

Pivotal.



Concourse

CI that scales with your project

Yadhav Jayaraman
Senior Platform Architect | Pivotal



Continuous Integration & Delivery

Benefits

AUTOMATION.

Integrate tools and automate processes from testing to builds and deployment

SPEED.

Release more frequently with smaller bits will reduce complexity and improve time-to-market

QUALITY.

Reduce feedback loop using test-driven development to surface problems sooner and be responsive

AGILITY.

Push updates on regular basis with no downtime to improve customer experience and time to market

Concepts

Commit Code Change

Automate Build & Test

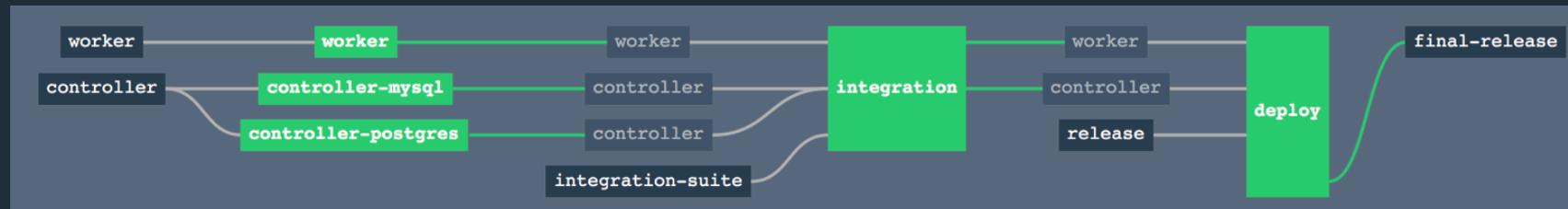
Store Binaries & Build Artifacts

Automated Integration Testing

Acceptance, Performance & Load

Zero Downtime Upgrade to Production

Pipeline



Concourse

© Copyright 2016 Pivotal. All rights reserved.

Pivotal™

Why Concourse?

Hudson



Travis CI



Jenkins



Concourse

© Copyright 2016 Pivotal. All rights reserved.

Pivotal™

What we found in other CI systems

Snowflakes

- lots of plugins
- system dependencies
- textbox scripting

Pipelines

- no first-class support
- complex job sequencing

Environment Parity

- works locally, breaks on server
- lots of debugging commits



Usability

- complicated UIs
- endless menus
- too many clicks to get logs

Execution Hierarchy

- deep and complex

Scalability

- hard to scale vertically or horizontally



Concourse

© Copyright 2016 Pivotal. All rights reserved.

Pivotal™

What if we could...

