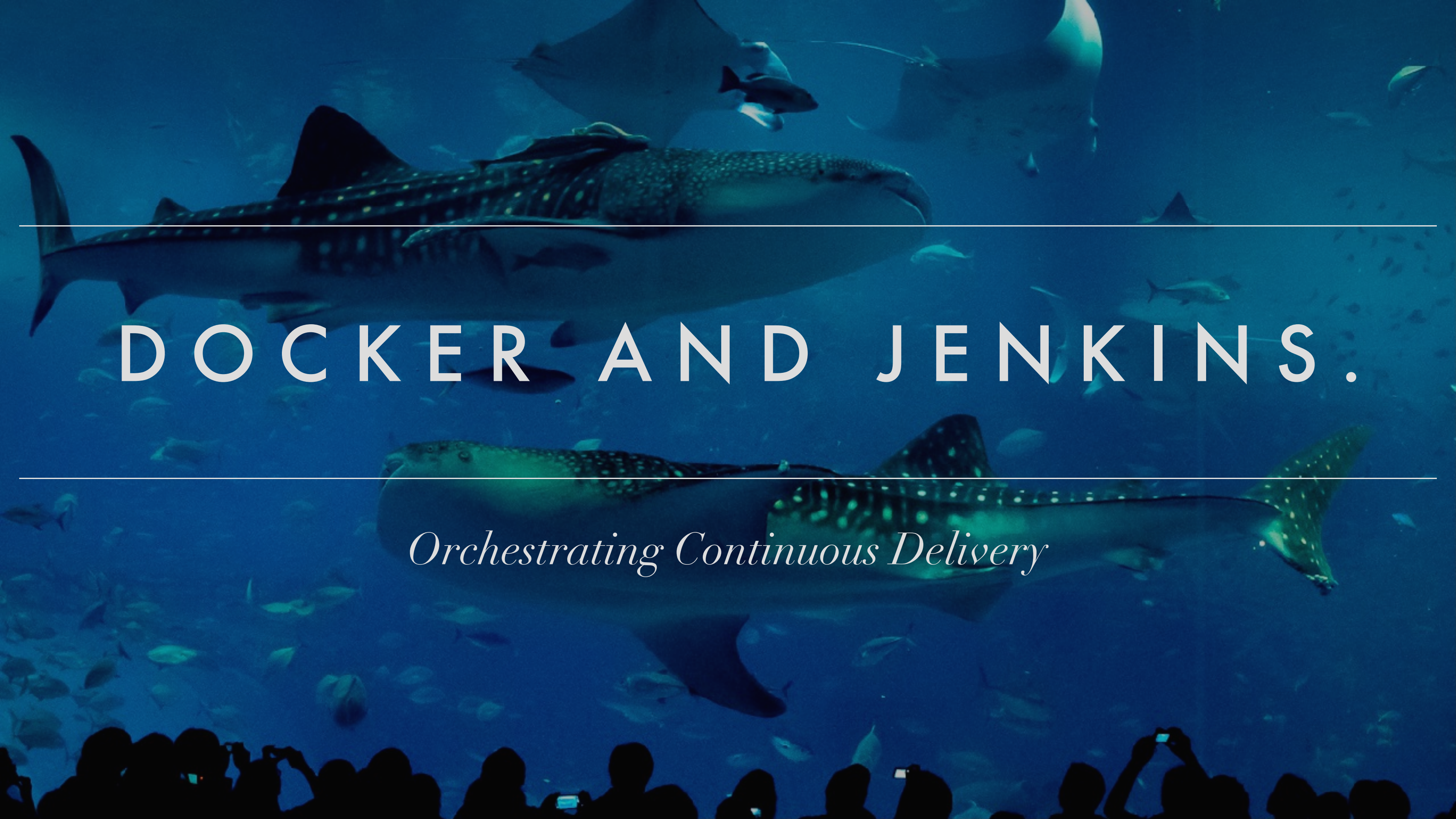


An underwater scene featuring two large whale sharks swimming in deep blue water. The sharks have characteristic white spots on their dark bodies. In the foreground, the silhouettes of a crowd of people are visible at the bottom, many holding up smartphones to take photos of the marine life. The overall atmosphere is serene and awe-inspiring.

DOCKER AND JENKINS.

Orchestrating Continuous Delivery



🐦 ALEXSOTOB – LORDOFTHEJARS.COM



SOFTWARE IS EATING THE WORLD







Software Update

There is a new version of your Tesla Model S software. Schedule installation, install now or close window to postpone.

12	00	
1	10	
2	20	AM
3	30	PM
4	40	

8 hr 47 min from now

SET FOR THIS TIME

INSTALL NOW



This update will take approximately 45 min

During the update process you will not be able to drive the vehicle or use the touchscreen, and your car alarm may be disarmed for a short duration. The car must be in PARK.

Tesla's Model S Safety Solution: Software

BY DOUG NEWCOMB DECEMBER 5, 2013 6 COMMENTS

A recent over-the-air upgrade increases highway ground clearance after vehicles struck road debris and caught fire.

44

SHARES



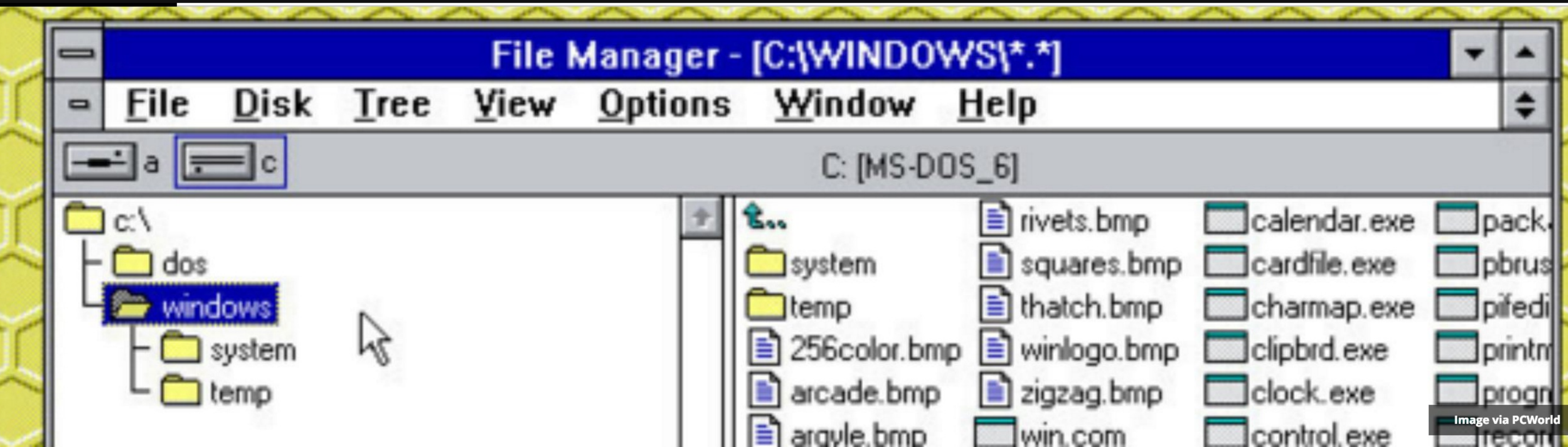
Most automakers typically respond to an investigation by government safety regulators by providing documents, and by approaching federal officials with a certain deference. Tesla, of course, is not a typical automaker.

