
codecentric

Continuous Integration for Frontend Code

Presentation

Marcel Birkner - Software Consultant
Bastian Krol - Software Consultant

Marcel Birkner

@marcelbirkner

github.com/marcelbirkner

marcel.birkner@codecentric.de

Bastian Krol

@bastiankrol

github.com/basti1302

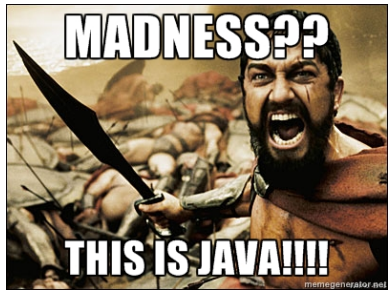
bastian.krol@codecentric.de

About codecentric

Big Data Nerds

Agile Ninjas

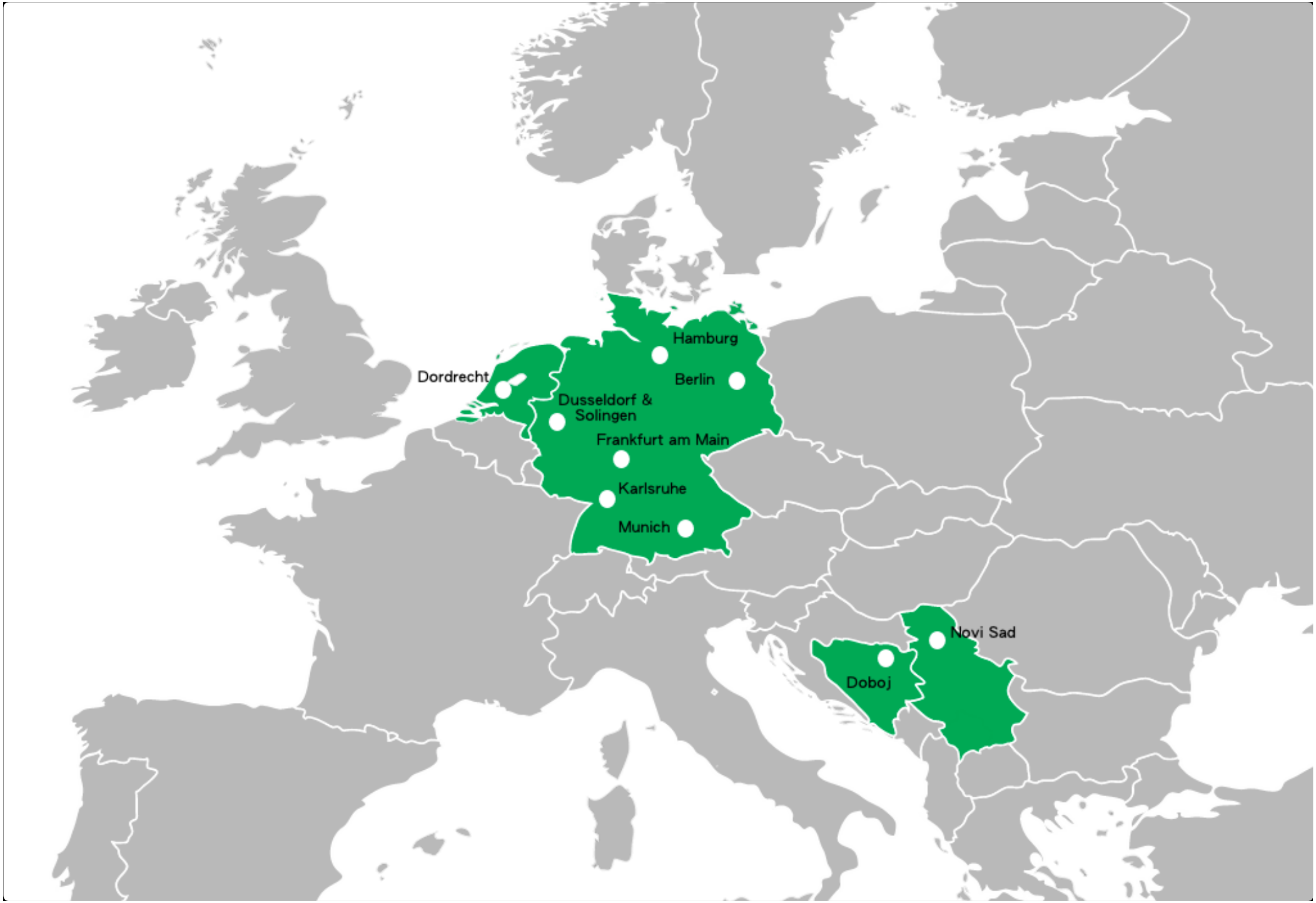
Continuous Delivery Gurus



Java Specialists

Performance Geeks

Hipster Developers


Hamburg
Dordrecht
Berlin
Dusseldorf &
Solingen
Frankfurt am Main
Karlsruhe
Munich
Novi Sad
Doboj

> 280 employees

And we are looking for more!

