DevOps

MICROSERVICES

Content

//WEEK 1 - Tobias

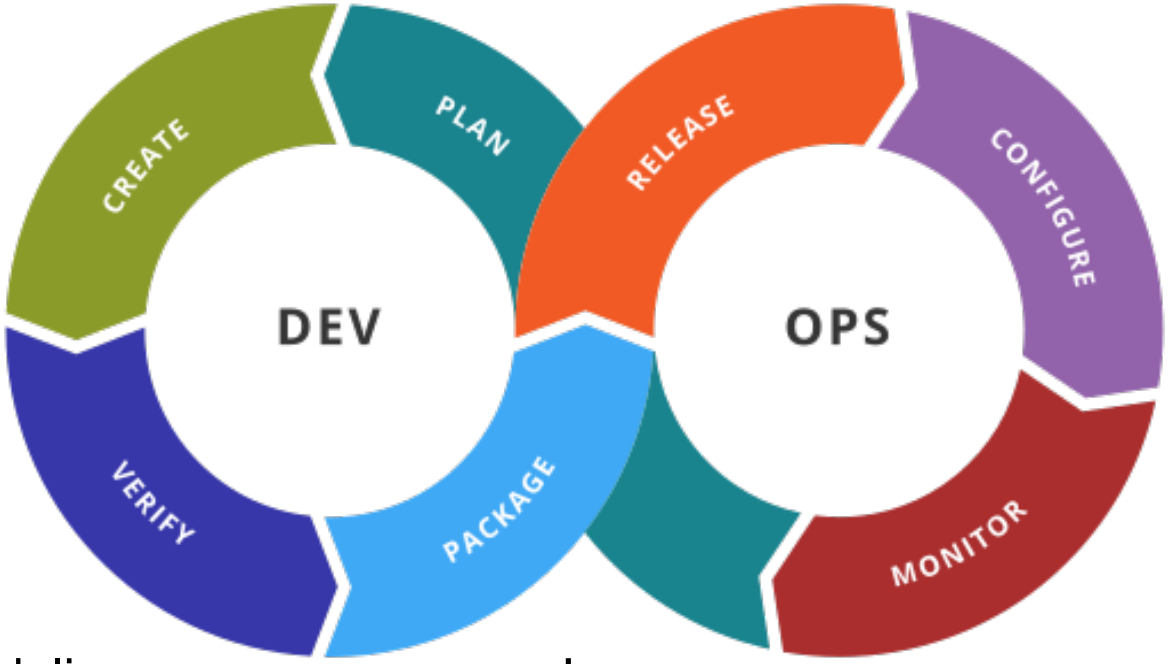
- DevOps Introduction
- Pipeline (continuous integration / continuous delivery / continuous deployment)
- Real World Examples

//WEEK 2 - Peter

- Infrastructure as Code (Puppet?, Chef?)
- Cloud Foundry
- tbd.

DevOps - Introduction

- **DevOps** is a software engineering practice that aims at unifying software development (Dev) and software operation (Ops)
  - People, processes and tools
  - working together
  - to enable continuous delivery of value
  - to the end users
  - fast(er)