The electric vehicle (EV) company and its CEO Elon Musk met news of a recent National Highway Traffic Safety Administration (NHTSA)

investigation into fires involving the Tesla Model S with a mix of bluster, statistics, and a

software upgrade. While these tactics are unorthodox in the staid automotive world, they're

in character for Tesla and particularly for Musk.



EUROPE

Windows 3.1 Is Still Alive, And It Just Killed a French Airport

By Pierre Longera

November 13, 2015 | 1:30 pm



\$57 Million Slot Machine Win Blamed on Software Error, Casino Won't Pay

ANDY CHALK | 13 DECEMBER 2011 11:18 PM

65

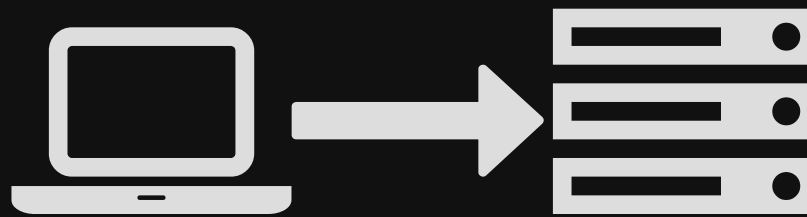
A man who won \$57 million from a slot machine is taking the casino to court after it told him it was a software error and refused to pay up.

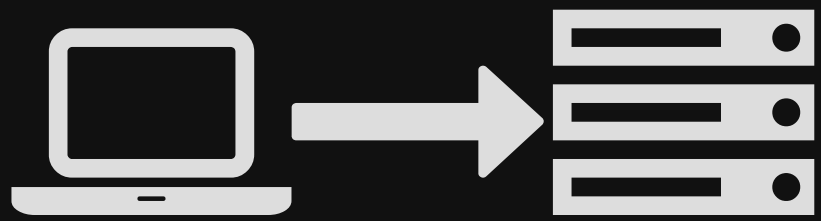
It was a very good day for Behar Merlaku of Switzerland when a slot machine at an Austrian casino went bells-and-whistles bananas, telling him he'd won the grand prize of \$57 million. But it quickly turned into a very bad day when he attempted to claim his prize and was told by the casino owners that a software error had triggered the machine and that they weren't going to pay up. They offered him \$100 and a free meal instead, and he, quite naturally, told them to get bent.



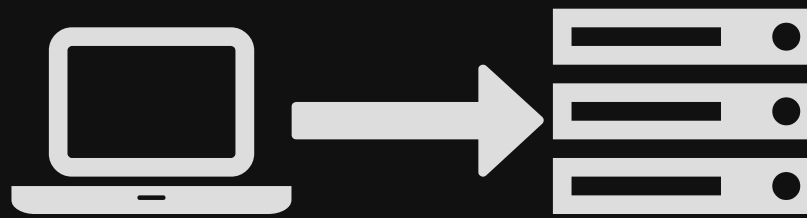
In the casino's defense, the top prize is meant for a five-slot match and Merlaku apparently only matched four slots, but the machine said he won and in his mind, that's all there is to it. So he's taking the casino to court in what is considered the largest claim of its kind ever and one to which casino operators around the world are paying very close attention.

"The jackpot came up loud and clear. There was music and the sum I had won - nearly €43 million - was displayed on the screen," Merlaku said in an interview on Austrian television. "I was so overjoyed and in my head I began calculating what I could do with all this money."

