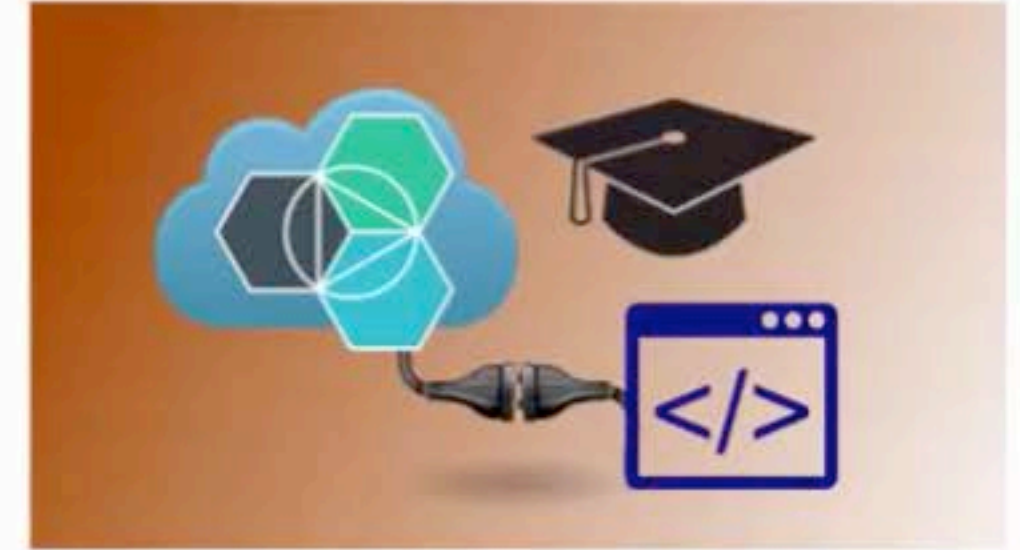


# Application Debugging



## Learning Objectives:

- 1 Tools, Features for identifying app issues
2. Demo - Eclipse Debugging

# Issues Resolution

- Use CF\_TRACE=true if a cf command fails
  - For example, `cf push CF_TRACE=true`
- Check **Logs** for runtime issues in the application
- Monitoring services, Auto scaling service
- Debugging using Eclipse

# Bluemix PaaS App Scaling



[raj@acloudfan.com](mailto:raj@acloudfan.com)

