Pivotal

SpringOne Platform by Pivotal

re:Cap Seoul 2019

2019년 12월 17일(화) • 한화 드림플러스 강남

Pivotal

SpringOne Platform by Pivotal

re:Cap Seoul 2019

Application CI/CD re:Cap

: Concourse / Spinnaker를 사용한 배포 자동화

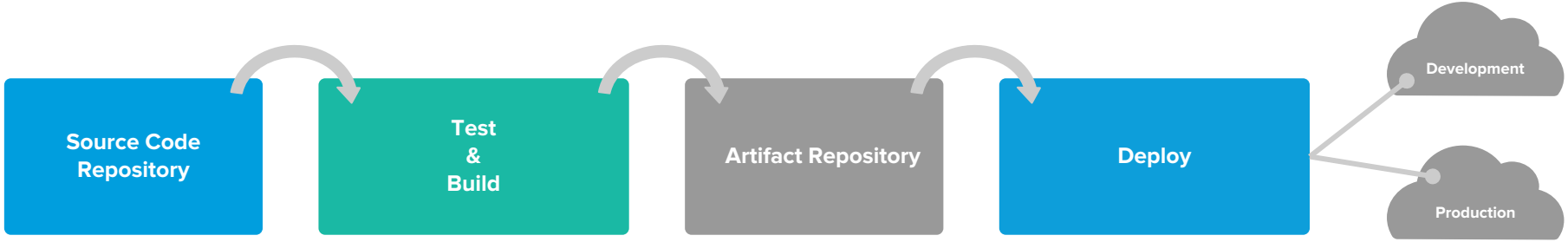
이동희

Senior Platform Architect, Pivotal

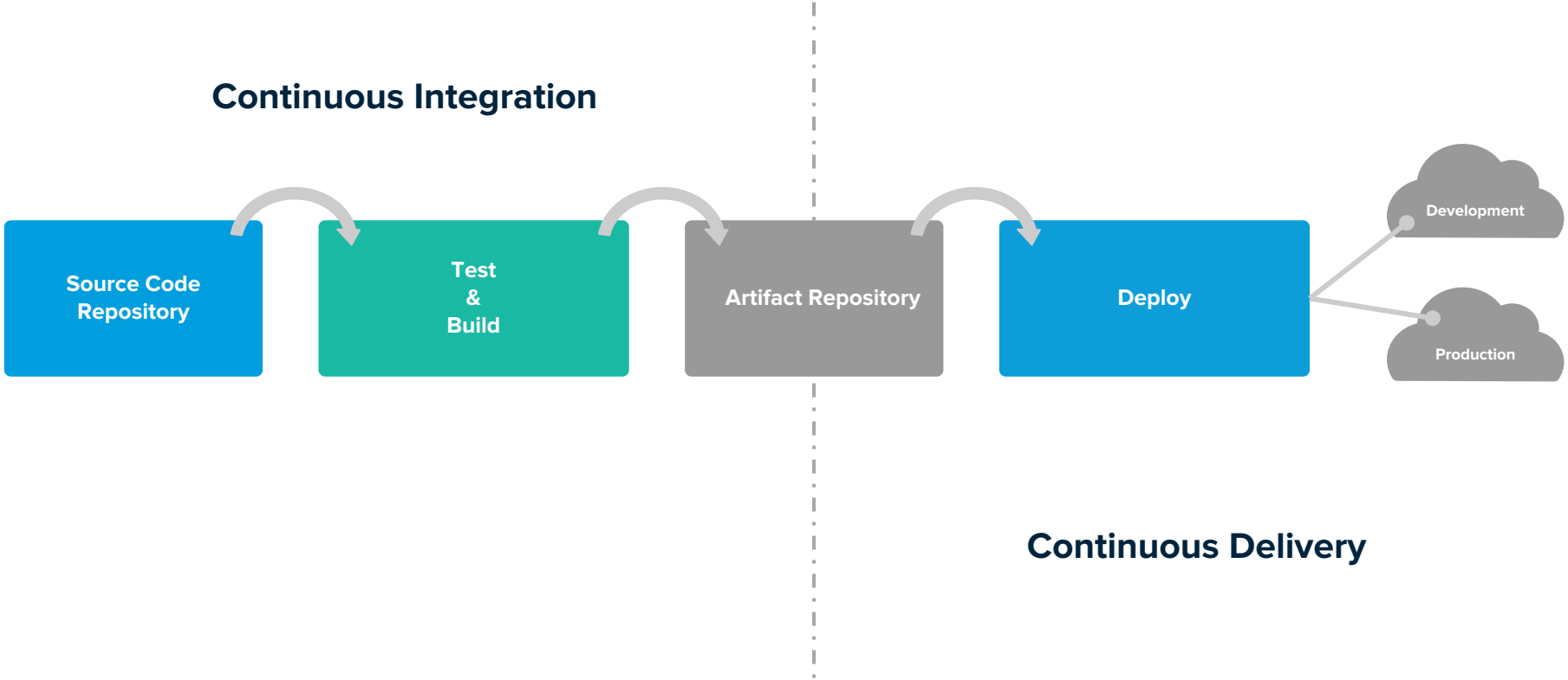
SpringOne Platform 2019 | CI/CD Sessions