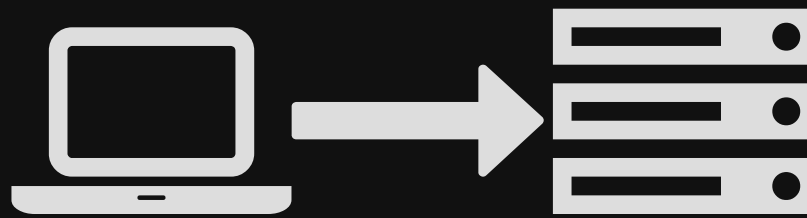




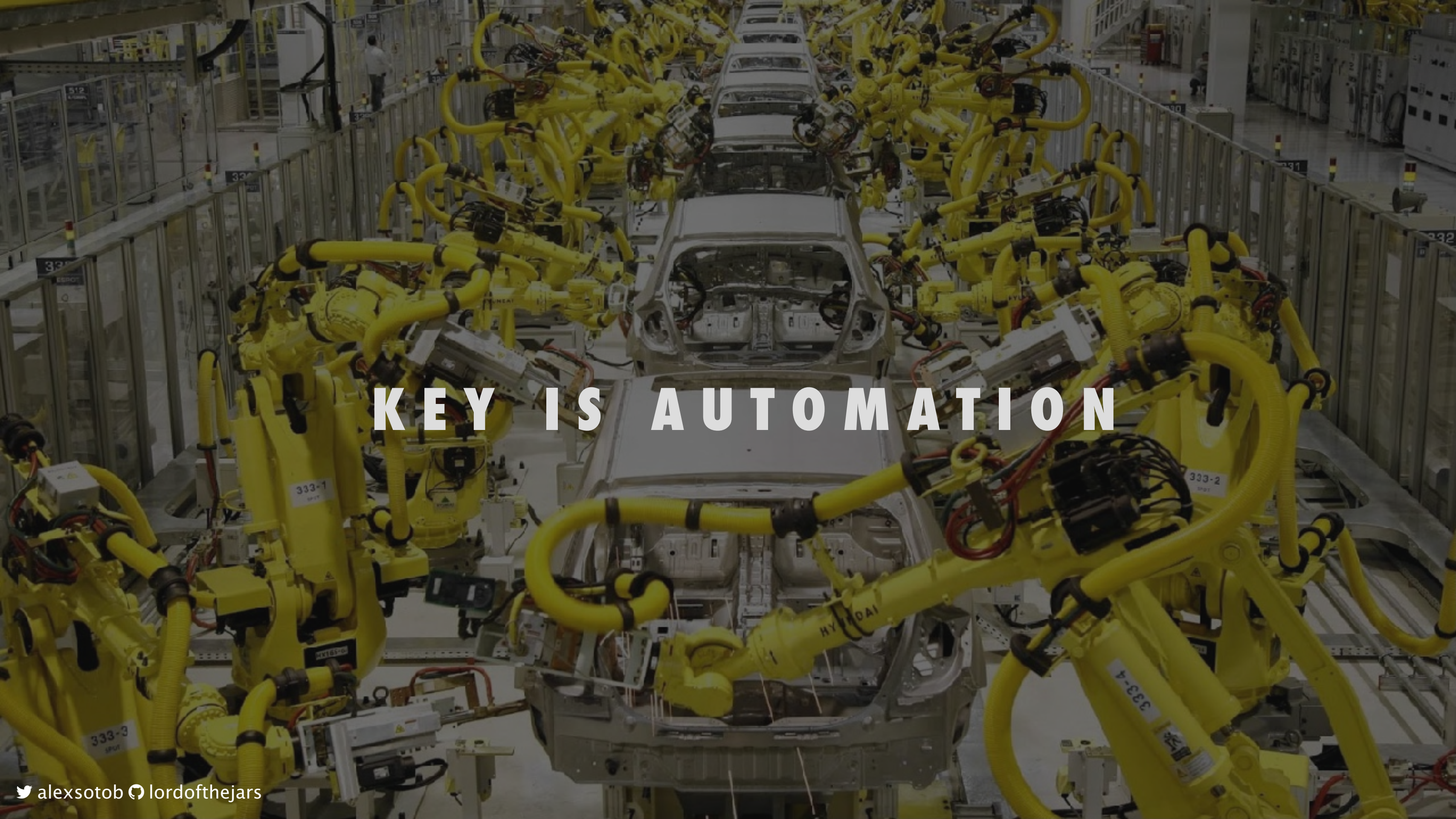
Faster



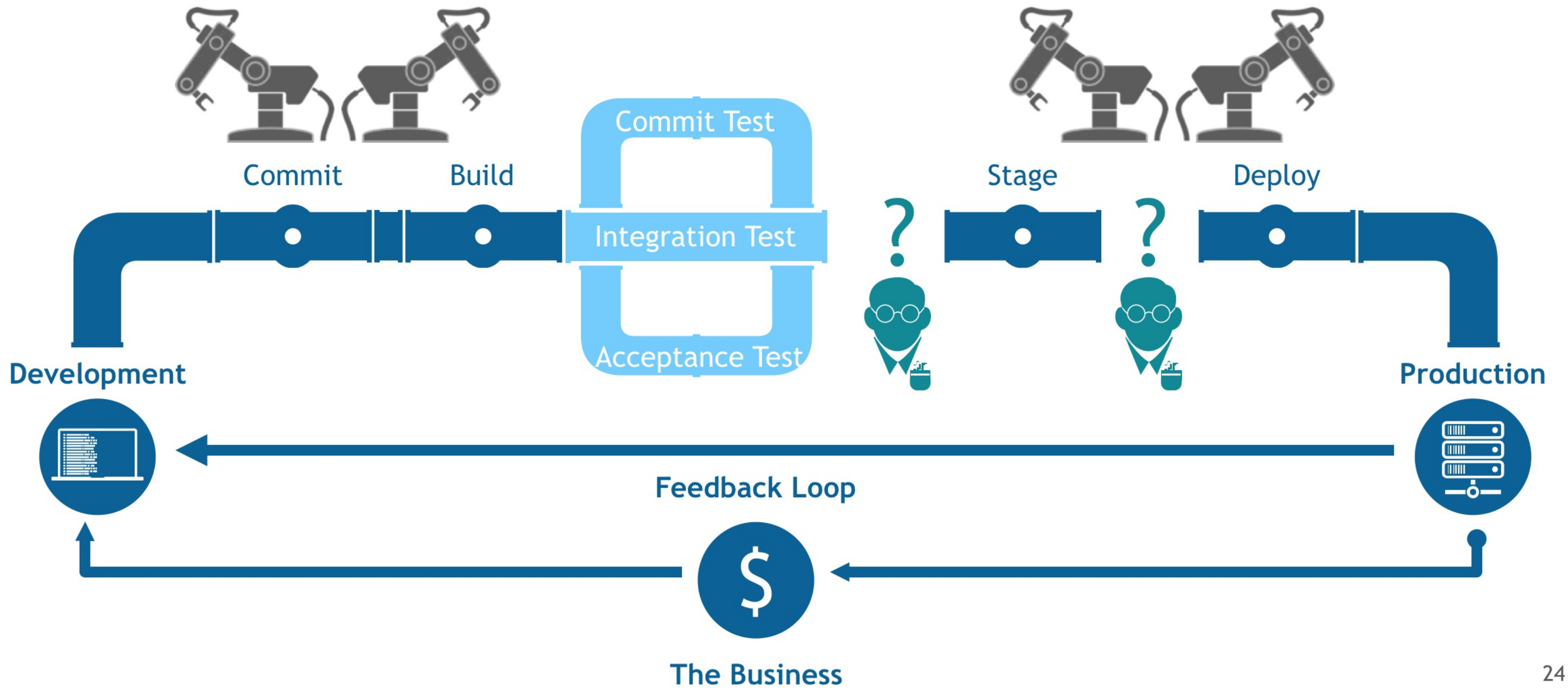
Faster
Sooner



Faster
Sooner
Better



KEY IS AUTOMATION



Over
1000 Jenkins
Plugins

Integration
with over 100
DevOps Tools

Orchestration
of the DevOps
Toolchain

End-to-End
CD Pipeline
Management



Code & Commit

Build & Config

Scan & Test

Release

Deploy



Dev



<DEV OS config>



DEV Server/VM

QA



<QA OS config>



QA Server/VM

Staging



<STG OS config>



STG Server/VM

Prod



<PROD OS config>



PROD Server/VM

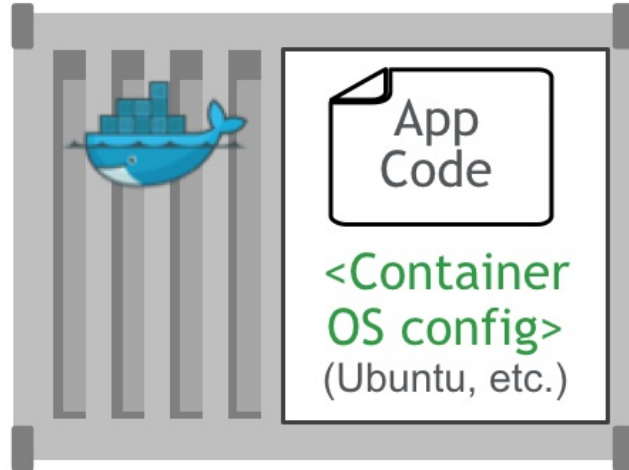


SPEED + FRICTION = HEAT

G O - G O - G O



Dev

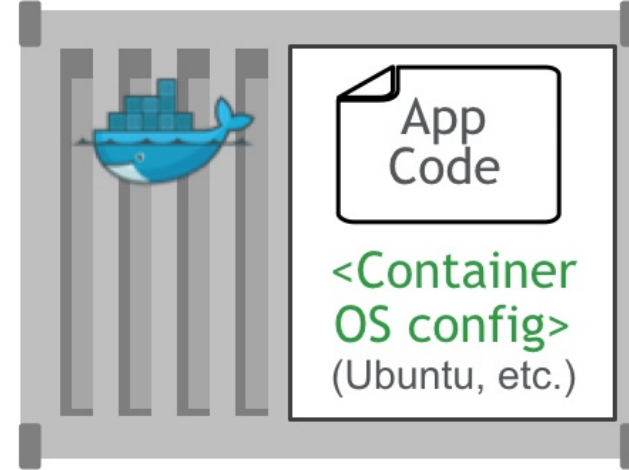


<DEV OS config>



DEV Server/VM

QA

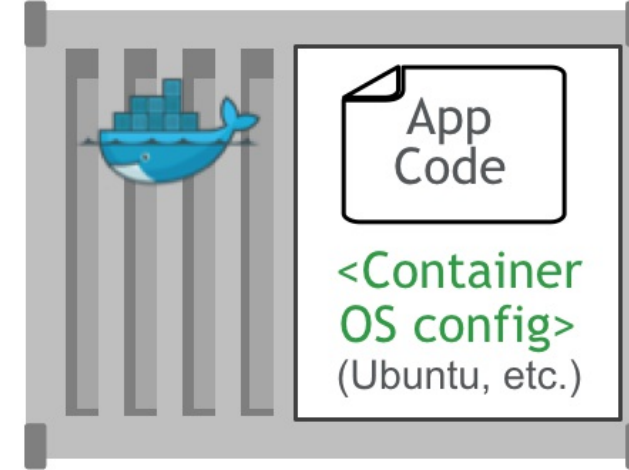


<QA OS config>



QA Server/VM

Staging

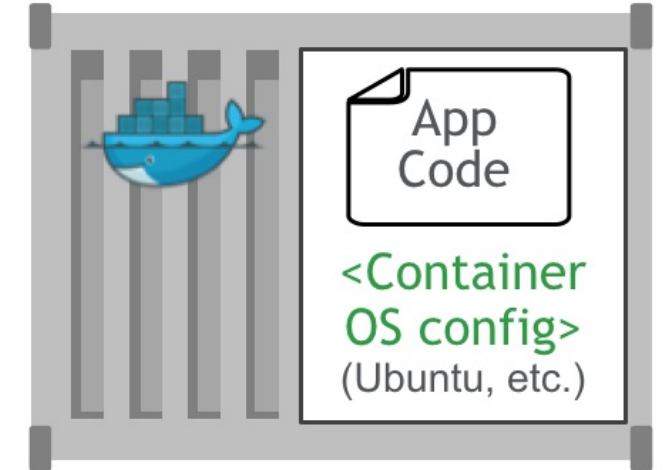


<STG OS config>



STG Server/VM

Prod



<PROD OS config>



PROD Server/VM

HOW I BUILD A DOCKER IMAGE?

```
$ docker run -ti ubuntu:12.10 bash
```

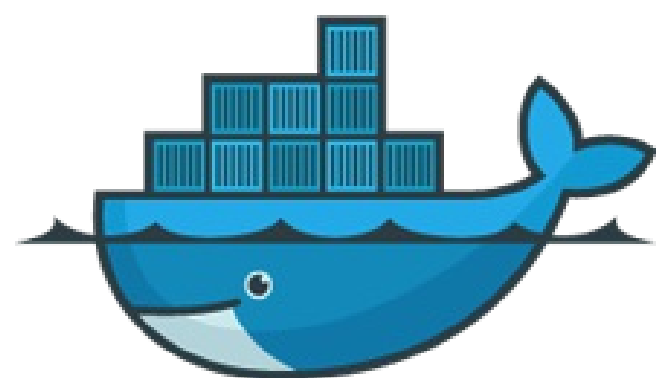
```
[a2b] apt-get update
```

```
[a2b] apt-get install curl
```

```
[a2b] exit
```

```
$ docker commit -m "curl installed" a2b username/image
```

```
$ docker build .
```



+



docker

JENKINS: DOCKER AND CD

Docker Hub Notification: triggers downstream jobs when a tagged container is pushed to Docker Hub

JENKINS: DOCKER AND CD

Docker Hub Notification: triggers downstream jobs when a tagged container is pushed to Docker Hub

Docker Traceability: identifies which build pushed a particular container, displays on Jenkins builds page

JENKINS: DOCKER AND CD

Docker Hub Notification: triggers downstream jobs when a tagged container is pushed to Docker Hub

Docker Traceability: identifies which build pushed a particular container, displays on Jenkins builds page

Docker Custom Build Environment: specifies customized build environments as Docker containers

JENKINS: DOCKER AND CD

Docker Hub Notification: triggers downstream jobs when a tagged container is pushed to Docker Hub

Docker Traceability: identifies which build pushed a particular container, displays on Jenkins builds page

Docker Custom Build Environment: specifies customized build environments as Docker containers

Docker: use a docker host to dynamically provision a slave, run a single build, then tear-down

JENKINS: DOCKER AND CD

Docker Hub Notification: triggers downstream jobs when a tagged container is pushed to Docker Hub

