

The background of the slide is a faded, blue-tinted image of the Golden Gate Bridge in San Francisco. The bridge's iconic towers and suspension cables are visible, stretching across the frame from the left towards the right. The water of the bay is at the bottom, and the distant hills are visible in the background.

Pivotal

PCF Operations Workshop - Platform & Application Scaling

Dan Herold – Platform Architect

Nenad Momcilovic – Platform Architect

Operations Workshop Agenda

- PCF Introduction
- Services Overview
- Platform Installation & Setup
- Role Based Access Control
- **Platform & Application Scaling**
- Platform & Application Health
- Patching & Updates
- Security
- Advanced BOSH

Platform Scaling

- Ops Manager allows administrators to quickly and easily scale PCF components using the “Resource Config” workflow of the Elastic Runtime tile
- Administrators can select machine sizes for their components
- Additional information can be located at the following URL:
<http://docs.pivotal.io/pivotalcf/1-7/concepts/high-availability.html#availability>

Ops Manager

admin

time

Resource Config

JOB	INSTANCES	PERSISTENT DISK TYPE	VM TYPE	ELB BARETS
Consul	Automatic: 1	Automatic: 1 GB	Automatic: t2.small (cpu: 1, ram: 2 GB, disk: 2 GB)	1
NATS	Automatic: 1	None	Automatic: t2.small (cpu: 1, ram: 2 GB, disk: 2 GB)	
etcd	Automatic: 1	Automatic: 1 GB	Automatic: t2.small (cpu: 1, ram: 2 GB, disk: 2 GB)	
Diego BBS	Automatic: 1	Automatic: 1 GB	Automatic: t2.small (cpu: 1, ram: 2 GB, disk: 2 GB)	
NFS Server	Automatic: 1	Automatic: 100 GB	Automatic: t2.small (cpu: 1, ram: 2 GB, disk: 2 GB)	
Router	Automatic: 1	None	Automatic: t2.small (cpu: 1, ram: 2 GB, disk: 2 GB)	
MySQL Proxy	Automatic: 1	None	Automatic: m3.medium (cpu: 1, ram: 3.75 GB, disk: 3 GB)	
MySQL Server	Automatic: 1	Automatic: 100 GB	Automatic: m3.xlarge (cpu: 4, ram: 15 GB, disk: 3 GB)	
Backup Prepare Node	Automatic: 0	None	Automatic: r3.xlarge (cpu: 16, ram: 122 GB, disk: 3 GB)	
Cloud Controller Database (Postgres)	Automatic: 0	Automatic: 2 GB	Automatic: t2.small (cpu: 1, ram: 2 GB, disk: 2 GB)	
UAA Database (Postgres)	Automatic: 0	Automatic: 10 GB	Automatic: t2.small (cpu: 1, ram: 2 GB, disk: 2 GB)	
Apps Manager Database (Postgres)	Automatic: 0	Automatic: 1 GB	Automatic: t2.small (cpu: 1, ram: 2 GB, disk: 2 GB)	
Cloud Controller	Automatic: 1	None	Automatic: m3.xlarge (cpu: 4, ram: 15 GB, disk: 3 GB)	
HAProxy	Automatic: 1	None	Automatic: t2.small (cpu: 1, ram: 2 GB, disk: 2 GB)	

PCF Ops Manager v1.7.0.0; ©2013-2016 Pivotal Software, Inc. All Rights Reserved.

API Docs | End User License Agreement

Applications - Vertical Scaling

- Cloud Foundry enables vertical scaling of instances
- Users can modify Memory and Disk limits via:
 - Apps Manager console
 - cf CLI
 - manifest.yml

The screenshot displays the 'Scaling' dialog in the Cloud Foundry Apps Manager console. It features a title bar with 'Scaling' on the left, 'Cancel' in the center, and a blue 'Scale App' button on the right. Below the title bar, there are three columns: 'Instances', 'Memory Limit', and 'Disk Limit'. Each column contains a text input field. The 'Instances' field contains the number '1'. The 'Memory Limit' field contains '1 GB' and has a small up/down arrow icon to its right. The 'Disk Limit' field contains '1 GB' and also has a small up/down arrow icon to its right.