Configurable

- declarative CI (no more snowflakes)
- store in version control

Unpolluted builds

- previous builds do not affect subsequent builds

Environment Parity

- run tasks locally before committing

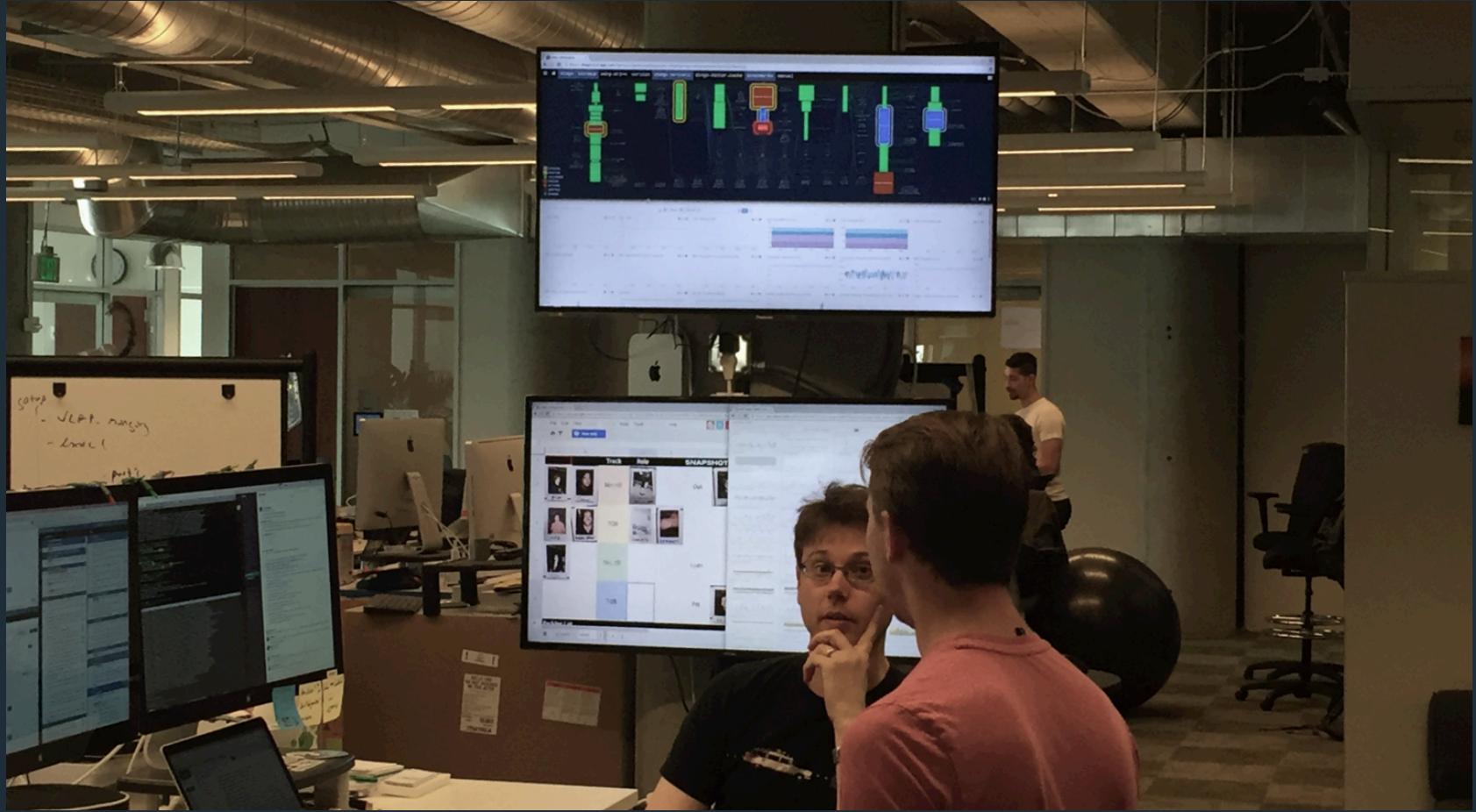
Usability

- visualize pipeline
- simple ui (click less!)



Scalability

- scale up to increase performance
- scale down to decrease cost



Concourse

© Copyright 2016 Pivotal. All rights reserved.

Pivotal™

Concourse Principles

Simple

Concourse is a response to the complexity introduced by other systems. It is built on the idea that the best tools can be learned in one sitting.

Usable

Concourse is optimized for quickly navigating to the pages you most care about. From the main page, a single click takes you from a pipeline view to the log of a job's latest failing build.

Isolated Builds

Every build task is executed in a container defined by its own configuration, by stateless workers. This eliminates build pollution and ensures multiple teams can use the same Concourse deployment without worrying about the state of the worker VMs.



Concourse

© Copyright 2016 Pivotal. All rights reserved.