- **SDLC for Pivotal Platform, Powered by Spring Initializr, Concourse, and Spinnaker**
- Cutting-Edge Continuous Delivery: Automated Canary Analysis Through Spring-Based Spinnaker
- **Square Pegs, Square Holes: CI/CD That Fits**
- **The Reality of Managing Microservice Deployments at Scale: You Need a Spinnaker**
- Highly Available and Resilient Multi-Site Deployments Using Spinnaker

Continuous Integration & Delivery



Continuous Integration & Delivery



Different Requirements

Continuous Integration

- Accelerate developer feedback
- Continuous testing—fail fast
- Daily code integration practices
- Iterate until your code is “ready to release”

Continuous Delivery

- Accelerate software release process
- Security & compliance
- Safe deployment strategies that can scale
- Operationalize apps

Concourse is a CI Tool
(Sometimes also used for CD)

Concourse Concepts

Resources

Detecting, fetching, creation of externally versioned “things”

Jobs

Compose resources and tasks together to do something (run tests, ship, etc).

Tasks

Run a script in a container with its dependent inputs

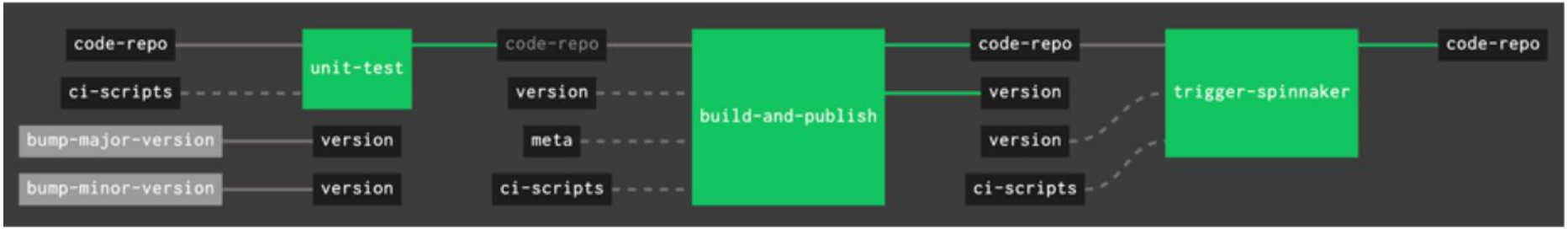
```
# pipeline.yml
resources:
- name: source-code
  type: git
  source:
    uri: https://github.com/...
    branch: master

- name: source-code
  type: git
  source:
    uri: https://github.com/...
    branch: master
```

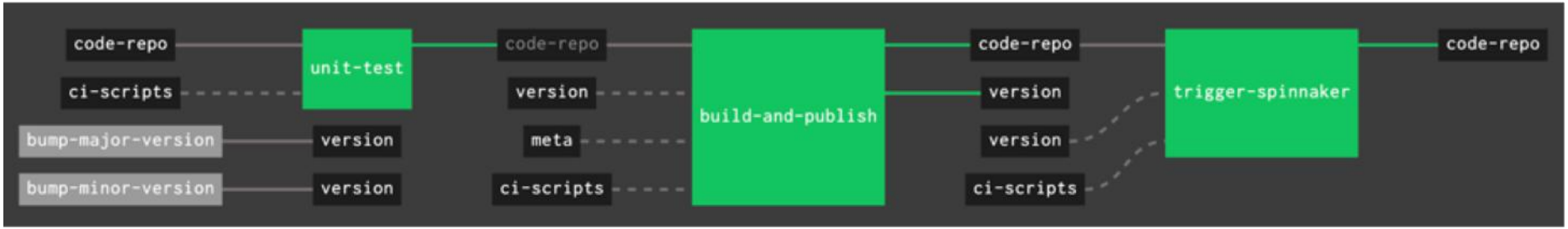
```
# pipeline.yml
jobs:
- name: unit
  plan:
  - get: source-code
    trigger: true
  - task: unit-tests
    file: source-code/ci/unit.yml
```

```
# unit.yml
platform: linux
image_resource:
  type: docker-image
  source:
    repository: java
    tag: '8'
inputs:
  - name: source-code
run:
  path: source-code/mvnw
  args: [ clean, test ]
```