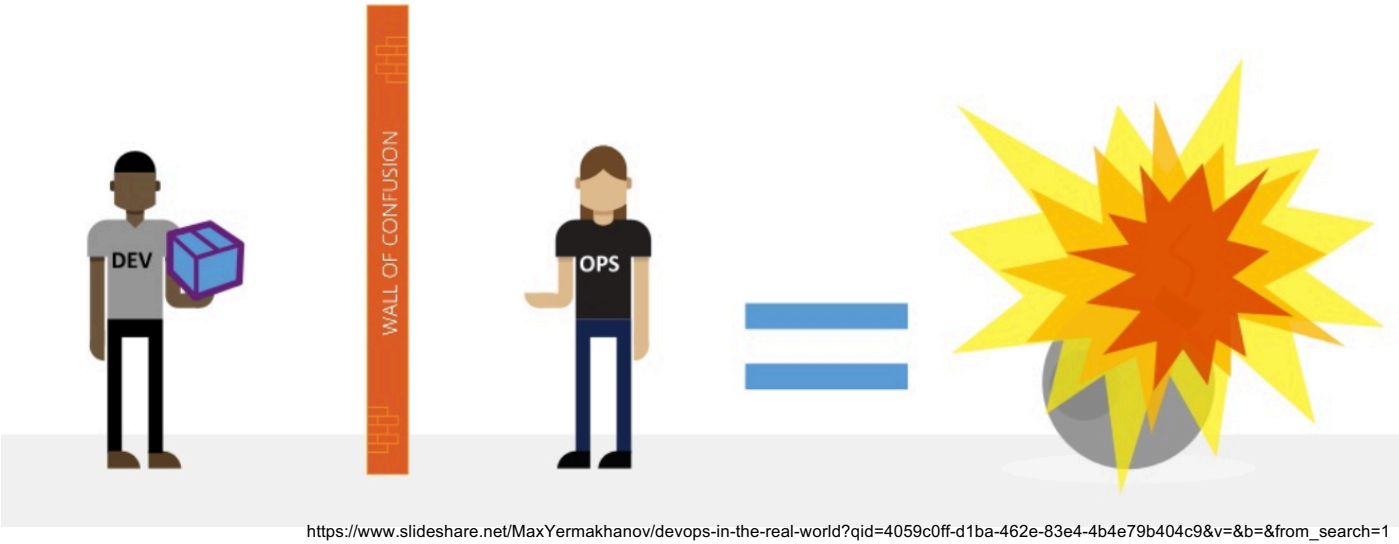
➔ Cultural change required

➔ We just focus on the software delivery process process!

https://upload.wikimedia.org/wikipedia/commons/thumb/0/05/Devops-toolchain.svg/512px-Devops-toolchain.svg.png

DevOps - Challenges

- development vs. operation
- agile vs. stability
- a lot iterations
- a lot of releases
- monitoring of all services
- code quality



https://www.slideshare.net/MaxYermakhanov/devops-in-the-real-world?qid=4059c0ff-d1ba-462e-83e4-4b4e79b404c9&v=&b=&from_search=1

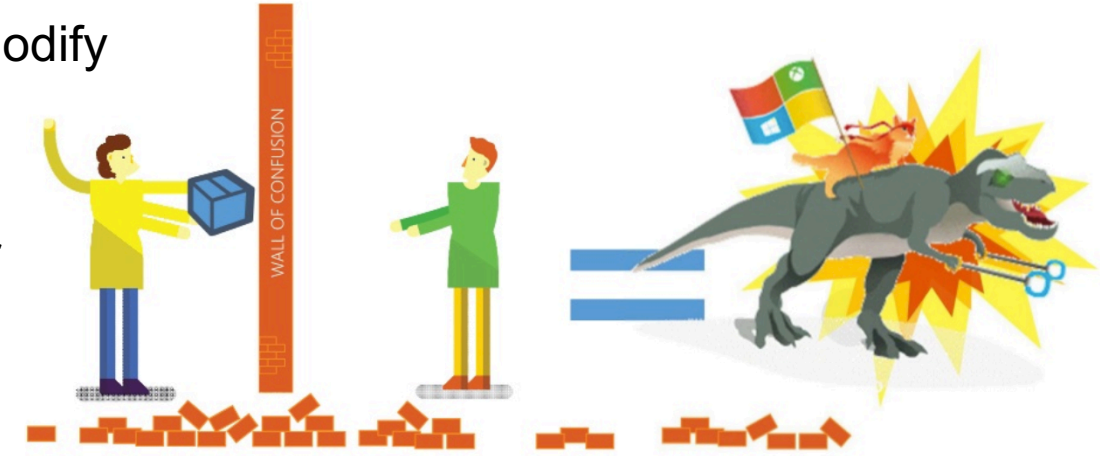
Goals

- The goals of DevOps span the entire delivery pipeline.
  - Improved deployment frequency
  - Faster time to market
  - Lower failure rate of new releases
  - Shortened lead time between fixes
  - Faster mean time to recovery
  - Easier onboarding for new developers