## Learning Objectives:

1. Bluemix PaaS vertical and horizontal scaling
2. Setting up a auto scaling policy for the application

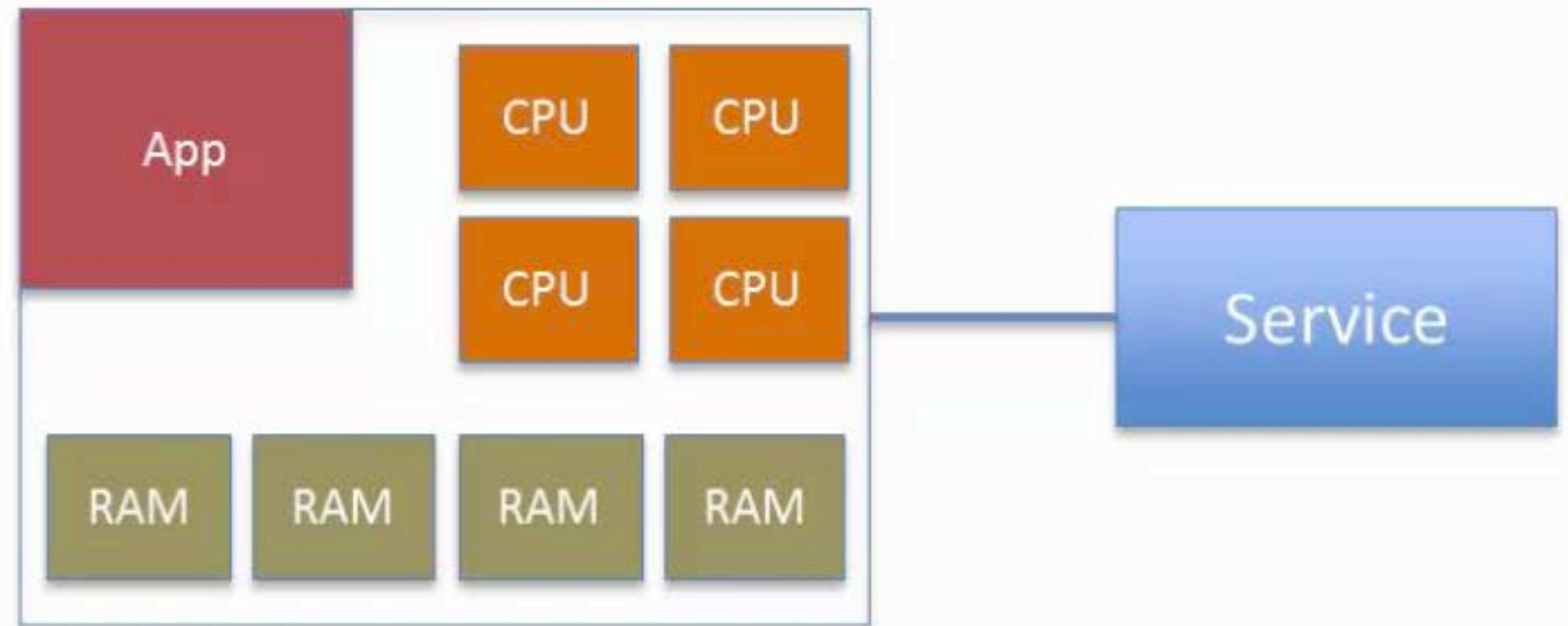
Breach Duration, Cooldown Period

Proactive scaling & Reactive scaling

# Vertical Scaling | Scale Up

- Increases the resource available on the node (runtime container of app) that is hosting the application instance