Continuous Integration for Frontend Code

Why should you use it?

What is the benefit?

JavaScript & CSS have grown up

Frontend Code is Mission Critical!

- Broken JavaScript → broken app
- Broken CSS → broken layout and/or broken app
- It impacts the perceived performance drastically
- Even more so on mobile

And Still ...

- We often build superior CI/CD pipelines for our backends
- But frontend code is often neglected
- For our users, it's always the whole package

We can do better!

What we will cover

- Asset Optimization
- Testing
- The Delivery Pipeline
- Local Development (Docker)

This is frontend only

Ideal: backend & frontend in one CD pipeline

Asset Optimization

Common Problems

- Bad Code Quality
- Misconfigured Caching
- Lots of assets (JS, CSS, images) ⇒ Lots of HTTP requests ⇒ Slow

Tools

- 

Grunt — Task Runner
- 
ESLint — Static Code Analysis
- grunt-contrib-concat — Concatenation
- grunt-contrib-uglify/UglifyJS — Minification
- 

SASS — CSS preprocessor
- 

compass — images → base 64 data URIs
- grunt-version-assets — Versioned File Names

Alternatives

- Grunt: Gulp | Broccoli | npm | make | ...
- ESLint: JSHint | JSLint
- SASS: Less | Stylus
- Module System + Bundler: Webpack | Browserify

of HTTP Requests — Concatenation

GRUNT

```
concat: {
  options: {
    separator: '\n;'
  },
  app: {
    src: [
      '<%= jsSrcDir %>/js/**/*.js',
    ],
    dest: '<%= jsTargetDir %>/app.js'
  }
}
```

Download Size — Minification

GRUNT

```
uglify: {
  app: {
    files: {
      '<%= jsTargetDir %>/app.min.js': [ '<%= jsTargetDir %>/app.js' ],
    }
  }
}
```

of HTTP Requests — Embed Images in CSS

SCSS

```
.img-foo {
  background-image: inline-image("foo.png");
}
```

GRUNT

```
sass: {
  app: {
    options: { compass: true, },
    files: { '<%= cssTargetDir %>/master.css': '<%= cssSrcDir %>/master.scss', }
  }
},
```

CSS

```
.img-foo {
  background-image:
    url('...');
}
```

Use the Browser Cache

Versioned File Names

- time stamp: app.min.20150619-0913.js
- hash: app.min.2412fbca2a07a.js
- content changes ⇒ file name changes
- file can be kept in browser cache forever

Versioned File Names (cont'd)

GRUNT

```
versioning: {
  options: {
    grepFiles: [ '<%= appTargetDir %>/**/*.html', ]
  },
  css: {
    src: [
      '<%= cssTargetDir %>/app.min.css',
    ]
  },
  js: {
    src: [
      '<%= jsTargetDir %>/app.min.js',
      '<%= jsTargetDir %>/vendor.min.js',
    ]
  },
},
```

Versioned File Names — Before

HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    ...
    <link href="css/app.min.css" rel="stylesheet">
    ...
  </head>
  <body>
    ...
    <script src="js/vendor.min.js" type="text/javascript"></script>
    <script src="js/app.min.js" type="text/javascript"></script>
  </body>
</html>
```

Versioned File Names — After

HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    ...
    <link href="css/app.min.b678e30139fc04.css" rel="stylesheet">
    ...
  </head>
  <body>
    ...
    <script src="js/vendor.min.dda09628f6e1da.js" type="text/javascript"></script>
    <script src="js/app.min.8e46534a4f66158.js" type="text/javascript"></script>
  </body>
</html>
```

Development: Turnaround Time Is Important

- Production Mode vs. Development Mode
- grunt watch
- Live Reload

Production Mode versus Development Mode

	Production	Development
JS	concatenated, minified	source files, not minified
CSS	compiled (SASS), concatenated, minified	only compiled (SASS)
Images	Embedded into CSS	Embedded into CSS (by Sass/Compass)
HTML	references optimized assets	references source assets

