

A background image showing a business meeting. Two people are seated at a table, looking at documents. One person is holding a pen and pointing at a document. The documents contain charts and graphs.

Pivotal Cloud Foundry – Managing Applications

Objectives of PCF

- Purpose:
 - To learn how to manage applications in cloud founry.
- Product:
 - Log Management
 - Application Performance management
 - Autoscaling
 - Zero-Downtime Deployments
- Process:
 - Understand 3rd party log, auto scaling, APM and zero-downtime deployments.

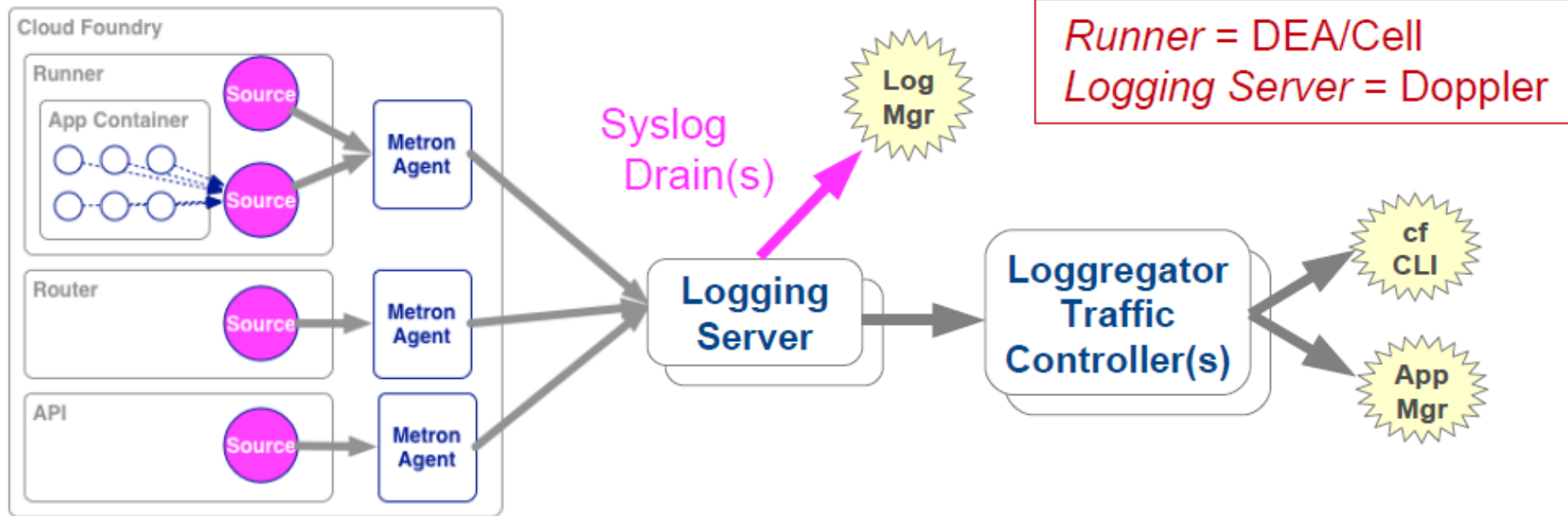
Table of Contents

- Log Management
- Application Performance management
- Autoscaling
- Zero-Downtime Deployments

LOG MANAGEMENT

Recall: Log Aggrgation Architecture

- **Collects** log output from app instance, CF components
- **Aggregates** into a consolidated log
- **Sinks** to cf logs, App Mgr, third-party log managers



Why Third-Party Log Managers?

- Recommended approach
 - Can store far more logging information than CF
 - Allow for persistence, storage, searching, analyzing, metrics
- Variety of third-party log managers supported:

 **sumologic** **papertrail**

 **logstash**

vmware
vCenter Log Insight

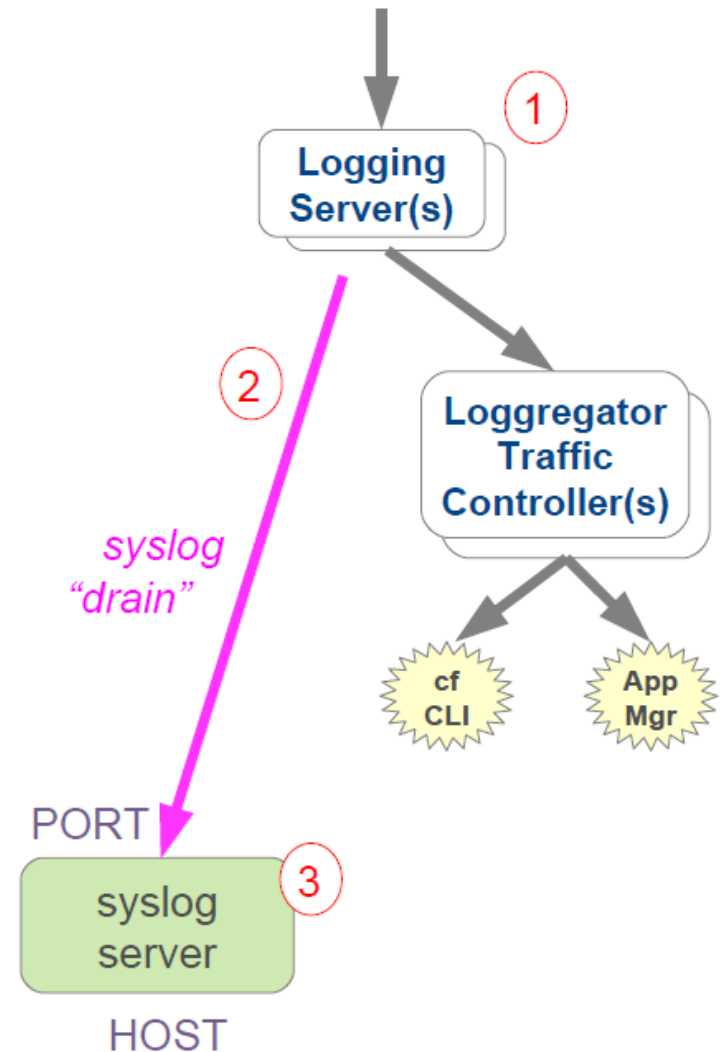
splunk > **storm**[™]

Connecting to Third-Party Log Managers

- Setup Log Managers, determine HOST and PORT.
 - Process varies according to vendor
- Create User ProvidedService with a Syslog drain:
 - **cf cups <SERVICE> -l syslog://<HOST>:<PORT>**
- Bind to application, restage
 - Cloud Foundry sinks loggregator output to this drain for this application

How It Works

- All output for app collected by Logging(Doppler) server (1)
- Loggregator opens socket to **HOST:PORT** (2)
 - sends all log info for app to socket in syslog format
- Received by third-party syslog server (3)



Example: PWS and PaperTrail

- **PaperTrail: Cloud-based Log Manager**

- a) Create account at <https://papertrailapp.com>
- b) Use the "Add System" button
 - a) Papertrail will provide you the URL to use for your syslog drain
 - b) Example: logs2.papertrailapp.com:41845

Example: PWS and PaperTrail

c) Click the
"Alternatives" link

d) Select the
"Heroku" option

e) Name your
system

Choose your situation:

☐ **A** My syslogd only uses the default port

GNU syslogd and some embedded OSes will only log to port 514. A few Linux distros shipped with GNU syslogd (mostly older CentOS and Gentoo).

☒ **B** I use Heroku

Register each app separately.

☐ **C** My system's hostname changes

In rare cases, one system may change hostnames frequently. For example, a roaming laptop which sets its hostname based on DHCP (and roams across networks).

Heroku uses TCP without TLS. We'll provide an app-specific Heroku syslog drain and [step-by-step setup](#) .

Let's create a destination for this app.

What should we call it?