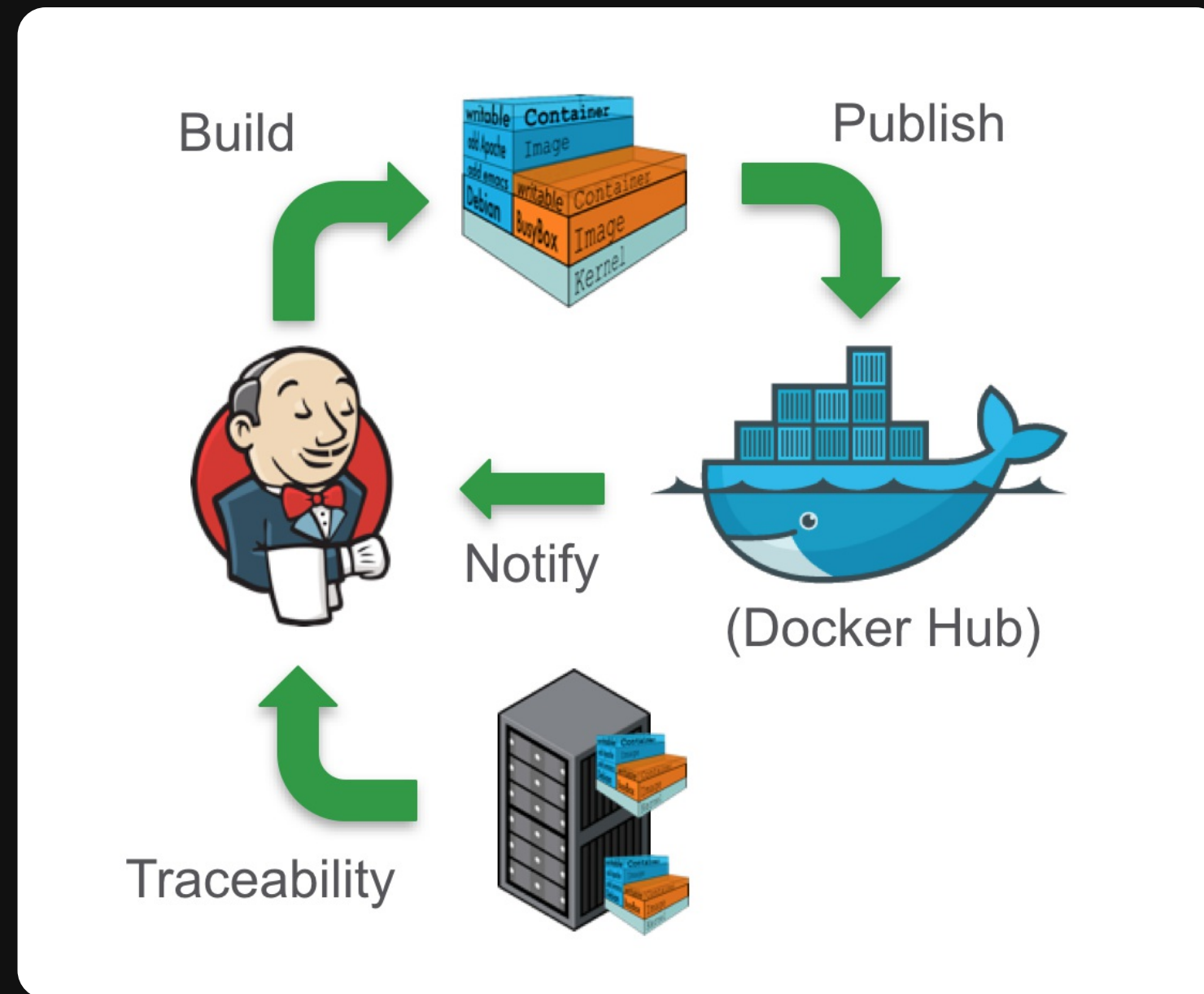
Docker Traceability: identifies which build pushed a particular container, displays on Jenkins builds page

Docker Custom Build Environment: specifies customized build environments as Docker containers

Docker: use a docker host to dynamically provision a slave, run a single build, then tear-down

Build and Publish : builds projects that have a Dockerfile and pushes the resultant tagged image to Docker Hub

BUILD, TEST, & DEPLOY DOCKER IMAGE FROM JENKINS



DOCKER HUB NOTIFICATION PLUGIN

Add Webhook

Webhook Name:

Hook URL 0



Create

Add URL

Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts)
- ☐ Build after other projects are built
- ☐ Build periodically
- ☐ Build pull requests to the repository
- ☐ Build when a change is pushed to GitHub
- ☐ Build when another project is promoted

☒ Monitor Docker Hub for image changes

- ☒ Any referenced Docker image can trigger this job
- ☒ Specified repositories will trigger this job

Repositories

harniman/mobile-deposit-api

Configure Trigger

DOCKER TRACEABILITY PLUGIN



[Back to Project](#)

[Status](#)

[Changes](#)

[Console Output](#)

[Edit Build Information](#)

[Delete Build](#)

[Environment Variables](#)

[See Fingerprints](#)

[Docker Fingerprints](#)

[Previous Build](#)



Docker fingerprints for Docker-demos » create and publish docker image #26 harniman/mobile-deposit-api:26

Image Summary	Created on	Introduced
ID: b66a31c7b100c0916ee86a0f2ca4295dfa057fdb341624a089ff93be08c973f8	23 min	Docker-demos/create and publish docker image #26
ID: 1e7c8f53a87dcd96a141e8a534873813867f408f2e255421ee8c7a5175aa976f	23 min	Docker-demos/create and publish docker image #26

CloudBees Jenkins Enterprise

search

log in | sign up

Jenkins

Docker Traceability

ENABLE AUTO REFRESH

New Item

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

Support

Credentials

Failure Cause Management

Pooled Virtual Machines

Docker Traceability

Build Queue

No builds in the queue.

Build Executor Status

master

1 idle

2 idle

jenkins-beedemo-local-a37896b67a6d

1 idle

2 idle

linux-slave-1

1 idle

linux-slave-2

1 idle

linux-slave-3

1 idle

osx-slave-1

1 idle

vagrant (offline)

windows-webstart (offline)

Docker Traceability

Provides information about Docker deployments related to this Jenkins installation. The detailed info about Docker images and containers is available through links in the list below.

The plug-in provides an extended API. See [API docs](#)

Registered containers

Container Info	Deployment summary
Name: /furious_varahamihira ID/Source: 2119264ec3a41817339cdd6fedfcccc1e8fad3b0d991db5df7fedb69abb893a1 (outside Jenkins) Base image: "harniman/mobile-deposit-api:25" Base image ID: f40dd14f5fde291cd5e886fb5be63c6b41057900951708f6dd23b60cc64dd88c (from Docker-demos/create and publish docker image #25) Parent images: None	

On: Blue (ID: unknown)

Last status: DEPLOYED

Running: true

Exit code: 0

Started at: 2015-10-08T13:04:46.452217263Z

Finished at: 0001-01-01T00:00:00Z

| Name: /drunk_kilby ID/Source: [29a4025532dfc54b3570277ee985f85c59aa69eba2320ea2d7bb0503c1162505](#) (outside Jenkins) Base image: "harniman/mobile-deposit-api:25" Base image ID: [f40dd14f5fde291cd5e886fb5be63c6b41057900951708f6dd23b60cc64dd88c](#) (from [Docker-demos/create and publish docker image #25](#)) Parent images: None |

On: unknown (ID: unknown)

Last status: DEPLOYED

Running: true

Exit code: 0

Started at: 2015-10-08T13:08:45.79075068Z

Finished at: 0001-01-01T00:00:00Z

| Name: /reverent_hawking ID/Source: [4397f115d5f9c786ea0f4e226b7351361a2f1444953b5cb494b70cd095e92be3](#) (outside Jenkins) Base image: "harniman/mobile-deposit-api:27" Base image ID: [fb5889de7e8e4a3681acc17c6bd9e7de901cf8e61c525b97014491347cdd9767](#) (from [Docker-demos/create and publish docker image #27](#)) Parent images: None |

On: mymac (ID: unknown)

Last status: STOPPED

Running: false

Exit code: 143

Started at: 2015-10-09T08:40:32.525257255Z

Finished at: 2015-10-09T08:40:33.108667878Z

| Name: /desperate_banach ID/Source: [7a1d1bb25e4d8e4743194affc849f9c0ae3ca6c54bb834d58e0ac31c70526336](#) (outside Jenkins) Base image: "harniman/mobile-deposit-api:24" Base image ID: [3ca3d81bfc3d7349bbbf6f1e90be3683151d33420e8e616eaacd082c4dd22309](#) (from [Docker-demos/create and publish docker image #24](#)) Parent images: None |

