

Welcome to

dreamOle[®]



omega



Cognizant



FORMTITAN

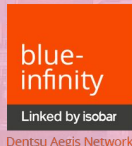
resco.net
technology on the move



MAKEMECLLOUD
consulting



aircall



FINANCIALFORCE



wefox

BLACKBIRD

#dreamOle18

Barcelona 2018



From Zero to CI in 30 minutes

Christian Szandor Knapp
Lead Salesforce Developer, appero
Salesforce MVP
@ch_sz_knapp

Agenda

Why CI?

What are the benefits of Continuous Integration?

Prerequisites

What do I need to get CI running?

CircleCI Basics

Why CCI? Setup, Workflows, jobs, steps, and gotchas

Deep Dive

Complete script for feature to production



DX Application Lifecycle



Why CI?

Why CI?

What are the benefits of Continuous Integration?

Prerequisites

What do I need to get CI running?

CircleCI Basics

Why CCI? Setup, Workflows, jobs, steps, and gotchas

Deep Dive

Complete script for feature to production



Why Continuous Integration?

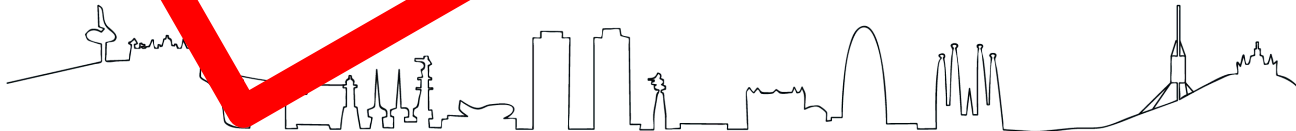
Changesets are evil

- manual work
- error prone
- time consuming

Agile works just better with CI

- shorter feedback cycle
- a new feature is useless if it cannot be deployed
- each feature needs to be built from scratch to detect errors early

**TIME IS
MONEY**



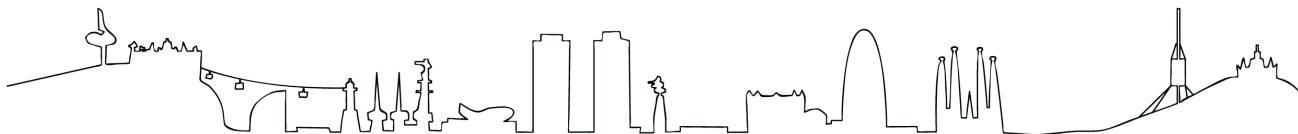
Why CircleCI 2.0?

I felt too dumb for other tools (Travis, Jenkins, Heroku)

- platform only person who has little to no clue about bash scripts
- there were tutorials around for CircleCI 1.0
- Circle 2.0 enabled me to wrap my head around parallelism
- AND get it to work with DX

Free for open source projects

- up to 4 containers (machines)
- ideal to experiment with CI concepts and parallelism



Prerequisites

Why CI?

What are the benefits of Continuous Integration?

Prerequisites

What do I need to get CI running?

CircleCI Basics

Why CCI? Setup, Workflows, jobs, steps, and gotchas

Deep Dive

Complete script for feature to production



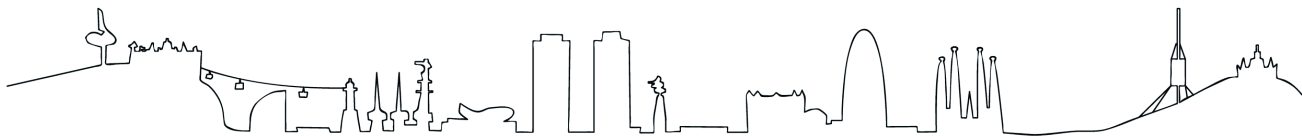
Prerequisites and the role of DX

Version Control System

- you need to have some repo and basic git knowledge
- you need to have a branch strategy
 - today we will use: *feature* and *master*

DX as CLI tool, not as new paradigm how to do stuff

- tomorrow we will build *developer controlled packages*
- but today we still deploy the happy soup to production
- DX source format is not necessary
-