Development Mode - Replace References

HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    ...
    <!-- build:css css/app.min.css -->
    <link rel="stylesheet" href="css/master.css">
    <link rel="stylesheet" href="css/dashboard.css">
    ...
    <!-- /build -->
  </head>
  <body>
    <!-- build:js js/app.min.js -->
    <script src="js/app.js" type="text/javascript"></script>
    <script src="js/routes.js" type="text/javascript"></script>
    ...
    <!-- /build -->
 </body>
</html>
```

Development Mode - Replace References (cont'd)

GRUNT

```
processhtml: {
  dist: {
    files: {
      '<%= appTargetDir %>/index.html': ['<%= appSrcDir %>/index.html']
    }
  }
},
```

Alternative: grunt-usemin to concat, minify & replace in one step

grunt watch

GRUNT

```
watch: {
  files: [
    '<%= jsSrcDir %>/**/*.js',
    '<%= cssSrcDir %>/**/*.scss',
    '<%= htmlSrcDir %>/**/*.html',
  ],
  tasks: [
    'dev-build',
  ],
  options: {
    livereload: true,
  }
}
```

dev-build

GRUNT

```
grunt.registerTask('dev-build', [
  'copy:cssThirdParty',
  'sass',
]);
```

Live Reload

- See changes instantly
- Never press F5 again
- Let's see this in action!

Measure it

- Google PageSpeed
- Yslow
- Fiddler

Comparison

Unoptimized Version

29 requests | 579 KB transferred | Finish: 319 ms |

Optimized Version

5 requests | 206 KB / 206 KB transferred | Finish: 155 ms

Testing

Front end unit tests

Karma

- Open Source Test Runner
- Created by the AngularJS team
- Write tests in Jasmine, Mocha, QUnit
- CI support (Jenkins, Travis)
- Based on Node.js and Socket.io
- Run in Headless Modus with PhantomJS
- Supported Browsers: Firefox, Chrome, Safari, IE (Desktop and Mobile)

Karma and Mocha (JS Test Framework)

JAVASCRIPT

```
var expect = chai.expect;
beforeEach(module('project-staffing'));
describe('UpperCase Test', function() {
  it('should convert first charactor to UpperCase', inject(function(uppercaseFilter) {
    expect(uppercaseFilter('a')).to.equal('A');
    expect(uppercaseFilter('hello world')).to.equal('Hello World');
  }));
});
```

Running Unit Tests with Karma

BASH

```
npm install -g karma-cli
karma start karma.conf.js
// or
grunt karma
```

Sinon (Mocking Framework)

JAVASCRIPT

```
var ActivityService;
var $http;

beforeEach(inject(function(_ActivityService_, _$http_) {
  ActivityService = _ActivityService_;
  $http = _$http_;
  sinon.stub($http, 'post', function(){});
}));

describe('Activity Service', function() {
  it('should have send http POST to backend after saving one activity',
      inject(function(ActivityService) {
    ActivityService.saveActivity('user', 'action', 'object');
    expect($http.post.callCount).to.equal(1);
  }));
});
```

Chai Assertion Library (BDD/TDD framework)

Should

```javascript
chai.should();
foo.should.be.a('string');
foo.should.equal('bar');
```

Expect

```javascript
var expect = chai.expect;
expect(foo).to.be.a('string');
expect(foo).to.equal('bar');
```

Assert

```javascript
var assert = chai.assert;
assert.typeOf(foo, 'string');
assert.equal(foo, 'bar');
```

End2End tests

Protractor

- Open Source E2E Testframework for AngularJS Apps
- Tests run in a real browser
- Tests can be written with Jasmine (default), Mocha, Cucumber
- No more waits and sleeps
- Build with Node.js on top of WebdriverJS

Protractor and Jasmine (BDD framework)

JAVASCRIPT

```
describe('Manage customer', function() {
  var ptor;

  beforeEach(function() {
    browser.get('/');
    ptor = protractor.getInstance();
    element(by.id('navEmployees')).click();
    element(by.id('navListEmployees')).click();
  });

  it('should navigate to list employees page', function() {
    expect(ptor.getCurrentUrl()).toMatch(/#\/list-employees/);
  });

