Pivotal

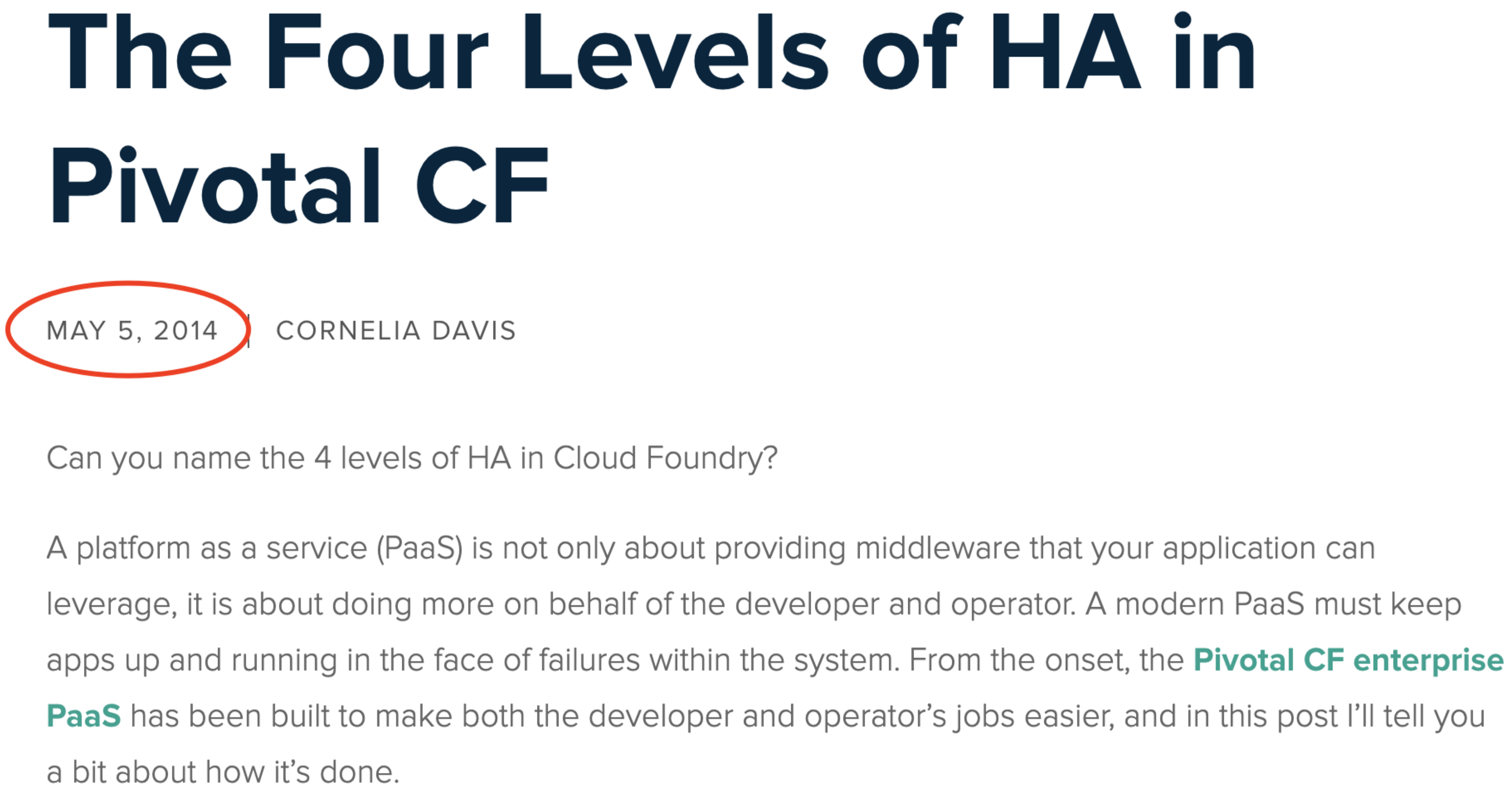
PCF Dev Enablement
- HA, Scaling and Observability

Derrick Chua
Senior Platform Architect
tchua@pivotal.io

July 2019

Pre-requisites

- Jmeter – Download from https://www-us.apache.org/dist//jmeter/binaries/apache-jmeter-5.1.1.zip



The Four Levels of HA in Pivotal CF

MAY 5, 2014 | CORNELIA DAVIS

Can you name the 4 levels of HA in Cloud Foundry?

A platform as a service (PaaS) is not only about providing middleware that your application can leverage, it is about doing more on behalf of the developer and operator. A modern PaaS must keep apps up and running in the face of failures within the system. From the onset, the **Pivotal CF enterprise PaaS** has been built to make both the developer and operator's jobs easier, and in this post I'll tell you a bit about how it's done.

DBS

Faster time to market

6 months

to

< 4 weeks

for

major releases

Increased release cadence

10X

improvement in lead time

(32 hours -> 3 hours)

6X

improvement in effort needed

(12 hours -> 2 hours)

Self provisioning infrastructure

1 week

To

Intraday

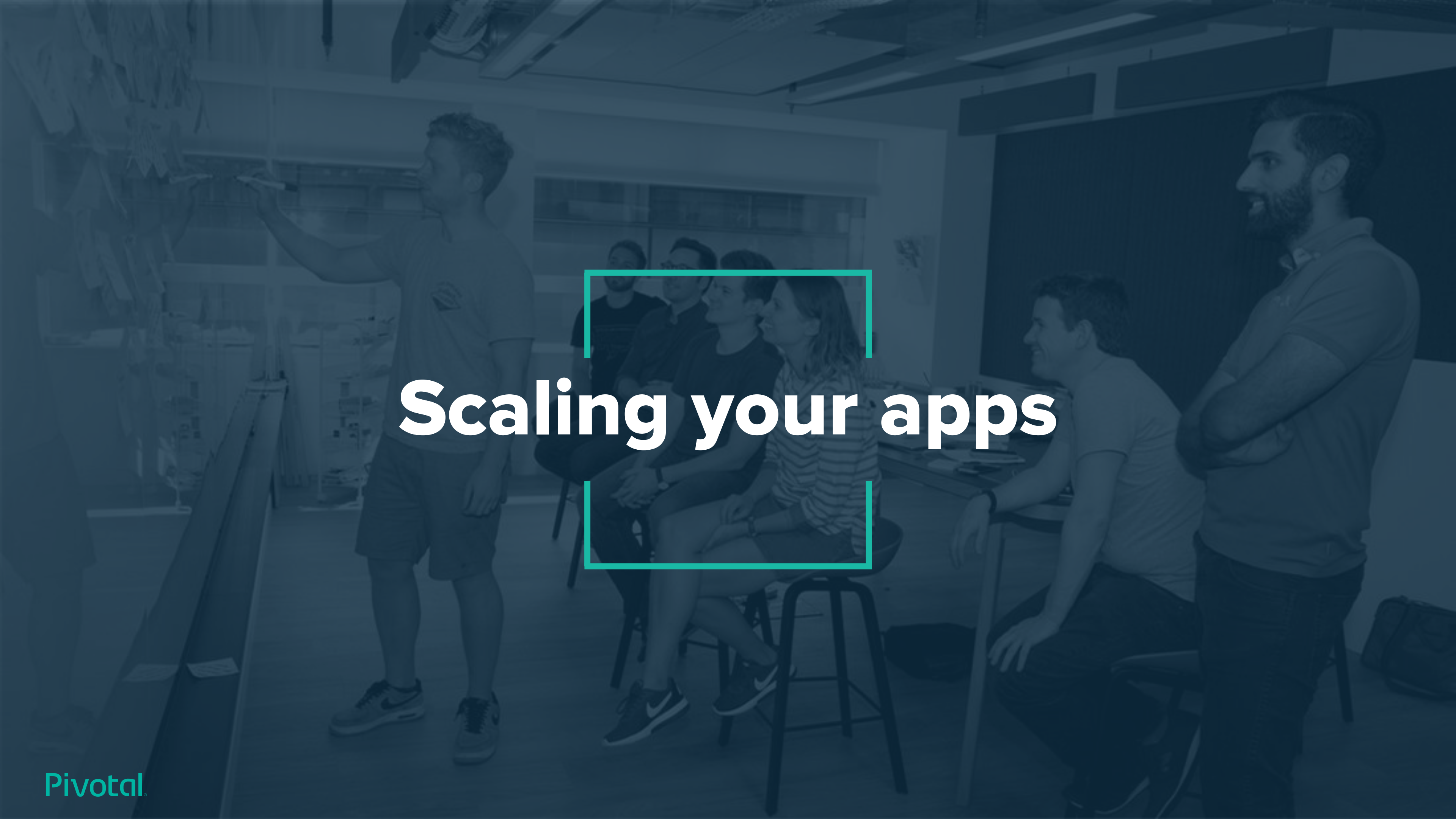
Stability and maintainability

Zero minutes downtime in Production

(since 2016)

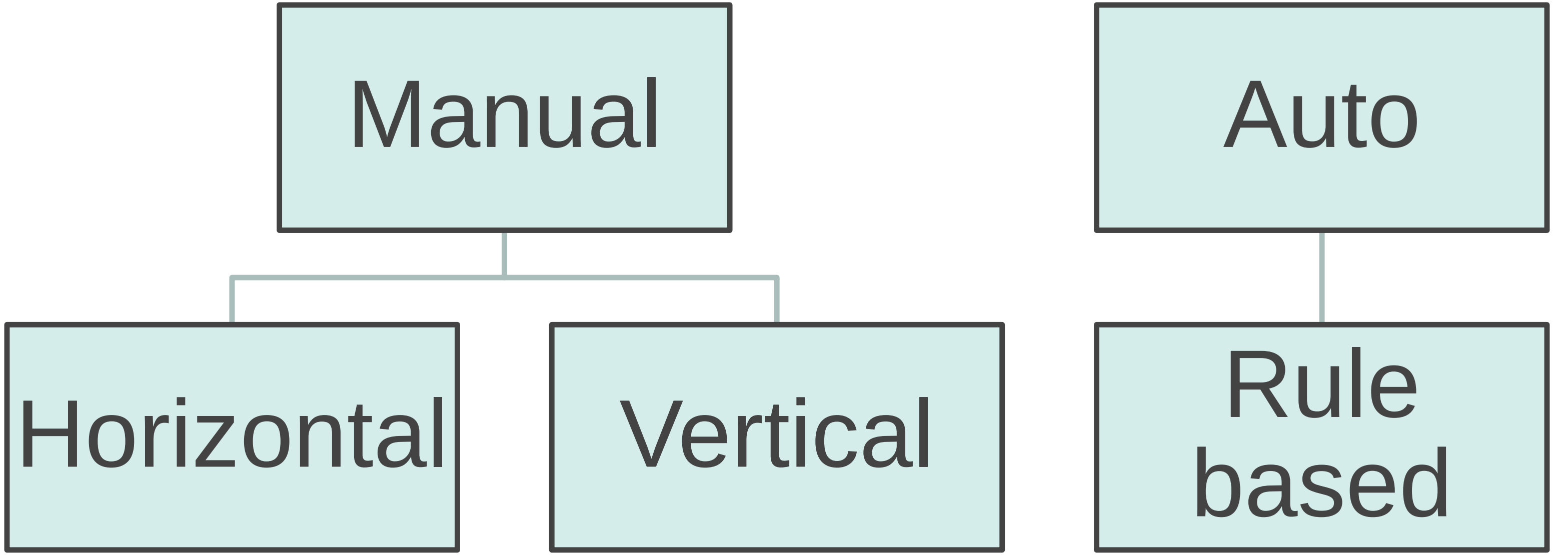
WHITEBOARD TIME!
makeameme.org

ENOUGH TALK
SHOW ME THE DEMO!
memegenerator.net



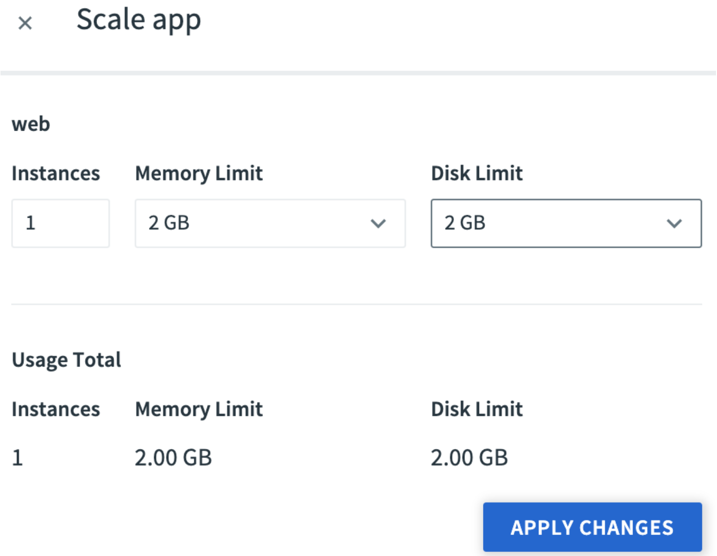
Scaling your apps

Scaling options



Hands-On: Step 1

- Scale employee-api microservice vertically to 2GB memory and 2GB disk limit, **using the Apps Manager UI**. Observe the changes and test that it works.
- Scale back down to original 1GB memory and 1GB disk limit, **using cf cli**.
    - cf scale employee-api -m 2G -k 2G