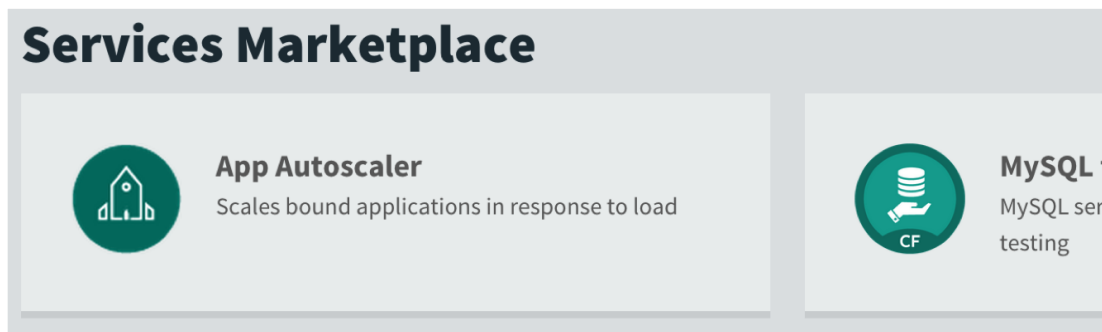
Instances	Memory Limit	Disk Limit
1	1 GB	1 GB

Applications - Horizontal Scaling

- Cloud Foundry enables rapid horizontal scaling of applications via containers
- Users can modify the number of instances via:
 - App Autoscaler tile
 - Apps Manager console
 - cf CLI
 - manifest.yml

App Autoscaler

- CPU usage or time-based schedule can be used to scale application instances
- It is a service in the marketplace



App Autoscaler Service – Steps

- 1) Create a service instance in for the space (if necessary)
- 2) Bind to an application
- 3) Set the desired scaling parameters
 - a) Add instance whenever high CPU threshold is reached
 - b) Subtract instance whenever low threshold is reached

The screenshot shows the Pivotal Autoscale service interface for an application named 'my-app'. The interface is divided into several sections:

- my-app**: The application name at the top right, with a pause button icon.
- INSTANCES**: A table showing the current number of instances.

	INSTANCES
min	2
max	5
- CPU THRESHOLDS**: A table showing the CPU thresholds for scaling.

	CPU THRESHOLDS
low	20%
high	80%
- LAST EVENT**: A section showing the most recent scaling event.

Scaled app from 1 to 2 instances
09/11/14 @ 23:15:56 UTC
- SCHEDULING**: A section showing the current scheduling rules.

0 rules Next: No Upcoming Events

App Autoscaling – Scheduling

- Autoscaling events can be scheduled
- Changes autoscaling behavior on the given date / time
- May be a single or recurring event