On: unknown (ID: unknown)

Last status: STOPPED

Running: false

Exit code: 143

Started at: 2015-10-08T13:11:49.471354074Z

Finished at: 2015-10-08T13:11:49.740646905Z

| Name: /angry_bhaskara ID/Source: [7aa8cbe1ce83d6e1c8441a900db5b5214721fb21582447446e798ece7f480f97](#) (outside Jenkins) Base image: "harniman/mobile-deposit-api" Base image ID: [b43afa407c288e8d82c8e98c259b5137999ad53840c468747c876366e98828ed](#) (from [Docker-demos/create and publish docker image #20](#)) Parent images: None |

On: prod-server-1 (ID: unknown)

Last status: DEPLOYED

Running: true

Exit code: 0

Started at: 2015-10-08T12:52:23.87176479Z

Finished at: 0001-01-01T00:00:00Z

alex

soto

lordofthejars



Container /reverent_hawking

MD5: 4397f15d5fetc786ea0f4e226b735136

Introduced 1 hr 28 min ago outside Jenkins

Usage

This file has not been used anywhere else.

Container Info

The information below has been retrieved from the latest report (2015-10-09T09:40:33Z).

- Name: /reverent_hawking
- Container ID: 4397f15d5fetc786ea0f4e226b7351361a2f1444953b5cb494b70cd095e92be3
- Origin: (outside Jenkins)
- Created on: 2015-10-09T08:40:32.065977462Z

Sources

- Image ID: [fb5889de7e6e4a3681acc17c6bd9e7de901cf8e61c525b97014491347cdd9767](#) (from [🌐 Docker-demos/create and publish docker image #27](#))
- Image name: "harniman/mobile-deposit-api:27"
- Parent images: None

Status

- Running on: mymac (ID: unknown)
- Last status: STOPPED
- Pid: 0
- Exit code: 143
- Started at: 2015-10-09T08:40:32.525257255Z
- Finished at: 2015-10-09T08:40:33.108667878Z

Raw data

- [Last record in JSON](#)



Deployment Events

Time	Event	State
2015-10-09T09:40:32Z	DEPLOYED	running: true, paused: false, exit code:0
2015-10-09T09:40:33Z	STOPPED	running: false, paused: false, exit code:143

DOCKER PLUGIN

Images

Docker Template

ID	kmadel/dind-jenkins-slave:1.0	
Labels	docker	
Usage	Only build jobs with label restrictions matching this node	
Credentials	jenkins   Add	
Remote Filing System Root	/home/jenkins	
Remote FS Root Mapping		
Instance Cap	4	
DNS		
Port bindings	1234	
Bind all declared ports	☒	
Hostname		



Advanced...

Delete Docker Template

Add Docker Template 

List of Images to be launched as slaves

DOCKER CUSTOM BUILD ENVIRONMENT PLUGIN

Build Environment

☐ Abort the build if it's stuck

☒ Build inside a Docker container



Docker image to use

☐ Build from Dockerfile



☒ Pull docker image from repository



Image id/tag

maven:3.3.3-jdk-8

Docker Host URI



Server credentials

- none -

Add

Docker registry credentials

- none -

Add



Volumes

Add



force Pull

☐

Run in privileged mode

☐

Verbose

☐

User group





Build History

[trend](#)

[#1](#)

Jun 12, 2015 6:13 PM



[RSS for all](#)



[RSS for failures](#)

BUILD AND PUBLISH PLUGIN

Post-build Actions

Archive the artifacts

Files to archive 



Build #75 (Oct 8, 2015 12:03:10 PM)

 [add description](#)



Build Artifacts

 [Dockerfile](#)

218 B  [view](#)

 [mobile-deposit-api-0.0.19-SNAPSHOT.jar](#)

17.71 MB  [view](#)

00000000

Copy artifacts from another project

Project name	Docker-demos/build and test	
Which build	Latest successful build ☐ Stable build only	
Artifacts to copy	**/*.jar, **/Dockerfile	
Artifacts not to copy		
Target directory		
Parameter filters		
	☐ Flatten directories ☐ Optional ☒ Fingerprint Artifacts	
Result variable suffix		

Docker Build and Publish

Repository Name

example/test

Tag

\${BUILD_NUMBER}

Docker server URI

tcp://127.0.0.1:2345

Server credentials

- none -

Add

Docker registry URL

https://docker.example.com

Registry credentials

example/***** (docker.example.com)

Add

Skip Push

☐

Do not push image to registry/index on successful completion

No Cache

☐

Force rebuild - do not use docker cache (may be slower)

Force Pull

☒

Update the source image before building even when it exists locally

Skip Build

☐

Do not build the image

Create fingerprints

☒

Skip Decorate

☐

Do not decorate the build name

Skip tag as latest

☐

Do not tag this build as the latest

Build Context

The project root path for the build. Defaults to the workspace root if not specified.

Dockerfile Path

Path to the Dockerfile for this build. File must be relative to build context. Can be used to specify a Dockerfile with a non-standard filename. Uses Docker client default if not specified.

Delete



**NO MORE BUILD
JOBS / UP / DOWNSTREAMS PLEASE!!!!**



PIPELINE PLUGIN

Continuous Delivery as Code

FEATURES OF PIPELINE PLUGIN

Configuration in Source Repositories

FEATURES OF PIPELINE PLUGIN

Configuration in Source Repositories

Less click-and-type, more code

FEATURES OF PIPELINE PLUGIN

Configuration in Source Repositories

Less click-and-type, more code

From simple to complex

FEATURES OF PIPELINE PLUGIN

Configuration in Source Repositories

Less click-and-type, more code

From simple to complex

Survives Jenkins restarts & connection losses (\$)

FEATURES OF PIPELINE PLUGIN

Configuration in Source Repositories

Less click-and-type, more code

From simple to complex

Survives Jenkins restarts & connection losses (\$)

Reusable Definitions

FEATURES OF PIPELINE PLUGIN

Configuration in Source Repositories

Less click-and-type, more code

From simple to complex

Survives Jenkins restarts & connection losses (\$)

Reusable Definitions

Build history/trend segregated per branches

FEATURES OF PIPELINE PLUGIN

Configuration in Source Repositories

Less click-and-type, more code

From simple to complex

Survives Jenkins restarts & connection losses (\$)

Reusable Definitions

Build history/trend segregated per branches

UI (\$ -> OSS)

