
NOVATEC
Concourse in Action

Agenda

- Why concourse?
- Working with concourse
  - Concourse Concepts
  - How to work with concourse
- Concourse in Production
  - Learnings and problems
- Whats up in the Future
  - Concourse v7 – Pull Request / feature branches Workflow
  - Roadmap to v10

Why concourse?

History and design principle

Why was concourse made?

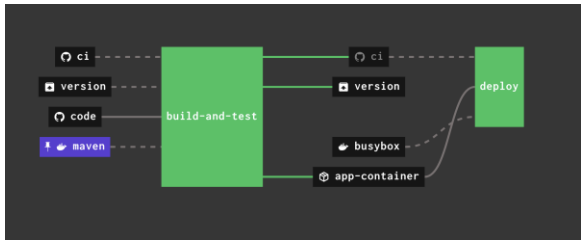
- Building Cloud Foundry with Jenkins (2014)
  - Managing Plugins and dependencies
  - Config hidden behind UI flags
  - Unclear who change what and why
  - State management on workers

Writing software to manage CI instead of developing the product

"Concourse's goal is to solve automation, once and for all, without becoming part of the problem."

- Concourse Design Principles

```
resources:
- name: code
- name: ci
- name: version
- name: maven
- name: busybox
- name: app-container
jobs:
- name: build-and-test
- name: deploy
```



Key Features

- Configuration as Code
  - Pipelines and tasks as yaml
- Visualize to verify
  - Web UI renders pipeline
  - If it looks wrong, it probably is
- CI under source control
  - See who changed what and when by already known tools
- Reproducible, debuggable builds
  - Everything is a container, minimize side effects

NOVATEC

Working with concourse

Concourse concepts and live demo

Concepts

- Pipelines
  Highest abstraction layer you manage
  - Resources
    External Resources, which flow through the pipeline jobs
    Atomically versioned artifact
    Resources configured in different pipelines behave the same
    Examples: Git Repositories, Sonar Server, Kubernetes Cluster
  - Jobs
    Actions to do in one build
    Each job has a plan, which consists of steps and tasks
    Examples: build my application, deploy to Kubernetes
  - Steps and Tasks
    One function to fulfill the goal of the job
    Examples: get the git repo, run a maven build, push the created artifact to artifactory

Working with concourse

Live Demo – Build and test an application

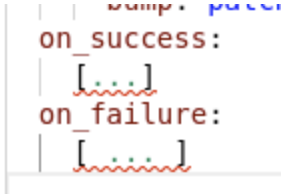
Concourse in Production

Learnings and problems

Learnings

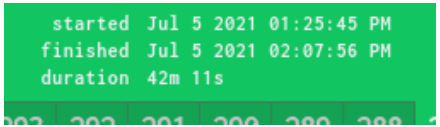
Lots of duplicated YAML

- Using YAML anchors and across to reduce common patterns, like on_success and on_failure hooks



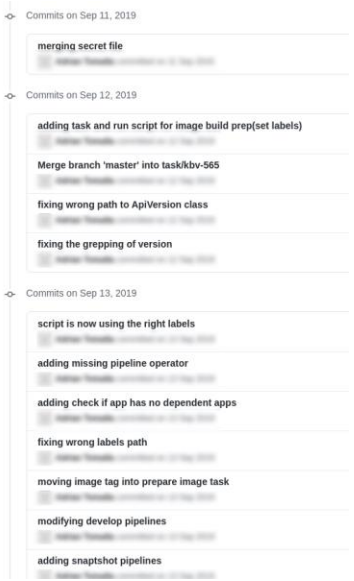
"Debugging" commits

- fly execute to run changed task with inputs provided
- Only works for tasks, not for other step types



Long build times

- Caching maven dependencies on workers reduced the time a lot



NOVATEC

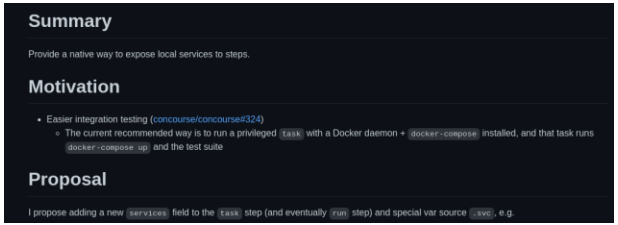
Open Problems

Tests with need of other services

- Since everything runs in containers, running testcontainers is not possible.
- Open RFC - Services

Even with YAML Anchors, LOTS of YAML

- Using templating Engines and Instanced Pipelines reduces the amount



Summary

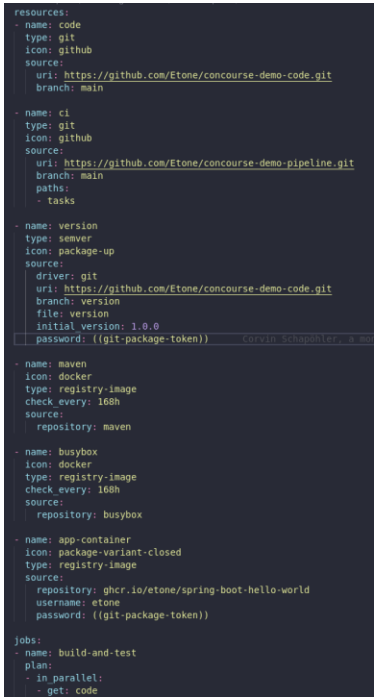
Provide a native way to expose local services to steps.