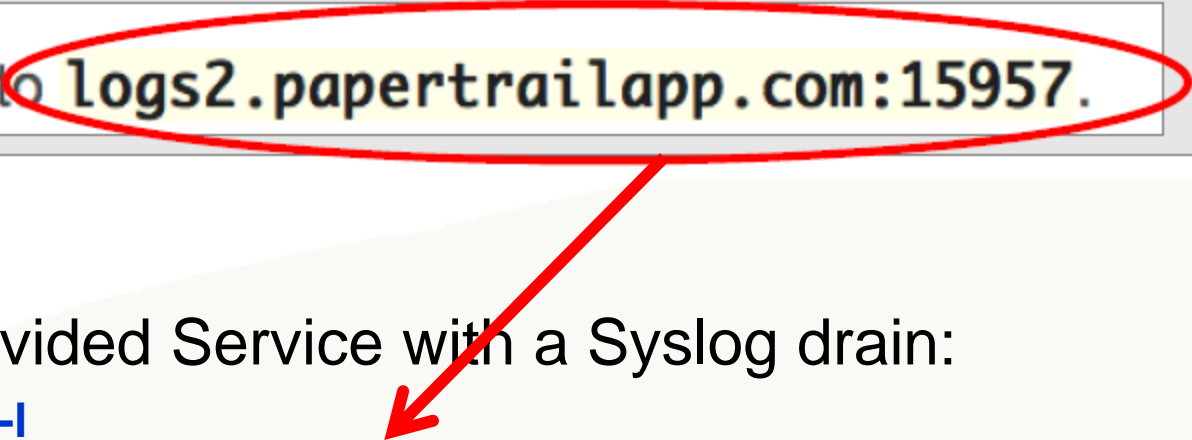
Alphanumeric. Does not need to match app name.

Save →

Example: PWS and PaperTrail

- f) Setup user defined service using Papertrail's URL

MyCFSystem will log to **logs2.papertrailapp.com:15957.**



- g) Create User Provided Service with a Syslog drain:

```
cf cups the-drain -l  
syslog://logs2.papertrailapp.com:15957
```

- h) Bind to application, restart:

```
cf bind-service the-app the-drain  
cf restart the-app
```

About Syslog

- De facto standard for logging on Unix/Linux
 - Can log to a file or a server *syslogd* (via a protocol)
 - Splunk, Papertrail and others provide syslog servers
- To log to syslog
 - Generate a TCP and UDP message in the right format
 - Open a socket to your syslog server and send
- Higher level logging options exists
 - <https://github.com/cloudfoundry-community/java-loggregator>
 - Output handlers for Java logging or log4j/ logback

APPLICATION PERFORMANCE MANAGEMENT

Application Performance Management



- Logs and analysis only takes you so far
- Important to have real-time monitoring of applications
 - Uptime, performance, etc.
- Application Performance Monitoring (APM) Tools
 - Monitor your application while running
 - Several choices available in Cloud Foundry
 - PWS - New Relic and AppDynamics
 - Pivotal Spring Insight

- PWS offers simple interface to New Relic
 - Available as Marketplace Service
 - Tracks different instances of application
 - Monitors down to the line of code

- How to Use:
 - Create New Relic service in desired space
 - Bind to desired Application(s)
 - Re-stage application
 - Java Buildpack includes New Relic Agent, others may not
 - APM available as a link from within PWS

Creating the New Relic Service



New Relic

Manage and monitor your apps

ABOUT THIS SERVICE

New Relic is the all-in-one web app performance tool that lets you see performance from the end user experience, through servers, and down to the line of code.

[Documentation](#) | [Support](#)

COMPANY

New Relic

SERVICE PLANS

Standard free

PLAN FEATURES

- ✓ JVM Performance analyzer
- ✓ Database call response time & throughput
- ✓ Performance data API access

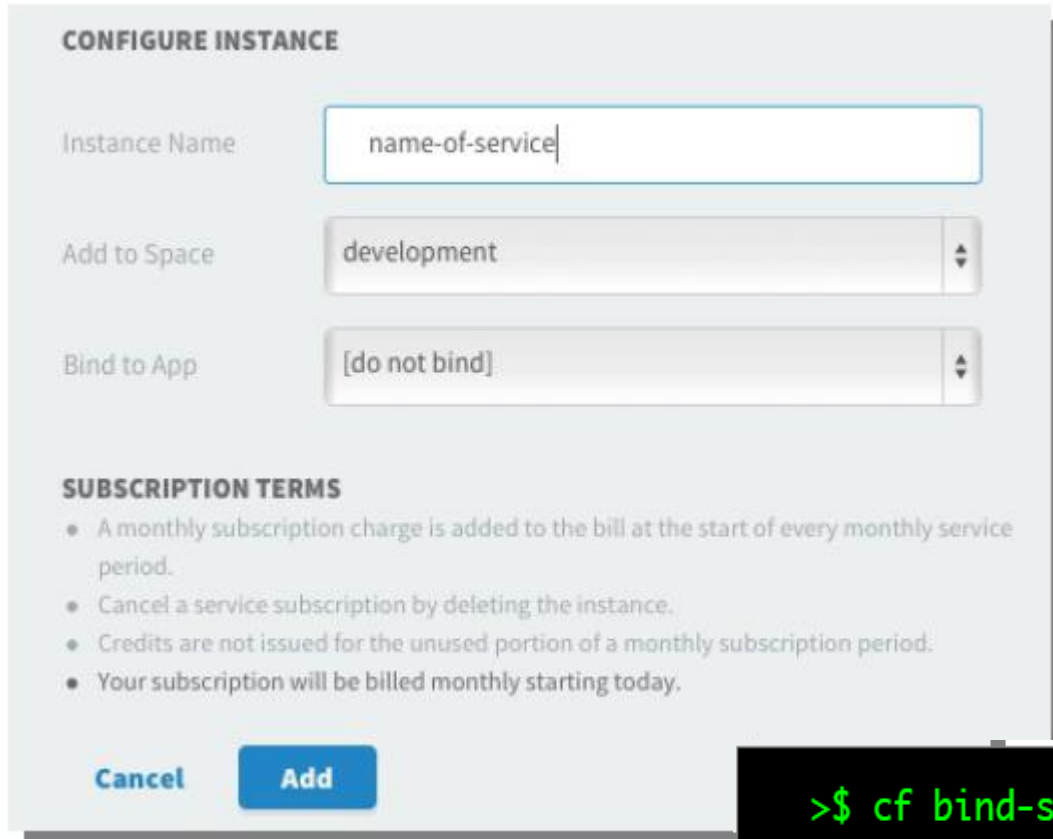
Select this plan

- Use App Manager Console

- Use **CF CLI**

```
>$ cf create-service newrelic standard apm
```


- Use CF CLI or App Manager Console:



CONFIGURE INSTANCE

Instance Name

Add to Space

Bind to App

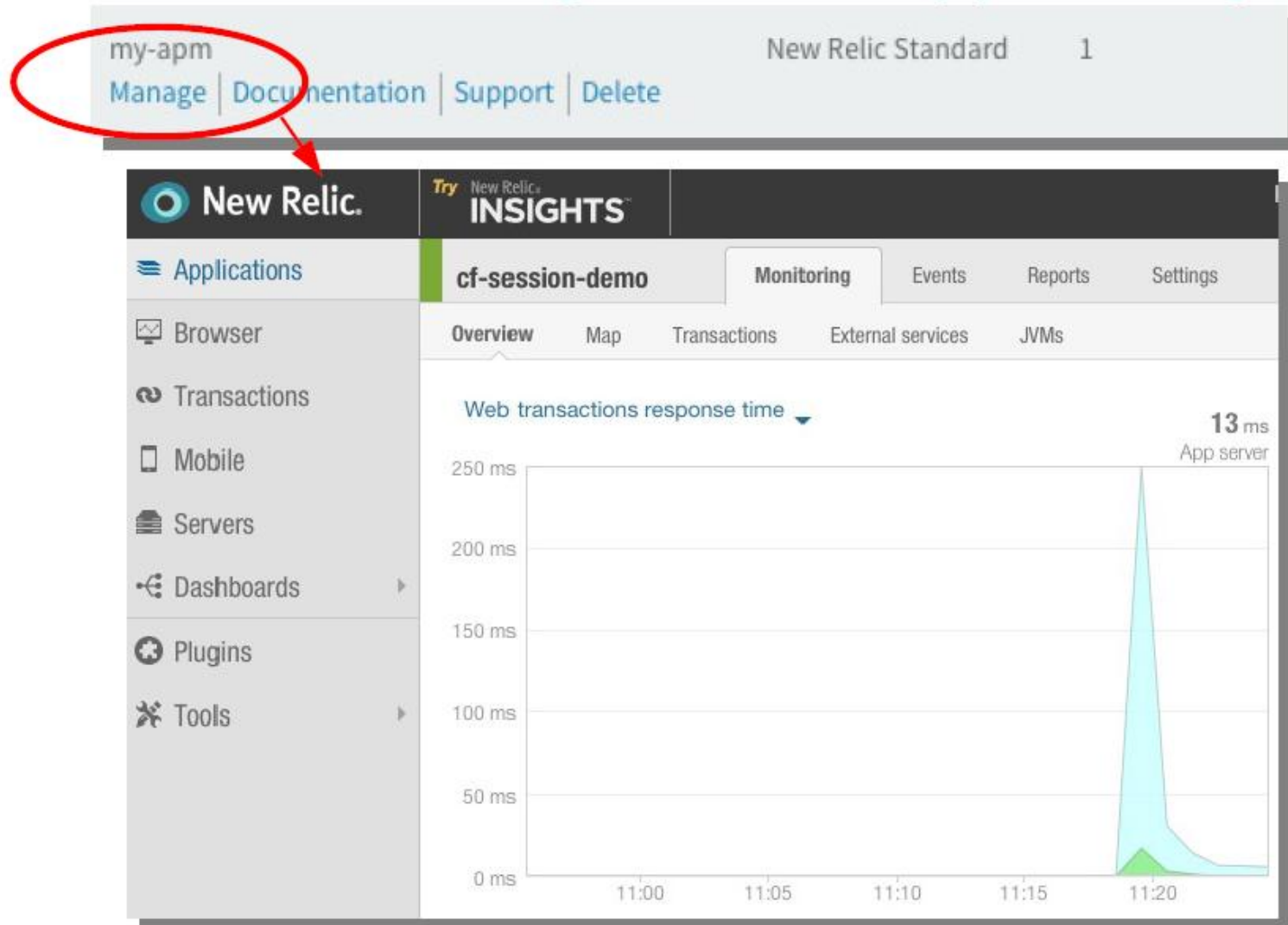
SUBSCRIPTION TERMS

- A monthly subscription charge is added to the bill at the start of every monthly service period.
- Cancel a service subscription by deleting the instance.
- Credits are not issued for the unused portion of a monthly subscription period.
- Your subscription will be billed monthly starting today.

[Cancel](#) [Add](#)

```
>$ cf bind-service some-app apm
```

Access via Manage Link in App Manger



The screenshot displays the New Relic App Manager interface. At the top, a header bar shows the application name 'my-apm' (circled in red), the plan 'New Relic Standard', and the number of instances '1'. Below this, a navigation bar includes links for 'Manage', 'Documentation', 'Support', and 'Delete'. The main interface is divided into a left sidebar with navigation options (Applications, Browser, Transactions, Mobile, Servers, Dashboards, Plugins, Tools) and a central content area. The content area shows the 'cf-session-demo' application with tabs for 'Monitoring', 'Events', 'Reports', and 'Settings'. Under 'Monitoring', there are sub-tabs for 'Overview', 'Map', 'Transactions', 'External services', and 'JVMs'. The 'Overview' tab is active, displaying a line graph titled 'Web transactions response time'. The graph shows a sharp spike in response time around 11:20, reaching a peak of 13 ms. The y-axis represents response time in milliseconds (ms), ranging from 0 to 250. The x-axis shows time in minutes and seconds, from 11:00 to 11:20. A legend indicates that the green area represents the 'App server'.

AUTOSCALING

Scaling Options

- CF allows horizontal scaling
 - Controlling the # of instances of an application running
 - All behind a common router (load balancer)
 - Controllable via the manifest, cf command line, or App Manager console
- All options require manual intervention

AutoScaling

- CF can allow applications to be automatically scaled
 - "AutoScaling"
- System load can be used as a trigger in place of manual interaction.



- Autoscaling Service
 - Must be installed by administrator
 - Not available in PWS

AutoScaling Service - Steps

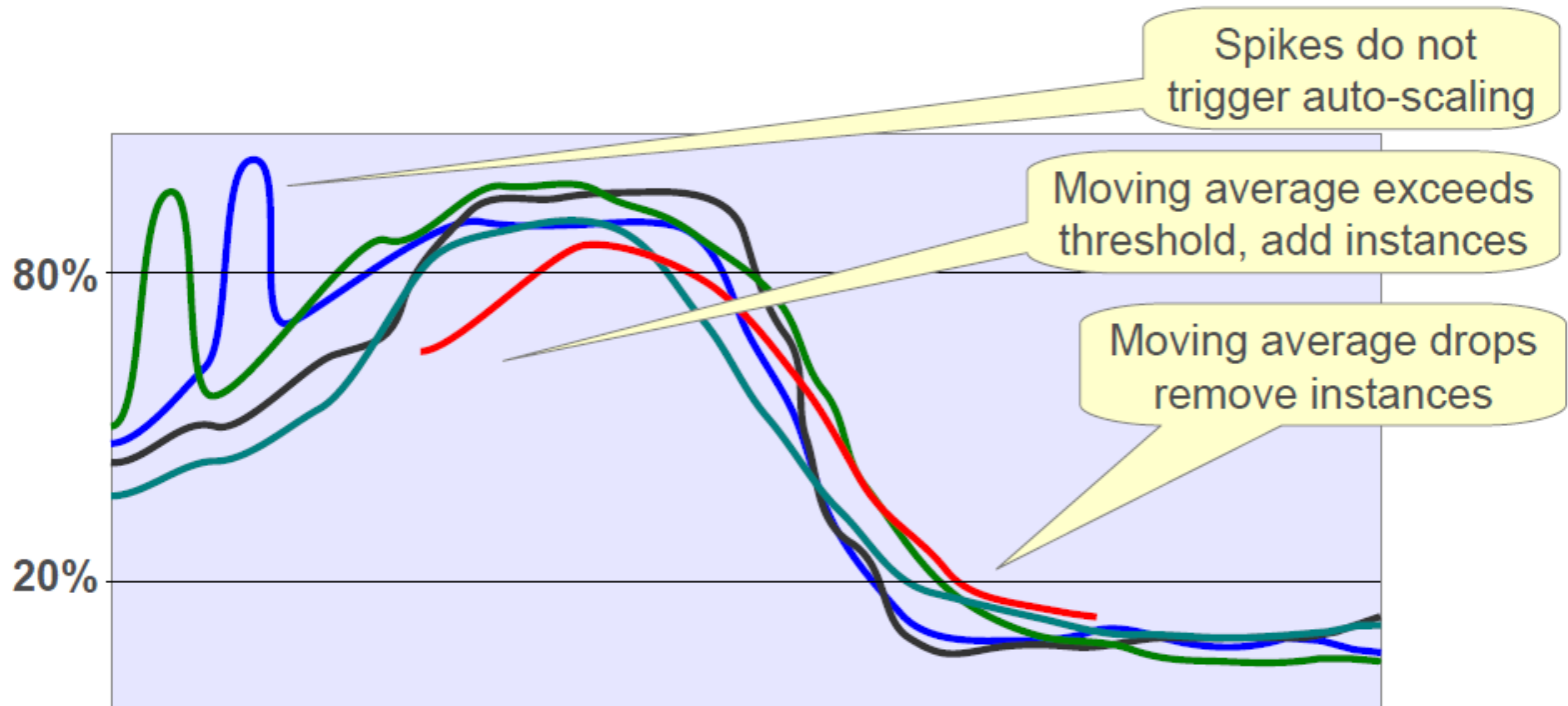
- Create the service
 - Select the desired plan
- Bind to Application
- Set desired scaling parameters
 - Add instance whenever high threshold is reached
 - Subtract instance whenever low threshold is reached

The screenshot displays the Pivotal™ Autoscale interface for an application named 'my-app'. It includes a status icon (two vertical bars) and a settings icon (wrench). The 'INSTANCES' section shows a minimum of 2 and a maximum of 5 instances. The 'CPU THRESHOLDS' section shows a low threshold at 20% and a high threshold at 80%. The 'LAST EVENT' section shows a scaling event from 1 to 2 instances on 09/11/14 at 23:15:56 UTC. The 'SCHEDULING' section shows 0 rules and the next event is 'No Upcoming Events'.