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

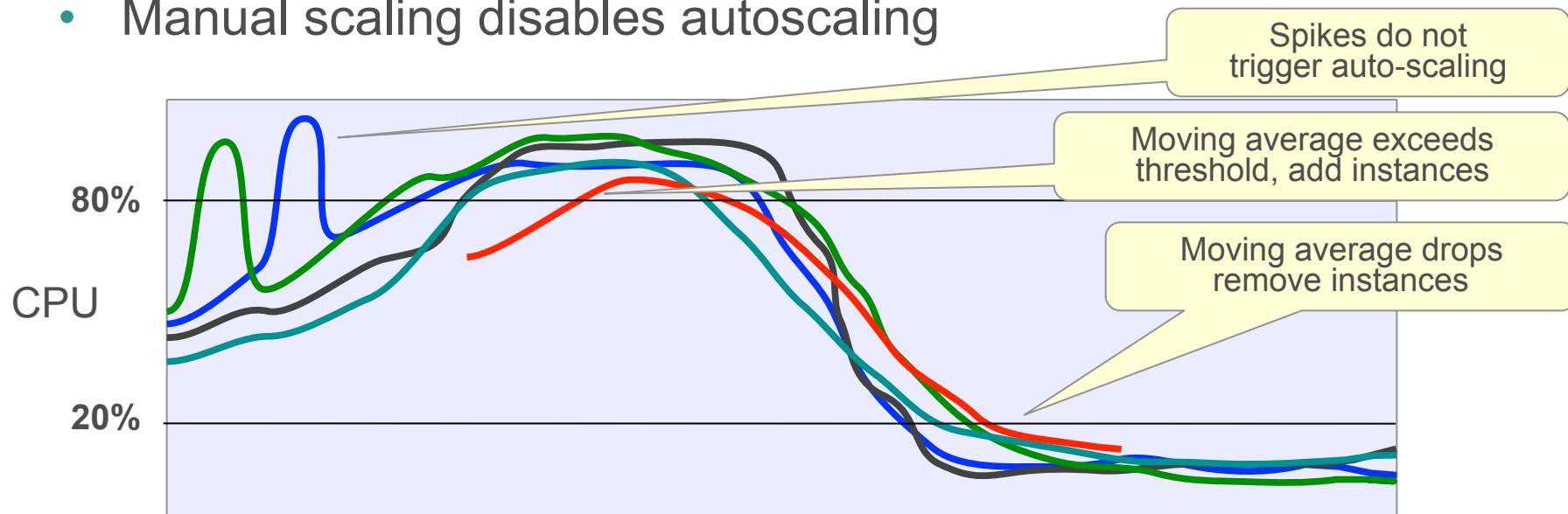
min low

max high


ADD

Autoscaling – Moving Average

- Scaling activity based on moving averages
 - Softens effect of temporary spikes
- Manual scaling disables autoscaling



Apps Manager Scaling

 Pivotal
Apps Manager

ORG

student20-org ▾

SPACES

development

production

test

Marketplace

Docs



Tools

student20-org > development > spring-music

student20 ▾

APP

spring-music




last push: 07/11/16 @ 18:41 UTC
<https://spring-music-hoofbound-p...>



ABOUT

BUILDPACK java-buildpack=v3.6-offline-https://github....
START CMD Set by the buildpack


CONFIGURATION

 Scale App


Instances

2  

Memory Limit

512 MB 

Disk Limit

1 GB 

STATUS

CLI Scaling

A terminal window with a title bar containing three colored circles (red, yellow, green) and the text "1. bash". The terminal text shows a command being executed in a shell. The prompt is "Nicks-MacBook-Pro-2:spring-music nsanfratello\$". The command is "cf scale -i 2 spring-music". The output is "Scaling app spring-music in org student20-org / space development as student20..." followed by "OK" on a new line.

```
1. bash
Nicks-MacBook-Pro-2:spring-music nsanfratello$ cf scale -i 2 spring-music
Scaling app spring-music in org student20-org / space development as student20...
OK
```

`cf scale -i 2 "my-application-name"`

Manifest Scaling

```
---
applications:
- name: spring-music
  memory: 64M
  instances: 5 # <-- Increase
  host: myspringmusic
  domain: example.com
  # comment
- name: nextapp
  memory: 256M
  ...
```

Platform Consumption & Chargeback

Usage Report PERIOD: Current: June 1 - Today [Download ZIP](#)

ORG	TOTAL APP MEMORY USAGE	TOTAL SERVICE INSTANCE USAGE
pivot-bboe	7.79 GB hrs	567.40 hrs
SPACE	APP MEMORY USAGE	SERVICE INSTANCE USAGE
development	7.79 GB hrs	567.40 hrs

SPACES DETAILS

development

APPS	7.79 GB hrs
spring-music	7.79 GB hrs
SERVICES	567.40 hrs
redis : p-redis : shared-vm	567.40 hrs

©2016 Pivotal Software, Inc. All Rights Reserved.

Download ZIP