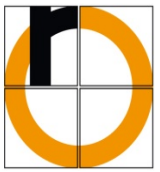
Happy DevOps

Dev ➔ Create Change, Add or modify features

Ops ➔ Create stability, Create or enhance services



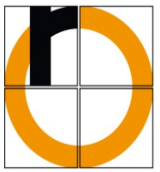
https://www.slideshare.net/MaxYermakhanov/devops-in-the-real-world?qid=4059c0ff-d1ba-462e-83e4-4b4e79b404c9&v=&b=&from_search=1



DevOps

- general project management tasks
- backlog
- documentation
- scrum planning
- Retrospectives
- use mvp's – don't develop the hole software without test! (from lean startup)

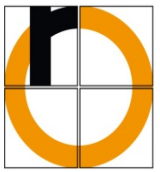
Tools: JIRA, Trello, Confluence, Wiki, github readme, Planning poker, Whiteboard /w Post-it, User Story Mapping, Business Model Canvas, …



DevOps

- coding with git (hopefully not with svn anymore)
- code reviews

Tools: github, codacy, bitbucket, …

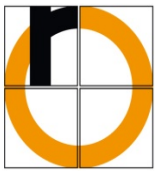


DevOps

- Continuous Integration
- different build tools for your project
- Automatic unit tests

Tools: sbt, maven, gradle, docker build, …

➔ CI: Jenkins, Travis CI, Circle CI, AWS CodeBuild



DevOps

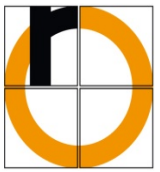
- Continuous Integration
- Code coverage report
- automatic test
- acceptance testing
- Integration testing

➔ CI: Jenkins, Travis CI, Circle CI, AWS CodeBuild

DevOps

- Packaging like building a jar or a docker container
- Pre-Deployment-Staging
- Release automation

➔ CD: Jenkins, Travis CI, Circle CI, AWS CodeDeploy, heroku, CloudFormation, Docker

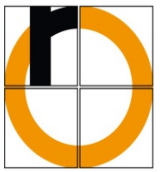


DevOps

- configuration
- production staging
- Infrastructure as a Code (IaaC)

Tools: Chef, Ansible, Cloud Formation, Elastic Beanstalk, Reactive Platform, ...

➔ CD: Jenkins, Travis CI, Circle CI, AWS CodeDeploy, heroku



DevOps

- logging
- exception handling
- performance monitoring
- support / service desk
- feature toggles
- Metrics / measure success / …

Tools: Jira Service Desk, CloudWatch, ELK Stack, Reactive Platform, ECS, EC2, Kubernetes, …

DevOps Pipeline

Agile development

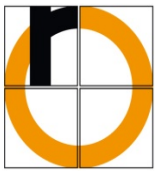
Continuous Integration

Continuous Delivery

Continuous Deployment

Continuous Configuration Automation

MORE INFO: https://www.scrum.de/unterschiede-zwischen-continuous-integration-continuous-delivery-und-continuous-deployment/



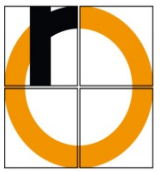
DevOps: Automate Almost Everything for Microservice

- The build
- database change (flyway)
- Deployment to test/staging/production environments
- Tests
- Remediation plans
- Monitoring
- Infrastructure as code
- Service discovery, DNS, Load Balancing, Auto Scaling, …

➔ Your continuous deployment pipeline should be a model of your process for getting software from version control into the hands of your users.

➔ A visualization (in real-time) of the status of software codebase after every change, for all stages from chek-in to release

@SPOTIFY: https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/ !!!



Travis CI Sample

```
language: scala

scala:
  - 2.11.8

jdk:
  - oraclejdk8

branches:
  only:
  - master

notifications:
  email: false

script:
  - sbt clean coverage test coverageReport

after_success:
  - bash <(curl -s https://codecov.io/bash)
```

You can try this travis ci sample.
Just fork innfactory/bootstrap-akka-graphql and activate travis ci (its free for open source!)