Concourse Pipeline Visualization

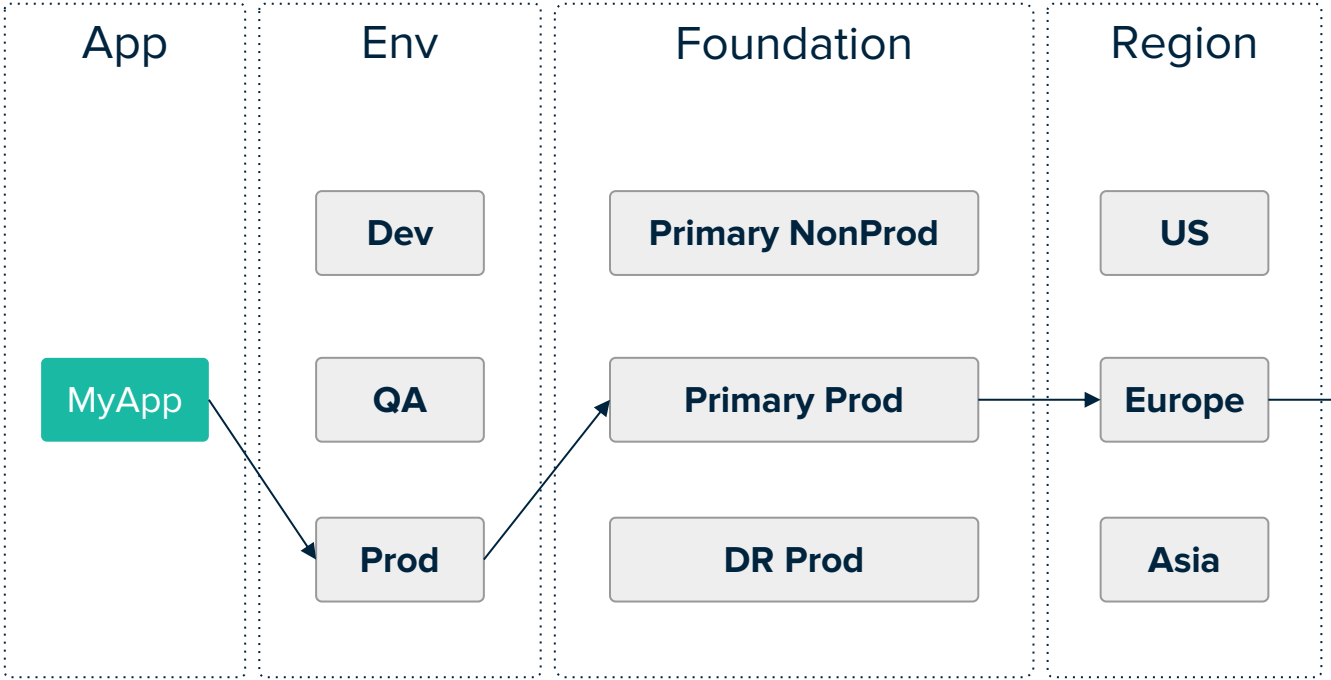


Automated Pipeline



Complex Env - Handling permutations?

Define your configuration and then create files per permutation. An example:



3 x 3 x 3 =
27 configs ??
27 pipelines ??

```
MyApp config

for Prod
Primary datacenter
in Europe
```

Automation requires APIs

To automate the setup of a new service in all components, the components need to provide an API or other way to configure it

- REST and/or Web-service **APIs**
- Command Line Interfaces (**CLI**)
- The capability to define the configuration in a text format (YAML for example) Provide a **templating** mechanism would be great

Pivotal Platform - PAS manifest file example

App

```
# app.yml
name: my-app
```

Env

```
# prod.yml
env: prod
```

Foundation

```
# primary.yml
host: prim-host.abc.com
```

Region

```
# europe.yml
region: europe
```

Template

```
# template.yml
---
Applications:
  name: ((name))
  type: 2G
  routes:
  - route: ((name))-((env)).((region))-((host))
```

Generated Config

```
---
Applications:
  name: my-app
  type: 2G
  routes:
  - route: my-app-prod.europe-prim-host.abc.com
```

```
cf push --vars-file app.yml --vars-file prod.yml --vars-file primary.yml
 --vars-file europe.yml -f template.yml
```