Pivotal™

Concourse Principles

Scalable, reproducible deployment

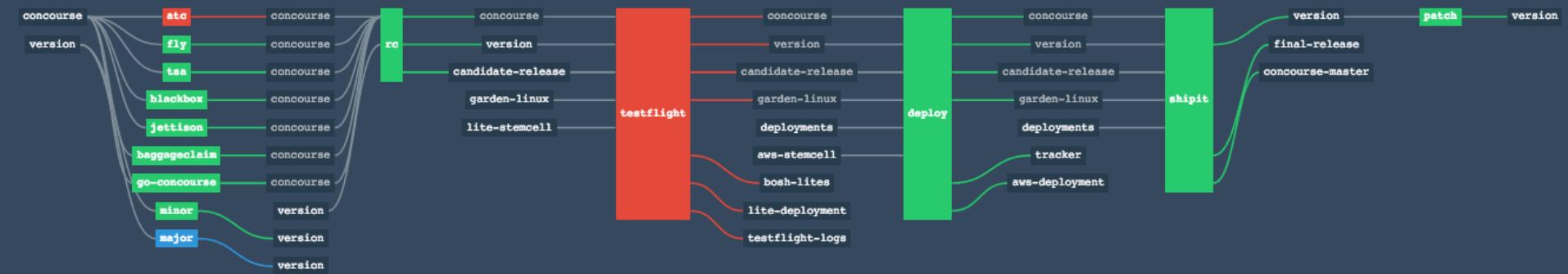
No Concourse deployment is a snowflake. There are no boxes to check; no configuration happens at runtime.

Flexible

Features that other systems implement in the core of the product, Concourse implements in "userland", as resources. This keeps the core of Concourse small and simple, and proves out the extensibility introduced by this simple interface.

Local iteration

Concourse supports running one-off builds from local task configuration that allows you to trust that your build running locally runs exactly the same way that it runs in your pipeline.



- [] pending
- [■] started
- [■] succeeded
- [■] failed
- [■] errored
- [■] aborted
- [■] paused

concourse publish bosh-lite rubbish-bin

atc #2111 started 24m 18s ago
finished 19m 5s ago
duration 5m 13s

2111 2110 2109 2108 2107 2106 2105 2104 2103 2102 2101 2100 2099 2098 2097 2096 2095 2094 2093 2092 2091 2090 2089

↓ concourse

ref 5e36e3b4e7fa15cbd90a7cc19e22ba67d67f1a62 ✓

↳ go-unit ×

↳ js-unit ×

```
+ pushd concourse/src/github.com/concourse/atc/web
/tmp/build/cb7ccdb4/concourse/src/github.com/concourse/atc/web /tmp/build/cb7ccdb4
+ npm install
npm [WARN] package.json atc@ No description
npm [WARN] package.json atc@ No README data
npm [WARN] deprecated react-tools@0.12.2: react-tools is deprecated. For more information, visit https://fb.me/react-tools-deprecated
npm [WARN] deprecated gulp-minify-css@1.2.3: Please use gulp-cssnano instead.
npm [WARN] optional dep failed, continuing fsevents@0.3.8
npm [WARN] deprecated lodash@1.0.2: lodash@<2.0.0 is no longer maintained. Upgrade to lodash@^3.0.0
\
> elm@0.16.0 install /tmp/build/cb7ccdb4/concourse/src/github.com/concourse/atc/web/node_modules/elm
> node install.js

Downloading Elm Reactor assets from https://dl.bintray.com/elmlang/elm-platform/0.16.0/elm-reactor-assets.tar.gz
Downloading Elm binaries from https://dl.bintray.com/elmlang/elm-platform/0.16.0/linux-x64.tar.gz
Successfully downloaded and processed linux-x64.tar.gz
Successfully downloaded and processed elm-reactor-assets.tar.gz
jasmine-jquery@2.1.1 node_modules/jasmine-jquery

react-immutable-render-mixin@0.6.5 node_modules/react-immutable-render-mixin

node-ansiparser@2.1.0 node_modules/node-ansiparser

jasmine-ajax@3.2.0 node_modules/jasmine-ajax

immutable@3.7.6 node_modules/immutable

moment-duration-format@1.3.0 node_modules/moment-duration-format

jshint-stylish@1.0.2 node_modules/jshint-stylish
```

Concourse Concepts: simple primitives

resources

detecting, fetching, creation of externally versioned “things”