});
```

Protractor and Jasmine

JAVASCRIPT

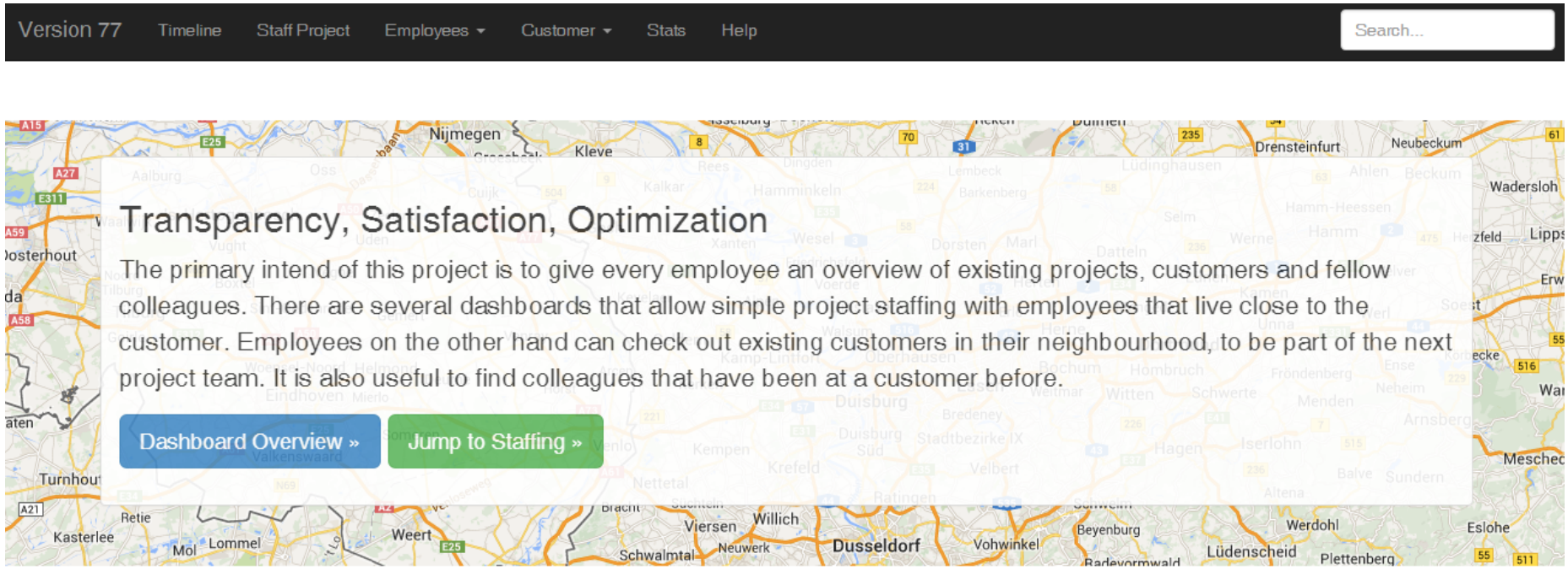
```
it('should find employee Maria on list search page', function() {
  createMultipleEmployees(); // Creates employees: Max, Maria, Daniel, John
  ...
  element(by.id('searchText')).sendKeys('Ma');
  expect(element.all(by.id('employee')).count()).toBe(2);
  element(by.id('searchText')).sendKeys('ria');
  expect(element.all(by.id('employee')).count()).toBe(1);
});
```

Running End2End Tests with Protractor

BASH