Concourse example

App	Env	Org	Foundation	Region
`# app.yml` `name: my-app`	`# prod.yml` `env: prod`	`# myOrg.yml` `org: MY_ORG`	`# primary.yml` `host: prim-host.abc.com`	`# europe.yml` `region: europe`

Template

```
# perf-pipeline-template.yml
resources:
- name: cf-performance
  type: cf
  source:
    api: https://api.system.((region))-((host))
    username: ((cf_username))
    password: ((cf_password))
    organization: ((org))
```

Generated Config

```
resources:
- name: cf-performance
  type: cf
  source:
    api: https://api.system.europe-prim-host.abc.com
    username: my_deployment_user
    password: the_secret
    organization: MY_ORG
```

```
fly -t np_us set-pipeline -p perf_my-app -c perf-pipeline-template.yml -l myOrg.yml
 -l europe.yml -l primary.yml
```

Spinnaker provides template inheritance

```
{
"schema": "v2",
"application": "{{name}}",
"name": "Deploy Red/Black",
"template": {
  "artifactAccount": "front50ArtifactCredentials",
  "reference": "spinnaker://RedBlackDeplTemplate",
  "type": "front50/pipelineTemplate",
},
"variables": {
  "waitTime": 4
},
"exclude": [],
…
```

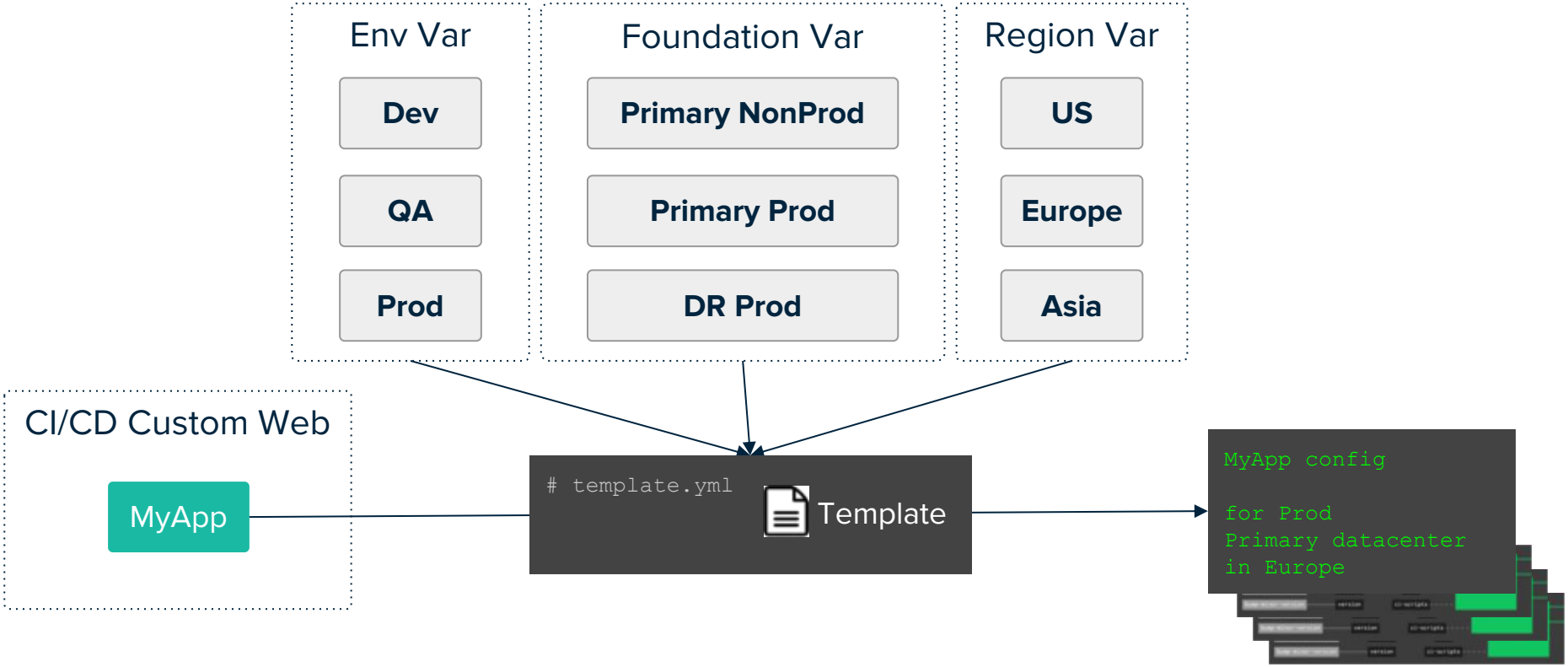
BEFORE

```
{
"schema": "v2",
"application": "myApp",
"name": "Deploy Red/Black",
"template": {
  "artifactAccount": "front50ArtifactCredentials",
  "reference": "spinnaker://RedBlackDeplTemplate",
  "type": "front50/pipelineTemplate",
},
"variables": {
  "waitTime": 4
},
"exclude": [],
…
```