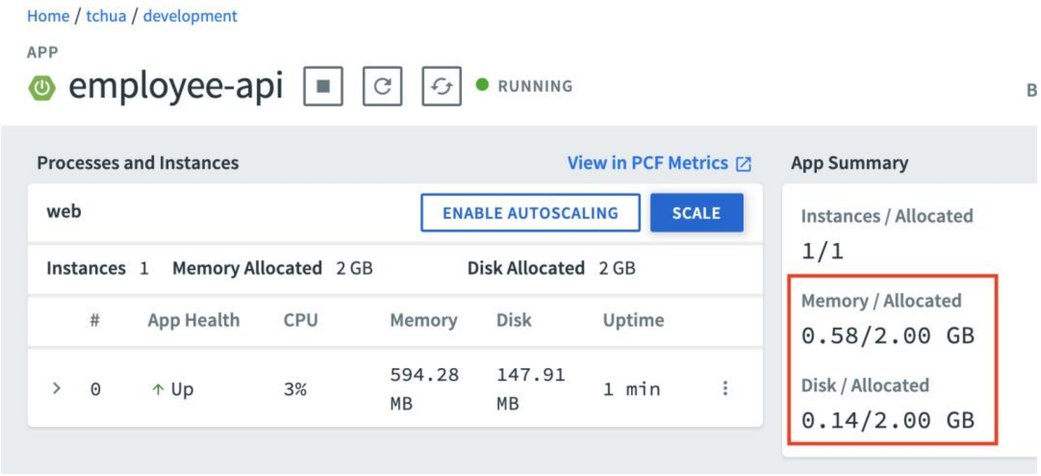
× Scale app

web

Instances	Memory Limit	Disk Limit
1	2 GB	2 GB

Usage Total

Instances	Memory Limit	Disk Limit
1	2.00 GB	2.00 GB



Home / tchua / development

APP

employee-api RUNNING

web

Instances 1 Memory Allocated 2 GB Disk Allocated 2 GB

#	App Health	CPU	Memory	Disk	Uptime
0	↑ Up	3%	594.28 MB	147.91 MB	1 min

App Summary

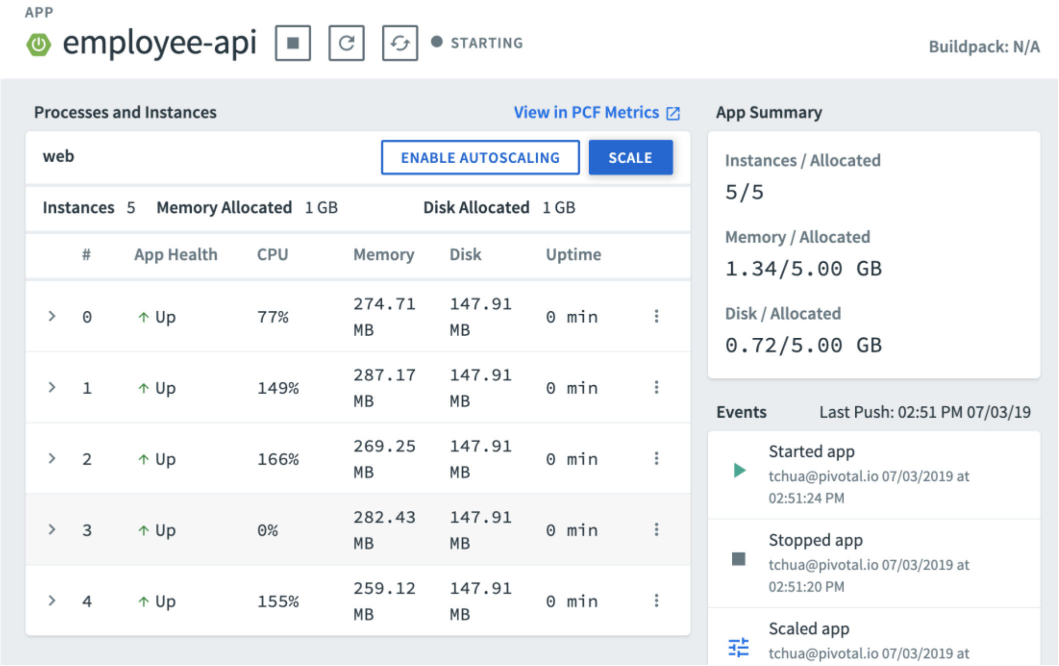
Instances / Allocated
1/1

Memory / Allocated
0.58/2.00 GB

Disk / Allocated
0.14/2.00 GB

Hands-On: Step 2

- Scale employee-api microservice horizontally to 5 instances using either the Apps Manager or the cf cli. Observe the changes and test that it works.
- Scale it back down to 1 instance.



Buildpack: N/A

Processes and Instances

web

Instances 5 Memory Allocated 1 GB Disk Allocated 1 GB

#	App Health	CPU	Memory	Disk	Uptime
0	↑ Up	77%	274.71 MB	147.91 MB	0 min
1	↑ Up	149%	287.17 MB	147.91 MB	0 min
2	↑ Up	166%	269.25 MB	147.91 MB	0 min
3	↑ Up	0%	282.43 MB	147.91 MB	0 min
4	↑ Up	155%	259.12 MB	147.91 MB	0 min

App Summary

Instances / Allocated
5/5

Memory / Allocated
1.34/5.00 GB

Disk / Allocated
0.72/5.00 GB

Events Last Push: 02:51 PM 07/03/19

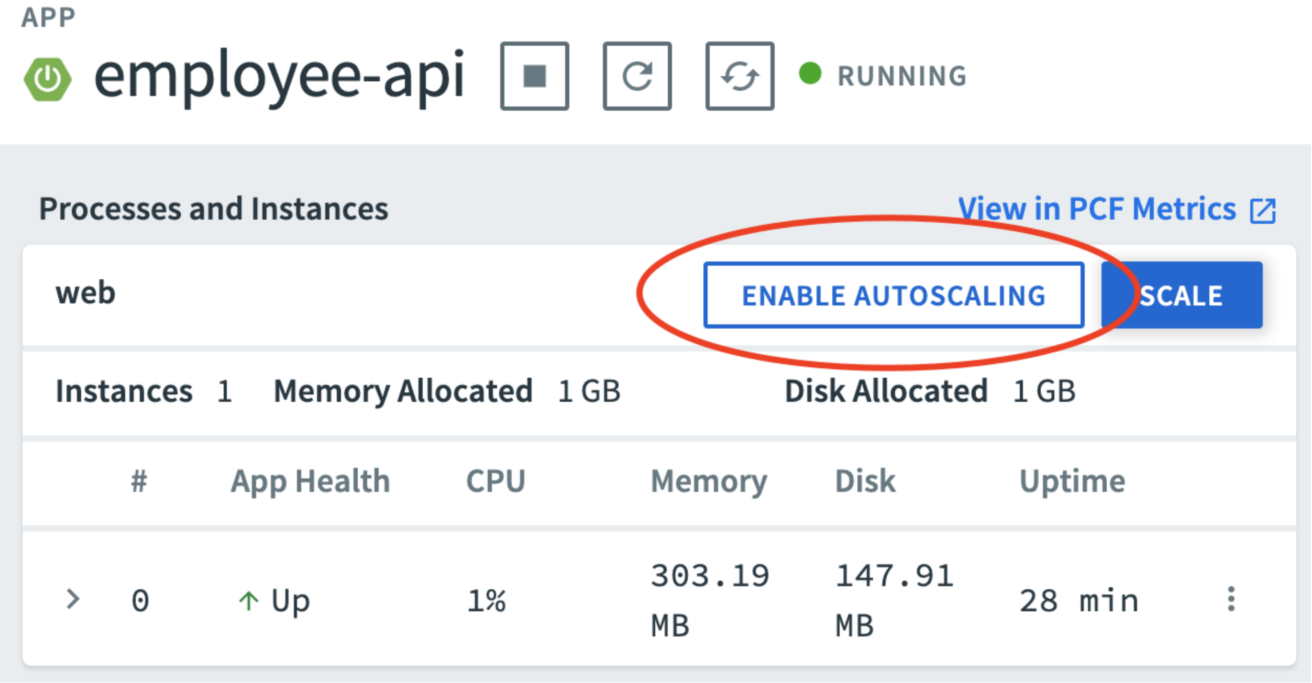
Started app
tchua@pivotal.io 07/03/2019 at 02:51:24 PM

Stopped app
tchua@pivotal.io 07/03/2019 at 02:51:20 PM

Scaled app
tchua@pivotal.io 07/03/2019 at

Hands-On: Step 3

- Enable autoscaling



Processes and Instances

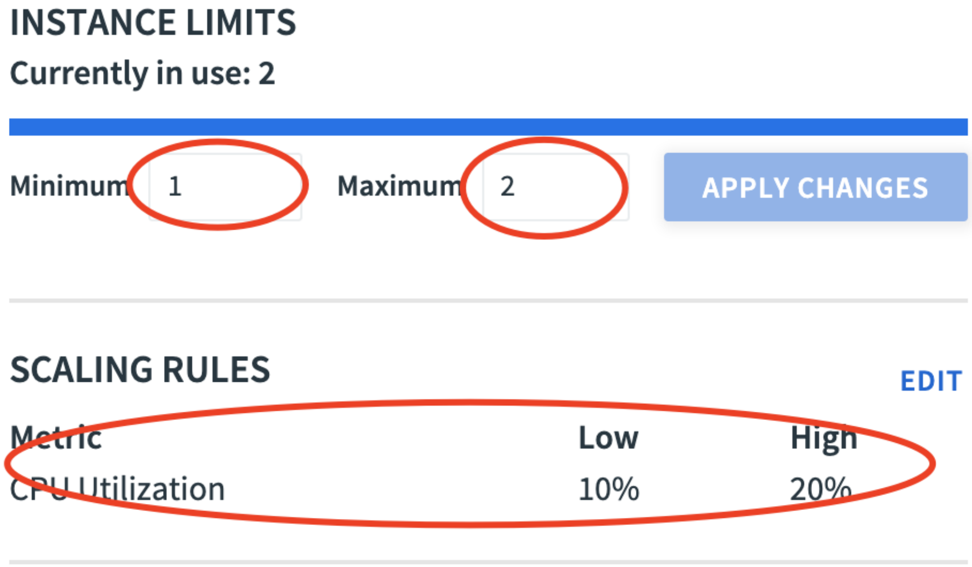
web

Instances 1 Memory Allocated 1 GB Disk Allocated 1 GB

	#	App Health	CPU	Memory	Disk	Uptime
>	0	↑ Up	1%	303.19 MB	147.91 MB	28 min

Hands-On: Step 4

- Select “Manage Autoscaling”
- Set the minimum and maximum instances to 1 and 2 respectively, and add a CPU utilization rule that scales up on CPU going beyond 20% and scaling down on CPU dropping to less than 10%



Currently in use: 2

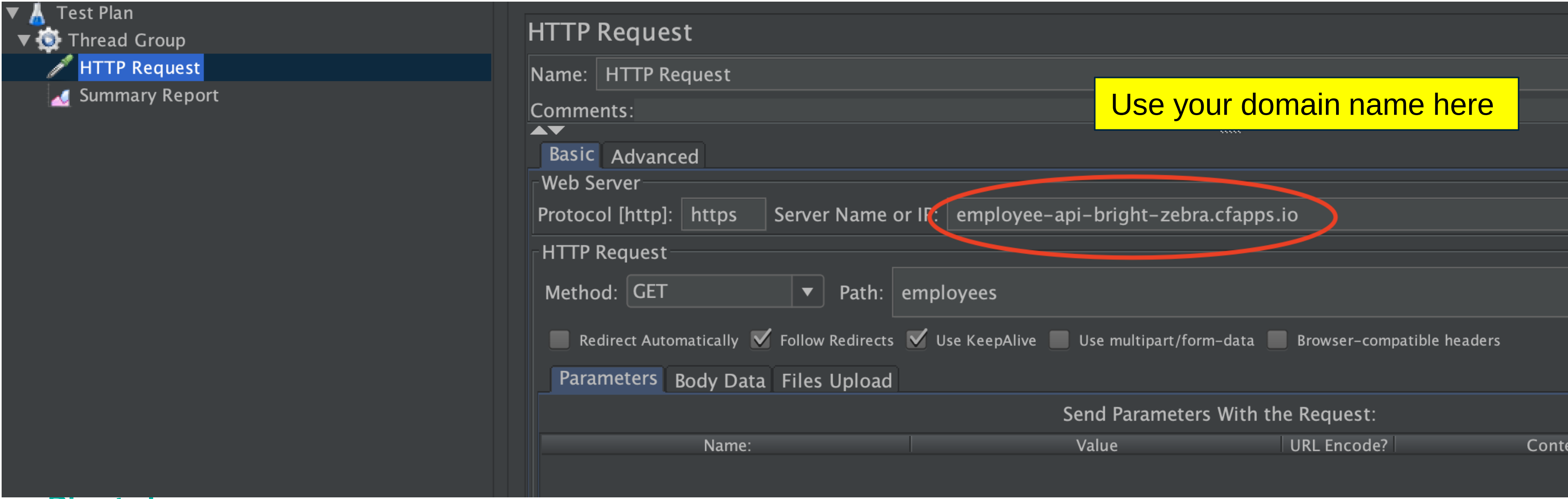
Minimum 1

Maximum 2

Metric	Low	High
CPU Utilization	10%	20%

Hands-On: Step 5

- Start JMeter and open autoscale.jmx in the scripts folder
- Update the HTTP Request sampler to use the domain name mapped to your employee-api microservice app
- Run the Thread Group