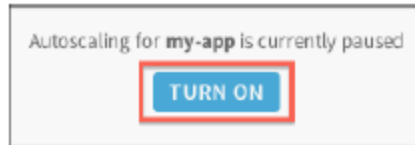
| PIVOTAL™ AUTOSCALE | | | |
|---|---|--------------------------|-----|
| my-app | | | |
| INSTANCES | | CPU THRESHOLDS | |
| min | 2 | low | 20% |
| max | 5 | high | 80% |
| LAST EVENT | | | |
| Scaled app from 1 to 2 instances 09/11/14 @ 23:15:56 UTC | | | |
| SCHEDULING | | | |
| 0 rules | | Next: No Upcoming Events | |

AutoScaling - Moving Average

- Scaling activity based on moving averages
 - Softens effect of temporary spikes



- Manual scaling disables AutoScaling
 - Re-enable:



AutoScaling - Scheduling

- Autoscaling events can be scheduled
- Changes auto scaling behaviour on the given date / time.
- May be single event or recurring

SCHEDULING: MY-APP SERVER TIME: 09/12/14 @ 21:27:22 UTC ✕

✚ New UNSAVED

11/21/2014 / 02:00 ✕

5 to 10 instances

Mon, Fri / 04:00 || ✕

10 to 20 instances

on at ⬆ ⬆

*All times are UTC

repeats every

☐ S ☐ M ☐ T ☒ W ☐ T ☐ F ☐ S

min

low

max

high

ADD

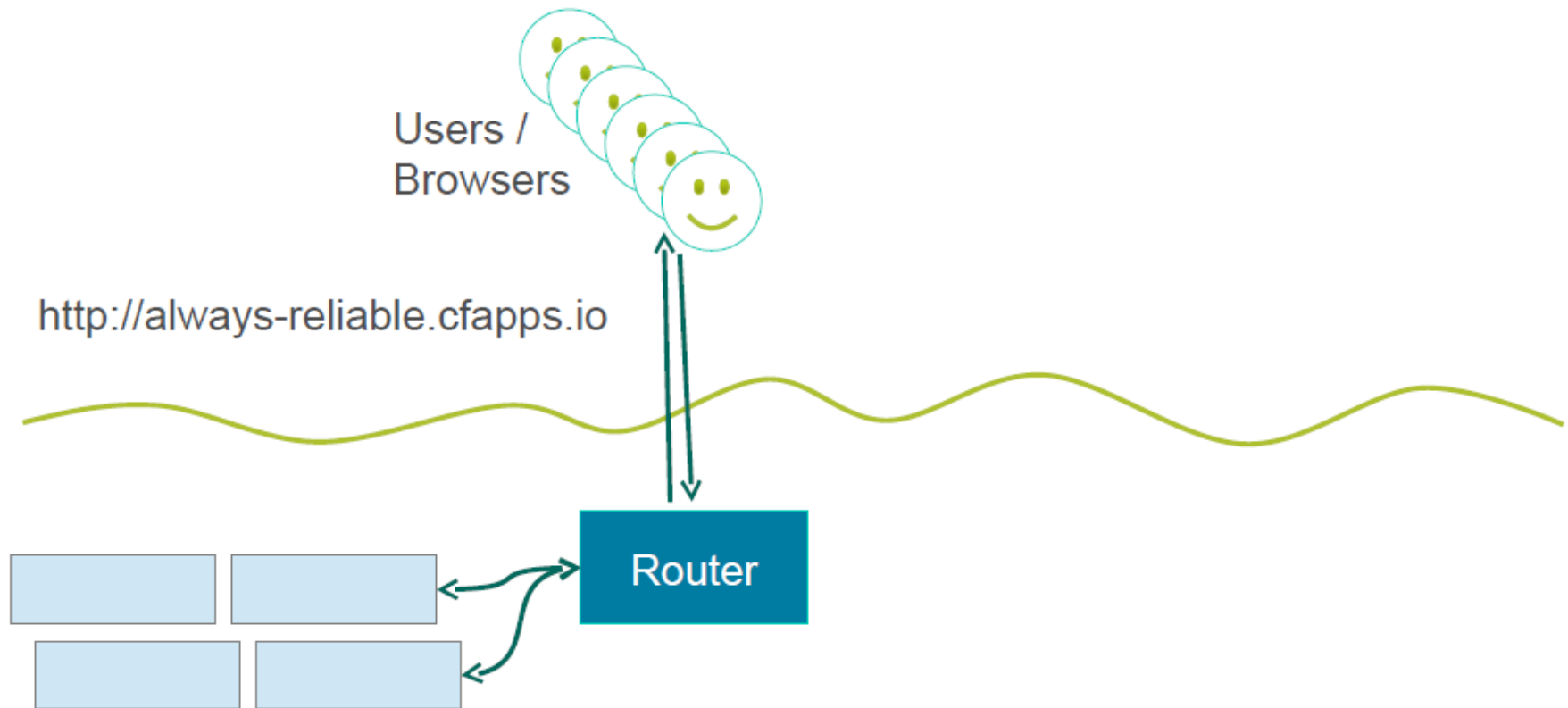
ZERO-DOWN TIME DEPLOYMENT

Blue Green Deployments

- **cf** push causes CF to stop old instances, then start new
 - Bad news if you are a user
- Blue/Green Deployment eliminates user downtime
 - Also known as "zero-downtime" or "A/B" Deployment
 - Avoids "Site Temporarily Down for Maintenance"
- How it works:
 - Run 2 versions of an application (new /old)
 - Not merely multiple instances
 - Alter routes for applications to transfer traffic
 - **Note:** Users can still experience session loss.