AFTER

```
spin application save --application-name myApp --owner-email theemail@company.com
 --cloud-providers "cloudfoundry”
spin pipeline save --file thepipeline.json
```

Template with variables



Spinnaker is a CD Tool for Multi-Cloud

Spinnaker Embeds CD Expertise

2012

Netflix built Asgard for managing cloud-based applications and infrastructure

NETFLIX

Netflix expanded Asgard ideas and introduced Spinnaker internally

Spinnaker open sourced

Google Cloud

Spinnaker community grows

Canary analysis and chaos monkey added to Spinnaker

Pivotal

Pivotal becomes third largest contributor to Spinnaker OSS

Spinnaker joins Continuous Delivery Foundation

2019

Spinnaker

"We want to provide guardrails, not gates."

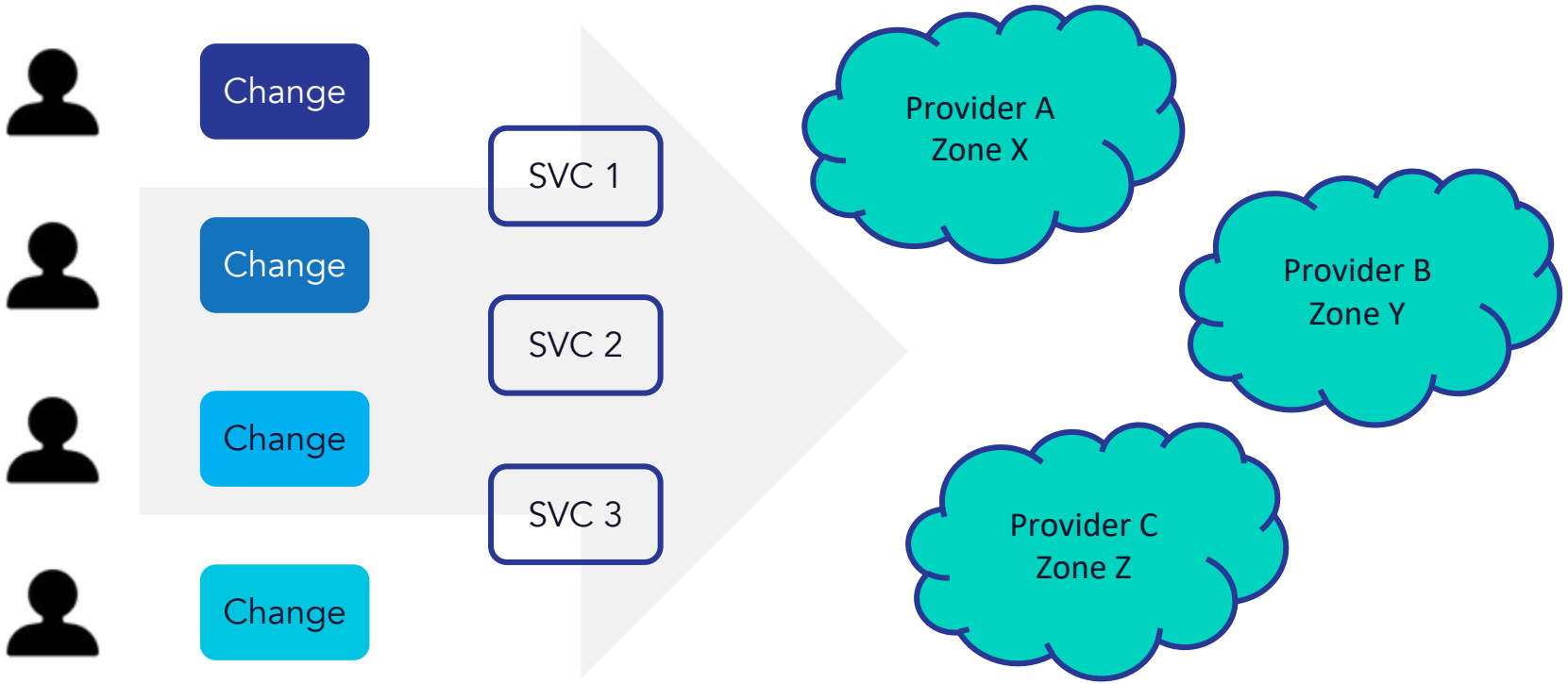
—Dianne Marsh, Netflix

Spinnaker Is an OSS Multi-Cloud Delivery Platform

Spinnaker Community

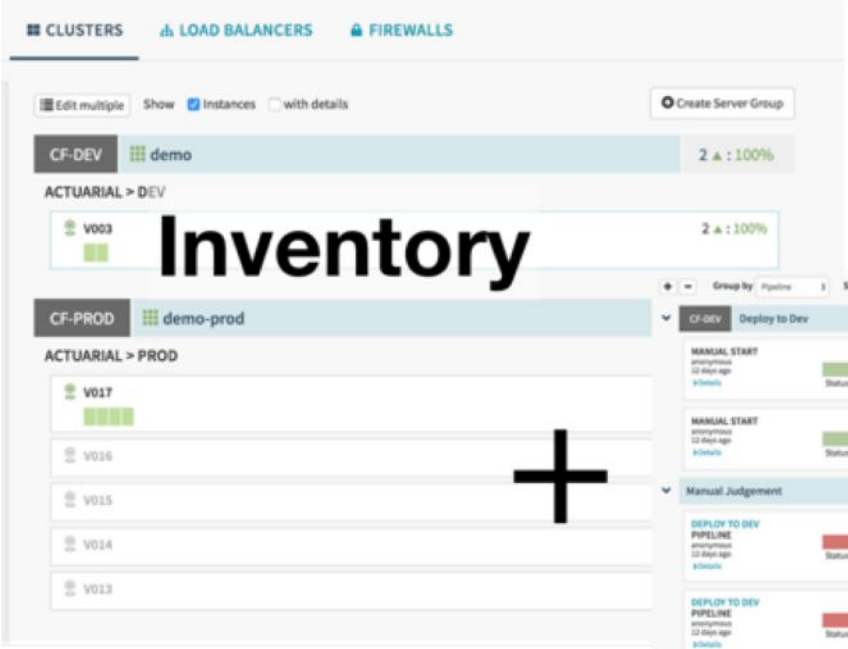
"...the passionate open source community dedicated to making deployment pain go away."