HTTP Request

Name: HTTP Request

Comments:

Use your domain name here

Web Server

Protocol [http]: https Server Name or IP: employee-api-bright-zebra.cfapps.io

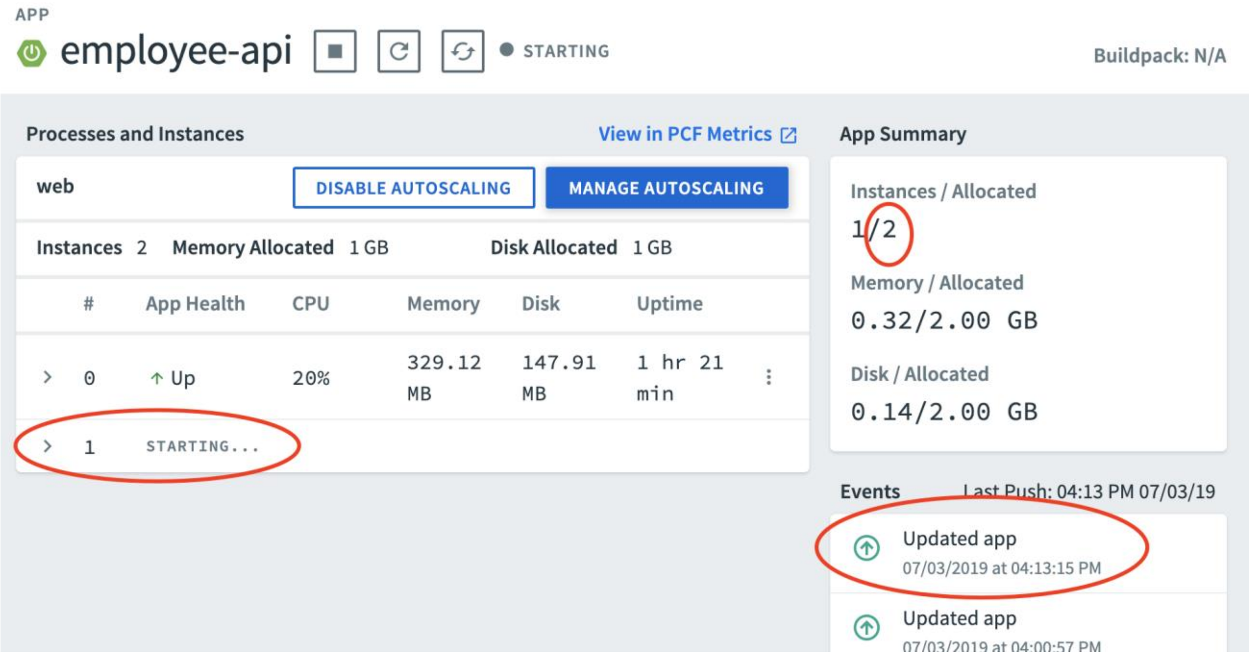
HTTP Request

Method: GET Path: employees

□ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive □ Use multipart/form-data □ Browser-compatible headers

Hands-On: Step 6

- Observe your application in Apps Manager. You should see the CPU usage increase to the point where another instance is automatically started after a few minutes.
- Stop the JMeter Thread Group. After a few minutes, you should observe that the additional instance of your app has been automatically terminated.



Processes and Instances

web

Instances 2 Memory Allocated 1 GB Disk Allocated 1 GB

#	App Health	CPU	Memory	Disk	Uptime
0	↑ Up	20%	329.12 MB	147.91 MB	1 hr 21 min
1	STARTING...				

App Summary

Instances / Allocated
1/2

Memory / Allocated
0.32/2.00 GB

Disk / Allocated
0.14/2.00 GB

Events Last Push: 04:13 PM 07/03/19

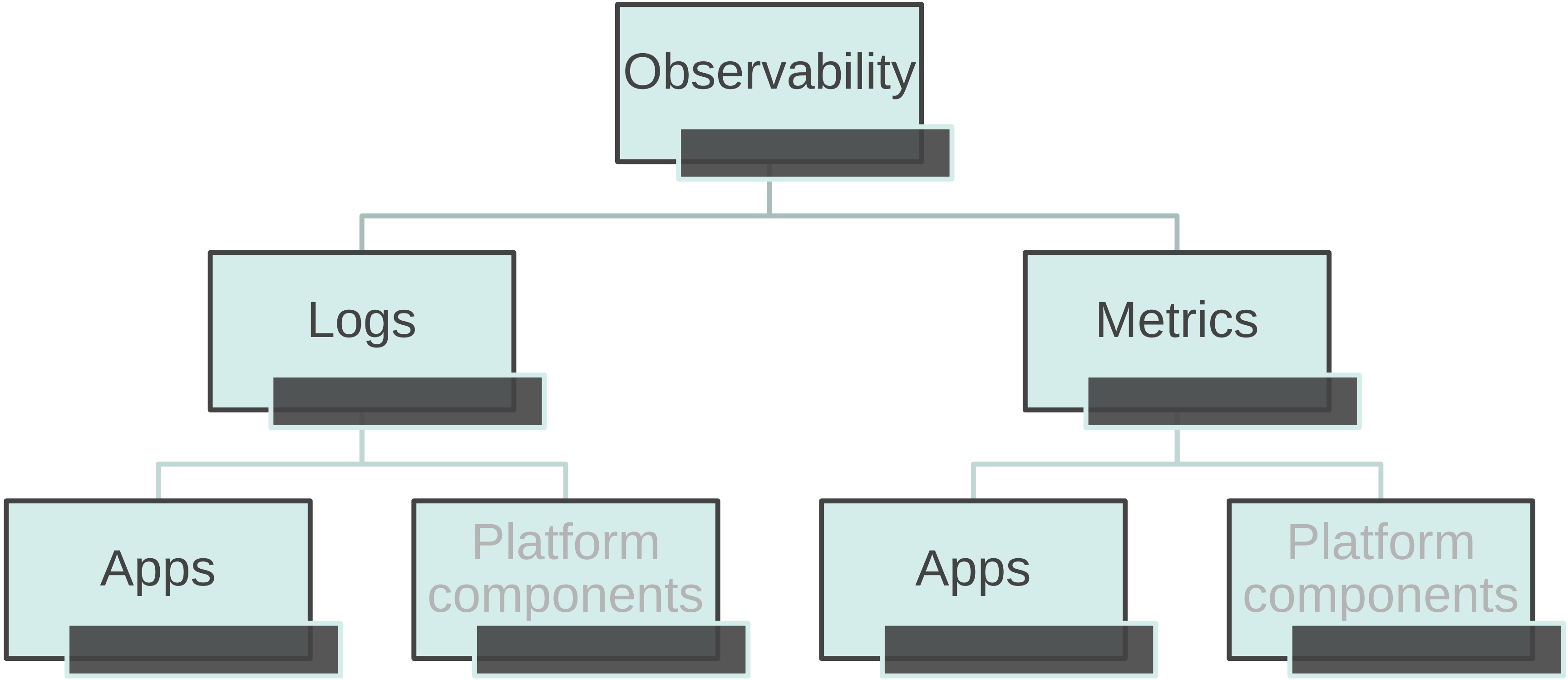
Updated app
07/03/2019 at 04:13:15 PM

Updated app
07/03/2019 at 04:00:57 PM

Think about...

- When do I use vertical vs horizontal scaling?
- Why does autoscaling not support vertical scaling?
- How do I configure and manage autoscaling through the cf cli?
- How can I schedule scaling to cater for known periods of high traffic?

Observability



Observability

Logs

Metrics

Apps

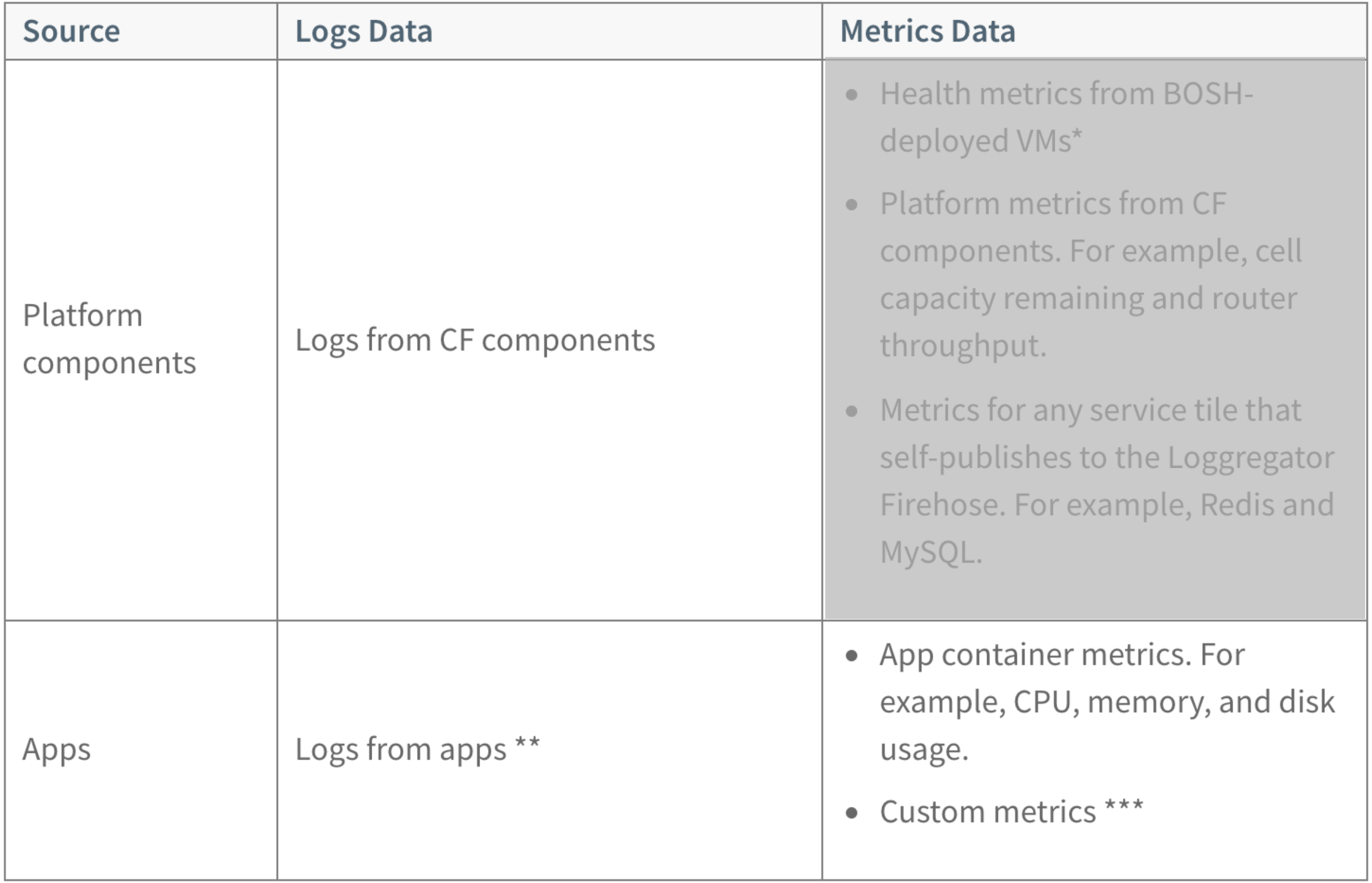
Platform components

Apps

Platform components

Cloud Foundry **aggregates logs for all instances** of your apps as well as for requests made to your apps through internal components of Cloud Foundry.