```
npm install -g protractor
webdriver-manager start
protractor test/client/e2e/conf.js
```

Demo :: Project Staffing App



Transparency

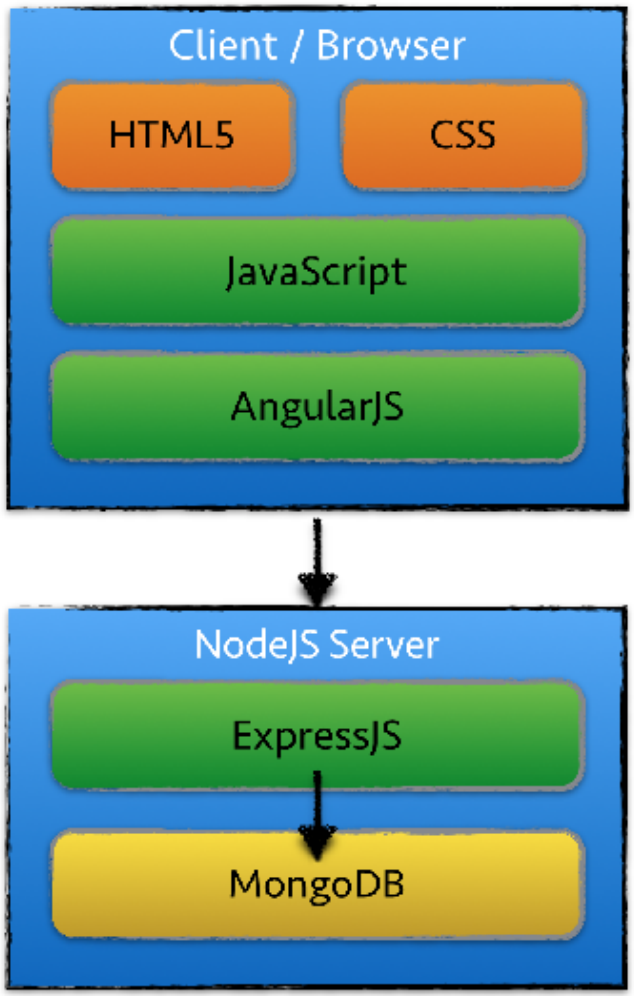
Everybody can see current projects, customers and the location of fellow colleagues. Therefore it is easier to get in contact with the right person when new projects come up.

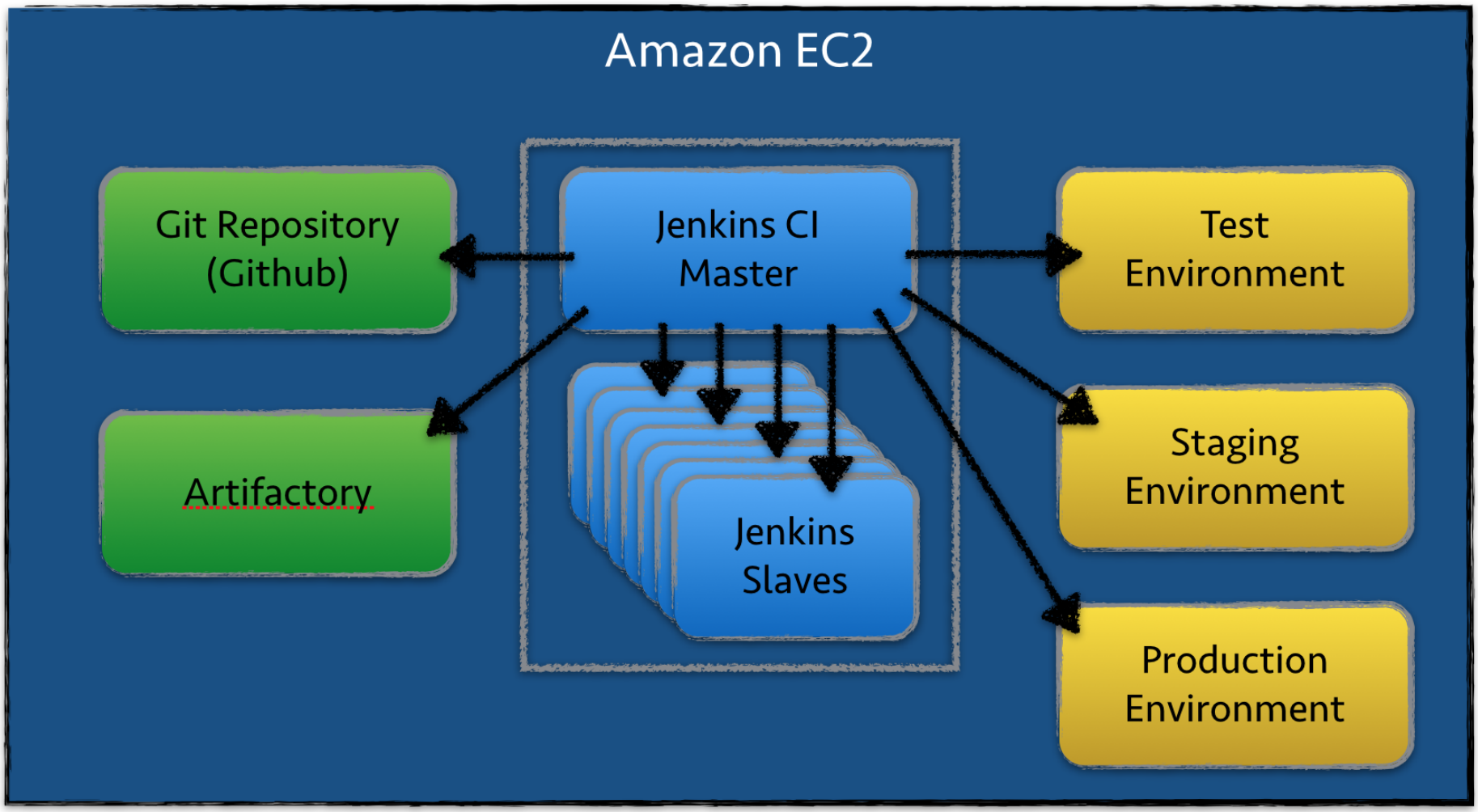
Satisfaction

Every time managers need to staff new projects they can check who will be available at the project start and decide which possible candidate lives closest to the customer. The dashboards allow employees to proactively check out projects in their area or technologies used in projects that they would like to use as well.

Optimization

By choosing employees for projects that live close to the customers, travel expenses will be reduced and travel times of consultants will be minimized.


Client / Browser
HTML5
CSS
JavaScript
AngularJS
NodeJS Server
ExpressJS
MongoDB


Amazon EC2
Git Repository (Github)
Jenkins CI Master
Test Environment
Artifactory
Jenkins Slaves
Staging Environment
Production Environment

Project Staffing App - TEST Environment

http://54.170.140.7:9000/

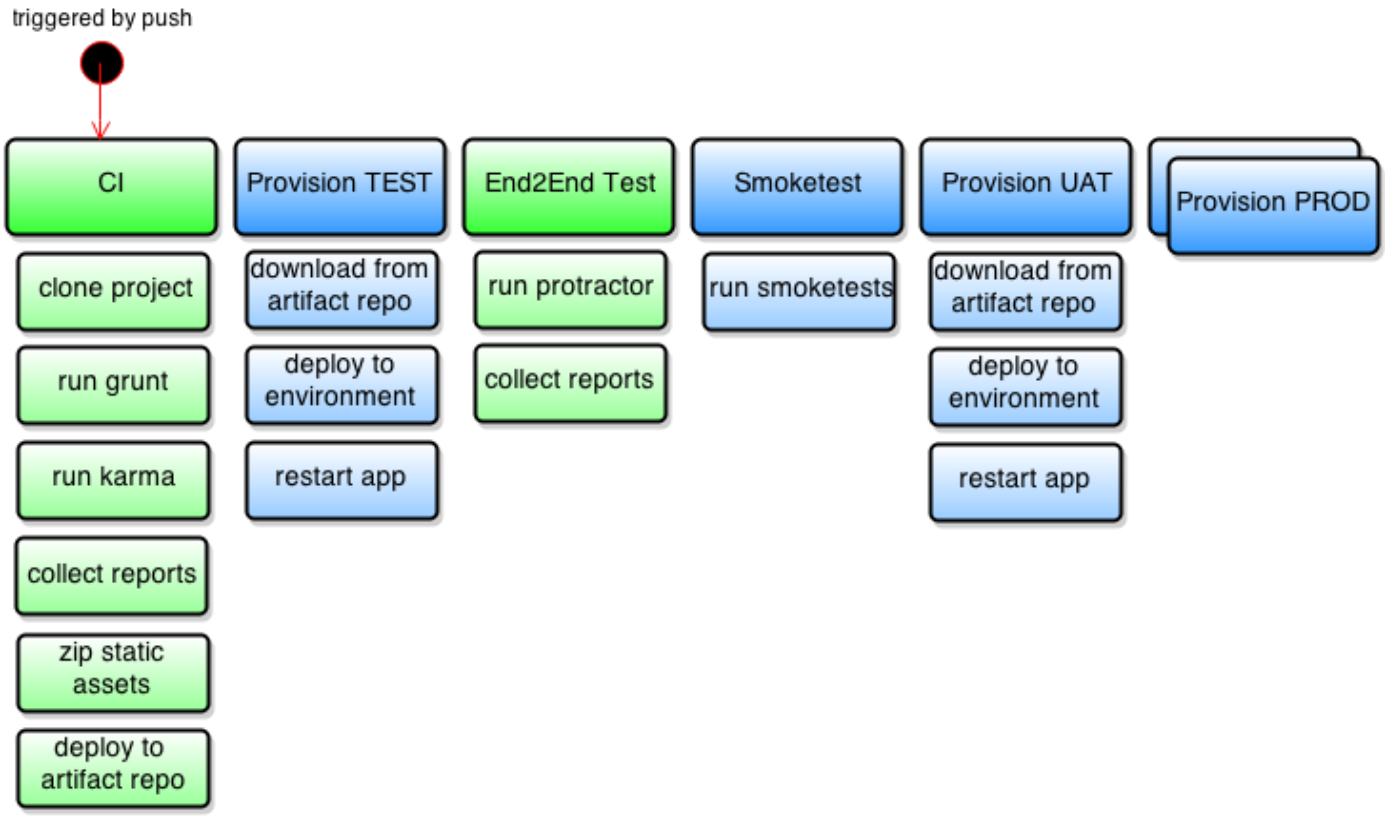
Delivery Pipeline

http://54.75.209.193/jenkins/view/EnterJS-Pipeline/