Cloud Deployments are Complex



Application Centric Control Plane

=

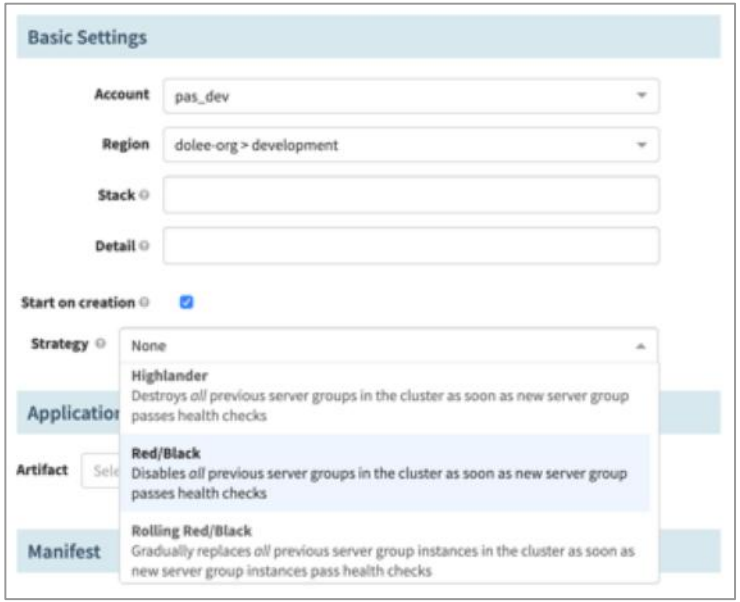


+

Turn monitoring into action

Deploy safely with speed

Spinnaker Deployment Strategies

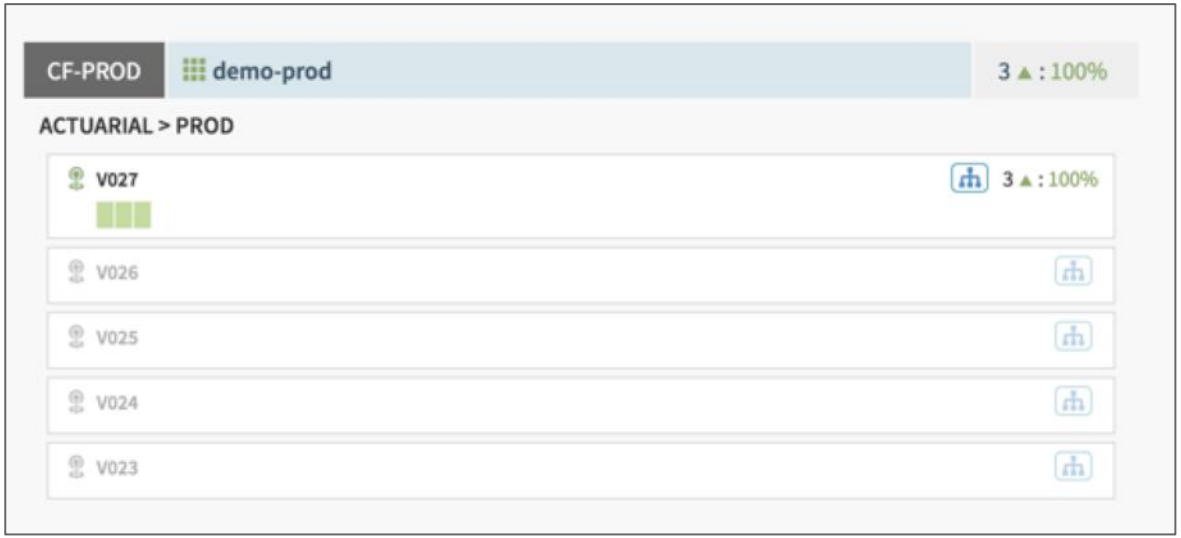


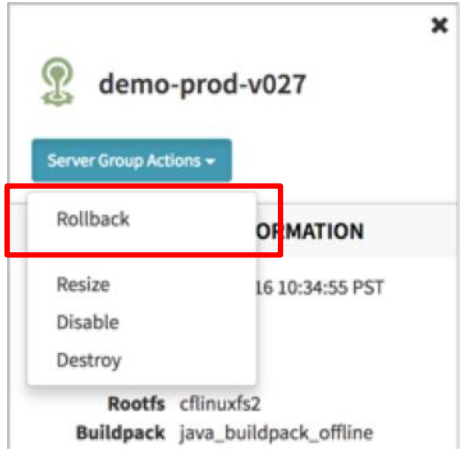
https://www.spinnaker.io/concepts/

Multiple Application Versions for Rollback

Spinnaker CLOUDFOUNDRY

Allows zero-downtime rollbacks to a set number of previous application versions without rebuilding the code





Kubernetes Deployment Strategies

Strategy	What does it do	Downtime?	Simultaneous traffic?
Recreate	1. Terminate V1 2. Start V2	Yes	No
Rolling Update	1. Create V2 ReplicaSet 2. Replace V1 pods with V2, few at a time, until all pods are fully replaced	No	Yes

