



Concourse

Roll your-own CI

What is it? How to use it?

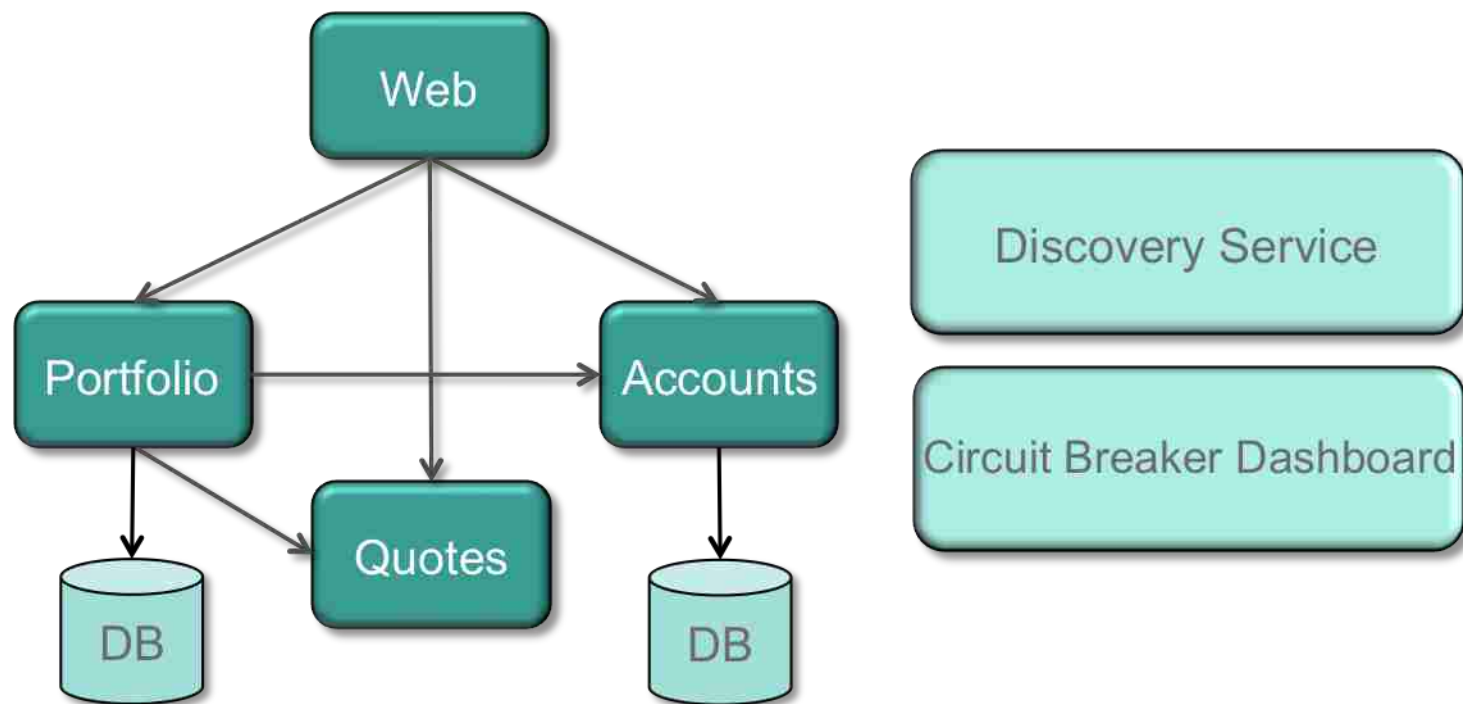
Agenda

- **Sample Application**
- Concourse

Sample Application: Spring Trader

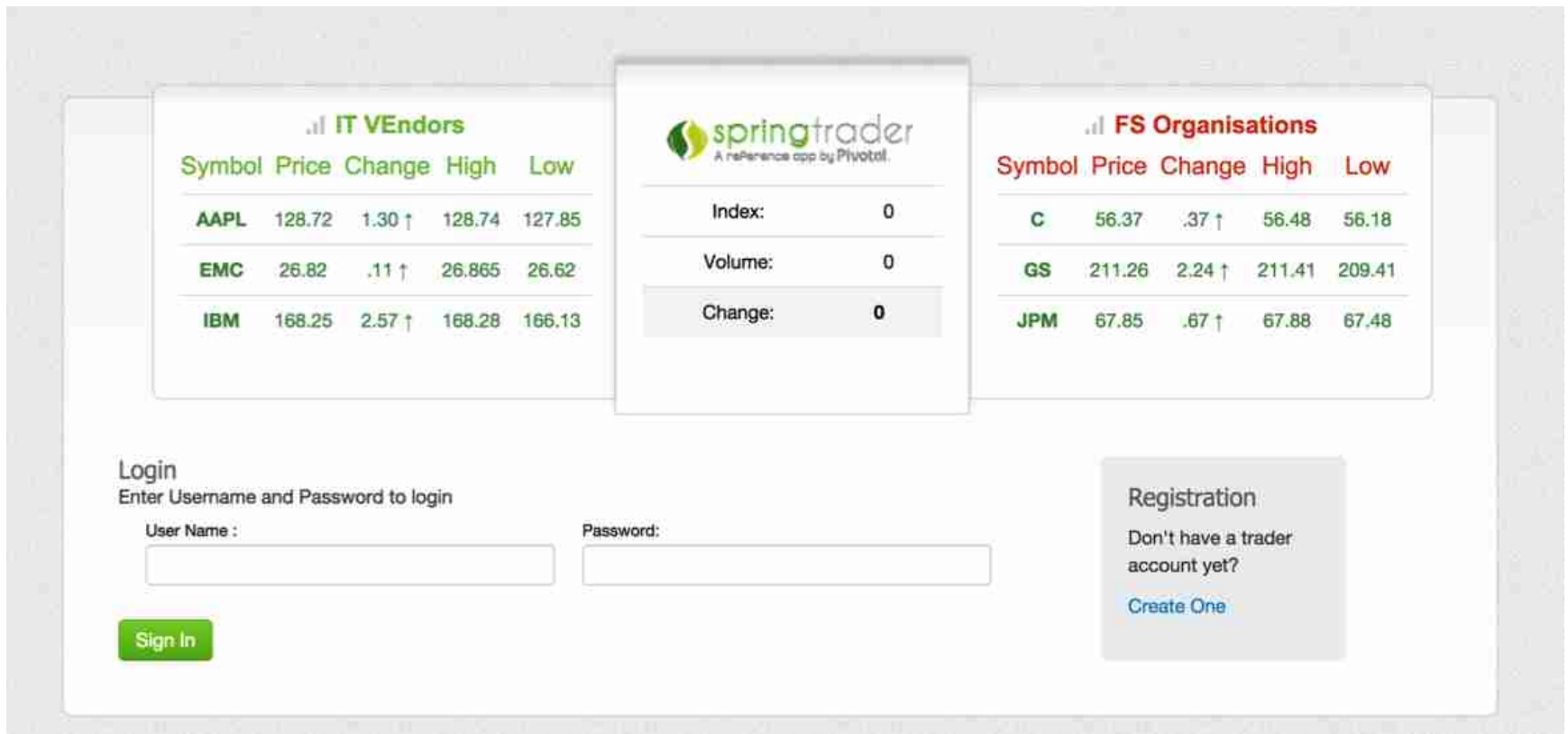
- Pre-existing microservices application
 - Uses Spring Boot and Spring Cloud
 - Incorporates Netflix projects
 - *Eureka* (service discovery)
 - *Ribbon* (load-balancer)
 - *Hystrix* (circuit-breaker, detects failed processes)
 - Deployed to Cloud Foundry
 - <https://github.com/pivotal-anz/cf-SpringBootTrader>

Architecture



Spring Trader Home Page

- Shows prices when NY Stock Exchange trading



The screenshot shows the Spring Trader home page. At the top, there are three main sections: 'IT Vendors', 'springtrader', and 'FS Organisations'. Each section contains a table of stock prices. The 'springtrader' section also includes a summary of market data (Index, Volume, Change). Below these sections is a login area with fields for 'User Name' and 'Password', and a 'Sign In' button. To the right of the login area is a 'Registration' box with a link to 'Create One'.

IT Vendors

Symbol	Price	Change	High	Low
AAPL	128.72	1.30 ↑	128.74	127.85
EMC	26.82	.11 ↑	26.865	26.62
IBM	168.25	2.57 ↑	168.28	166.13

springtrader
A reference app by Pivotal.

Index: 0
Volume: 0
Change: 0

FS Organisations

Symbol	Price	Change	High	Low
C	56.37	.37 ↑	56.48	56.18
GS	211.26	2.24 ↑	211.41	209.41
JPM	67.85	.67 ↑	67.88	67.48

Login
Enter Username and Password to login

User Name :

Password:

[Sign In](#)

Registration
Don't have a trader account yet?
[Create One](#)

Trade Your Portfolio

The screenshot shows a web browser window with three tabs: Pivotal CF, Eureka, and Spring Trader. The address bar shows the URL `webtrader.emea.fe.gopivotal.com/trade`. The page header includes the Spring Trader logo and navigation links for home, portfolio, trade, and a user profile for 'davpin'.

Below the header, there is a search bar with the text 'Enter partial company name or symbol: AAPL' and a 'Get Quotes' button. Below this is a 'Quotes' section with a table showing the quote for AAPL (Apple Inc.).