Sources of logs in PCF

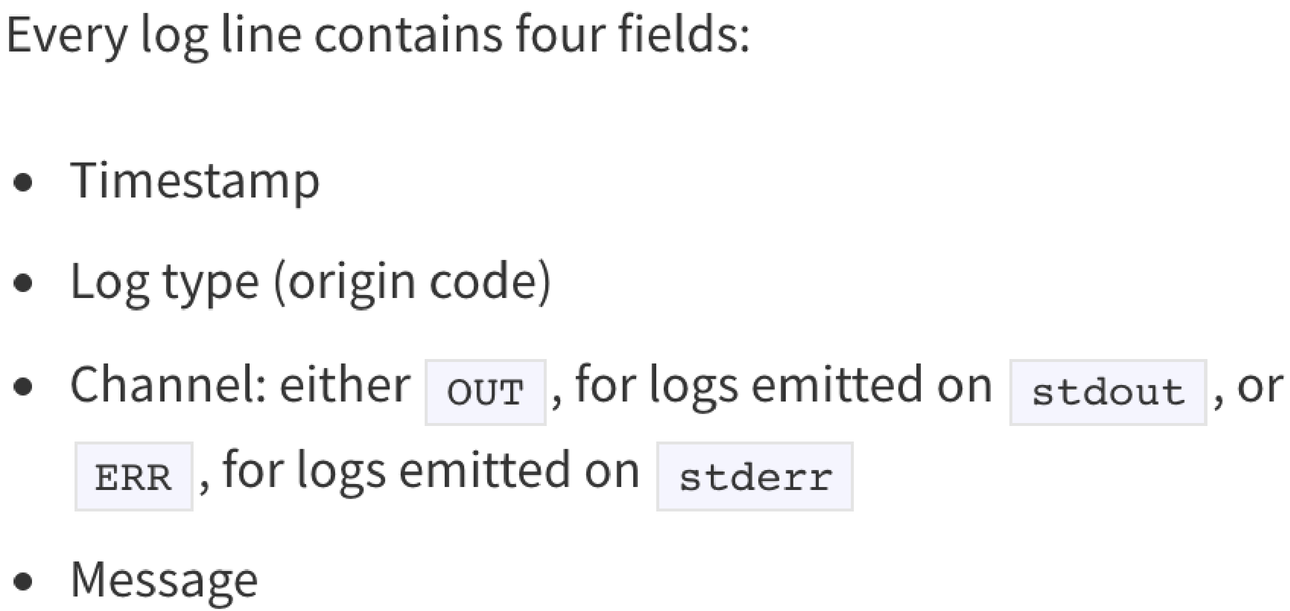


Source	Logs Data	Metrics Data
Platform components	Logs from CF components	- Health metrics from BOSH-deployed VMs* - Platform metrics from CF components. For example, cell capacity remaining and router throughput. - Metrics for any service tile that self-publishes to the Loggregator Firehose. For example, Redis and MySQL.
Apps	Logs from apps **	- App container metrics. For example, CPU, memory, and disk usage. - Custom metrics ***

Viewing logs in PCF

- cf cli
- Apps Manager
- PCF Metrics

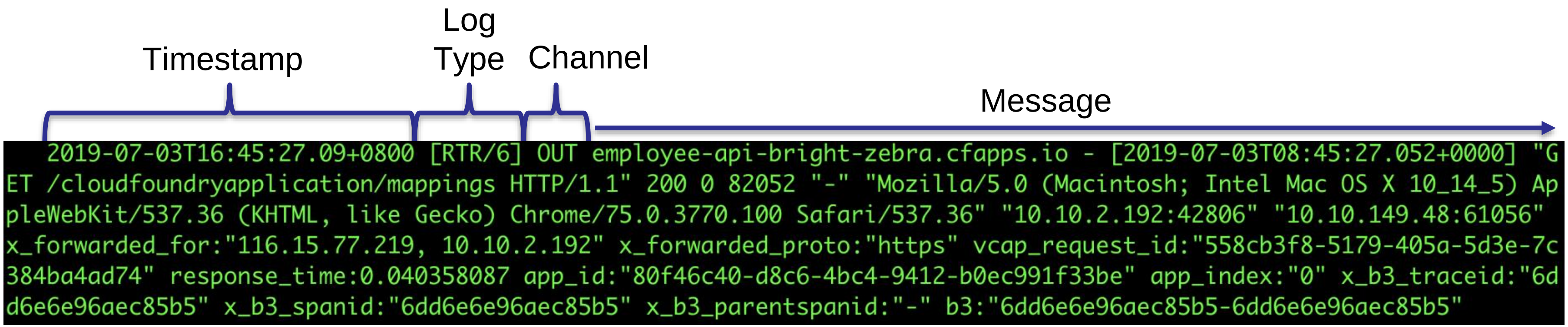
Log format



Every log line contains four fields:

- Timestamp
- Log type (origin code)
- Channel: either `OUT`, for logs emitted on `stdout`, or `ERR`, for logs emitted on `stderr`
- Message

Log format



Timestamp

Log Type

Channel

Message

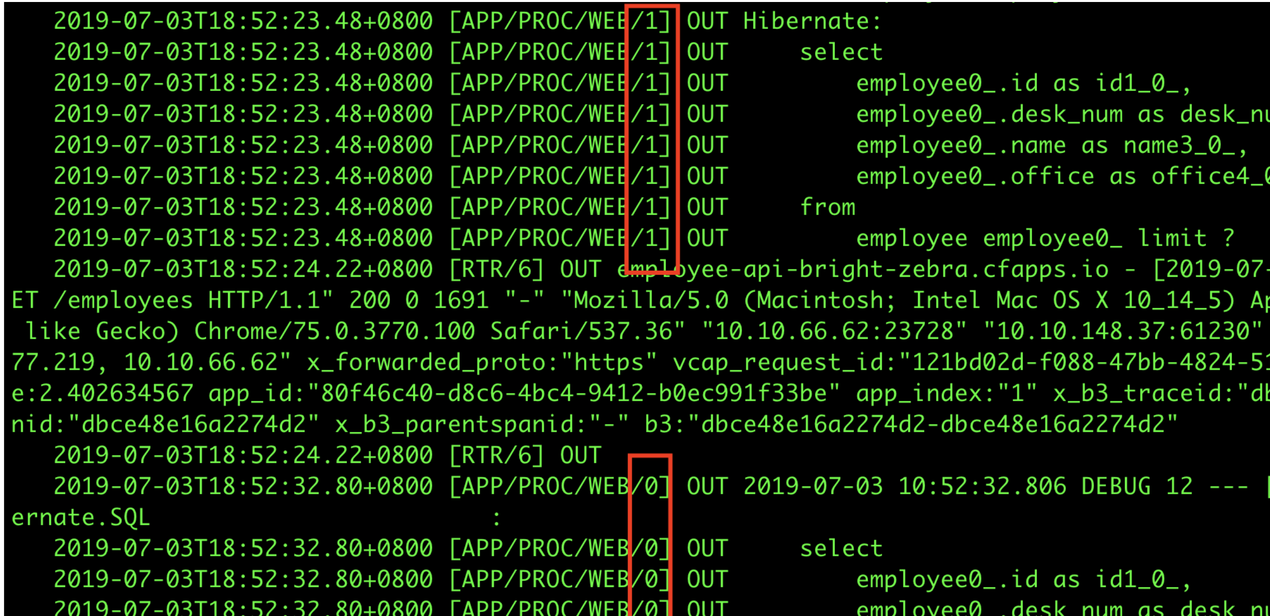
```
   2019-07-03T16:45:27.09+0800 [RTR/6] OUT employee-api-bright-zebra.cfapps.io - [2019-07-03T08:45:27.052+0000] "G
ET /cloudfoundryapplication/mappings HTTP/1.1" 200 0 82052 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) Ap
pleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36" "10.10.2.192:42806" "10.10.149.48:61056"
x_forwarded_for:"116.15.77.219, 10.10.2.192" x_forwarded_proto:"https" vcap_request_id:"558cb3f8-5179-405a-5d3e-7c
384ba4ad74" response_time:0.040358087 app_id:"80f46c40-d8c6-4bc4-9412-b0ec991f33be" app_index:"0" x_b3_traceid:"6d
d6e6e96aec85b5" x_b3_spanid:"6dd6e6e96aec85b5" x_b3_parentspanid:"-" b3:"6dd6e6e96aec85b5-6dd6e6e96aec85b5"
```

Log types

- **API** - Users make API calls to request changes in app state. Cloud Controller, the Cloud Foundry component responsible for the API, logs the actions that Cloud Controller takes in response.
- **STG** - The Diego cell or the Droplet Execution Agent emits STG logs when staging or restaging an app.
- **RTR** - The Router emits RTR logs when it routes HTTP requests to the app.
- **LGR** - Loggregator emits LGR to indicate problems with the logging process.
- **APP** - Every app emits logs according to choices by the developer.
- **SSH** - The Diego cell emits SSH logs when a user accesses an app container through SSH by using the Cloud Foundry Command Line Interface (cf CLI) cf ssh command.
- **CELL** - The Diego cell emits CELL logs when it starts or stops the app.

Hands-On: Step 1

- Scale the employee-api microservice up to 3 instances (You may have to first disable autoscaling)
- Execute the following cf command to tail the logs
  - cf logs employee-api
- Make a number of requests to the /employees endpoint
- Identify
  - Log types
  - Instance index



```
   2019-07-03T18:52:23.48+0800 [APP/PROC/WEB/1] OUT Hibernate:
   2019-07-03T18:52:23.48+0800 [APP/PROC/WEB/1] OUT     select
   2019-07-03T18:52:23.48+0800 [APP/PROC/WEB/1] OUT         employee0_.id as id1_0_,
   2019-07-03T18:52:23.48+0800 [APP/PROC/WEB/1] OUT         employee0_.desk_num as desk_nu
   2019-07-03T18:52:23.48+0800 [APP/PROC/WEB/1] OUT         employee0_.name as name3_0_,
   2019-07-03T18:52:23.48+0800 [APP/PROC/WEB/1] OUT         employee0_.office as office4_0
   2019-07-03T18:52:23.48+0800 [APP/PROC/WEB/1] OUT     from
   2019-07-03T18:52:23.48+0800 [APP/PROC/WEB/1] OUT         employee employee0_ limit ?
   2019-07-03T18:52:24.22+0800 [RTR/6] OUT employee-api-bright-zebra.cfapps.io - [2019-07-
ET /employees HTTP/1.1" 200 0 1691 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) Ap
 like Gecko) Chrome/75.0.3770.100 Safari/537.36" "10.10.66.62:23728" "10.10.148.37:61230"
77.219, 10.10.66.62" x_forwarded_proto:"https" vcap_request_id:"121bd02d-f088-47bb-4824-51
e:2.402634567 app_id:"80f46c40-d8c6-4bc4-9412-b0ec991f33be" app_index:"1" x_b3_traceid:"db
nid:"dbce48e16a2274d2" x_b3_parentspanid:"-" b3:"dbce48e16a2274d2-dbce48e16a2274d2"
   2019-07-03T18:52:24.22+0800 [RTR/6] OUT
   2019-07-03T18:52:32.80+0800 [APP/PROC/WEB/0] OUT 2019-07-03 10:52:32.806 DEBUG 12 --- [
ernate.SQL                              :
   2019-07-03T18:52:32.80+0800 [APP/PROC/WEB/0] OUT     select
   2019-07-03T18:52:32.80+0800 [APP/PROC/WEB/0] OUT         employee0_.id as id1_0_,
   2019-07-03T18:52:32.80+0800 [APP/PROC/WEB/0] OUT         employee0_.desk_num as desk_nu
```