- Node will need a restart

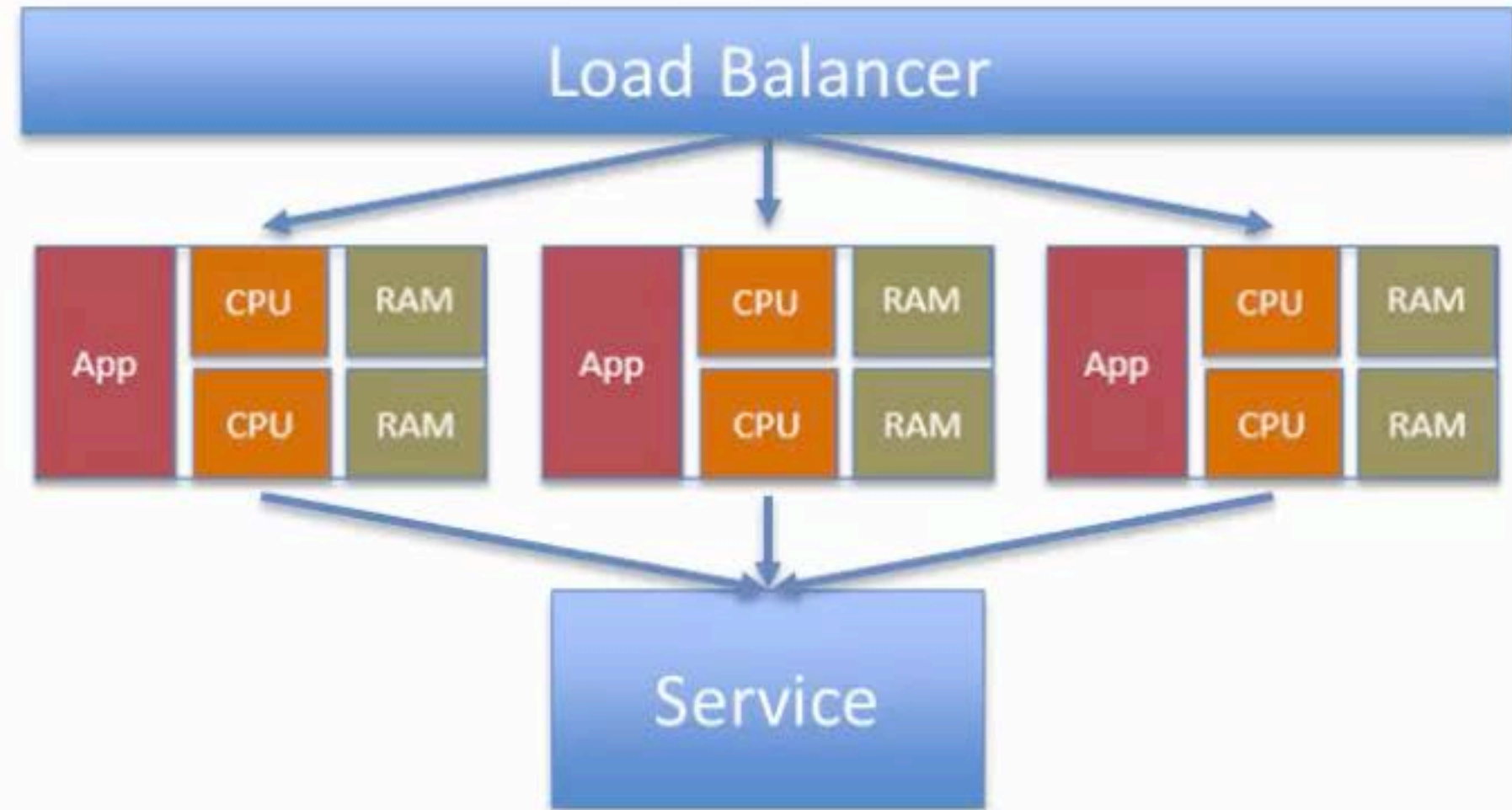


- Generally performance improvement is seen with vertical scaling but the gains are typically not linear



# Horizontal Scaling | Scale Out

- Overall capacity is increased by adding nodes. All nodes look the same in terms of capacity, configuration, code



- No downtime
- Nearly linear scaling for well designed apps

# Bluemix PaaS | Scaling support

- Vertical scaling
    - User controls the memory and disk allocated to the application; restart needed
  - Horizontal scaling
    - User controls the number of instances of the app; NO restart needed
1. Manual scaling using the Bluemix console & cf commands
  2. Scaling through the Manifest file
  3. Using the Auto scaling service