Spinnaker Deployment Strategies

https://www.spinnaker.io/concepts/

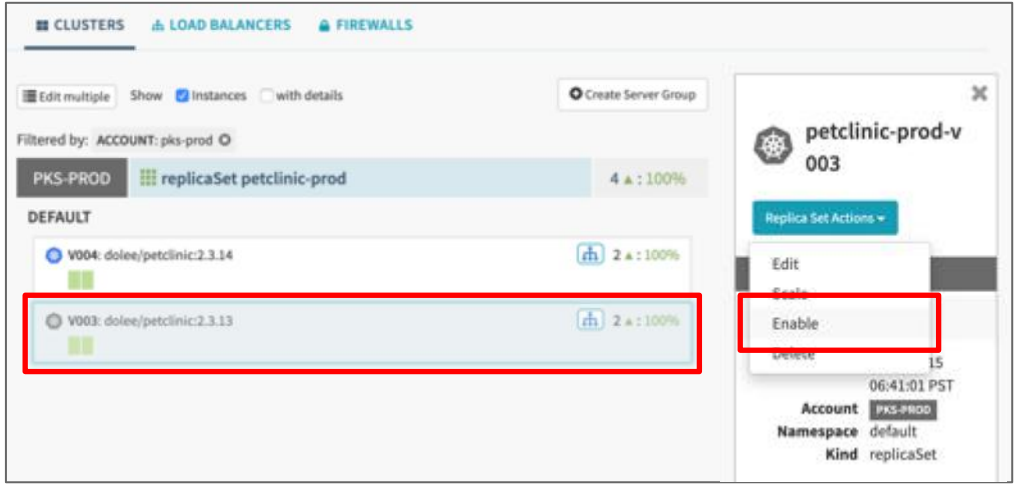
Spinnaker Deployment Strategies

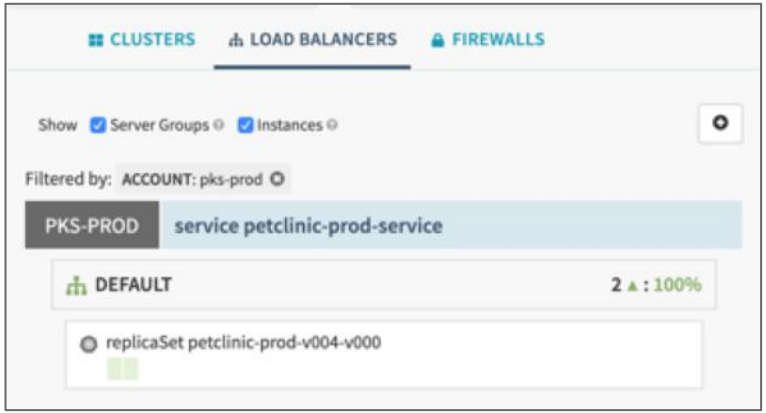
Strategy	What does it do	Downtime?	Simultaneous traffic?
Dark Rollout	1. Deploy V2 alongside V1 but don't send traffic to V2 immediately. 2. Add separate stage to enable traffic to V2 3. Disable traffic to V1	No	Yes
Highlander	1. Deploy V2 alongside V1 2. Send traffic to V2 3. Disable and destroy V1	No	Yes