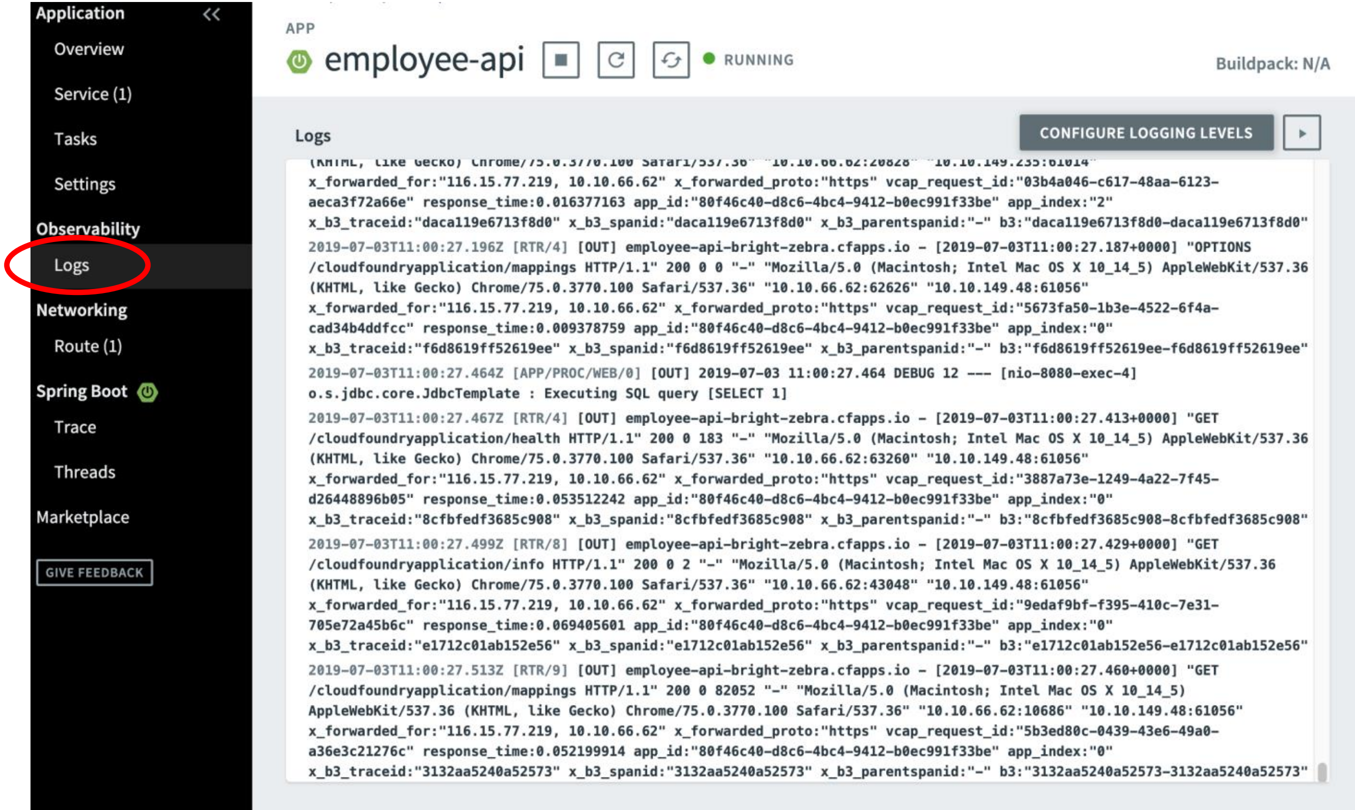
Hands-On: Step 2

- Get a dump of recent logs using the following cf command
  - cf logs employee-api --recent
  - To pipe it to file, you can use this instead

    cf logs employee-api --recent > emp.logs

Hands-On: Step 3

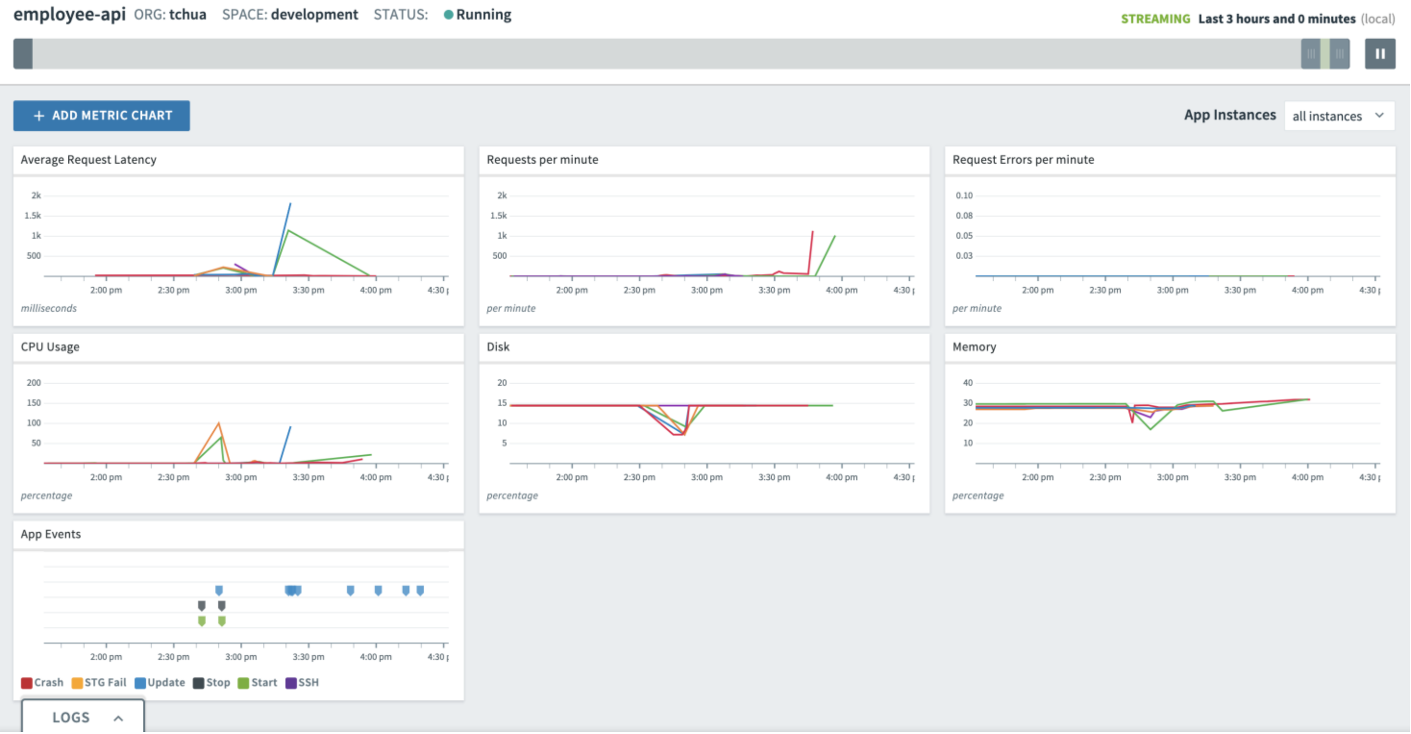
- View logs through the Apps Manager UI



```
(KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36" "10.10.66.62:20828" "10.10.149.235:61014"
x_forwarded_for:"116.15.77.219, 10.10.66.62" x_forwarded_proto:"https" vcap_request_id:"03b4a046-c617-48aa-6123-
aeca3f72a66e" response_time:0.016377163 app_id:"80f46c40-d8c6-4bc4-9412-b0ec991f33be" app_index:"2"
x_b3_traceid:"daca119e6713f8d0" x_b3_spanid:"daca119e6713f8d0" x_b3_parentspanid:"-" b3:"daca119e6713f8d0-daca119e6713f8d0"
2019-07-03T11:00:27.196Z [RTR/4] [OUT] employee-api-bright-zebra.cfapps.io - [2019-07-03T11:00:27.187+0000] "OPTIONS
/cloudfoundryapplication/mappings HTTP/1.1" 200 0 0 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36" "10.10.66.62:62626" "10.10.149.48:61056"
x_forwarded_for:"116.15.77.219, 10.10.66.62" x_forwarded_proto:"https" vcap_request_id:"5673fa50-1b3e-4522-6f4a-
cad34b4ddfcc" response_time:0.009378759 app_id:"80f46c40-d8c6-4bc4-9412-b0ec991f33be" app_index:"0"
x_b3_traceid:"f6d8619ff52619ee" x_b3_spanid:"f6d8619ff52619ee" x_b3_parentspanid:"-" b3:"f6d8619ff52619ee-f6d8619ff52619ee"
2019-07-03T11:00:27.464Z [APP/PROC/WEB/0] [OUT] 2019-07-03 11:00:27.464 DEBUG 12 --- [nio-8080-exec-4]
o.s.jdbc.core.JdbcTemplate : Executing SQL query [SELECT 1]
2019-07-03T11:00:27.467Z [RTR/4] [OUT] employee-api-bright-zebra.cfapps.io - [2019-07-03T11:00:27.413+0000] "GET
/cloudfoundryapplication/health HTTP/1.1" 200 0 183 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36" "10.10.66.62:63260" "10.10.149.48:61056"
x_forwarded_for:"116.15.77.219, 10.10.66.62" x_forwarded_proto:"https" vcap_request_id:"3887a73e-1249-4a22-7f45-
d26448896b05" response_time:0.053512242 app_id:"80f46c40-d8c6-4bc4-9412-b0ec991f33be" app_index:"0"
x_b3_traceid:"8cfbfedf3685c908" x_b3_spanid:"8cfbfedf3685c908" x_b3_parentspanid:"-" b3:"8cfbfedf3685c908-8cfbfedf3685c908"
2019-07-03T11:00:27.499Z [RTR/8] [OUT] employee-api-bright-zebra.cfapps.io - [2019-07-03T11:00:27.429+0000] "GET
/cloudfoundryapplication/info HTTP/1.1" 200 0 2 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36" "10.10.66.62:43048" "10.10.149.48:61056"
x_forwarded_for:"116.15.77.219, 10.10.66.62" x_forwarded_proto:"https" vcap_request_id:"9edaf9bf-f395-410c-7e31-
705e72a45b6c" response_time:0.069405601 app_id:"80f46c40-d8c6-4bc4-9412-b0ec991f33be" app_index:"0"
x_b3_traceid:"e1712c01ab152e56" x_b3_spanid:"e1712c01ab152e56" x_b3_parentspanid:"-" b3:"e1712c01ab152e56-e1712c01ab152e56"
2019-07-03T11:00:27.513Z [RTR/9] [OUT] employee-api-bright-zebra.cfapps.io - [2019-07-03T11:00:27.460+0000] "GET
/cloudfoundryapplication/mappings HTTP/1.1" 200 0 82052 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36" "10.10.66.62:10686" "10.10.149.48:61056"
x_forwarded_for:"116.15.77.219, 10.10.66.62" x_forwarded_proto:"https" vcap_request_id:"5b3ed80c-0439-43e6-49a0-
a36e3c21276c" response_time:0.052199914 app_id:"80f46c40-d8c6-4bc4-9412-b0ec991f33be" app_index:"0"
x_b3_traceid:"3132aa5240a52573" x_b3_spanid:"3132aa5240a52573" x_b3_parentspanid:"-" b3:"3132aa5240a52573-3132aa5240a52573"
```

PCF Metrics

Pivotal Cloud Foundry (PCF) Metrics stores logs, metrics data, and event data from apps running on PCF for the past 14 days. It graphically presents this data to help operators and developers better understand the health and performance of their apps.