Pipeline

Definition

Pipeline script from SCM

SCM

None

Script Path

Jenkinsfile

Pipeline DSL Reference

☒ Snippet Generator

Steps

Sample Step

archive: Archive artifacts

Includes

Excludes

Generate Groovy

JENKINSFILE (1/3)

```
stage 'compileAndUnit'
node {
    git branch: 'master', url: 'https://github.com/lordofthejars/starwars.git'
    gradle 'clean test'
    stash excludes: 'build/', includes: '**', name: 'source'
    stash includes: 'build/jacoco/*.exec', name: 'unitCodeCoverage'
    step([$class: 'JUnitResultArchiver', testResults: '**/build/test-results/*.xml'])
}

stage 'codeQuality'
parallel 'pmd' : {
    node {
        unstash 'source'
        gradle 'pmdMain'
        step([$class: 'PmdPublisher', pattern: 'build/reports/pmd/*.xml'])
    }
}, 'jacoco': {
    node {
        unstash 'source'
        unstash 'unitCodeCoverage'
        gradle 'jacocoTestReport'
    }
}
```

JENKINSFILE (2/3)

```
stage 'assemble-binaries'

node('linux') {
    unstash 'source'
    withEnv(["SOURCE_BUILD_NUMBER=${env.BUILD_NUMBER}"]) {
        gradle 'assemble'
    }
}

input message: "Deploy Application to QA ?"

stage name: 'Deploy to QA', concurrency: 1
node {
    echo "Application Deployed to QA."
}
```

JENKINSFILE (3 / 3)

```
gradle = load 'jenkins/gradle.groovy'

withCredentials([[ $class: 'UsernamePasswordMultiBinding',
                    credentialsId: 'MyID', usernameVariable: 'GIT_USERNAME',
                    passwordVariable: 'GIT_PASSWORD' ]]) {
    sh "git tag -a some_tag -m 'Jenkins'"
    sh "git push https://${env.GIT_USERNAME}:${env.GIT_PASSWORD}@ --tags"
}

def mvnHome tool 'Maven 3.3.3'
sh "${mvnHome}/bin/mvn -B verify"
```

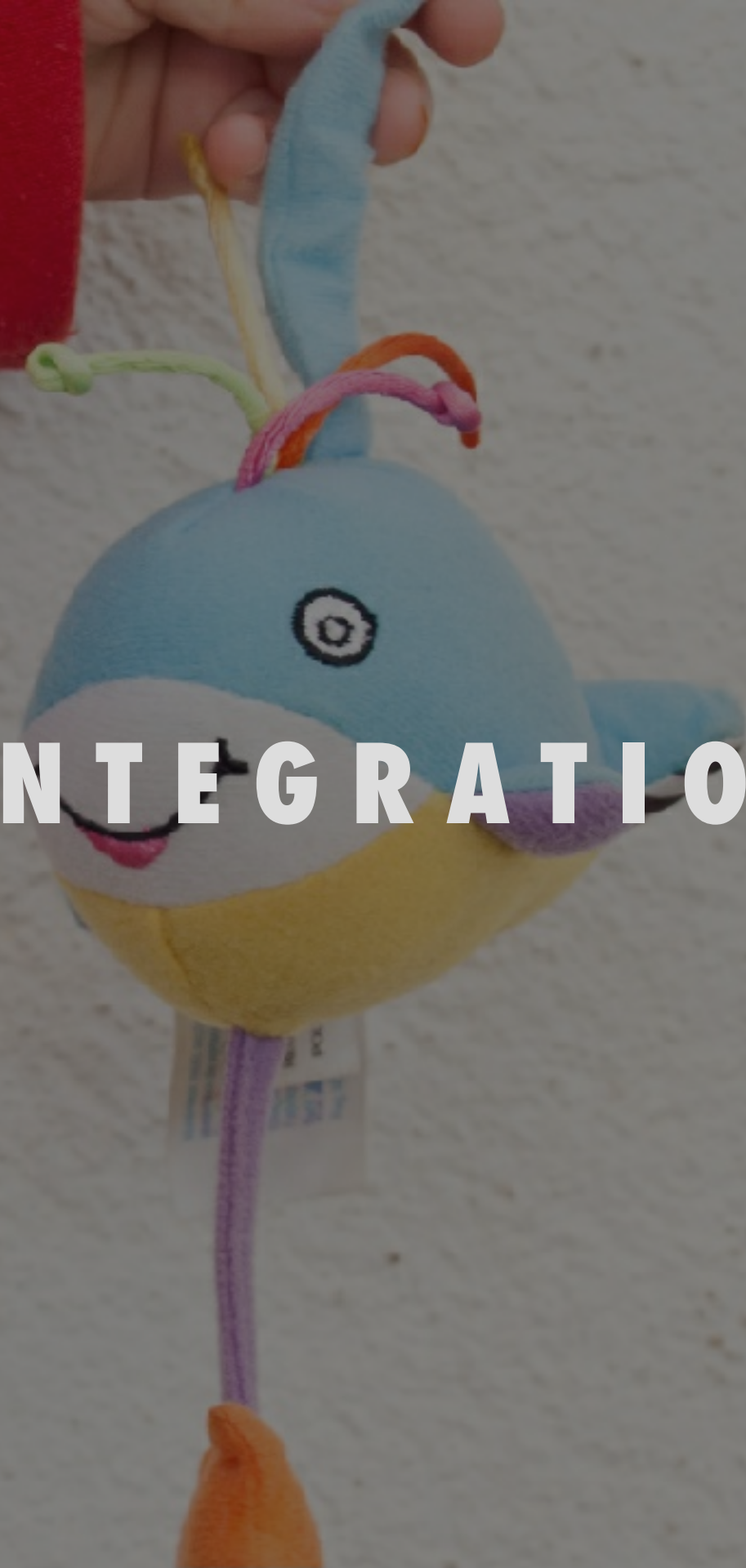


Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

	Name	Kind
	test/***** (slave)	Username with password
	aaa/***** (MyID)	Username with password

DOCKER PIPELINE INTEGRATION



JENKINSFILE (1 / 2)

```
docker.withRegistry('https://lordofthejars-docker-continuous_delivery.bintray.io', 'd4fc3fa9-39f7-47ea-a57c-795642f90989') {  
    git 'git@github.com:lordofthejars/busybox.git'  
    def newApp = docker.build "lordofthejars-docker-continuous_delivery.bintray.io/lordofthejars/javatest:${env.BUILD_TAG}"  
    newApp.push()  
}  
  
docker.image('lordofthejars/javatest').withRun {c ->  
    sh './executeTests.sh'  
}
```

JENKINSFILE (2/2)

```
docker.image('maven:3.3.3-jdk-8').inside('-v ~/.m2/repository:/m2repo -u 1000:50') {  
    git '...your-sources...'  
    sh 'mvn -Dmaven.repo.local=/m2repo clean install'  
}  
  
docker.withServer('tcp://swarm.mycorp.com:2376', 'swarm-certs') {  
    docker.image('httpd').withRun('-p 8080:80') {c ->  
        sh "curl -i http://${hostIp(c)}:8080/"  
    }  
}  
  
def hostIp(container) {  
    sh "docker inspect -f {{.Node.Ip}} ${container.id} > hostIp"  
    readFile('hostIp').trim()  
}
```