Demo I: Pull Request on Master

<https://github.com/szandor72/dreamole18-ci-in-30-minutes>



Pull Request / Master Build

Workflows » Szandor72 » dreamole18-ci-in-30-minutes » master » 8374e299-b1d6-4ecc-b1b9-670ba4667d1f

✓ SUCCEEDED

master / salesforcedx

👤 Merge pull request #4 from Szandor72/feature/AwesomeFeature

📅 3 days ago

🕒 02:50

🔗 f5a29c0

🔄 Rerun

4 jobs in this workflow

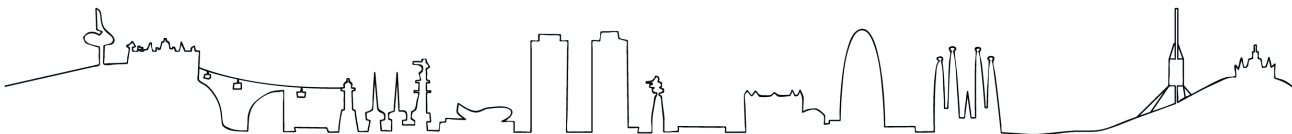
✓ setup-dx-env... 🕒 00:40

✓ master-conve... 🕒 00:34

✓ master-Check... 🕒 00:42

✓ master-Deploy 🕒 00:40

#dreamOle18



dreamOle.

CircleCI Basics

Why CI?

What are the benefits of Continuous Integration?

Prerequisites

What do I need to get CI running?

CircleCI Basics

Why CCI? Setup, Workflows, jobs, steps, and gotchas

Deep Dive

Complete script for feature to production



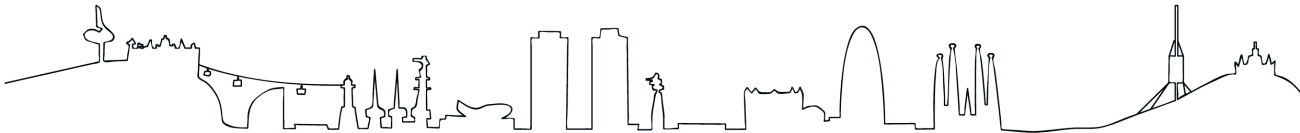
CircleCI - creating account is done in a jiffy

3 easy steps

1. Go to circleci.com Login with your BitBucket or Github Account

2. 'Follow' the repo you want to build from

3. Add necessary settings/environment variables



These are all the settings you need

Add environment variables to the build. You can add sensitive data (e.g. API keys) here, rather than placing them in the repository. The values can be any bash expression and can reference other variables, such as setting `M2_MAVEN` to `${HOME}/.m2` .

Name	Value	Remove
DX_CONSUMER_KEY	xxxxxewO	×
DX_USER	xxxx.com	×
PRODUCTION	xxxx.com	×
SFDX_AUTOUPDATE_DISABLE	xxxvue	×
SSL_SERVER_KEY_HEX	xxxx2d0a	×




```
version: 2
```

```
jobs:  
  build:  
    machine: true  
    steps:  
      - checkout  
      - run: <command>
```

```
test:  
  machine: true  
  steps:  
    - checkout  
    - run: <command>
```

```
workflows:  
  version: 2  
  build_and_test:  
    jobs:  
      - build  
      - test
```

.circleci/config.yml - one to rule them all

Workflow(s)

Jobs

Steps

build_and_test

test, build

run command



```
version: 2
```

```
jobs:
```

```
  build:
```

```
    machine: true
```

```
    steps:
```

- checkout
- run: <command>

```
  test:
```

```
    machine: true
```

```
    steps:
```

- checkout
- run: <command>

```
workflows:
```

```
  version: 2
```

```
  build_and_test:
```

```
    jobs:
```

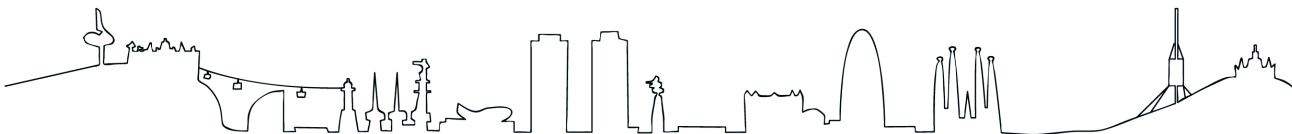
- build
- test

Unleashing the power of CI

These two jobs can run at the same time

CI is not only about automation

CI is also about parallelism



```
version: 2
```

```
jobs:
```

```
  build:
```

```
    machine: true
```

```
    steps:
```