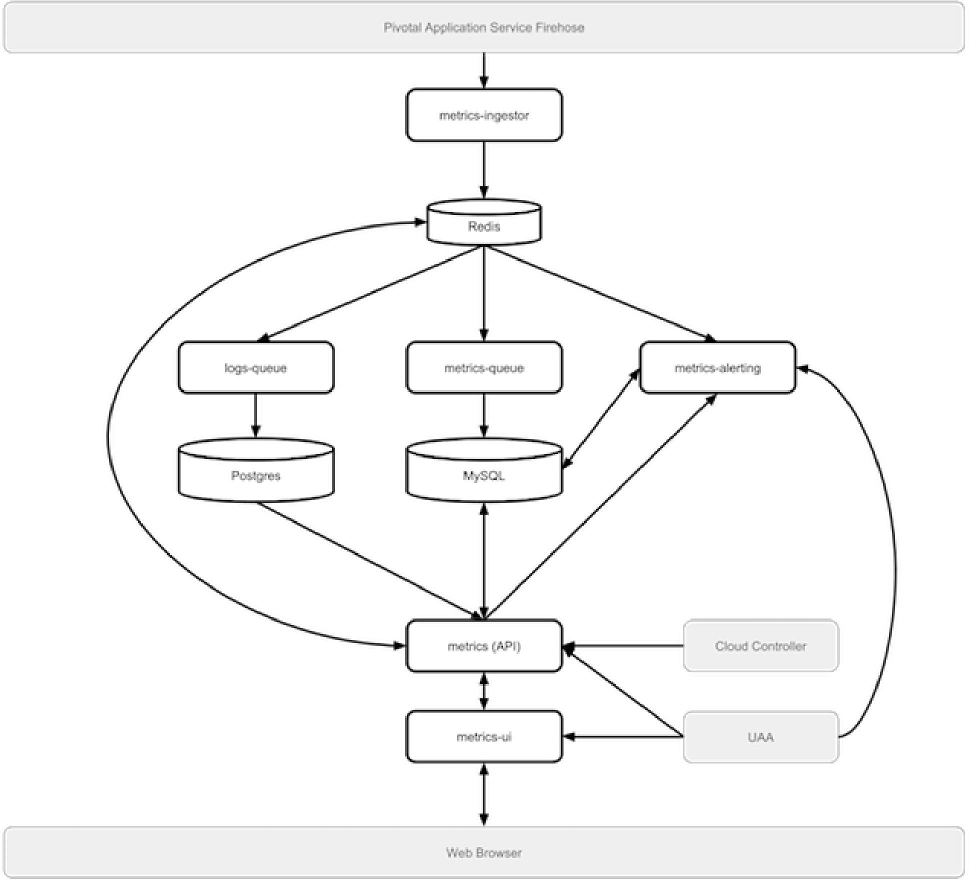


employee-api ORG: tchua SPACE: development STATUS: ● Running

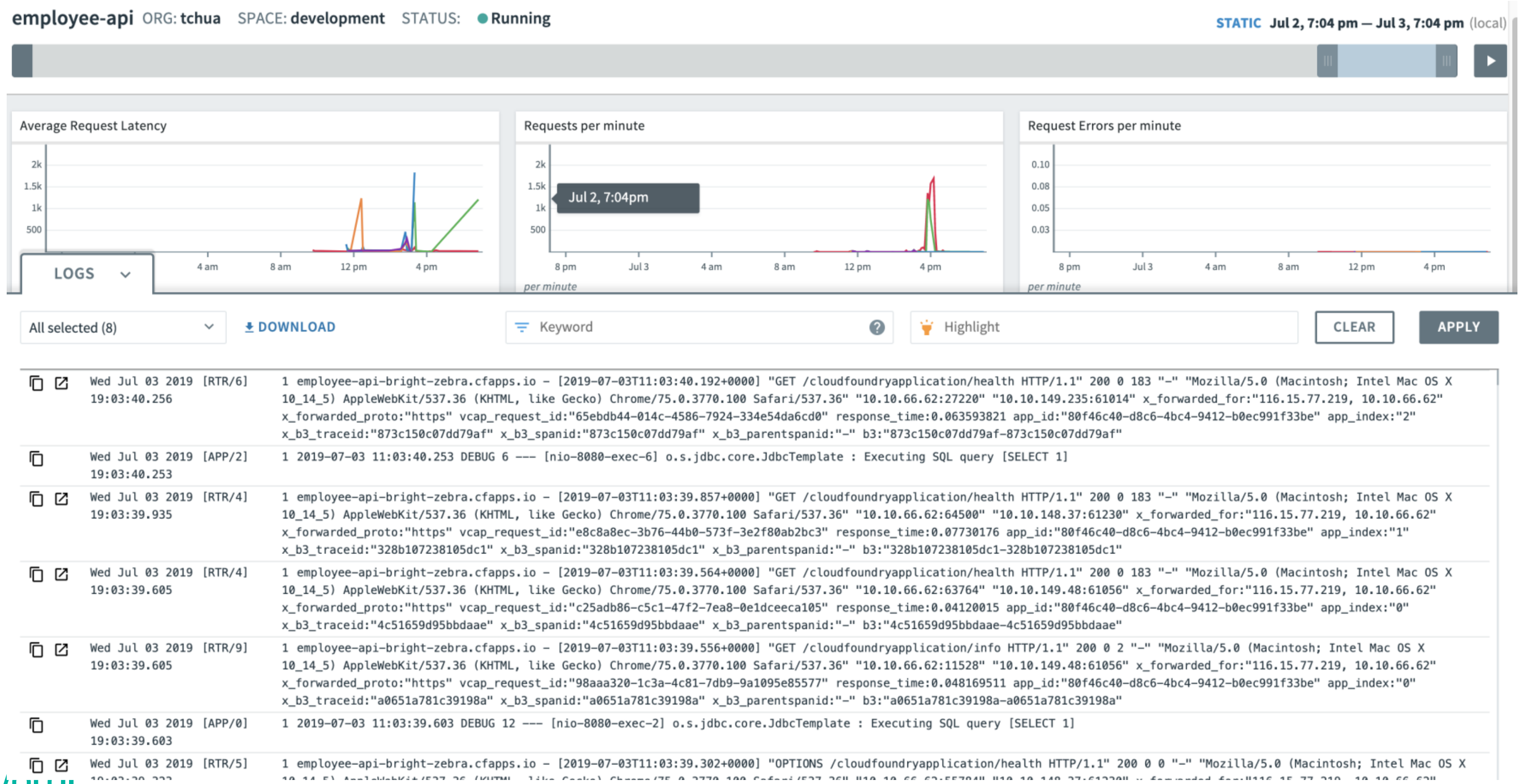
STREAMING Last 3 hours and 0 minutes (local)

App Instances: all instances

PCF Metrics Architecture



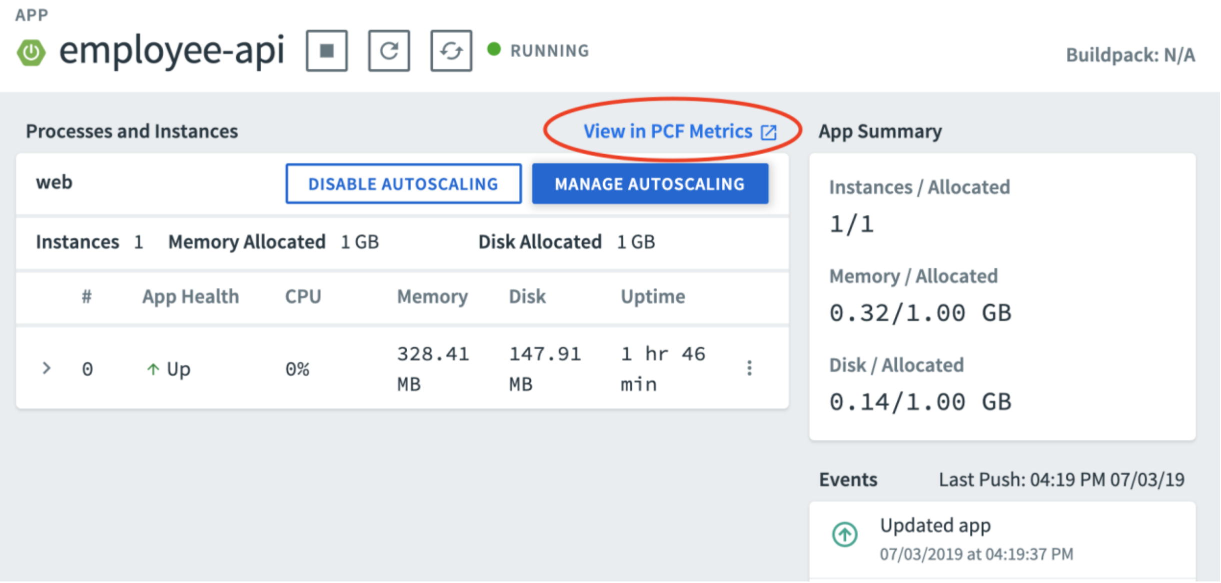
View logs in PCF Metrics



```
Wed Jul 03 2019  [RTR/6]   1 employee-api-bright-zebra.cfapps.io - [2019-07-03T11:03:40.192+0000] "GET /cloudfoundryapplication/health HTTP/1.1" 200 0 183 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X
19:03:40.256                 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36" "10.10.66.62:27220" "10.10.149.235:61014" x_forwarded_for:"116.15.77.219, 10.10.66.62"
                             x_forwarded_proto:"https" vcap_request_id:"65ebdb44-014c-4586-7924-334e54da6cd0" response_time:0.063593821 app_id:"80f46c40-d8c6-4bc4-9412-b0ec991f33be" app_index:"2"
                             x_b3_traceid:"873c150c07dd79af" x_b3_spanid:"873c150c07dd79af" x_b3_parentspanid:"-" b3:"873c150c07dd79af-873c150c07dd79af"
Wed Jul 03 2019  [APP/2]   1 2019-07-03 11:03:40.253 DEBUG 6 --- [nio-8080-exec-6] o.s.jdbc.core.JdbcTemplate : Executing SQL query [SELECT 1]
19:03:40.253
Wed Jul 03 2019  [RTR/4]   1 employee-api-bright-zebra.cfapps.io - [2019-07-03T11:03:39.857+0000] "GET /cloudfoundryapplication/health HTTP/1.1" 200 0 183 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X
19:03:39.935                 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36" "10.10.66.62:64500" "10.10.148.37:61230" x_forwarded_for:"116.15.77.219, 10.10.66.62"
                             x_forwarded_proto:"https" vcap_request_id:"e8c8a8ec-3b76-44b0-573f-3e2f80ab2bc3" response_time:0.07730176 app_id:"80f46c40-d8c6-4bc4-9412-b0ec991f33be" app_index:"1"
                             x_b3_traceid:"328b107238105dc1" x_b3_spanid:"328b107238105dc1" x_b3_parentspanid:"-" b3:"328b107238105dc1-328b107238105dc1"
Wed Jul 03 2019  [RTR/4]   1 employee-api-bright-zebra.cfapps.io - [2019-07-03T11:03:39.564+0000] "GET /cloudfoundryapplication/health HTTP/1.1" 200 0 183 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X
19:03:39.605                 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36" "10.10.66.62:63764" "10.10.149.48:61056" x_forwarded_for:"116.15.77.219, 10.10.66.62"
                             x_forwarded_proto:"https" vcap_request_id:"c25adb86-c5c1-47f2-7ea8-0e1dceeca105" response_time:0.04120015 app_id:"80f46c40-d8c6-4bc4-9412-b0ec991f33be" app_index:"0"
                             x_b3_traceid:"4c51659d95bbdaae" x_b3_spanid:"4c51659d95bbdaae" x_b3_parentspanid:"-" b3:"4c51659d95bbdaae-4c51659d95bbdaae"
Wed Jul 03 2019  [RTR/9]   1 employee-api-bright-zebra.cfapps.io - [2019-07-03T11:03:39.556+0000] "GET /cloudfoundryapplication/info HTTP/1.1" 200 0 2 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X
19:03:39.605                 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36" "10.10.66.62:11528" "10.10.149.48:61056" x_forwarded_for:"116.15.77.219, 10.10.66.62"
                             x_forwarded_proto:"https" vcap_request_id:"98aaa320-1c3a-4c81-7db9-9a1095e85577" response_time:0.048169511 app_id:"80f46c40-d8c6-4bc4-9412-b0ec991f33be" app_index:"0"
                             x_b3_traceid:"a0651a781c39198a" x_b3_spanid:"a0651a781c39198a" x_b3_parentspanid:"-" b3:"a0651a781c39198a-a0651a781c39198a"
Wed Jul 03 2019  [APP/0]   1 2019-07-03 11:03:39.603 DEBUG 12 --- [nio-8080-exec-2] o.s.jdbc.core.JdbcTemplate : Executing SQL query [SELECT 1]
19:03:39.603
Wed Jul 03 2019  [RTR/5]   1 employee-api-bright-zebra.cfapps.io - [2019-07-03T11:03:39.302+0000] "OPTIONS /cloudfoundryapplication/health HTTP/1.1" 200 0 0 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X
```

Hands-On: Step 1

- On your employee-api app page in Apps Manager, click the “View in PCF Metrics” link.
- Are you able to correlate the analytics provided by PCF Metrics to the various activities done earlier?



APP

employee-api ■ ⟳ RUNNING

Buildpack: N/A

Processes and Instances

View in PCF Metrics

web

DISABLE AUTOSCALING

MANAGE AUTOSCALING

Instances 1 Memory Allocated 1 GB Disk Allocated 1 GB

#	App Health	CPU	Memory	Disk	Uptime
0	↑ Up	0%	328.41 MB	147.91 MB	1 hr 46 min

App Summary

Instances / Allocated
1/1

Memory / Allocated
0.32/1.00 GB

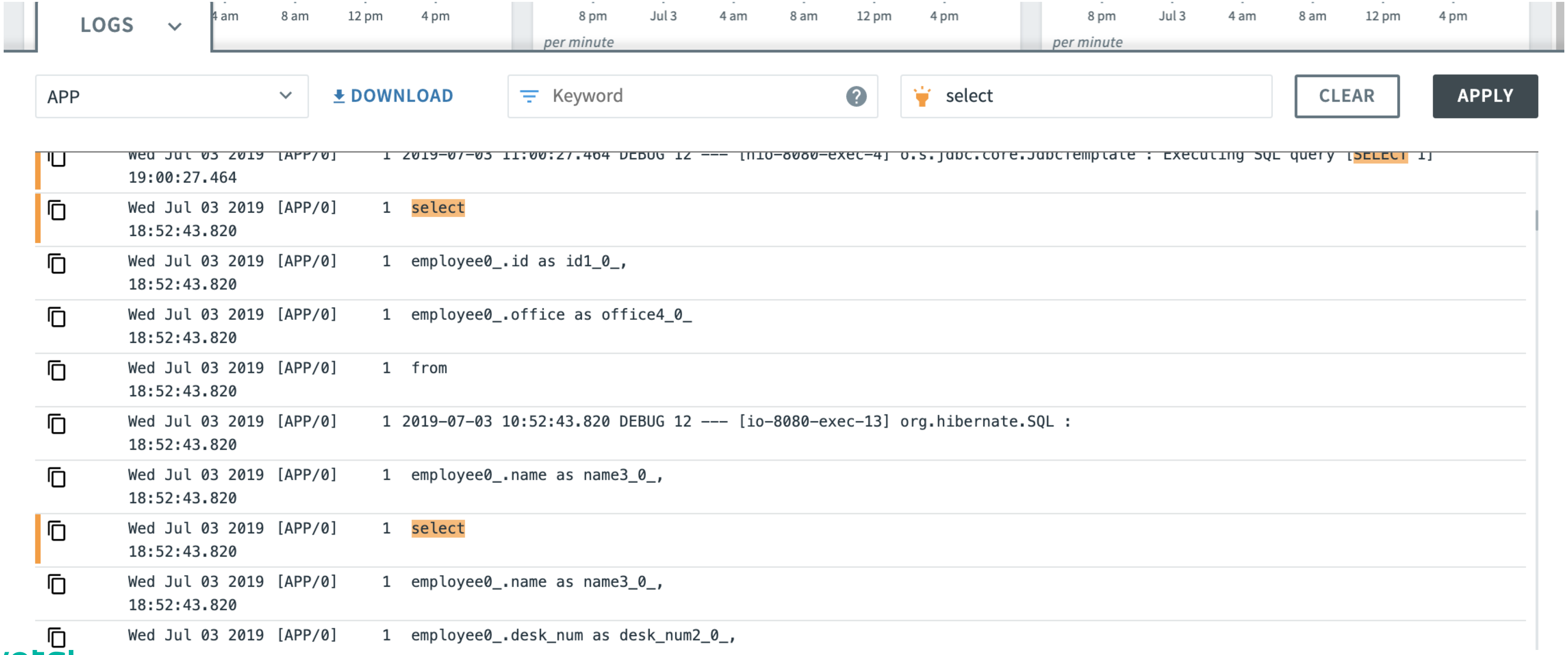
Disk / Allocated
0.14/1.00 GB

Events Last Push: 04:19 PM 07/03/19

Updated app
07/03/2019 at 04:19:37 PM

Hands-On: Step 2

- Using the logs feature in PCF Metrics, filter out Apps log, highlighting SQL select queries