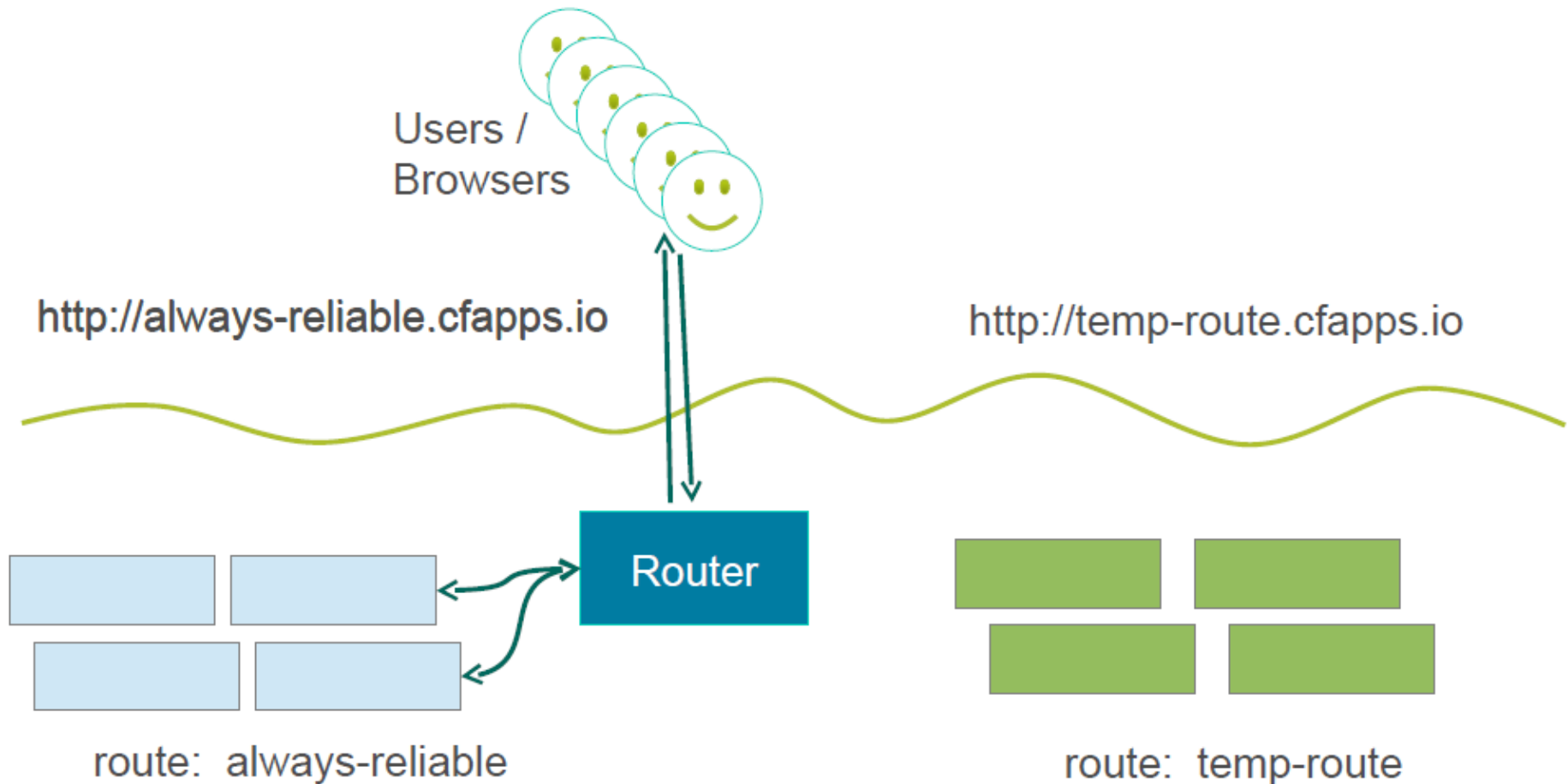
Blue Green Deployment – Existing App

cf push blue -p app.war -n always-reliable -i 4



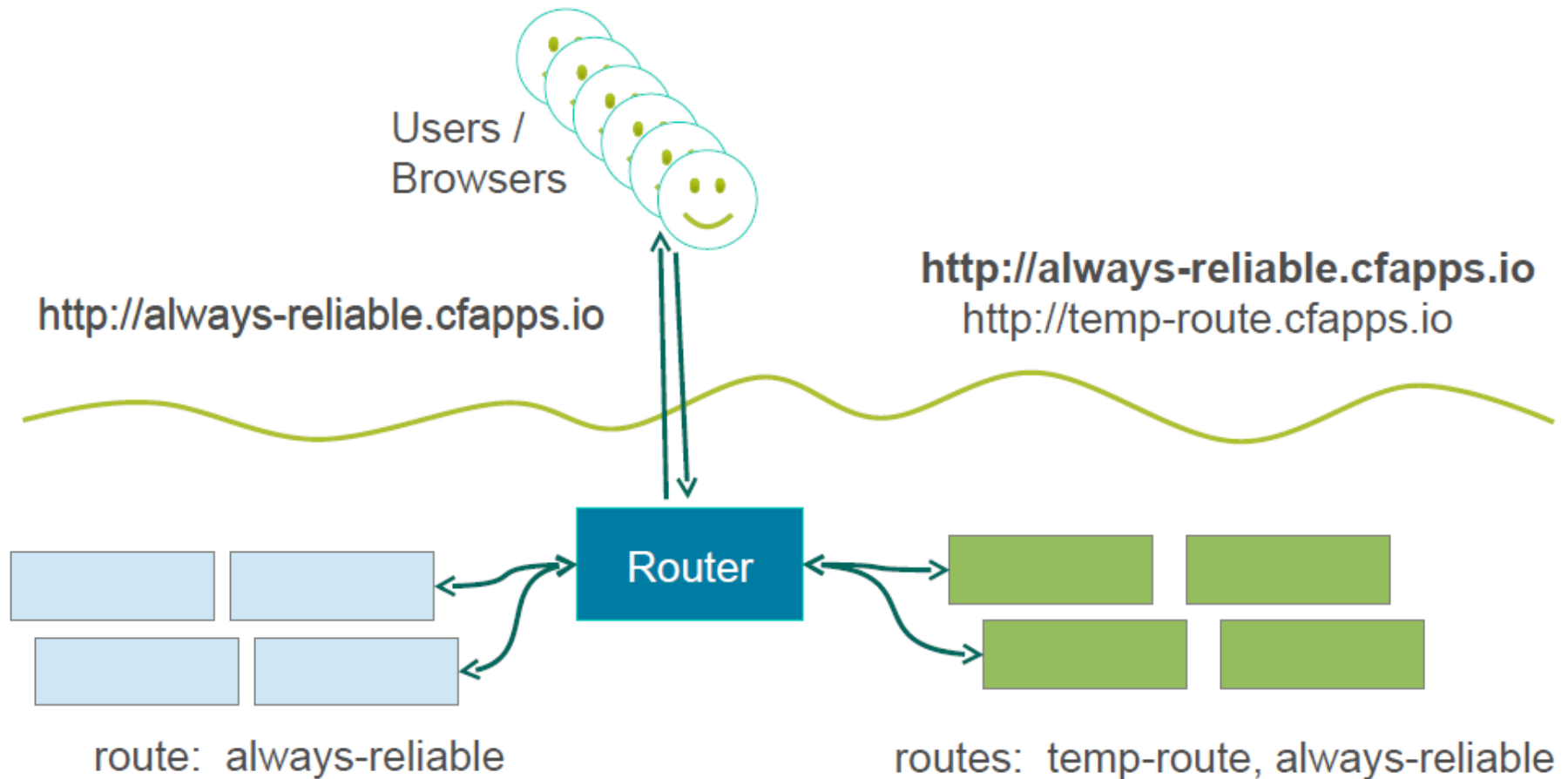
Blue Green Deployment – New App

```
cf push green -p app.war -n temp-route -i 4
```