Info: Server will be shutdown after talk

Delivery Pipeline

Delivery Pipeline Steps



Collect the reports

- Mocha reporter ⇒ unit test
- Jasmine reporter ⇒ end2end tests
- ESLint ⇒ static code analysis

Mocha Report (unit tests)

Test Result : Filter Test UpperCase Test

0 failures (±0)

2 tests (±0)
Took 22 ms.

All Tests

Test name	Duration	Status
should convert first charactor of each whitespace separated string to UpperCase	1 ms	Passed
should convert first charactor of string to UpperCase	21 ms	Passed

Protractor Report (end2end tests)

Jenkins ▸ CL-Jobs ▸ PS - End2End Test ▸ #52 ▸ Test Results ▸ (root) ENABLE AUTO REFRESH

- Back to Project
- Status
- Changes
- Console Output
- View as plain text
- Edit Build Information
- History
- Parameters
- Environment Variables
- Git Build Data
- No Tags
- Test Result
- Previous Build
- Next Build

Test Result : (root)

4 failures (-16)

20 tests (±0)
Took 2 min 54 sec.
add description

All Failed Tests

Test Name	Duration	Age
Delete all customer and add new customer.all customers should have been deleted	10 sec	3
Delete all customer and add new customer.should navigate to manage customers	10 sec	3
List customer on map tests:.navigate to cologne should find customer on map	10 sec	3
List customer on map tests:.should navigate to manage customers	16 sec	3