# Bluemix PaaS | Manual Scaling

- Vertical scaling

```
cf scale APP_NAME -m MEM_LIMIT -k DISK_LIMIT
```

- Horizontal scaling

```
cf scale APP_NAME -i INSTANCES
```





Getting Started

Overview

Runtime

Connections

Logs

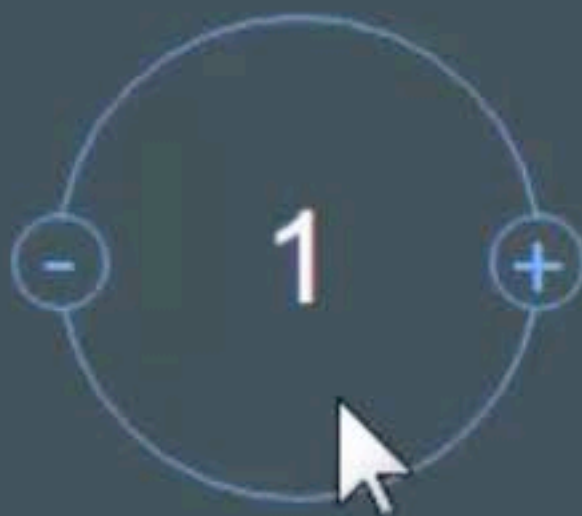
Monitoring

## Runtime



**BUILDPACK**

Liberty for Java™

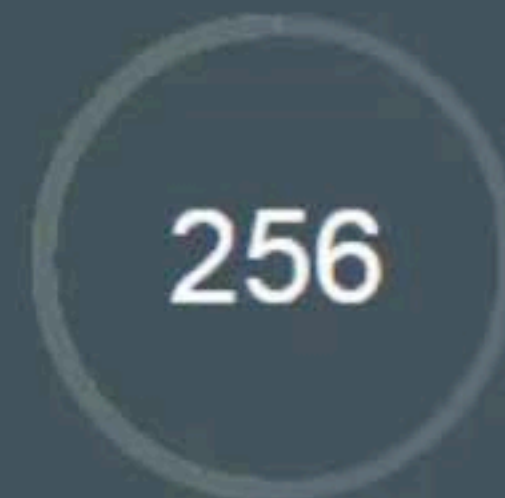


**INSTANCES**

All instances are running  
Health is 100%



**MB PER INSTANCE**



**TOTAL MB ALLOCATION**

524032 MB still available





Getting Started

Overview

Runtime

Connections

Logs

Monitoring

## Runtime



**BUILDPACK**

Liberty for Java™



**INSTANCES**

All instances are running  
Health is 100%



**MB PER INSTANCE**



**TOTAL MB ALLOCATION**

524032 MB still available



Getting Started

Overview

Runtime

Connections

Logs

Monitoring

## Runtime



**BUILDPACK**

Liberty for Java™



**INSTANCES**

Save

Reset



**MB PER INSTANCE**



**TOTAL MB ALLOCATION**

523776 MB still available



Getting Started

Overview

Runtime

Connections

Logs

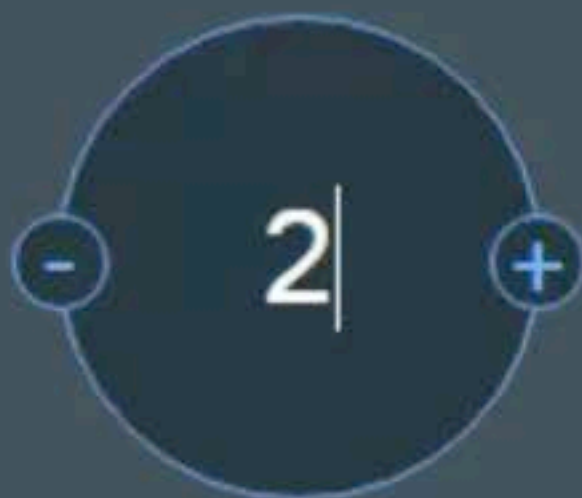
Monitoring

## Runtime



**BUILDPACK**

Liberty for Java™



**INSTANCES**

Save

Reset



**MB PER INSTANCE**



**TOTAL MB ALLOCATION**

523776 MB still available





Getting Started

Overview

Runtime

Connections

Logs

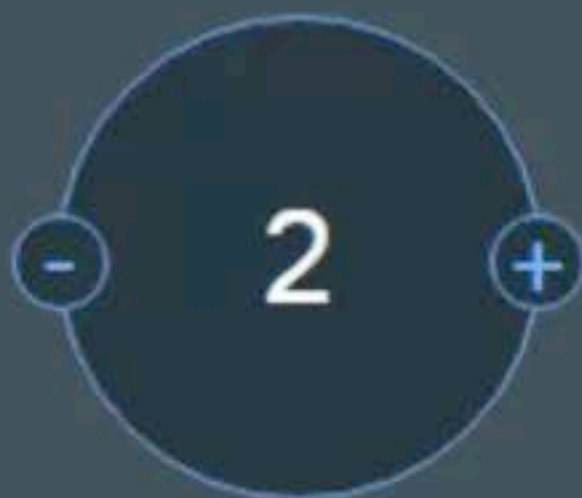
Monitoring

## Runtime

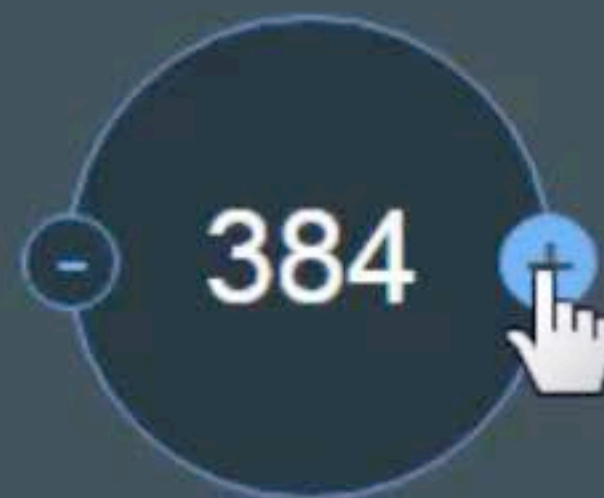


**BUILDPACK**

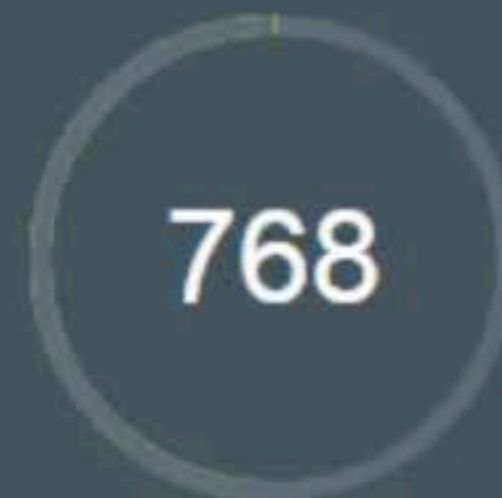
Liberty for Java™



**INSTANCES**



**MB PER INSTANCE**



**TOTAL MB ALLOCATION**

523520 MB still available

Save

Reset





Getting Started

Overview

Runtime

Connections

Logs

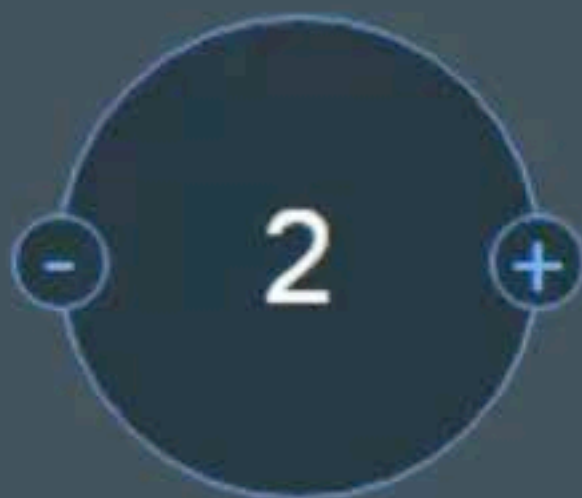
Monitoring

## Runtime



**BUILDPACK**

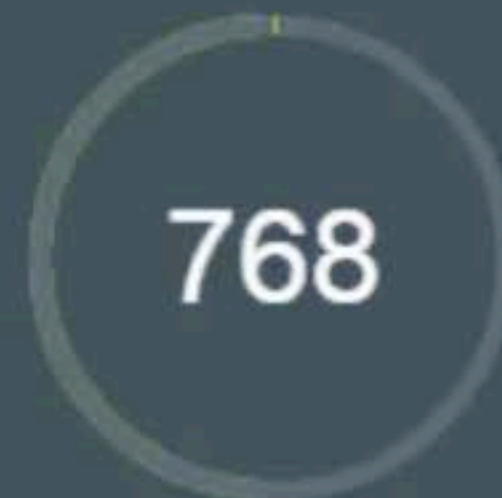
Liberty for Java™