StarWars - Stage View

		compileAndUnit	codeQuality	assemble-binaries	publish-binaries	Deploy to QA
Average stage times: (Average <u>full</u> run time: ~1min 7s)		37s	19s	9s	10s	67ms
#10	Jan 26 13:04 2 commits	31s	22s	9s	12s (paused for 1min 16s)	64ms
#9	Jan 26 12:58 4 commits Retry	1min 11s	27s	11s	10s (paused for 56s)	70ms
#8	Jan 26 12:49 2 commits	40s	24s	10s	12s (paused for 43s)	67ms
#7	Jan 26 12:13 4 commits	30s	15s	6s	8s	

DIFFERENT WAYS OF DOING SAME THINGS WITH DSL



SIMPLE BUILD FOR PIPELINE PLUGIN

JENKINSFILE

```
simpleBuild {  
    machine = "hi-speed"  
    docker = "java:1.9"  
  
    env = [  
        FOO : 42,  
        BAR : "YASS"  
    ]  
  
    git_repo = "https://github.com/cloudbeers/PR-demo"  
  
    before_script = "echo before"  
    script = 'echo after $FOO'  
    after_script = 'echo done now'  
  
    notifications = [  
        email : "foo@bar.com"  
    ]  
}
```



LET'S WIND
DOWN

CONCLUSIONS

Automation is the Key

CONCLUSIONS

Automation is the Key
Docker Simplifies Environment Problems

CONCLUSIONS

Automation is the Key
Docker Simplifies Environment Problems
Jenkins is Ready For Docker and CD

CONCLUSIONS

Automation is the Key

Docker Simplifies Environment Problems

Jenkins is Ready For Docker and CD

Pipeline as Code gives you Freedom



ONLY ONE PIECE OF THE JIGSAW

80:54

FCB 3 - 1 CEL

OGI
SPORTS

Nuevo Audi A4

beko.com

QATAR AIRWAYS

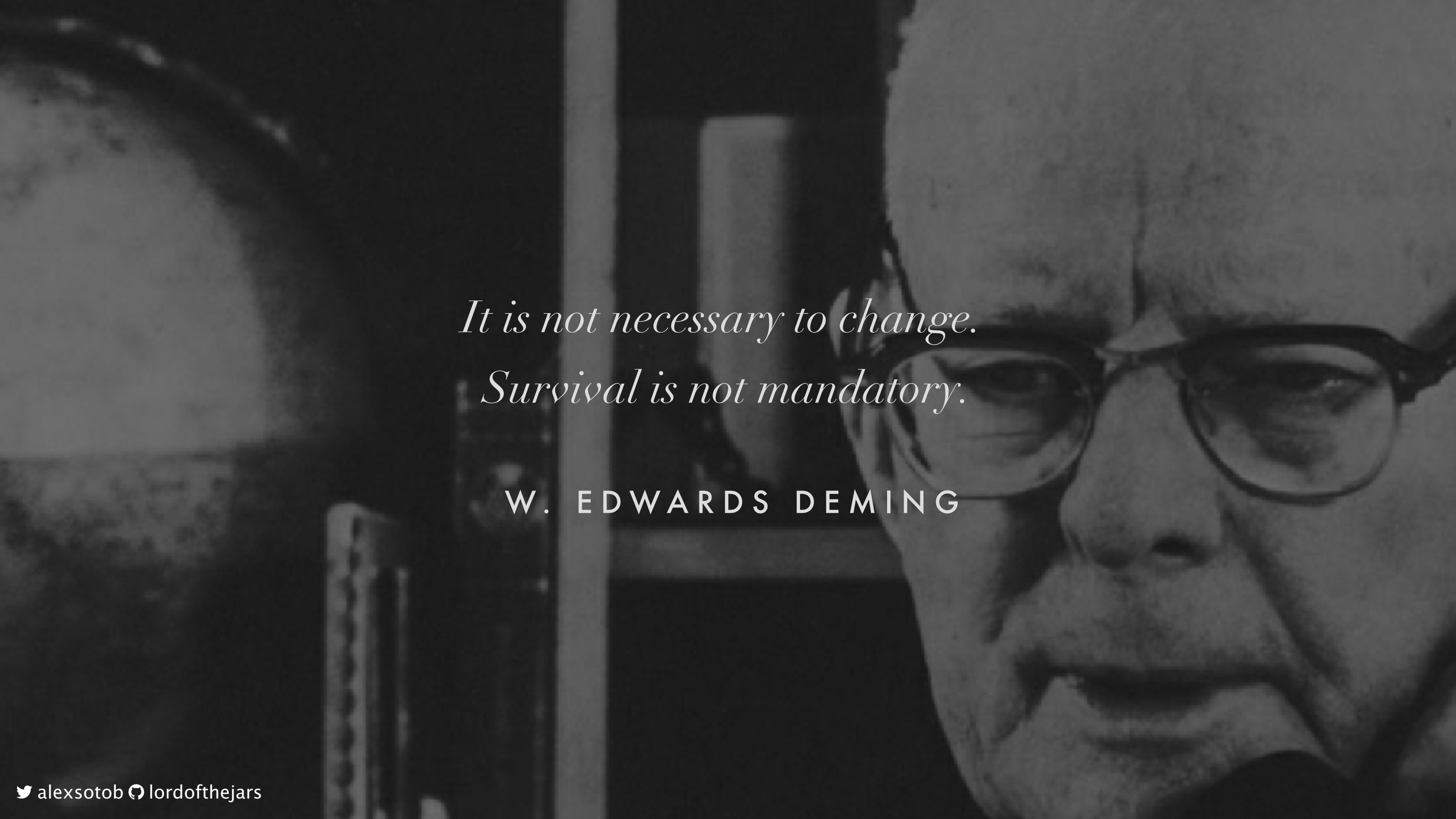
FOLLOW THE RULES, ADAPT TO YOUR
NEEDS

URHD

C L O U D B E E S



J E N K I N S



*It is not necessary to change.
Survival is not mandatory.*

W. EDWARDS DEMING



QUESTIONS



<http://www.slideshare.net/asotobu/jenkins-docker>



asotobu@gmail.com

<https://github.com/lordofthejars/starwars>
<https://wiki.jenkins-ci.org/display/JENKINS/Pipeline+Plugin>
<https://wiki.jenkins-ci.org/display/JENKINS/CloudBees+Docker+Pipeline+Plugin>
<https://wiki.jenkins-ci.org/display/JENKINS/CloudBees+Docker+Traceability>
<https://wiki.jenkins-ci.org/display/JENKINS/CloudBees+Docker+Build+and+Publish+plugin>
<https://wiki.jenkins-ci.org/display/JENKINS/CloudBees+Docker+Hub+Notification>
<https://wiki.jenkins-ci.org/display/JENKINS/Docker+Plugin>
<https://github.com/jenkinsci/simple-build-for-pipeline-plugin>