All Tests

Class	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
Delete all customer and add new customer	38 sec	2	-1	0		1	+1	3	
List customer on map tests:	27 sec	2		0		0		2	
Listing Employees Page	4.5 sec	0	-2	0		2	+2	2	
Manage Employees Page	1 min 26 sec	0	-5	0		5	+5	5	
Project Staffing Start Page	18 sec	0	-8	0		8	+8	8	

ESLint Report (static code analysis)

CheckStyle Result

Warnings Trend

All Warnings	New Warnings	Fixed Warnings
138	138	0

Summary

Total	High Priority	Normal Priority	Low Priority
138	138	0	0

Details

Folders | **Files** | Warnings | Details | New

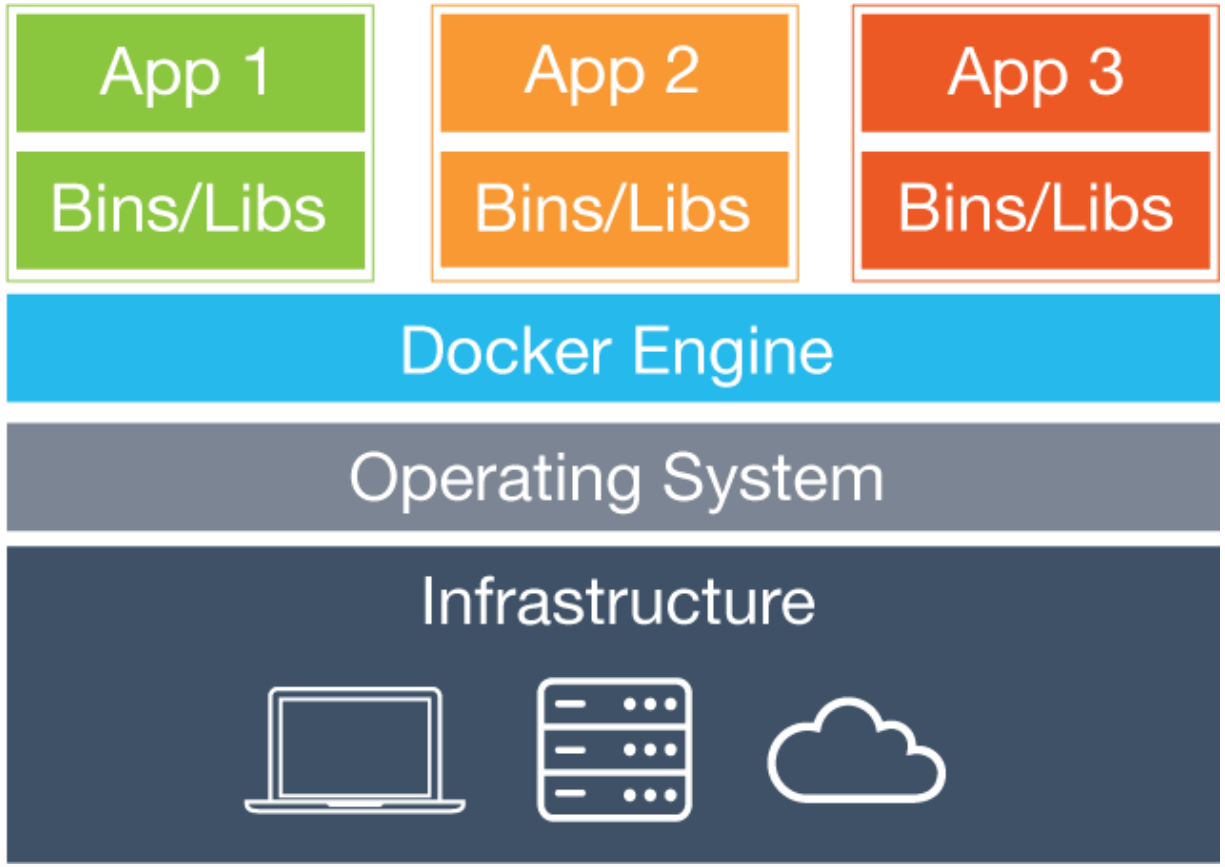
File	Total	Distribution
address_controller.js	2	
app.js	2	
customer_controller.js	30	
dashboard_controller.js	18	
employee-directives.js	1	
employee_controller.js	13	
filter.js	1	
navigation_controller.js	1	
ngAutocomplete.js	30	
project_controller.js	4	
routes.js	11	
skill_controller.js	3	
staffing_controller.js	20	
timeline_controller.js	2	
Total	**138**	

Using Container during Development

- Docker / boot2docker
- docker-compose aka fig
- Docker Hub / Registry

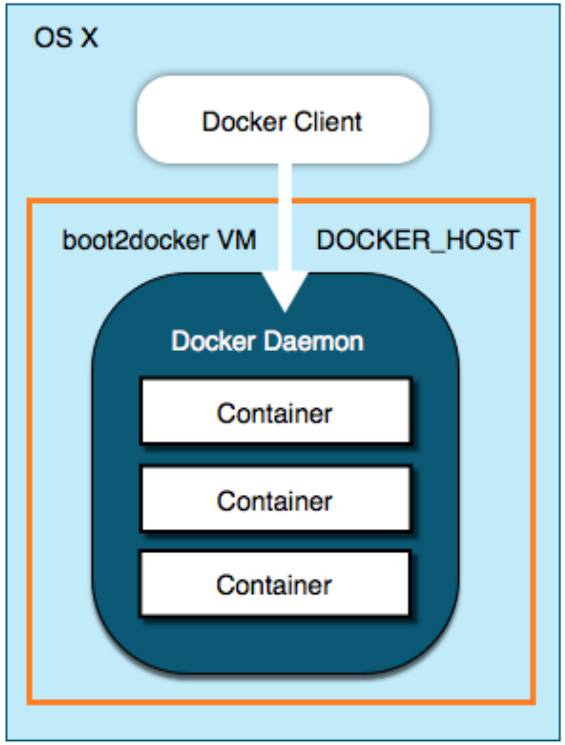
Docker

Container Technology, Lightweight, Portable



boot2docker

Based on Tiny Core Linux (required for MacOS and Windows)



docker-compose

BASH