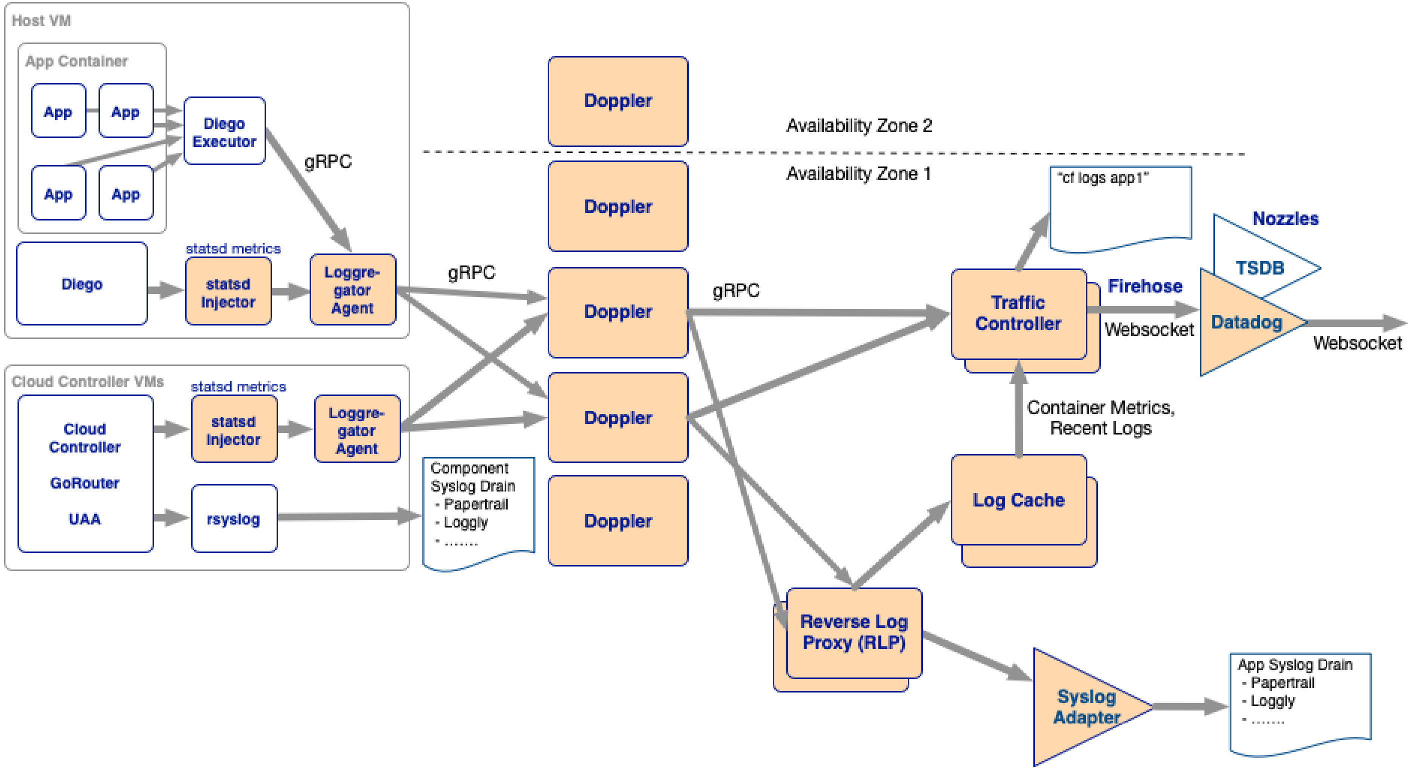


```
Wed Jul 03 2019 [APP/0]     1 2019-07-03 11:00:27.464 DEBUG 12 --- [nio-8080-exec-4] o.s.jdbc.core.JdbcTemplate : Executing SQL query [SELECT 1]
19:00:27.464
Wed Jul 03 2019 [APP/0]     1   select
18:52:43.820
Wed Jul 03 2019 [APP/0]     1   employee0_.id as id1_0_,
18:52:43.820
Wed Jul 03 2019 [APP/0]     1   employee0_.office as office4_0_
18:52:43.820
Wed Jul 03 2019 [APP/0]     1   from
18:52:43.820
Wed Jul 03 2019 [APP/0]     1 2019-07-03 10:52:43.820 DEBUG 12 --- [io-8080-exec-13] org.hibernate.SQL :
18:52:43.820
Wed Jul 03 2019 [APP/0]     1   employee0_.name as name3_0_,
18:52:43.820
Wed Jul 03 2019 [APP/0]     1   select
18:52:43.820
Wed Jul 03 2019 [APP/0]     1   employee0_.name as name3_0_,
18:52:43.820
Wed Jul 03 2019 [APP/0]     1   employee0_.desk_num as desk_num2_0_,
```

Streaming logs and metrics to third party service

- **Loggregator**
  - Gathers and streams logs and metrics from user apps in a PCF deployment
  - Gathers and streams metrics from PCF components and health metrics from BOSH-deployed VMs
  - Architecture includes components for collecting, storing, and forwarding logs and metrics

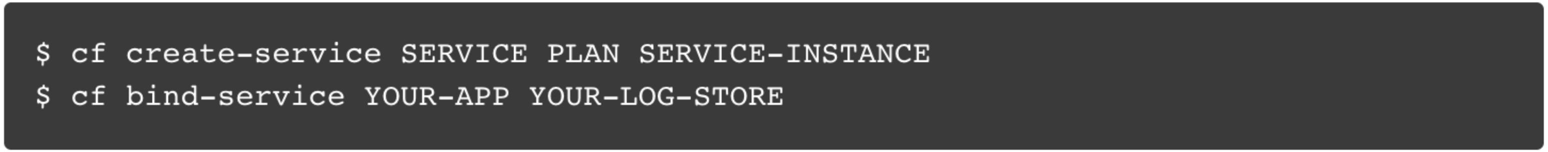
Loggregator architecture and components



Details at https://docs.pivotal.io/pivotalcf/2-6/loggregator/architecture.html#components

Streaming app logs to external services

- If external service is available in marketplace



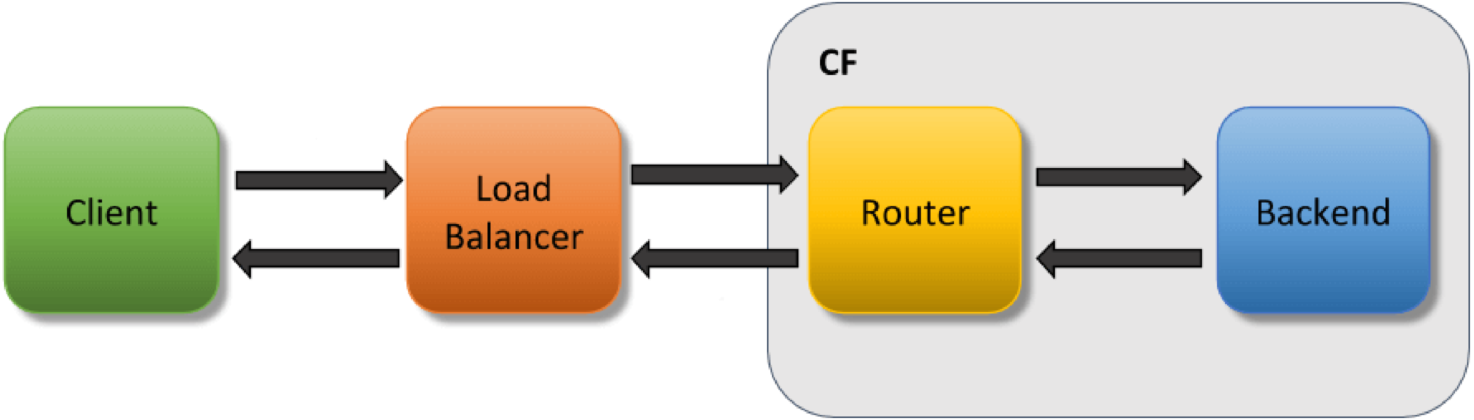
```
$ cf create-service SERVICE PLAN SERVICE-INSTANCE
$ cf bind-service YOUR-APP YOUR-LOG-STORE
```

- If external service is **not** available in marketplace
  See https://docs.pivotal.io/pivotalcf/2-6/devguide/services/log-management.html#user-provided

Troubleshooting

Troubleshooting slow requests

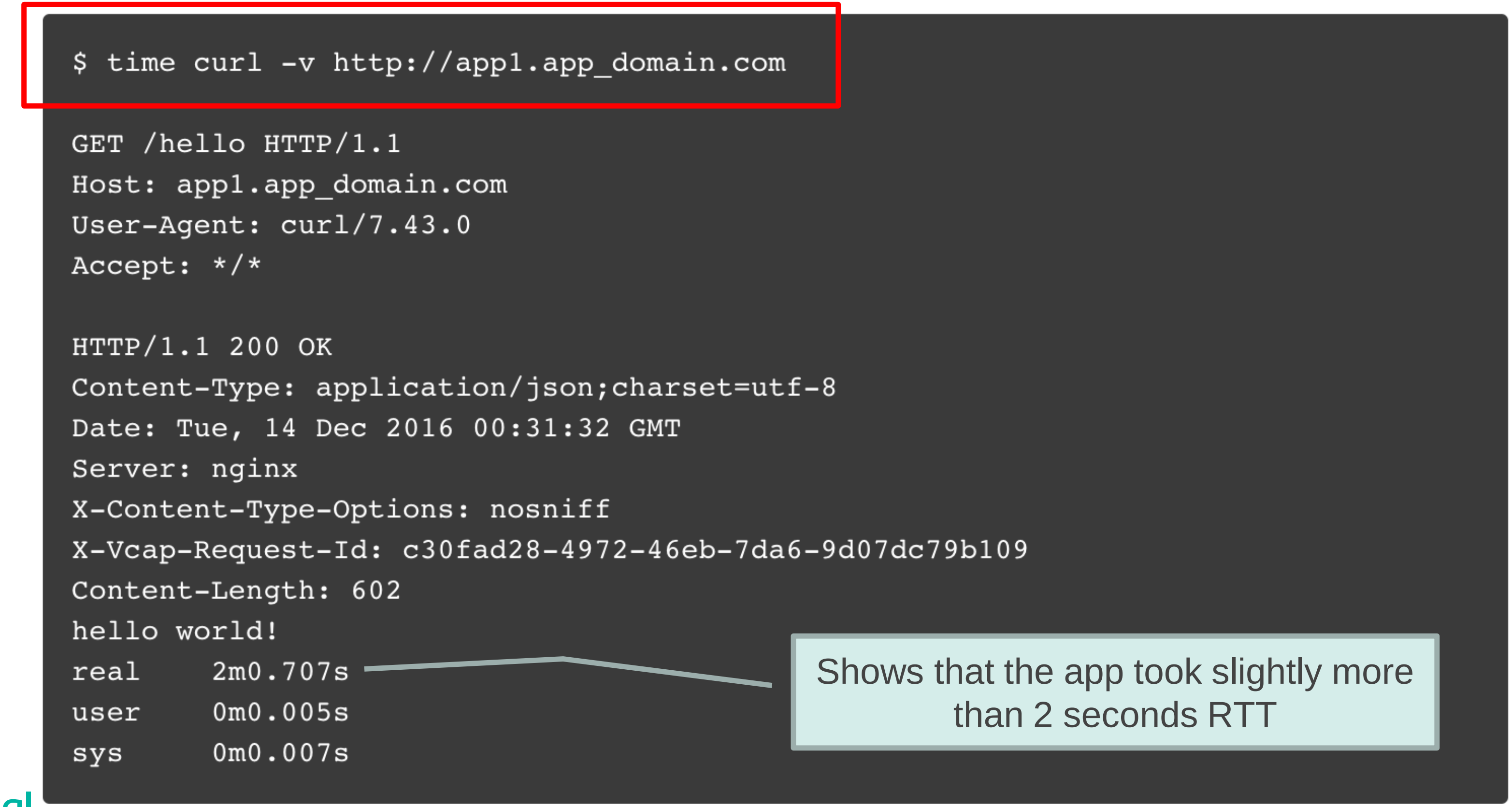
App request path



Only the GoRouter and your app is within the scope of Cloud Foundry

Troubleshooting slow requests

Measuring total round trip time (client point of view)