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

jobs

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

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

tasks

run a script in a container with its dependent inputs

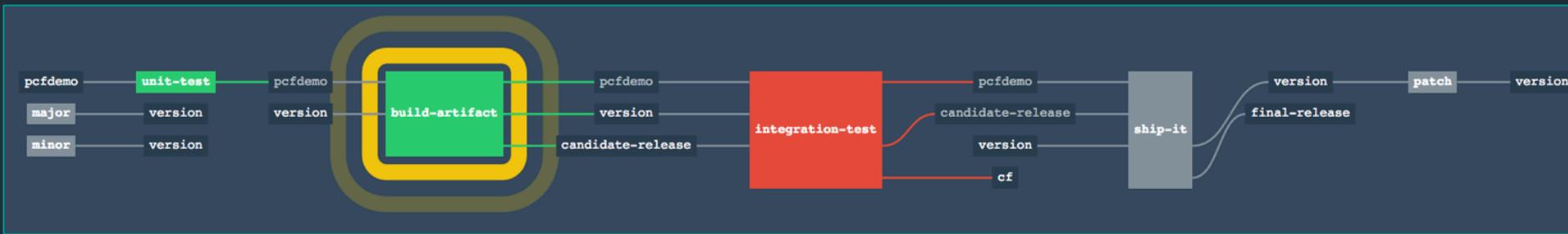
```
# unit.yml
platform: linux
image_resource:
  type: docker-image
  source:
    repository: java
    tag: '8'
inputs:
- name: source-code
run:
  path: source-code/mvnw
  args: [ clean, test ]
```

Concourse Concepts: pipelines

the resulting flow of resources through jobs

fancy visualization UI for build monitor

many isolated pipelines per deployment



Concourse Concepts: reproducible deployment

no clicking through wizards to configure

Concourse pipelines are just config files

deploy cluster with PCF, BOSH, or Vagrant

workers are stateless



reproducible deployment
and configuration

infrastructure failure resilient



Concourse

© Copyright 2016 Pivotal. All rights reserved.

Pivotal™

Concourse Concepts: reproducible builds

never hand-configure workers again

all builds run in stateless containers

one-off builds with custom inputs (local bits)

no ability to rely on CI state



portable CI scripts with
decent abstractions, less
coupled to Concourse

Concourse Concepts: resources

- encapsulation of some external resource
- replaces plumbing scripts
- results in intuitive pipeline semantics
- many first-class concepts from other systems are implemented in terms of resources (ex: timed triggers)
- only pluggable interface

Concourse Concepts: resource interface

```
get      ::  version  -> input  
put      ::  output   -> version  
check    ::  version? -> [version]
```

ex: git resource:

```
get      =  git clone git@github.com:...  
put      =  git push origin [branch]  
check   =  git pull && git log abcdef..HEAD
```

Concourse Concepts: available resources

- git repo
- s3 bucket
- docker image
- bosh deployment
- bosh.io release
- bosh.io stemcell
- Pivotal Tracker
- github release
- cf app
- vagrant cloud/atlas
- time
- semver

more coming: twitter, email, github-pr, rubygems, ...

fly execute: run task with local bits

```
~/git/PCF-demo » fly -t lite execute -c ci/tasks/build.yml -i pcfdemo=.  
executing build 92  
  % Total      % Received   % Xferd  Average Speed   Time     Time     Time  Current  
          Dload      Upload   Total   Spent   Left  Speed  
100 58.8M    0 58.8M    0      0  28.1M      0  --:--:--  0:00:02  --:--:-- 28.1M  
initializing with docker:///java#8  
running pcfdemo/ci/tasks/build.sh  
[INFO] Scanning for projects...  
[INFO]  
[INFO] -----  
[INFO] Building pcf-demo 1.0.0-BUILD-SNAPSHOT  
[INFO] -----  
...  
[INFO] Packaging webapp  
[INFO] Assembling webapp [pcf-demo] in [/tmp/build/e55deab7/pcfdemo/target/pcfdemo]  
[INFO] Processing war project  
[INFO] Copying webapp resources [/tmp/build/e55deab7/pcfdemo/src/main/webapp]  
[INFO] Webapp assembled in [61 msecs]  
[INFO] Building war: /tmp/build/e55deab7/pcfdemo/target/pcfdemo.war  
[INFO] WEB-INF/web.xml already added, skipping  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 33.423 s  
[INFO] Finished at: 2016-02-03T12:56:54+00:00  
[INFO] Final Memory: 20M/217M  
[INFO] -----  
succeeded
```