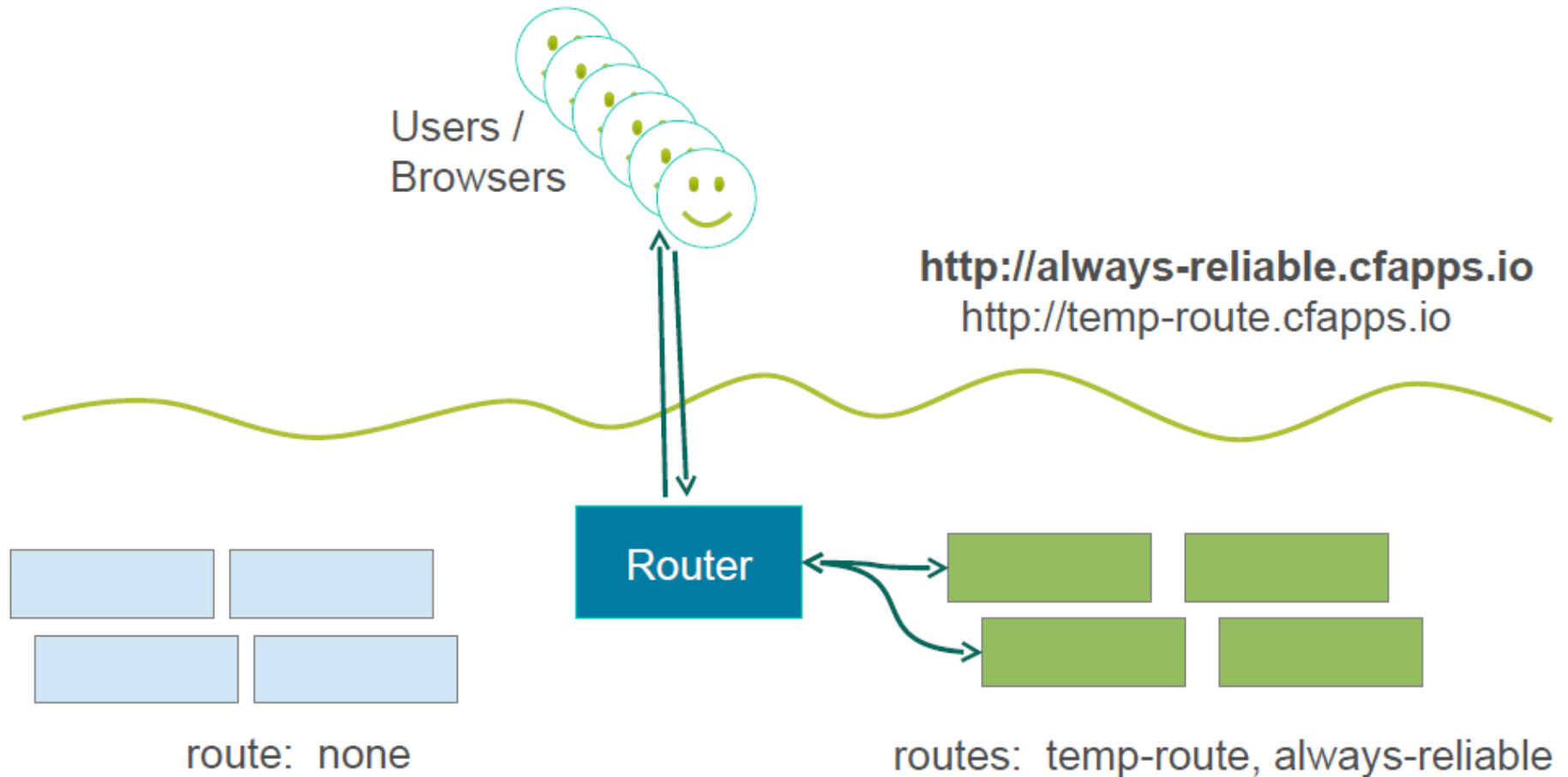
Blue Green Deployment – Duplicate Route

`cf map-route green cfapps.io -n always-reliable`



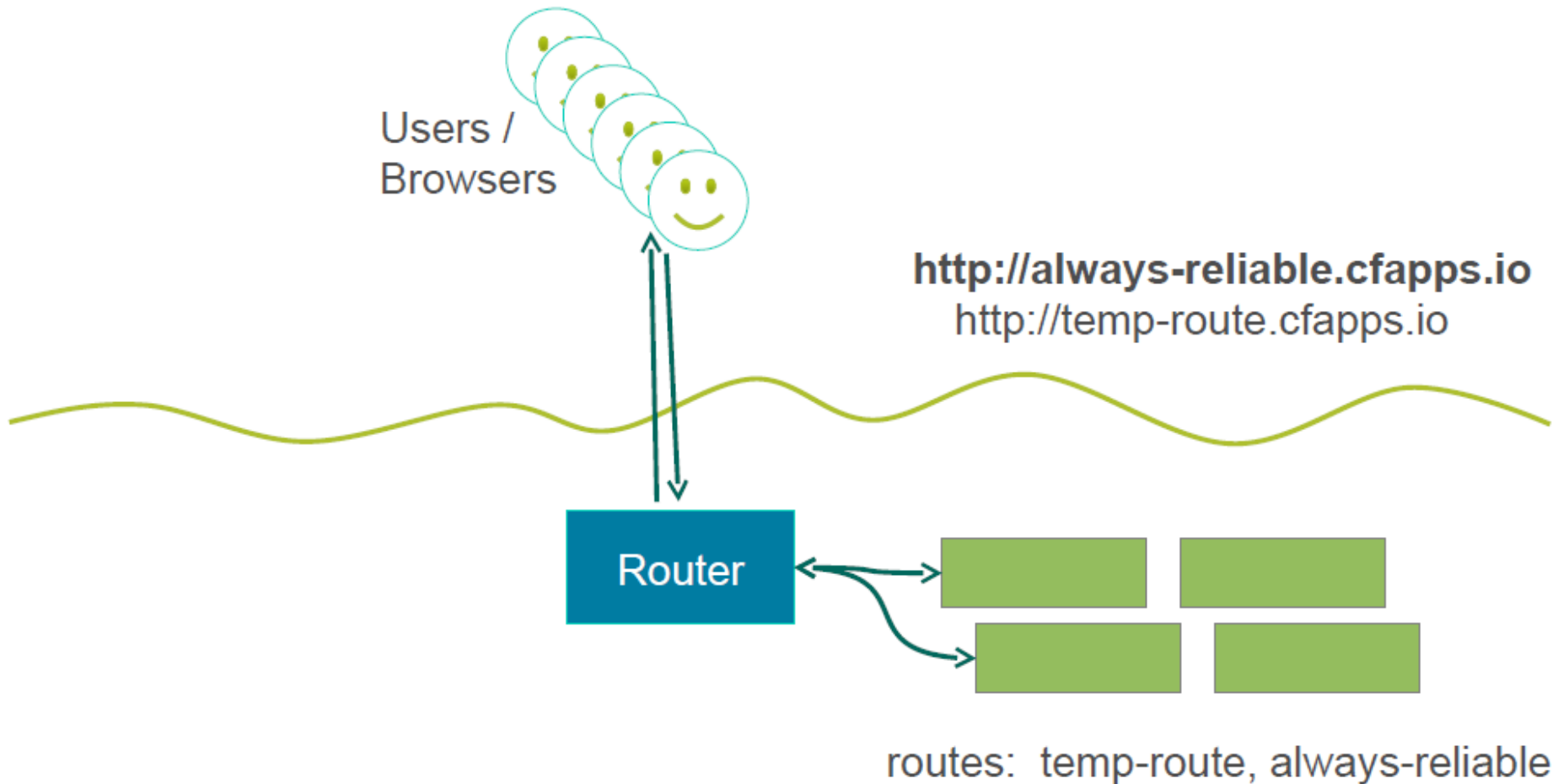
Blue Green Deployment – Disconnect Blue

cf unmap-route blue cfapps.io -n always-reliable



Blue Green Deployment – Remove Blue

cf delete blue

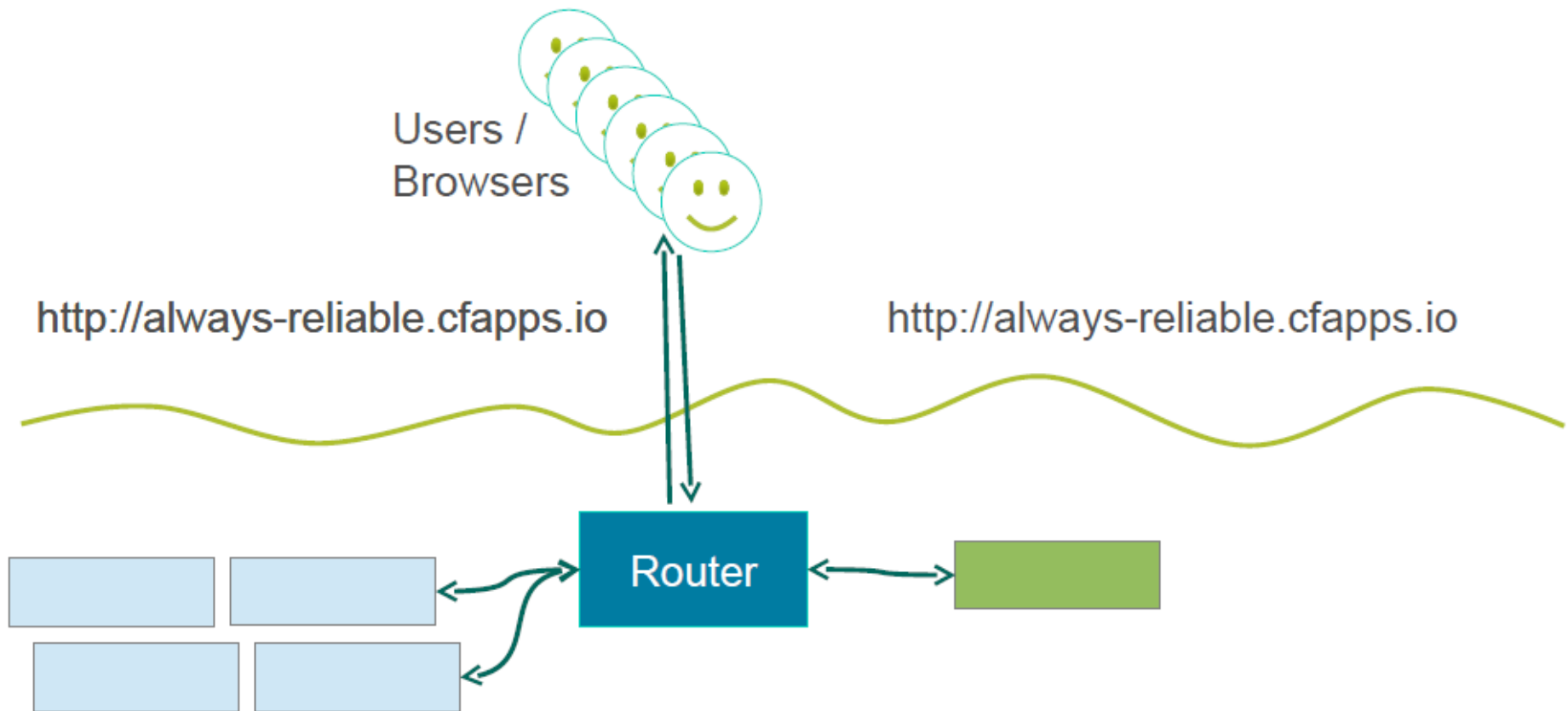


Canary Deployments

- variation on the Blue/Green deployment
 - "Canary in a coal mine"
- 1. Start with many 'blue' instances
- 2. Start a single 'green' instance, route traffic to both
 - Green instance is the 'Canary'
- 3. Watch the Canary
 - If it behaves, scale 'green' up /scale 'blue' down.
- 4. Continue monitoring and scaling until zero blue instances.