**INSTANCES**



**MB PER INSTANCE**



**TOTAL MB ALLOCATION**

523520 MB still available

Save

Reset

```
$ cf app CloudRocketApp
Showing health and status for app CloudRocketApp in org Cloud Rocket / space Development as
OK
```

```
requested state: started
```

```
instances: 1/1
```

```
usage: 256M x 1 instances
```

```
urls: CloudRocket.mybluemix.net
```

```
last uploaded: Sun Nov 15 13:38:13 UTC 2015
```

```
stack: cflinuxfs2
```

```
buildpack: Liberty for Java(TM) (WAR, liberty-2015.10.0_0, ibmjdk-1.8.0_20150828, env)
```

	state	since	cpu	memory	disk	details
#0	running	2015-11-22 07:10:57 AM	0.6%	177.4M of 256M	178.5M of 1G	

```
$
```



```
0 of 1 instances running, 1 starting
0 of 1 instances running, 1 starting
0 of 1 instances running, 1 starting
1 of 1 instances running
```

App started

OK

App CloudRocketApp was started using this command ``.liberty/initial_startup.rb``

Showing health and status for app CloudRocketApp in org Cloud Rocket / space Development as  
OK

requested state: started

instances: 1/1

usage: 512M x 1 instances

urls: CloudRocket.mybluemix.net

last uploaded: Sun Nov 15 13:38:13 UTC 2015

stack: cflinuxfs2

buildpack: Liberty for Java(TM) (WAR, liberty-2015.10.0\_0, ibmjdk-1.8.0\_20150828, env)

	state	since	cpu	memory	disk	details
#0	running	2015-11-22 07:14:35 AM	0.0%	193.7M of 512M	177.3M of 1G	

\$

\$

```
$ cf scale CloudRocketApp -i 2  
Scaling app CloudRocketApp in org Cloud Rocket / space Development as  
OK
```

```
$
```



# Bluemix PaaS | Manifest file Scaling attributes

- Vertical scaling

Memory quota attribute

Disk quota attribute

- Horizontal scaling

Instances attribute



```
1 applications:
2 - path: JavaHelloWorldApp.war
3   memory: 512M
4   disk_quota: 1024M
5   instances: 1
6   domain: mybluemix.net
7   name: CloudRocketApp
8   host: CloudRocket
```