fly hijack: hop into build's container

```
~/git/PCF-demo » fly -t lite hijack -j pcfdemo/build-artifact
1: build #10, step: version, type: get
2: build #10, step: prepare-build, type: task
3: build #10, step: candidate-release, type: put
4: build #10, step: version, type: put
5: build #10, step: pcfdemo, type: get
6: build #10, step: version, type: get
7: build #10, step: candidate-release, type: get
8: build #10, step: build, type: task
choose a container: 2
root@bqvgog0t9s0:/tmp/build/5020c204# ls -al
total 8644
drwxr-xr-x 1 root root      84 Feb  3 13:14 .
drwxr-xr-x 1 root root      16 Feb  3 13:14 ..
drwxr-xr-x 1 root root     14 Feb  3 13:14 build
-rw-r--r-- 1 root root 8849783 Feb  3 13:14 pcf-demo-1.1.0-rc.4.war
drwxr-xr-x 1 root root     242 Feb  3 11:24 pcfdemo
drwxr-xr-x 1 root root     12 Feb  3 13:14 version
root@bqvgog0t9s0:/tmp/build/5020c204# echo `cat version/number`
1.1.0-rc.4
root@bqvgog0t9s0:/tmp/build/5020c204#
```



Concourse

© Copyright 2016 Pivotal. All rights reserved.

Pivotal™

fly set-pipeline: iterate on pipeline

```
~/git/PCF-demo » fly -t lite set-pipeline -p pcfdemo -c ci/pipeline.yml -l ~/concourse/pcfdemo-properties.yml
resources:
  resource cf has changed:
    name: cf
    type: cf
    source:
      api: https://api.local.micropcf.io
      organization: micropcf-org
      password: admin
      skip_cert_check: true
      skip_cert_check: false
      space: micropcf-space
      username: admin

apply configuration? [yN] :
```

Pipelines Patterns in Concourse

- Trigger a pipeline on a time resource
- Trigger a pipeline when a depending resource changes
- Share artifacts among tasks of the same job using output/input options
- Use artifact or file repository resources to share artifacts across jobs
- Notify users with notification resources when a job fails or runs successfully



Concourse

© Copyright 2016 Pivotal. All rights reserved.

Pivotal™

Pipelines Patterns – Time Resource trigger

- Pipelines can be executed on a schedule with the “time” resource

<https://github.com/concourse/time-resource>

```
resources:  
- name: 5m  
  type: time  
  source: {interval: 5m}  
  
jobs:  
- name: run-something-every-5m  
  plan:  
    - get: 5m  
      trigger: true    # ← this is what defines the automatic trigger  
    something  
    config: # ...
```

Pipelines Patterns – Resource change trigger

- Trigger a pipeline when a depending resource changes

For example, when a file in a github repository resource changes:

<https://github.com/concourse/git-resource>

```
resources:  
- name: concourse-pipeline-samples  
  type: git  
  source:  
    branch: master  
    uri: https://github.com/pivotalservices/concourse-pipeline-samples.git  
  
jobs:  
- name: unit-tests  
  ...  
  
plan:  
- get: concourse-pipeline-samples  
  trigger: true    # ← this is what defines the automatic trigger for the resource  
- task: my-run-unit-tests  
  ...
```

Pipelines Patterns – Share artifacts among tasks

- Tasks within the same job can share artifacts (e.g. files)
- The output of a task can be used as the input of a later task
- When a task defines an item under its “outputs” entry, Concourse will automatically create a directory for that item in the running container. That item can be used as an entry for the “inputs” of a later task, where a directory with that same name and with the same content produced by the previous task will be available for the container of the running task.
- See inputs and outputs definitions for tasks in this document:
<http://concourse.ci/running-tasks.html>



Concourse

© Copyright 2016 Pivotal. All rights reserved.

Pivotal™

Pipelines Patterns – share artifacts across jobs

- Artifacts can only be shared across jobs with the use of resources
- A task of a job can only consume artifacts produced by another task of another job via resources such as git, S3 or other file repositories.
- The producer task needs to write its output to a resource before another task in another job is allowed to use it



Concourse

© Copyright 2016 Pivotal. All rights reserved.