Motivation

- Easier integration testing (concourse/concourse#324)
  - The current recommended way is to run a privileged `task` with a Docker daemon + `docker-compose` installed, and that task runs `docker-compose up` and the test suite

Proposal

I propose adding a new `services` field to the `task` step (and eventually `run` step) and special var source `.svc`, e.g.



```yaml
resources:
- name: code
  type: git
  icon: github
  source:
    uri: https://github.com/Etone/concourse-demo-code.git
    branch: main

- name: ci
  type: git
  icon: github
  source:
    uri: https://github.com/Etone/concourse-demo-pipeline.git
    branch: main
    paths:
    - tasks

- name: version
  type: semver
  icon: package-up
  source:
    driver: git
    uri: https://github.com/Etone/concourse-demo-code.git
    branch: version
    file: version
    initial_version: 1.0.0
    password: ((git-package-token))

- name: maven
  icon: docker
  type: registry-image
  check_every: 168h
  source:
    repository: maven

- name: busybox
  icon: docker
  type: registry-image
  check_every: 168h
  source:
    repository: busybox

- name: app-container
  icon: package-variant-closed
  type: registry-image
  source:
    repository: ghcr.io/etone/spring-boot-hello-world
    username: etone
    password: ((git-package-token))

jobs:
- name: build-and-test
  plan:
  - in_parallel:
    - get: code
```

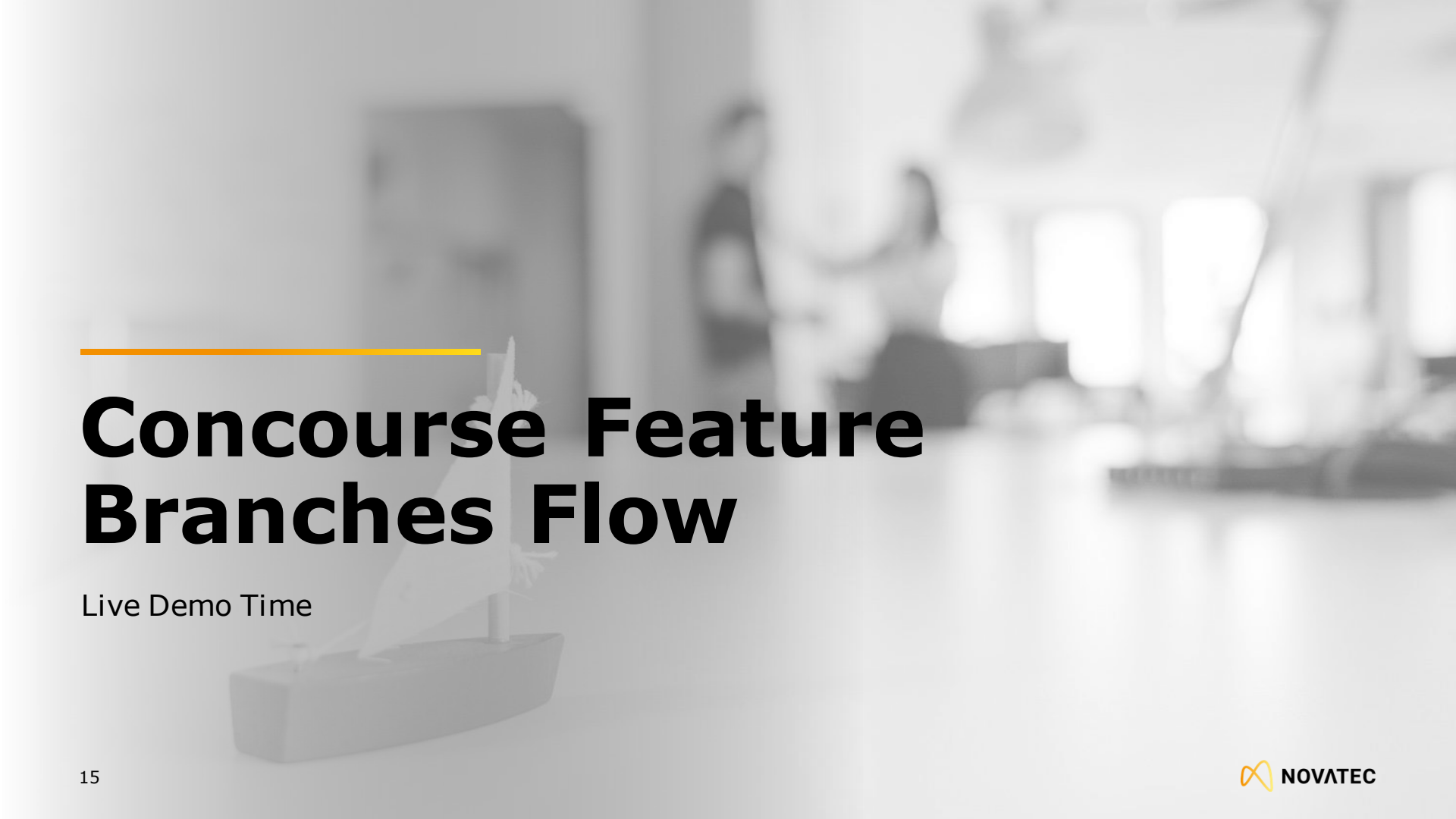

NOVATEC

What's in the future

Concourse v7, Roadmap v10, Live Demo

Future Steps and open RFCs

- **Spatial Resources**
  - Create steps over multiple, dynamic variables
- **Instanced pipelines**
  - Create pipelines from templates as instances
- **Set_pipeline step**
  - Manage pipelines in pipelines
- *Prototypes*
  - Reusable tasks and smaller resource checks
- *Projects*
  - Namespaced resources and pipelines



Concourse Feature Branches Flow

Live Demo Time

Corvin Schapöhler
Software Engineer

Twitter: @cschapoehler
Github: Etone

Novatec Consulting GmbH
Bertha-Benz-Platz 1
D-70771 Leinfelden-Echterdingen

T. +49 711 22040-700
info@novatec-gmbh.de
www.novatec-gmbh.de