# Bluemix PaaS | Auto Scaling



Auto-Scaling

- Elastic process whereby more resources are provisioned as the load increases and de-provisioned as the demand for the resource slackens
  - Match the performance requirements
  - Meet the Service Level Agreement (SLA)
- A free Service that can be bound to an application

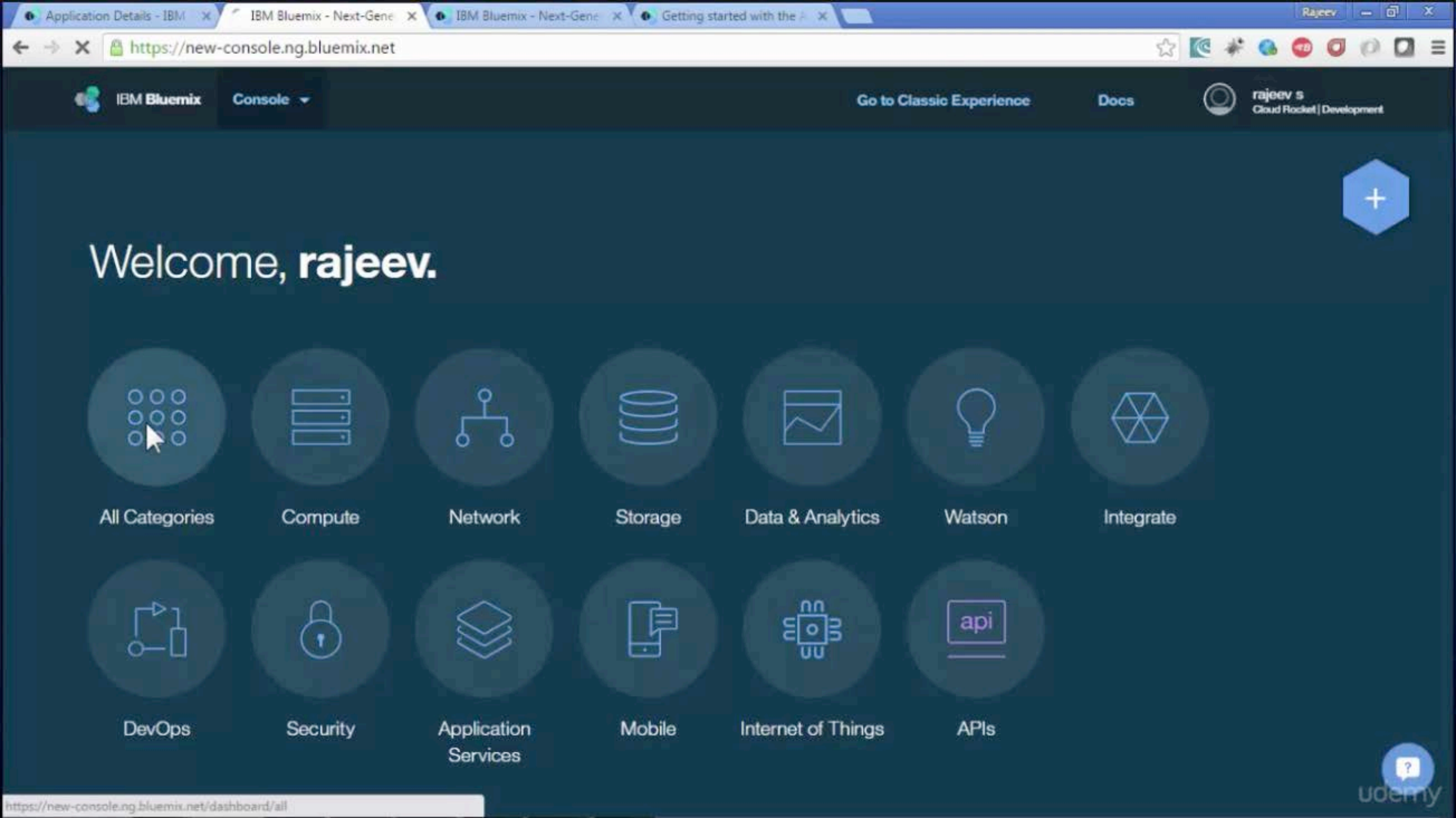
# Bluemix PaaS | Auto Scaling Policy



Auto-Scaling

1. Create & Bind app to the *Auto-Scaling* service
2. Set up the scaling Policy
  - Metrics based on Runtime  
e.g., Java you may use Heap, Throughput, Response time, Memory  
e.g., Node you may use Memory
  - Scale Out – specify the threshold for increasing the number of instances
  - Scale In – specify the threshold for decreasing the number of instances