Pivotal™

Pipelines Patterns – notify pipeline users of execution status

- Notification resources can be used to send messages to pipeline users when a job execution fails or succeeds
- Email Resource
<https://github.com/mdomke/concourse-email-resource>
- Slack Notifications
<https://github.com/cloudfoundry-community/slack-notification-resource>
- Twitter resource
<https://github.com/ECSTeam/twitter-resource>
- Metadata about running builds
<https://concourse.ci/implementing-resources.html#resource-metadata>



Concourse

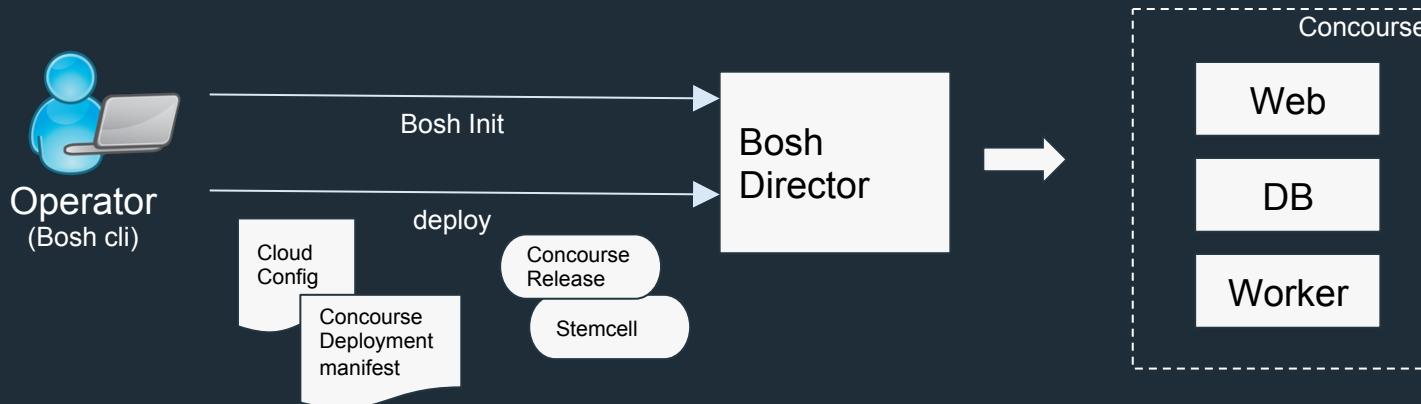
© Copyright 2016 Pivotal. All rights reserved.

Pivotal™

Deploying Concourse - Clusters with Bosh

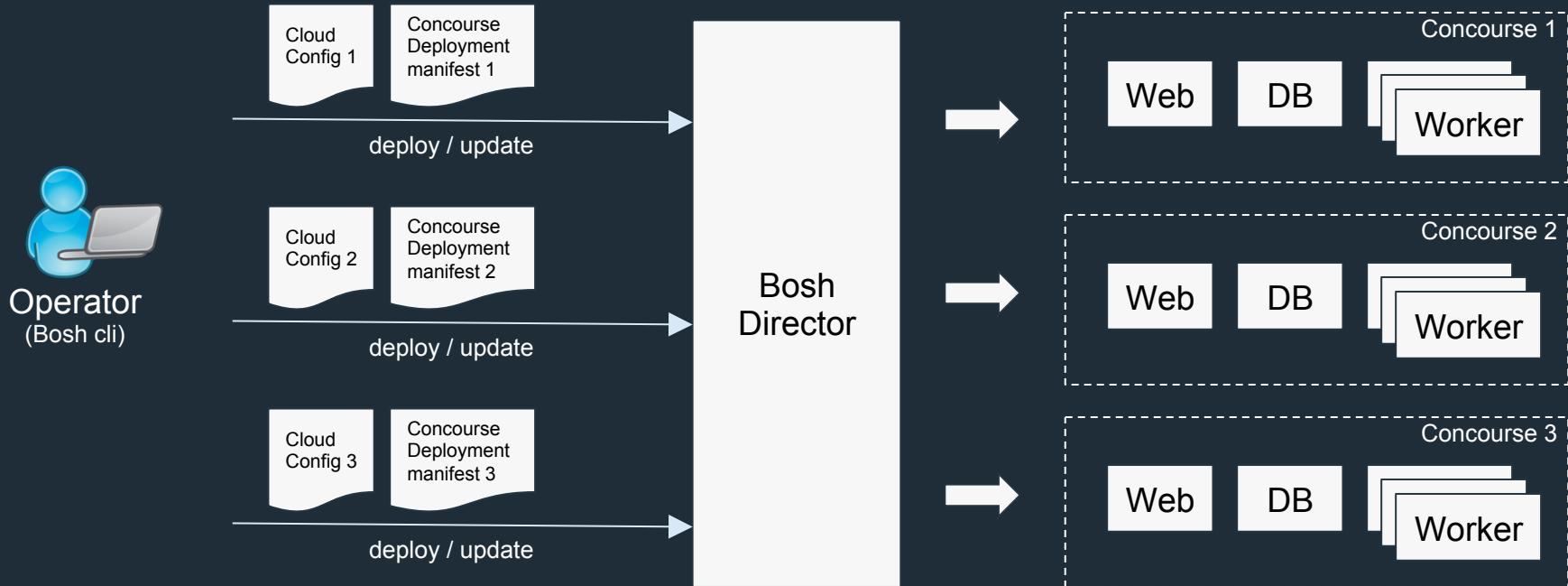
Scalable cluster with health management and rolling upgrades