Company name	Symbol	Price	Change	Change YTD	High	Low	Quantity
APPLE INC	AAPL	128.78	1.36	110.38	128.82	127.85	<input type="text"/> <input type="button" value="BUY"/>

Below the quotes section is a 'List of Holdings' section with a table showing the current holdings of the user.

Symbol	Held quantity	Total purchase price	Total sold value	Current unit value	Current holding value	Total Gain/Loss	Sell
IBM	1000	168350.00	.00	167.96	167,960.00	-390.00 ↓	<input type="text"/> <input type="button" value="Sell"/>
JPM	1000	66350.00	.00	67.79	67,790.00	1,440.00 ↑	<input type="text"/> <input type="button" value="Sell"/>
ORCL	1000	43780.00	.00	43.36	43,360.00	-420.00 ↓	<input type="text"/> <input type="button" value="Sell"/>
EMC	1000	27020.00	.00	26.80	26,800.00	-220.00 ↓	<input type="text"/> <input type="button" value="Sell"/>
Total					305,910.00	1,900.00	

Agenda

- Sample Application
- **Concourse**

What is Concourse?



- First and foremost it is not a CI tool, out-of-the-box
 - But they reckon you can build one
 - Maybe?!
- Concourse allows flows to be defined and run in a dedicated VM
 - Uses YAML files to define the flow
 - No GUI Definition too
 - Web interface to view flows
 - Similar to Spring Cloud Data Flow without the DSL or Flo

Setup - Local

- Run up Concourse VM using Vagrant

```
vagrant init concourse/lite    # creates ./Vagrantfile
vagrant up                     # downloads box & spins up VM
```

- Access at <http://192.168.100.4:8080>
 - Download the CLI “fly” from home page and add to your path

<https://github.com/pivotal-anz/ci-ultimate/blob/master/UsingConcourse.md>



Setup - AWS

- Create an AWS VM using a Concourse AMI
 - Only in default N. America zone
 - Ours is at <http://52.7.64.81:8080>
 - Can download fly again
 - If fly already installed, must run
 - `fly -target http://52.7.64.81:8080 sync`
- Set targets
 - `fly save-target --api http://52.7.64.81:8080 aws`
 - `fly save-target --api http://192.168.100.4:8080 local`
- Now use `fly -t aws` or `fly -t local`

Our Simple CI Process



- Concourse configuration + scripts
 - <https://github.com/pivotal-anz/ci-ultimate>
- Our application
 - <https://github.com/pivotal-anz/cf-SpringBootTrader>
- Our “flow”
 - Clone *ci-ultimate*
 - Clone *cf-SpringBootTrader*
 - Run build and tests using *gradle*
 - Push to PWS



Our Flow Definition ... (1) Resources

resources:

- **name: cf-spring-trader-repo** 1
type: git
source:
uri: https://github.com/pivotal-
anz/cf-SpringBootTrader
branch: master
- **name: ci-ultimate-repo** 2
type: git
source:
uri: https://github.com/pivotal-
anz/ci-ultimate
branch: master

- **name: pcf-bedazzle** 3

type: cf

source:

api: https://api.pcfdemo.info

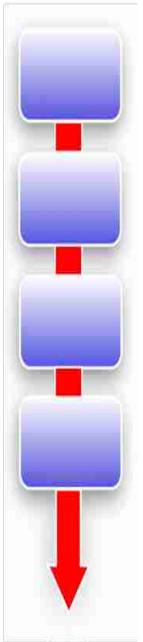
username: ci@pcf-demo.com

password: demo

organization: Bedazzle

space: development

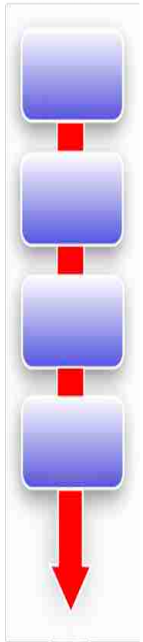
skip_cert_check: true



Our Flow Definition ... (2) The Job Plan

jobs:

- name: ci-pipeline
plan:
 - **get: ci-ultimate-repo**
trigger: true
 - **get: cf-spring-trader-repo**
trigger: true
 - **task: unit**
file: ci-ultimate-repo/ci/ci-task.yml
 - **put: pcf-bedazzle**
params:
 - manifest: cf-spring-trader-repo/manifest-registry-ci.yml
 - path:cf-spring-trader-repo/registryServer/build/libs/registry-0.2.jar
- ...



Resources

- Our repos
 - <https://github.com/pivotal-anz/ci-ultimate>
 - <https://github.com/pivotal-anz/cf-SpringBootTrader>
- Concourse
 - <http://concourse.ci>
- Overview of our CI flow
 - <https://github.com/pivotal-anz/ci-ultimate/blob/master/UsingConcourse.md>