Canary Deployment – Push The Canary

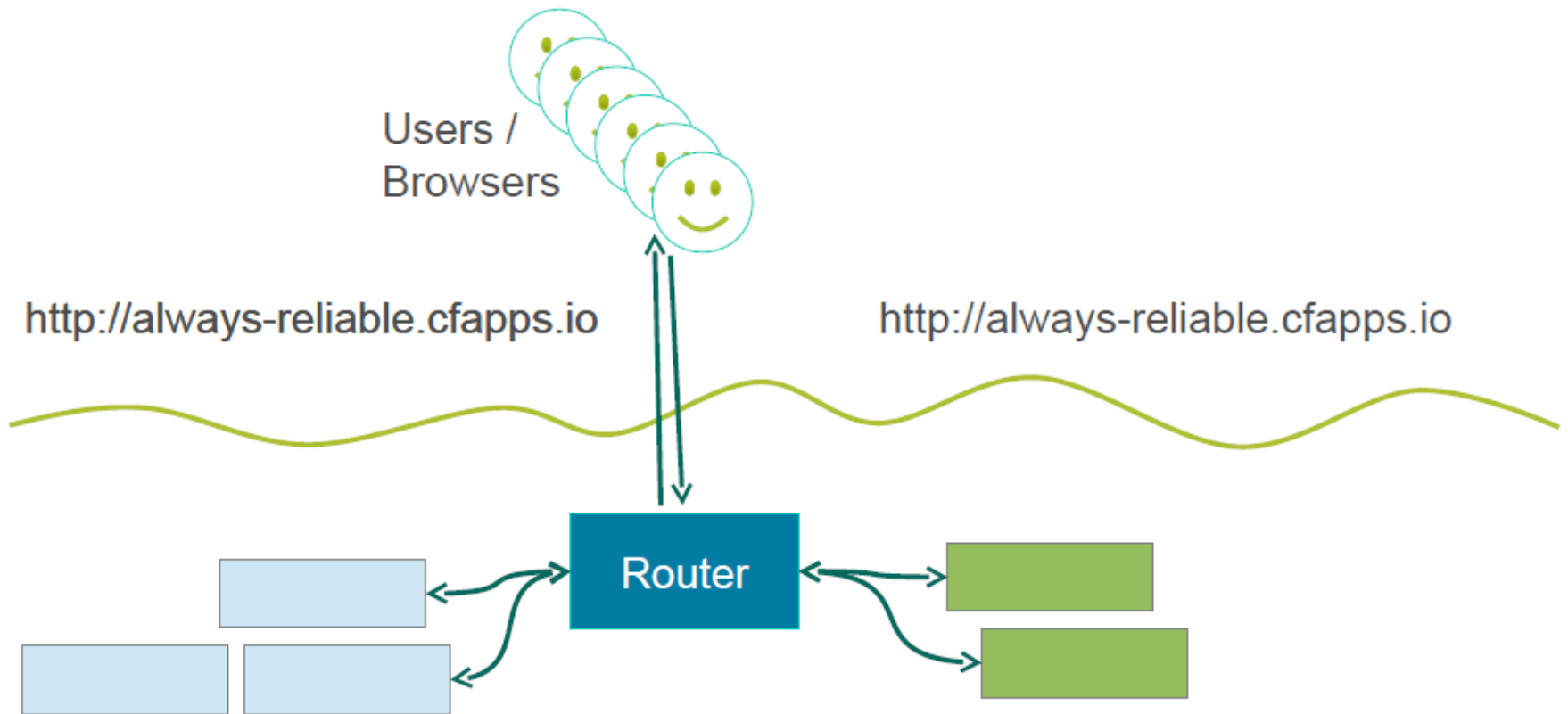
```
cf push green -p app.war -n always-reliable -i 1
```



