

@codecentric

# Jenkins Workshop @Commerzbank\_

Alexander Sachs



# About me

Alexander Sachs

IT-Consultant

codecentric AG  
Curiestr. 2  
70563 Stuttgart, Germany

📞 +49 (0) 151 1086 7089

✉️ alexander.sachs@codecentric.de

🌐 www.codecentric.de

💻 blog.codecentric.de



@codecentric



# About you

---

Who are you?

What do you expect?

What do you already know about Jenkins?



# Introduction

@codecentric



# Introduction

---

## What is Continuous Integration (CI)?

Continuous Integration is a development practice that requires developers to integrate code into a shared repository at regular intervals.

This concept was meant to remove the problem of finding later occurrence of issues in the build lifecycle.

Continuous integration requires the developers to have frequent builds.

The common practice is that whenever a code commit occurs, a build should be triggered.

But what does that mean?



# Introduction

---

What it doesn't mean:

Deploying Software manually

Deploying to a Production-like Environment only after Deployment is complete

Manual Config Management of Production Environments



# Introduction

---

But how to avoid those antipatterns?

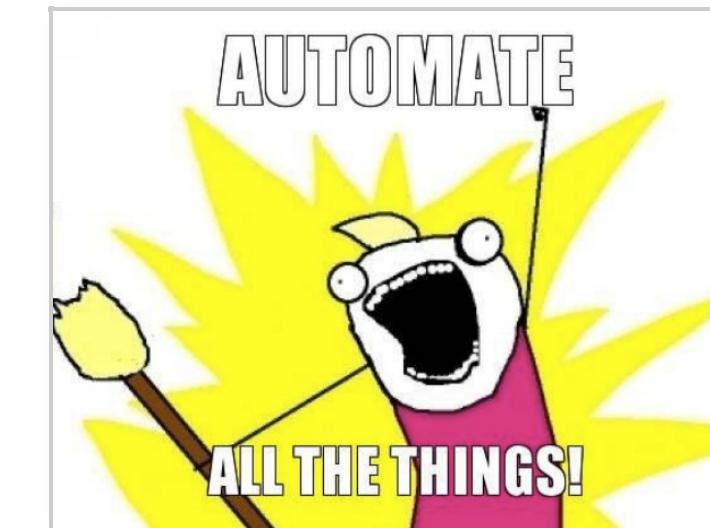
Every change should trigger the feedback process

The feedback must be received as soon as possible

The delivery team must receive feedback and then act on it

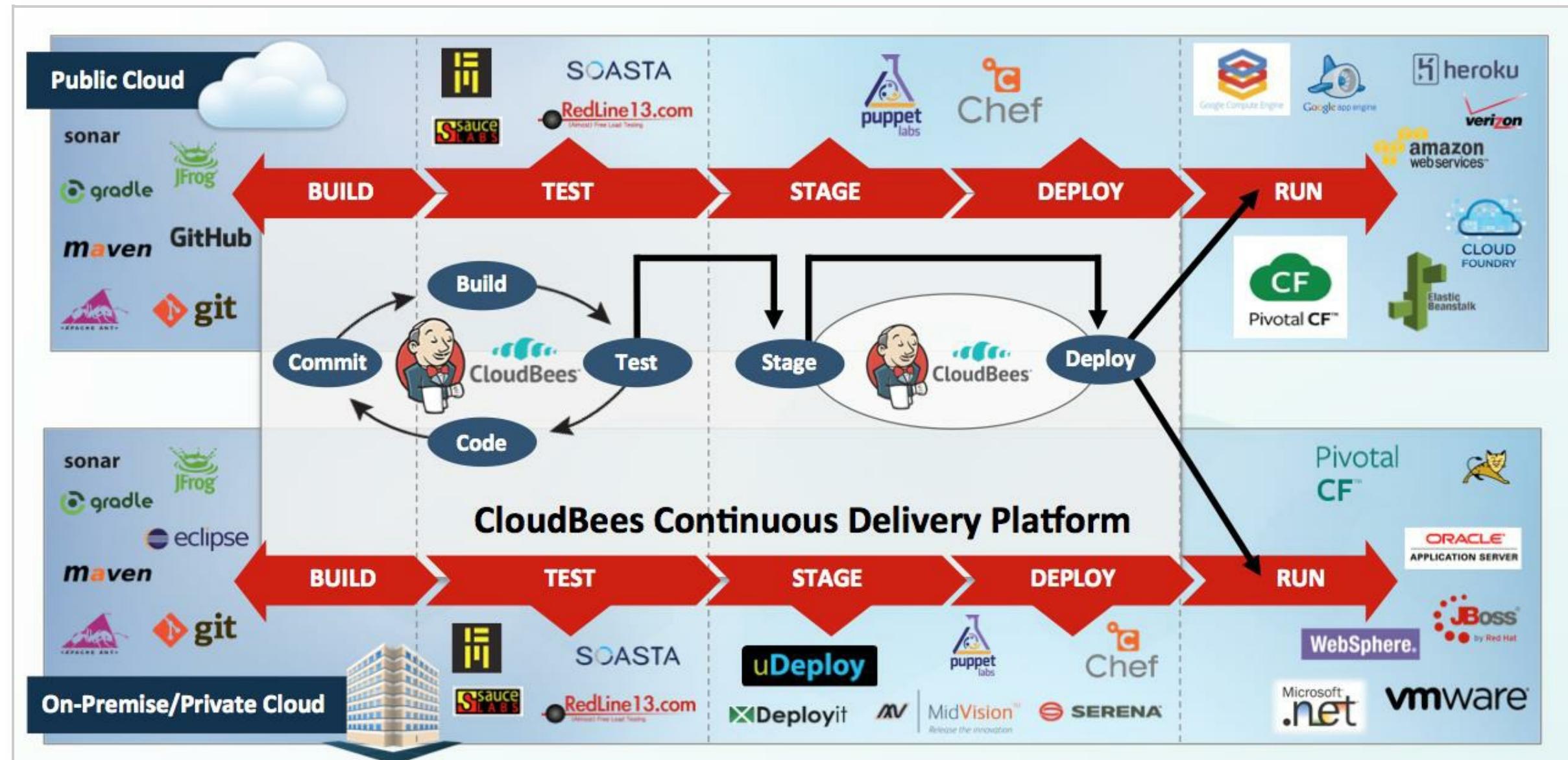
Create a repeatable, reliable process for releasing software

