1. Continuous Integration/ Deployment with Jenkins

Infrastructure Automation
HOGENT applied computer science
Bert Van Vreckem & Thomas Parmentier
2024-2025

Intro

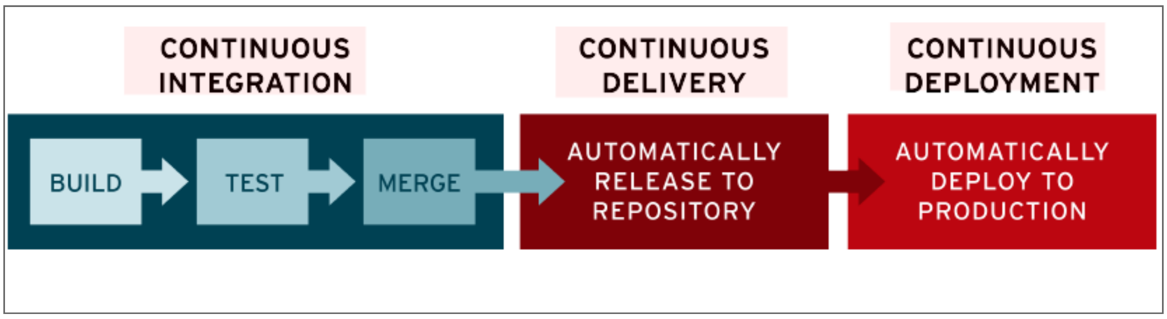
Traditional release

- Weeks/months apart
- Lots of changes!
- ⇒ Lots of bugs…
  - Wait for vN.1?
- Releases are painful, high-risk events

How e.g. Facebook does it

- “If it hurts, do it more often”
- Dozens of releases per day!
- Small, incremental changes go into production
- ⇒ Reduced risk
  Read more: **Rapid release at massive scale**

Continuous Integration/Delivery



CI vs CD. (**Redhat, 2020**)

Typical tasks (1)

- Linting: code style checks
- Static analysis (e.g. shellcheck)
- Compilation
- Unit tests
- Code coverage analysis
- Packaging

Typical tasks (2)

- Release to package repository
- Deploy in acceptance environment
- Integration/acceptance/load-tests…
- Deploy to production

CI/CD tooling

Overview

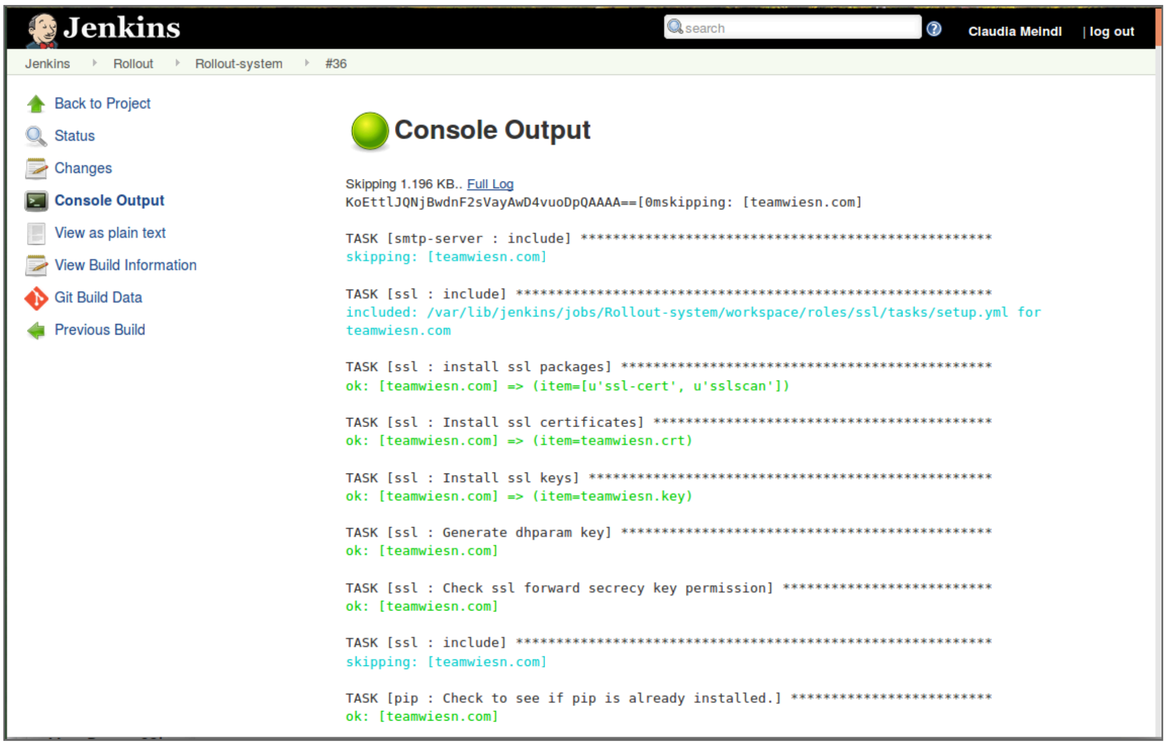
- Open source
  - **Concourse**
  - **Jenkins**
  - **Spinnaker**
- Commercial on-prem
  - **Atlassian Bamboo**
  - **Jetbrains TeamCity**