Welcome, **rajeev.**



All Categories



Compute



Network



Storage



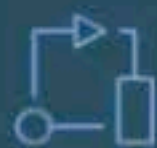
Data & Analytics



Watson



Integrate



DevOps



Security



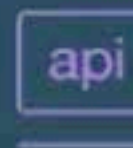
Application  
Services



Mobile



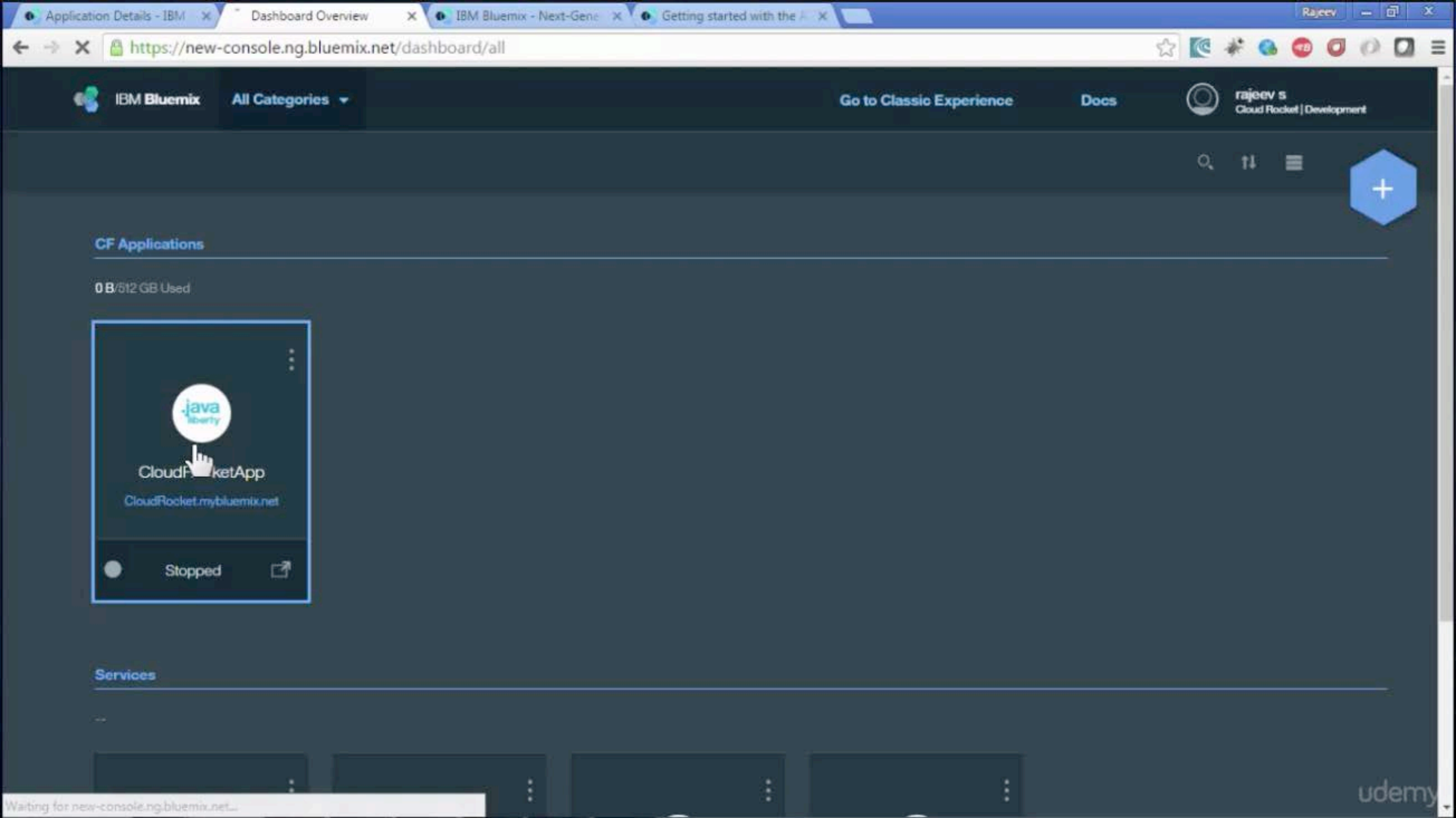
Internet of Things




APIs







← All Categories

 CloudRocketApp Status: ● Your app is stopped

[View App](#)


[Connections](#)

[Logs](#)

[Monitoring](#)


[Connect Existing](#)

[Connect New](#)




No Connected Services

Click the Connect Existing or Connect New button to add an item to your app.