Red/Black (Blue Green)	1. Deploy V2 alongside V1 2. Send traffic to V2, then disable V1. V1 is available for hot swap if V2 runs into issues.	No	Yes

Multiple Application Versions for Rollback

Spinnaker kubernetes

Allows zero-downtime rollbacks to a set number of previous application versions without rebuilding the code

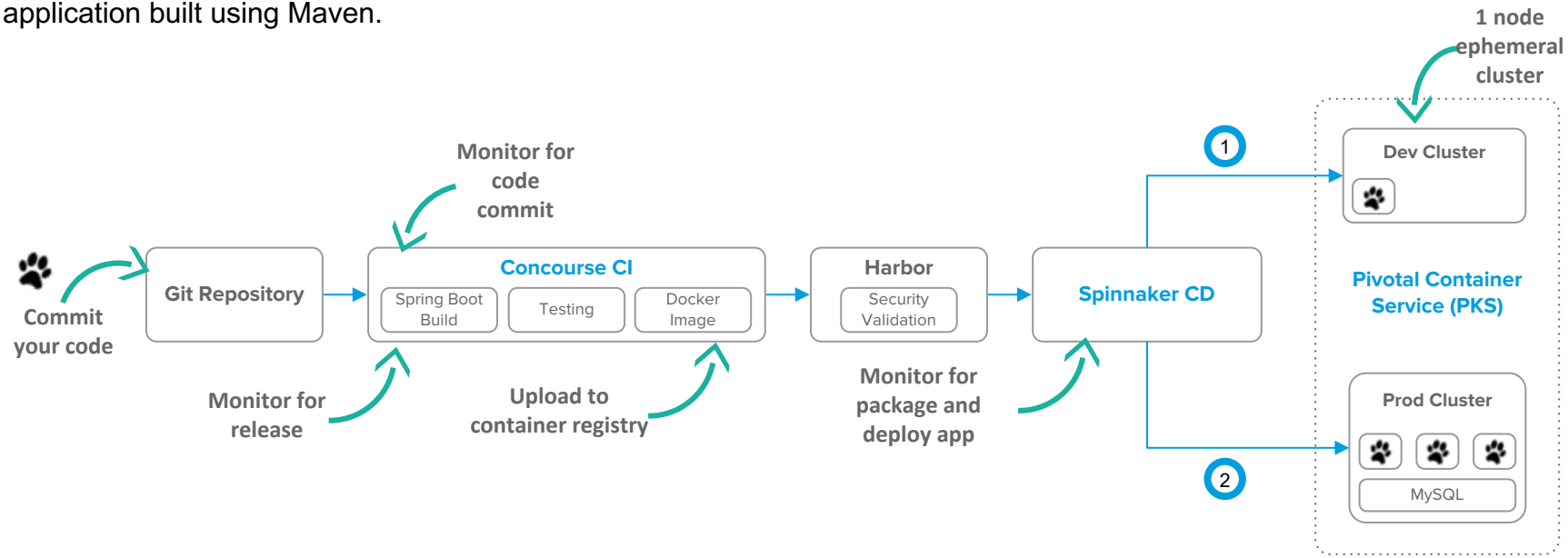




Demo

DEMO: Deploying the PetClinic App to PKS

PetClinic is a Spring Boot application built using Maven.



Pivotal

SpringOne Platform by Pivotal

re:Cap Seoul 2019

감사합니다