- checkout
- run: <command>

```
  test:
```

```
    machine: true
```

```
    steps:
```

- checkout
- run: <command>

```
workflows:
```

```
  version: 2
```

```
  build_and_test:
```

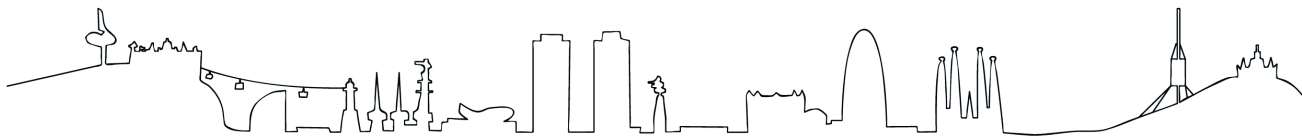
```
    jobs:
```

- build
- test

CCI Parallelism Gotchas

Parallelism presupposes a complete stack per job/container

Every machine starts empty. To persist DX CLI, authentication, ... a little work needs to be done



Demo II: dynamically distributed tests

<https://github.com/szandor72/dreamole18-ci-in-30-minutes>



Feature Build

Workflows » Szandor72 » dreamole18-ci-in-30-minutes » feature/AwesomeFeature » 448003d8-5580-44d7-b198-4d9405d1c833

RUNNING

Cancel

feature/AwesomeFeature / salesforcedx

Awesome new features

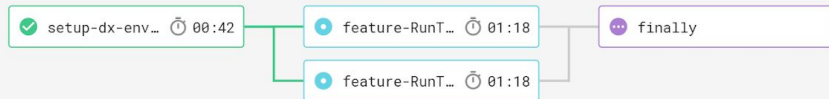
2 min ago

02:07

#5

304294f

4 jobs in this workflow



Deep Dive

Why CI?

What are the benefits of Continuous Integration?

Prerequisites

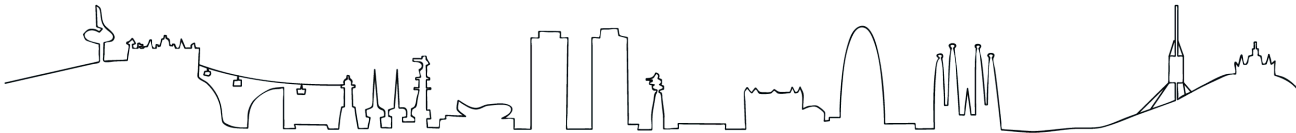
What do I need to get CI running?

CircleCI Basics

Why CCI? Setup, Workflows, jobs, steps, and gotchas

Deep Dive

Complete script for feature to production



workflows:

version: 2

salesforcedx:

jobs:

- setup-dx-environment: [...]
- feature-RunTestDistributed_1: [...]
- feature-RunTestDistributed_2: [...]
- master-convert-and-modify-source: [...]
- master-Deploy-checkonly: [...]
- master-Deploy:

requires:

- master-Deploy-checkonly

filters:

branches:

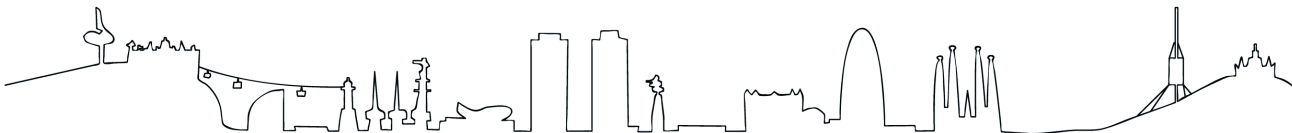
only:

- master

Main Workflow I

A job will run only when all requirements are fulfilled, i.e. indicated jobs completed successfully

if there are no requirements that preclude it and containers are available, jobs will run in parallel



workflows:

version: 2

salesforcedx:

jobs:

- setup-dx-environment: [...]
- feature-RunTestDistributed_1: [...]
- feature-RunTestDistributed_2: [...]
- master-convert-and-modify-source: [...]
- master-Deploy-checkonly: [...]
- master-Deploy:

requires:

- master-Deploy-checkonly

filters:

branches:

only:

- master

Main Workflow II

A job will run only on the indicated branch by using filters

hardcoded branch names

regular expressions



Exemplary step

finally:

machine: true

steps:

- attach_workspace:

 - at: ~/

- run:

 - name: DX Install

 - command: ~/tools/sfdx/install

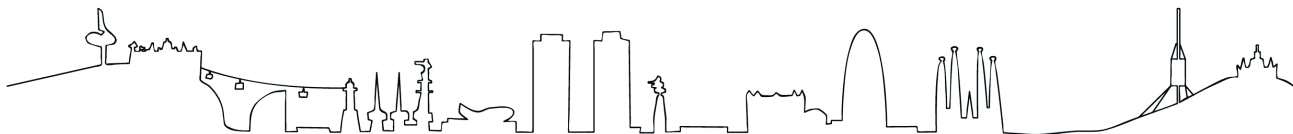
- run:

 - name: delete all scratch orgs

 - command: |

 - sfdx force:org:delete -u feature1 --no-prompt || true

 - sfdx force:org:delete -u feature2 --no-prompt || true



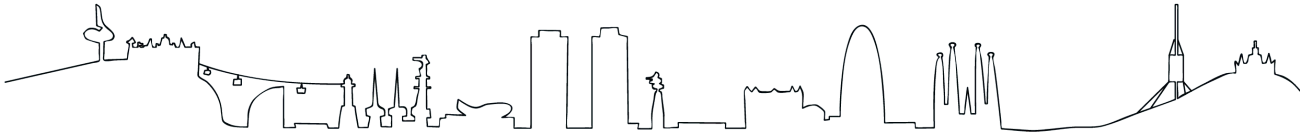
Run tests in parallel I

feature-RunTestsDistributed_1:

machine: true

steps:

- checkout
- attach_workspace: [...]
- run: DX Install [...]
- run: Create 1st feature Org and push source [...]
- run: generate password so we can login if something fails [...]
- run: Query Tests And Run 1st half of Tests [...]
- run: Delete Scratch
- store_test_results



Run tests in parallel II

- run:

name: Query Tests And Run 1st half of Tests

command: |

```
sfdx force:data:soql:query
```

```
-q "select Name from apexclass where name like '%Test'"
```

```
-u feature1
```

```
--json >> ~/tools/ApexTests.json
```

```
//setting env var
```

```
testsToRun=$(cat ~/tools/ApexTests.json |
```

```
jq -r '.result.records[0:length/2] |
```

```
.[].Name+"", "" --join-output)
```

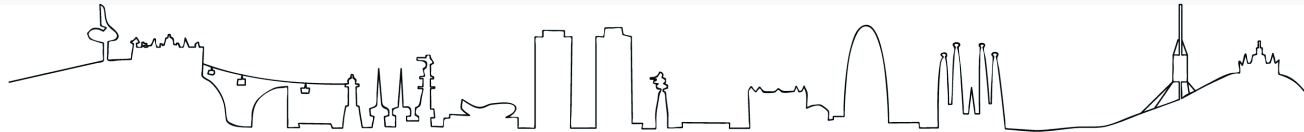
```
sfdx force:apex:test:run -n $testsToRun
```

```
-u feature1
```

```
-w 10
```

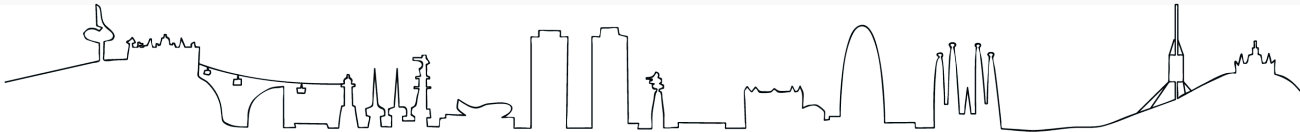
```
-c
```

```
-r human -d ~/test-results
```