- Commercial, hosted
  - **AWS CodePipeline**
  - **Azure DevOps**
  - **CircleCI**
  - **Codeship**
  - **Github Actions**
  - **GitLab CI**
  - **Travis CI**

Jenkins

https://www.jenkins.io/

- Open source
- Mature
- Most flexibility
  - on-prem/private/public cloud
  - rich feature set



Skipping 1.196 KB.. Full Log

```
KoEttlJQNjBwdnF2sVayAwD4vuoDpQAAAA==[0mskipping: [teamwiesn.com]

TASK [smtp-server : include] ****************************************************
skipping: [teamwiesn.com]

TASK [ssl : include] ************************************************************
included: /var/lib/jenkins/jobs/Rollout-system/workspace/roles/ssl/tasks/setup.yml for
teamwiesn.com

TASK [ssl : install ssl packages] ***********************************************
ok: [teamwiesn.com] => (item=[u'ssl-cert', u'sslscan'])

TASK [ssl : Install ssl certificates] *******************************************
ok: [teamwiesn.com] => (item=teamwiesn.crt)

TASK [ssl : Install ssl keys] ***************************************************
ok: [teamwiesn.com] => (item=teamwiesn.key)

TASK [ssl : Generate dhparam key] ***********************************************
ok: [teamwiesn.com]

TASK [ssl : Check ssl forward secrecy key permission] ***************************
ok: [teamwiesn.com]

TASK [ssl : include] ************************************************************
skipping: [teamwiesn.com]

TASK [pip : Check to see if pip is already installed.] **************************
ok: [teamwiesn.com]
```

Jenkins Example output (**Wikipedia**)

A Github Actions case

This slide deck was built on Github Actions & deployed to Github Pages!
https://github.com/HoGentTIN/infra-slides

Working with Github Actions

- Go to Actions, New workflow
- Or create `.github/workflows/workflow-name.yml`
- **RTFM**

Example workflow

https://github.com/HoGentTIN/infra-slides/blob/main/.github/workflows/compile.yml

```yaml
---
name: compile
on:
  push:
    branches:
      - main
jobs:
  convert_via_pandoc:
    runs-on: ubuntu-18.04
    steps:
      - name: Configure Git for Github
        run: |
          git config --global user.name
         "${GITHUB_ACTOR}"
          git config --global user.email
         "${GITHUB_ACTOR}@users.noreply.github.com"
      - uses: actions/checkout@v2
      - uses: r-lib/actions/setup-pandoc@v1
        with:
          pandoc-version: '2.9'
      - name: Publish Site
        env:
          REPOSITORY: "https://${{
         secrets.GITHUB_PAT }}@github.com/${{
         github.repository }}.git"
        run: ./publish.sh
```

Another case: testing Ansible roles

https://github.com/bertvv/ansible-role-bind/blob/master/.github/workflows/ci.yml

- Ansible role `bertvv.bind`
- Installs ISC BIND (a DNS server) on several Linux distros
- ~40 contributors, dozens of PRs
- Contributed code may break the role!

Testing Ansible roles in Gitlab CI

- On each push/PR:
  - Spin up Docker container for each supported distro
  - Apply role, use (most) functionality
  - Run acceptance tests (= DNS queries)

Get started with the lab assignment!

Jenkins lab assignment

```
$ cd dockerlab
$ vagrant up
$ vagrant ssh
```

Follow the steps in the assignment **https://github.com/HoGentTIN/infra-labs/blob/main/assignment/2-cicd.md**
Jenkins UI resides at **http://192.168.56.20:8080/**

Setup

- Jenkins runs in a Docker container
- Default installation, minimal configuration required
- Launch demo application in another Docker container
- Make change, rebuild & deploy!

Reflection

Lab setup vs reality

- Complete build server
  - Physical system or “traditional” VM
  - Worker nodes
- Central repo + build tools

Change in discipline needed!

- code coverage ⟶ 100%
- **feature flags**
- **canary deployment**
- **blue/green deployment**
- **trunk-based development**

Canary deployments

Canary deployments (Sato, 2014)

Canary deployments (**Sato, 2014**)

Blue-Green Deployment

Acceptance/Prod swap places (Fowler, 2010)

Acceptance/Prod swap places (**Fowler, 2010**)

Trunk-based development

Branches considered harmful! (Fowler, 2020)

Branches considered harmful! (**Fowler, 2020**)