Make everybody responsible for the delivery process



@codecentric

# Introduction



@codecentric



# Jenkins first steps

@codecentric



# Jenkins Installation

---

Docker (fastest way to get started)

Deploy a war file to any webserver

Use AWS to install Jenkins



# Jenkins - Getting started

---

Claim your AWS instance with preinstalled Jenkins

Enter initialAdminPassword

Choose "install suggested plugins"

Create account and note your username and password



# Jenkins exercises - Getting started

---

Go to <https://github.com/TheFutureStartsNow/jenkinsWorkshop>

This repo has an exercise folder where you can find the exercises



# Jenkins exercise 1 - Freestyle Projects

General      Source Code Management      Build Triggers      Build Environment      Build      Post-build Actions

Project name: build

Description:

[Plain text] [Preview](#)

Discard old builds

GitHub project

Project url: <https://github.com/TheFutureStartsNow/jenkinsWorkshop/ex1/>

[Advanced...](#)

This project is parameterized

Throttle builds

Disable this project

Execute concurrent builds if necessary

[Advanced...](#)

**Source Code Management**

None

Git

Repositories

Repository URL: <https://github.com/TheFutureStartsNow/jenkinsWorkshop.git>

Credentials: - none - [Add](#)

[Save](#) [Apply](#)

@codecentric

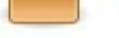


# Jenkins exercise 1 - Artifacts

Jenkins > build > #17

[Back to Project](#)  
[Status](#)  
[Changes](#)  
[Console Output](#)  
[Edit Build Information](#)  
[Delete Build](#)  
[Open Blue Ocean](#)  
[Git Build Data](#)  
[No Tags](#)  
[Previous Build](#)

**Build #17 (Jun 25, 2017 8:20:16 AM)**

 [Build Artifacts](#)  
 [build.jar](#) 1.63 KB [view](#)

 No changes.

 Started by user [alex](#)

 This run spent:

- 4 ms waiting in the queue;
- 8.4 sec building on an executor;
- 8.4 sec total from scheduled to completion.

 Revision: b601d7c976277851d0f662dcd28e29c778525608  
• refs/remotes/origin/ex1

@codecentric



# Jenkins exercise 1

---

Try yourself with the exercise



# Jenkins Plugins

@codecentric



# Jenkins exercise 2 - Plugins

1000+ plugins available

Available on <https://plugins.jenkins.io>



@codecentric



# Jenkins exercise 2 - Plugins - HTML Publisher

---

Project test

-  Back to Dashboard
-  Status
-  Changes
-  Workspace
-  Build Now
-  Delete Project
-  Configure
-  Favorite
-  HTML Report

 [HTML Report](#)

 [Workspace](#)

 [Recent Changes](#)

@codecentric



# Jenkins exercise 2 - Plugins - HTML Publisher

### Test Summary

1 tests	0 failures	0 ignored	0.060s duration
---------	------------	-----------	-----------------

**100%**  
successful

[Packages](#) [Classes](#)

#### Packages

Package	Tests	Failures	Ignored	Duration	Success rate
<a href="#">default-package</a>	1	0	0	0.060s	100%