Store test results

- run:
 - name: Delete Scratch
 - when: always
 - command: |
 - sfdx force:org:delete -u feature2 --noprompt
- store_test_results:
 - path: ~/test-results
- store_artifacts:
 - path: ~/test-results
 - destination: test-results
- store_artifacts:
 - path: ~/.sfdx/sfdx.log
 - destination: sfdx-logs



Q & A



Cognizant



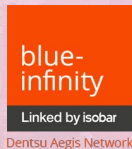
FORMTITAN



MAKEMECLLOUD
consulting



aircall



FINANCIALFORCE



wefox

BLACKBIRD

#dreamOle18



Barcelona 2018

Slide Title

Title

Enter slide information
here

Title

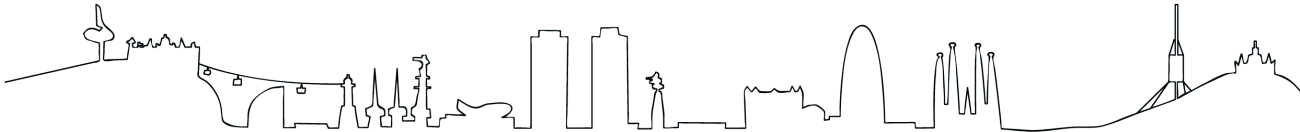
Enter slide information
here

Title

Enter slide information
here

Title

Enter slide information
here

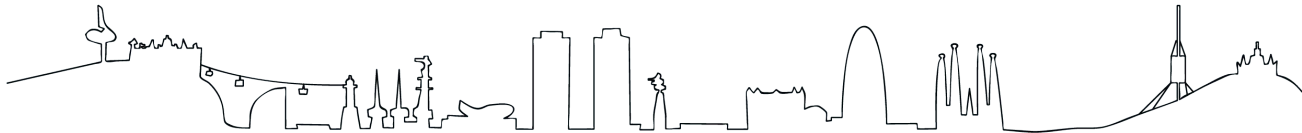


Basic slide

Subtitle

This placeholder can hold text and below bullet points if required. You can also add charts, smartart or media.

- Item 1
- Item 2
- Item 3



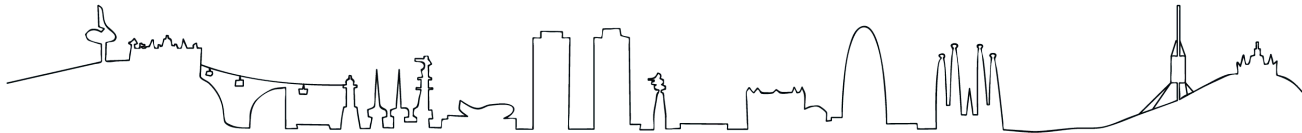
3 column slide

Subtitle

These three placeholders can hold text, charts, smartart, or media.

These three placeholders can hold text, table, charts, smartart, or media.

These three placeholders can hold text, table, charts, smartart, or media.



Slide Title



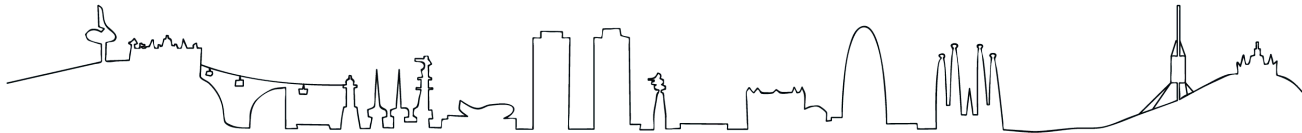
Title

This placeholder can hold text.



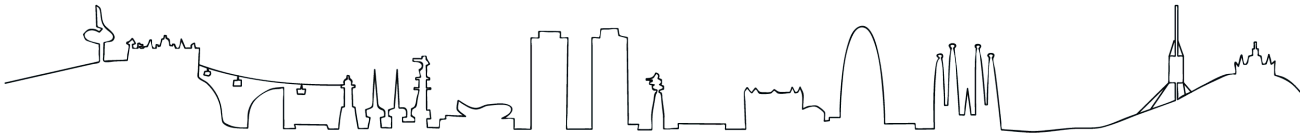
Title

This placeholder can hold text.



Slide Title

This placeholder can hold text.



Slide Title