Canary Deployment – Scale Traffic

cf scale green -i 2

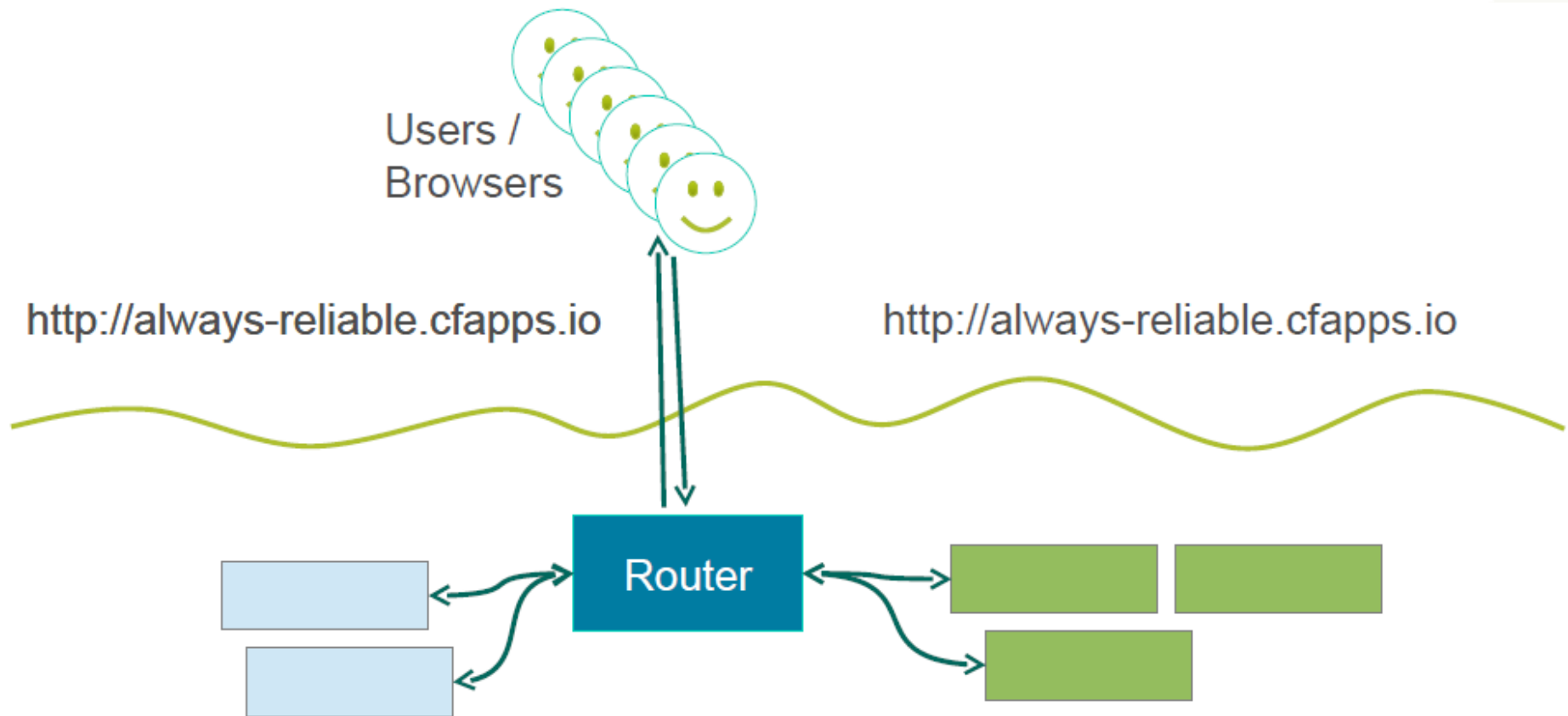
cf scale blue -i 3



Canary Deployment – Scale Traffic

cf scale green -i 3

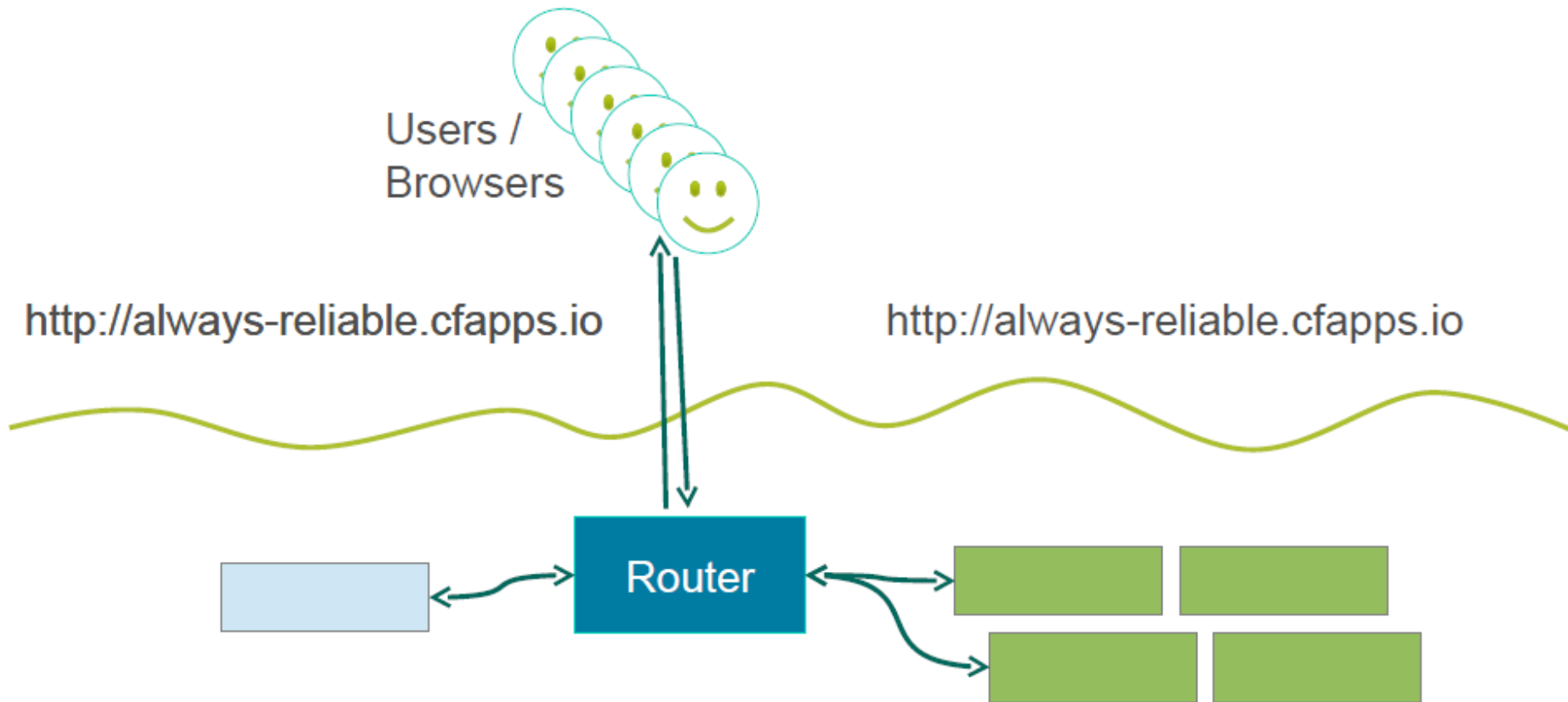
cf scale blue -i 2



Canary Deployment – Scale Traffic

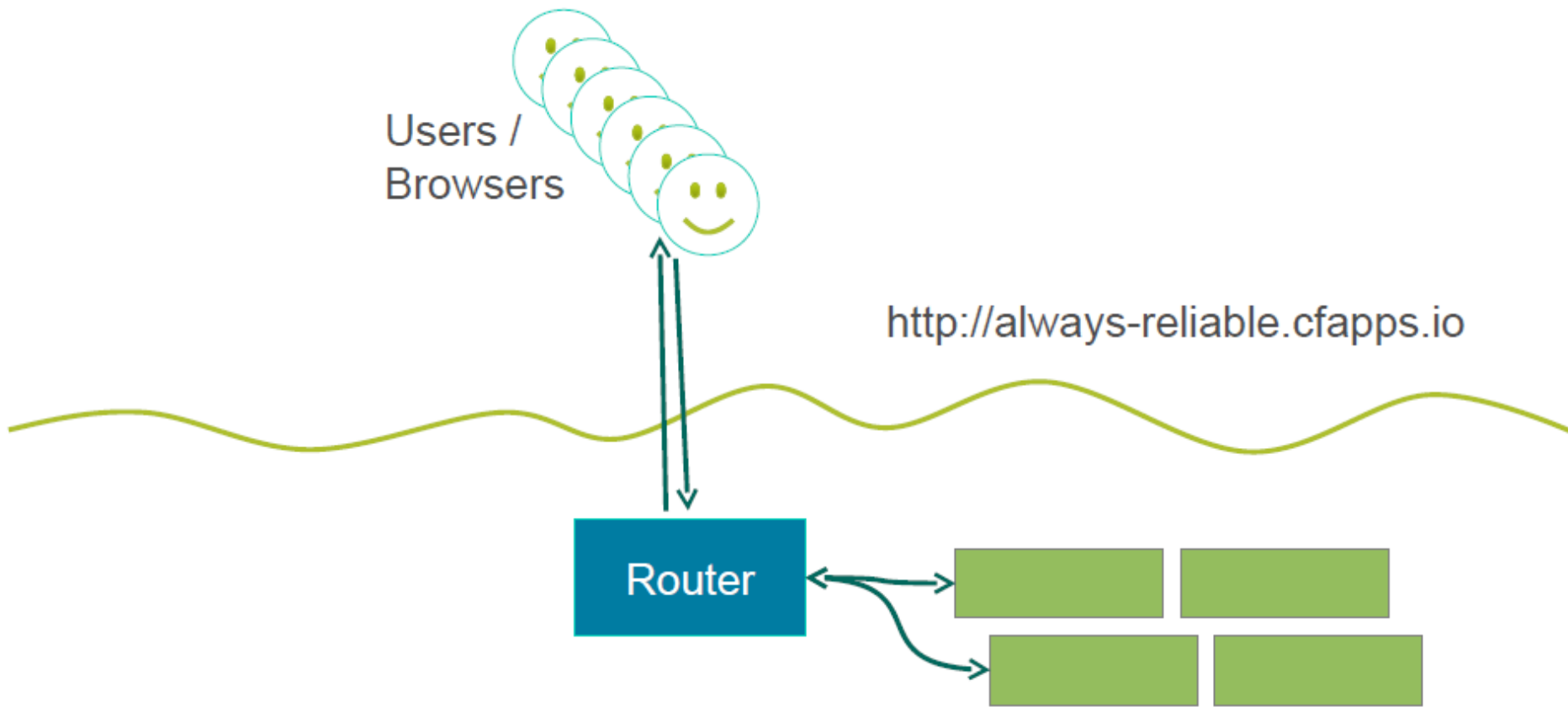
`cf scale green -i 4`

`cf scale blue -i 1`



Canary Deployment – Scale Traffic

cf delete blue



Summary

- How to integrate with third-party log manager
- How to integrate with APM services
- How to employ App Autoscaling
- How to deploy with zerotime

Recap

3rd party log

autoscale

ops manager

blue

green

zerodown-time

canary

People matter, results count.



About Capgemini

With more than 130,000 people in 44 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2012 global revenues of EUR 10.3 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.



www.capgemini.com