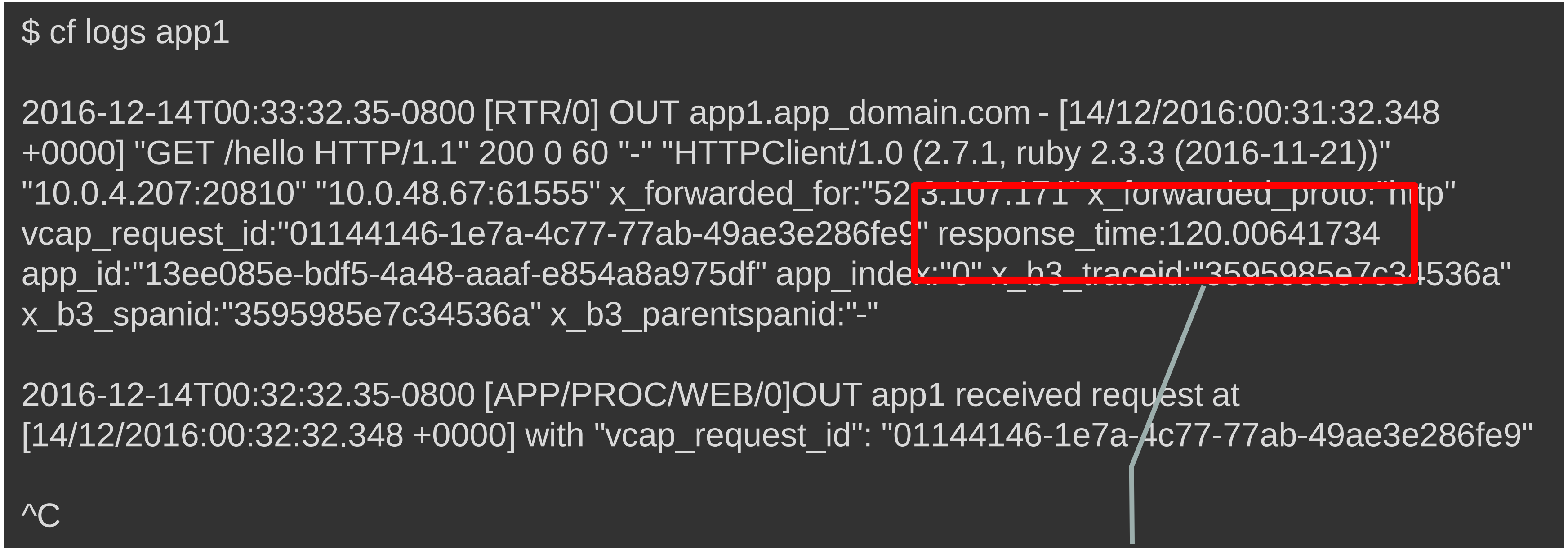


```
$ time curl -v http://app1.app_domain.com

GET /hello HTTP/1.1
Host: app1.app_domain.com
User-Agent: curl/7.43.0
Accept: */*

HTTP/1.1 200 OK
Content-Type: application/json;charset=utf-8
Date: Tue, 14 Dec 2016 00:31:32 GMT
Server: nginx
X-Content-Type-Options: nosniff
X-Vcap-Request-Id: c30fad28-4972-46eb-7da6-9d07dc79b109
Content-Length: 602
hello world!
real    2m0.707s
user    0m0.005s
sys     0m0.007s
```

Shows that the app took slightly more than 2 seconds RTT

Troubleshooting slow requests

Measuring time spent in Cloud Foundry



```
$ cf logs app1

2016-12-14T00:33:32.35-0800 [RTR/0] OUT app1.app_domain.com - [14/12/2016:00:31:32.348
+0000] "GET /hello HTTP/1.1" 200 0 60 "-" "HTTPClient/1.0 (2.7.1, ruby 2.3.3 (2016-11-21))"
"10.0.4.207:20810" "10.0.48.67:61555" x_forwarded_for:"52.3.107.171" x_forwarded_proto:"http"
vcap_request_id:"01144146-1e7a-4c77-77ab-49ae3e286fe9" response_time:120.00641734
app_id:"13ee085e-bdf5-4a48-aaaf-e854a8a975df" app_index:"0" x_b3_traceid:"3595985e7c34536a"
x_b3_spanid:"3595985e7c34536a" x_b3_parentspanid:"-"

2016-12-14T00:32:32.35-0800 [APP/PROC/WEB/0]OUT app1 received request at
[14/12/2016:00:32:32.348 +0000] with "vcap_request_id": "01144146-1e7a-4c77-77ab-49ae3e286fe9"

^C
```

GoRouter response_time shows total time spent in Cloud Foundry

Troubleshooting slow requests

Use app logs to locate delays within Cloud Foundry

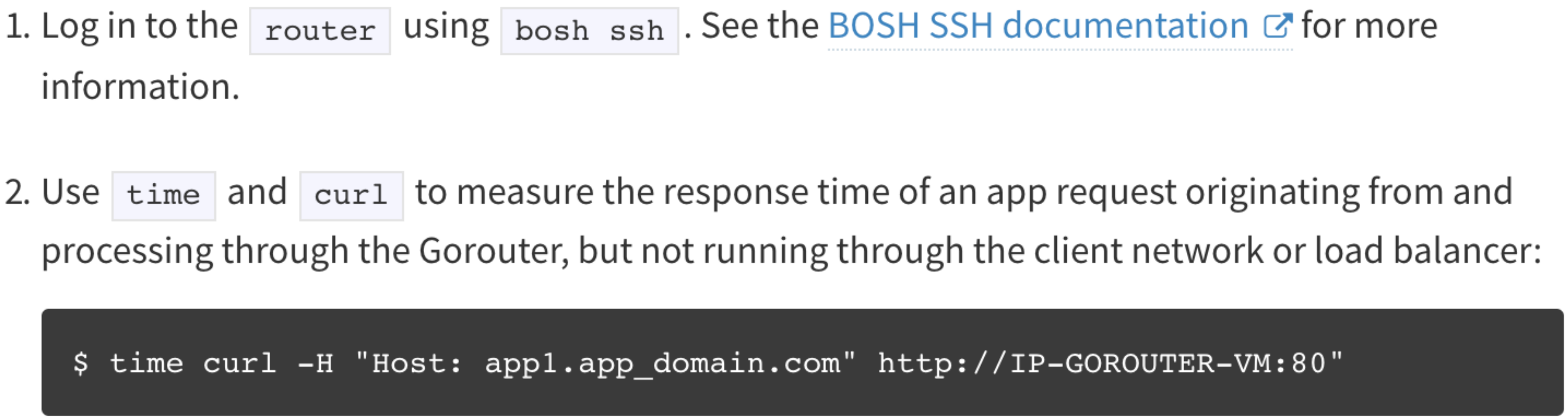
```
2016-12-14T00:33:32.35-0800 [RTR/0] OUT app1.app_domain.com - [14/12/2016:00:31:32.348
+0000] "GET /hello HTTP/1.1" 200 0 60 "-" "HTTPClient/1.0 (2.7.1, ruby 2.3.3 (2016-11-21))"
"10.0.4.207:20810" "10.0.48.67:61555" x_forwarded_for:"52.3.107.171" x_forwarded_proto:"http"
vcap_request_id:"01144146-1e7a-4c77-77ab-49ae3e286fe9" response_time:120.00641734
app_id:"13ee085e-bdf5-4a48-aaaf-e854a8a975df" app_index:"0" x_b3_traceid:"3595985e7c34536a"
x_b3_spanid:"3595985e7c34536a" x_b3_parentspanid:"-”

2016-12-14T00:32:32.35-0800 [APP/PROC/WEB/0]OUT app1 received request at
[14/12/2016:00:32:32.348 +0000] with "vcap_request_id": "01144146-1e7a-4c77-77ab-49ae3e286fe9"

2016-12-14T00:32:32.50-0800 [APP/PROC/WEB/0]OUT app1 finished processing req at
[14/12/2016:00:32:32.500 +0000] with "vcap_request_id": "01144146-1e7a-4c77-77ab-49ae3e286fe9"
```

Use custom logging at start and end of request to calculate time spent by app itself in Cloud Foundry

Pivotal

Troubleshooting slow requests

Measuring GoRouter processing (1/2)



1. Log in to the `router` using `bosh ssh`. See the BOSH SSH documentation for more information.

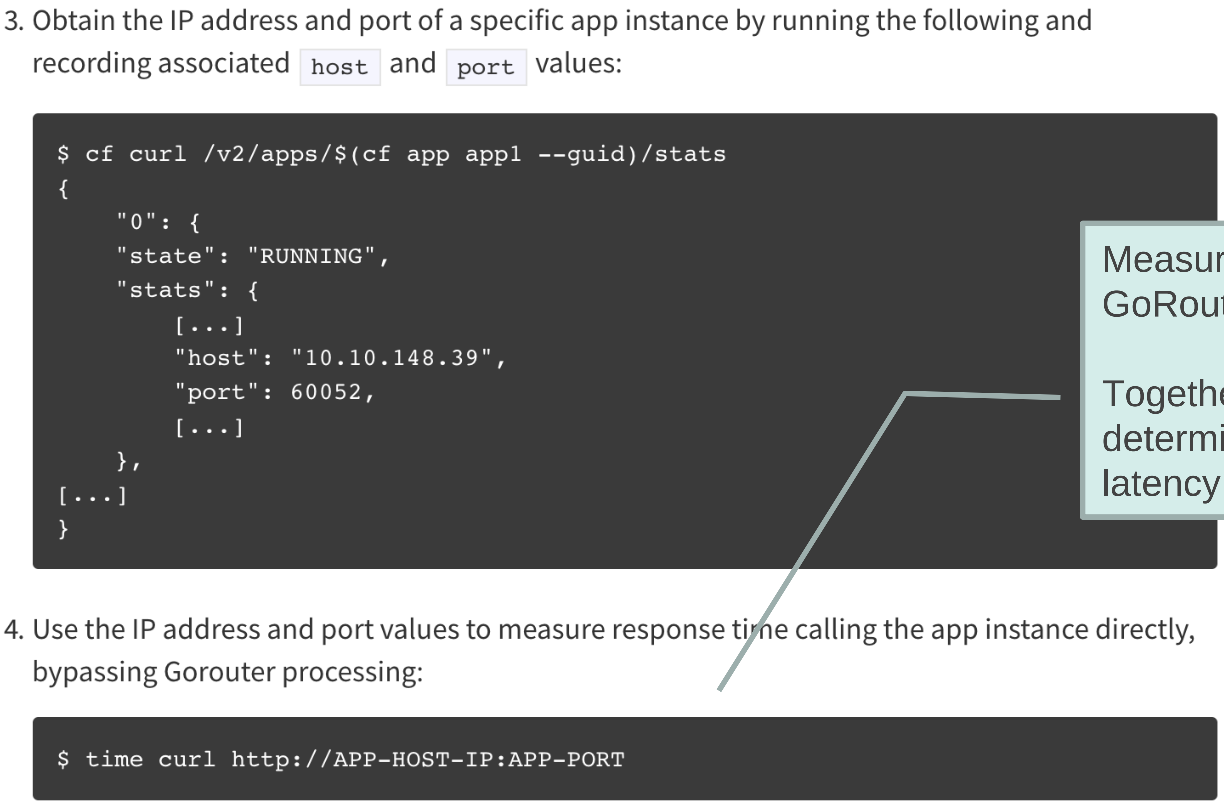
2. Use `time` and `curl` to measure the response time of an app request originating from and processing through the Gorouter, but not running through the client network or load balancer:

```
$ time curl -H "Host: app1.app_domain.com" http://IP-GOROUTER-VM:80"
```

Measures RTT from GoRouter

Troubleshooting slow requests

Measuring GoRouter processing (2/2)



3. Obtain the IP address and port of a specific app instance by running the following and recording associated `host` and `port` values:

```
$ cf curl /v2/apps/$(cf app app1 --guid)/stats
{
    "0": {
    "state": "RUNNING",
    "stats": {
        [...]
        "host": "10.10.148.39",
        "port": 60052,
        [...]
    },
[...]
}
```

4. Use the IP address and port values to measure response time calling the app instance directly, bypassing Gorouter processing:

```
$ time curl http://APP-HOST-IP:APP-PORT
```

Measures request time bypassing GoRouter.

Together with previous result, determines if GoRouter to App latency is culprit

Pivotal®

Transforming How The World Builds Software