1. Setup Bosh Director
2. Define deployment manifest files (Cloud Config + Concourse)
3. Upload Releases and Stemcell files to Bosh Director
4. Deploy!



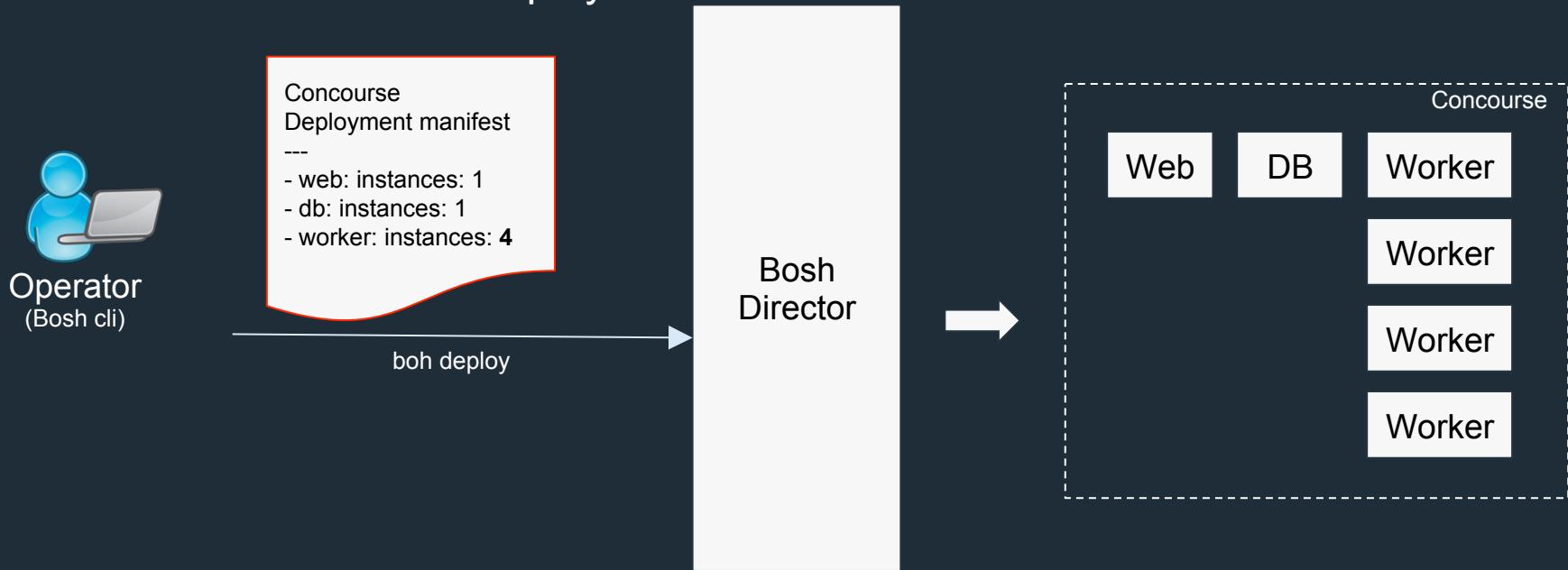
Deploying Concourse - Clusters with Bosh (cont.)