```
project-staffing git:(master) ✗ docker-compose

Commands:
  build     Build or rebuild services
  help      Get help on a command
  kill      Kill containers
  logs      View output from containers
  port      Print the public port for a port binding
  ps        List containers
  pull      Pulls service images
  rm        Remove stopped containers
  run       Run a one-off command
  scale     Set number of containers for a service
  start     Start services
  stop      Stop services
  restart   Restart services
  up        Create and start containers
```

docker-compose up

BASH

```
➜  project-staffing git:(master) ✗ docker-compose up
Recreating projectstaffing_mongodb_1...
Creating projectstaffing_nodejsserver_1...
Building nodejsserver...
Step 0 : FROM tcnksm/centos-ruby
 ---> 255207061af8
Step 1 : RUN yum install -y npm
 ---> Using cache
 ---> c8ca0ad1bec0
Step 2 : COPY . /opt/project-staffing/
 ---> dc70b159f357
...
Step 5 : CMD node /opt/project-staffing/server.js
 ---> Running in 78d831b9f0f0
 ---> 88b07ba248a0
Successfully built 88b07ba248a0
  ...
```

Docker Hub

- https://registry.hub.docker.com/
- Official Repositories: redis, ubuntu, WordPress, MySQL, mongoDB, nodeJS, ...
- Share your own Containers

ons?

nsultant
ultant