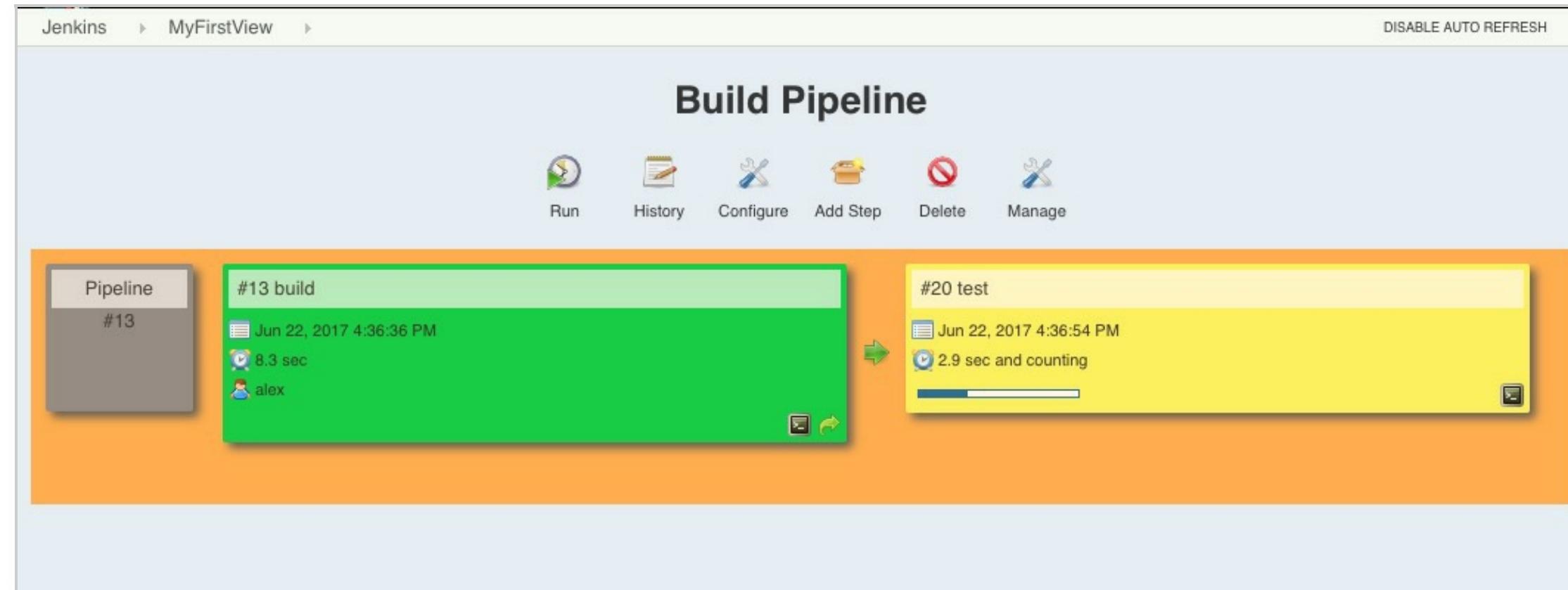
#### Classes

Class	Tests	Failures	Ignored	Duration	Success rate
<a href="#">AppSpec</a>	1	0	0	0.060s	100%

@codecentric



# Jenkins exercise 2 - Plugins - Build Pipeline



@codecentric



# Jenkins exercise 2 - Plugins - Blue Ocean

The screenshot shows the Jenkins Blue Ocean Pipelines interface. At the top, there is a blue header bar with the text "Pipelines" and a magnifying glass icon on the left, and a "New Pipeline" button on the right. Below the header is a table with the following data:

Name	Health	Branches	PR
build		-	
test		-	

@codecentric



# Jenkins exercise 2

---

Try yourself with the exercise



# The Jenkinsfile

@codecentric



# Jenkinsfile

---

What is a Jenkinsfile?

A file written in Groovy containing a pipeline definition

```
pipeline {  
    agent any  
    stages {  
        stage('Build') {  
            steps {  
                echo 'do some build steps'  
            }  
        }  
        stage('Test'){  
            steps {  
                sh './gradlew test'  
            }  
        }  
    }  
}
```



# Jenkinsfile

---

- Why you should use a Jenkinsfile
  - Code review/iteration on the Pipeline
  - Audit trail for the pipeline
  - Single source of truth for the pipeline



# Jenkinsfile - exercise

---

Try yourself with the exercise



# Best practice

@codecentric



# Jenkins - Best practice

Develop your pipeline as code  
All work within a stage



@codecentric



# Jenkins - Best practice

---

The most reliable builds are built fully from Source Code Control  
Archive unused jobs before removing them  
Setup a different job/project for each maintenance or development branch you create  
Avoid scheduling all jobs to start at the same time  
Set up email notifications mapping to ALL developers in the project, so that everyone on the team has his pulse on the project's current status



# Summary

@codecentric



# Summary

---

Create different types or projects

- freestyle project

- pipeline project

- multibranch project

Install and use plugins

- html publisher

- build pipeline

- blue ocean



# Summary

---

Jenkinsfile  
pipeline definition  
stages  
tasks

@codecentric



# Questions?

Alexander Sachs

IT-Consultant

codecentric AG  
Curiestr. 2  
70563 Stuttgart, Germany

📞 +49 (0) 151 1086 7089

✉️ alexander.sachs@codecentric.de

🌐 www.codecentric.de

💻 blog.codecentric.de



@codecentric