One Bosh Director can deploy and manage multiple Concourse instances



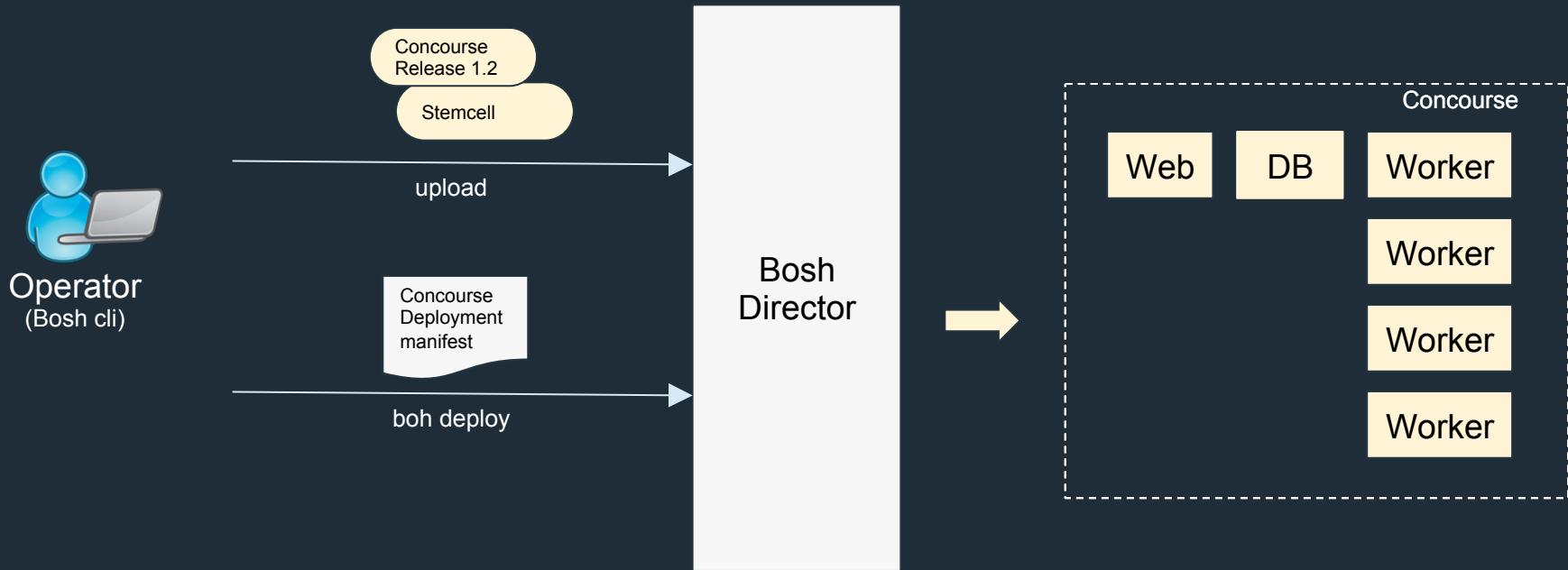
Deploying Concourse - Clusters with Bosh (cont.)

Scaling workers up or down consists of simply updating the Deployment Manifest and issue the deploy command:



Deploying Concourse - Clusters with Bosh (cont.)

Upgrading consists of uploading the new release and running Bosh Deploy command:



How To Get Started

- Go to: [concourse.ci](#)
 - [Getting Started](#) docs to vagrant up
 - [Versioned S3 Artifacts](#) example
 - [PCF-demo](#) example
- bosh [deploy it](#) somewhere
- [Concourse team](#) on Slack

Pivotal.

Open.
Agile.
Cloud-Ready.