← All Categories

 CloudRocketApp Status: ● Your app is stopped

[View App](#)

[Connections](#)

[Getting Started](#)

[Overview](#)


[Runtime](#)

[Logs](#)

[Monitoring](#)


[Connect Existing](#)

[Connect New](#)



No Connected Services

Click the Connect Existing or Connect New button to add an item to your app.





Application Details - IBMCatalog - IBM BluemixIBM Bluemix - Next-GenGetting started with the

https://new-console.ng.bluemix.net/catalog/?context=services&app=ebc74b1d-7fb1-41eb-a2c3-af1afdff78e6

IBM BluemixConsole

Go to Classic ExperienceDocs

rajeev sCloud Rocket | Development

All Categories >

Search allFilter

Network

Storage

Data & Analytics

Watson

Integrate

DevOps

Security

Application Services

Mobile

Internet of Things

APIs

Network

Content Delivery (CDN)Provides CDN functionalityIBM Beta

Virtual Private Network (VPN)Secured VPN connection for Containers.IBM

Storage

Object StorageProvides an unstructured cloud data store to build and deliver cloud applications.IBM

Data & Analytics

Apache SparkIBM Application for Apache Spark for

BigInsights for Apache HadoopProvision managed Hadoop on IBM Cloud

Cloudant NoSQL DBCloudant NoSQL DB provides access to a



Application Details - IBM

Catalog - IBM Bluemix

IBM Bluemix - Next-Gen

Getting started with the

https://new-console.ng.bluemix.net/catalog/?context=services&app=ebc74b1d-7fb1-41eb-a2c3-af1afdff78e6&category=devops

IBM Bluemix

Console

Go to Classic Experience

Docs

rajeev s  
Cloud Rocket | Development

Search all

Filter

All Categories

Network

Storage

Data & Analytics

Watson

Integrate

DevOps

Security

Application Services

Mobile

Internet of Things

APIs

**Active Deploy**  
Update your running apps with zero downtime, or quickly revert to your previous version.  
IBM Beta

**Auto-Scaling**  
Automatic increase or decrease the number of application instances based on demand.  
IBM

**Delivery Pipeline**  
Use IBM DevOps Services to automate builds and deployments, test execution, and deployment.  
IBM

**Globalization Pipeline**  
Quickly translate your web or mobile application UI into a selection of target languages.  
IBM Beta

**IBM Alert Notification**  
Never miss critical alerts. Notify the right people immediately. Speed up response time.  
IBM

**Monitoring and Analytics**  
Gain the visibility and control you need over your application. Determine the root cause of issues.  
IBM

**Track & Plan**  
Use IBM DevOps Services to create stories, tasks, and defects to describe your work.  
IBM

**BlazeMeter**  
Performance Testing Platform  
Third Party

**jKool**  
jKool provides real-time & historical visualization & analytics as a service.  
Third Party

**Load Impact**  
Performance and load testing for DevOps

**New Relic**  
Manage and monitor your apps

udemy

Application Details - IBM xAuto-Scaling - IBM Bluemix xIBM Bluemix - Next-Gen xGetting started with the x

https://new-console.ng.bluemix.net/catalog/services/auto-scaling/?context=services&app=ebc74b1d-7fb1-41eb-a2c3-af1afdff78e6

IBM BluemixConsoleGo to Classic ExperienceDocsrajeev sCloud Rocket | Development

View all

Auto-Scaling

X

The Auto-Scaling for Bluemix service enables you to automatically increase or decrease the compute capacity of your application. The number of application instances are adjusted dynamically based on the Auto-Scaling policy you define.

Connect to:

CloudRocketApp

[View Docs](#)

AUTHORIBM

PUBLISHED01/22/2016

TYPEService

LOCATIONUS South

Service name:

Auto-Scaling-Cloud-Rocket

Features

Dynamic scaling

Automatically add or remove resources to match the current workload.

Metric statistics

Visualize the current and historical values of performance metrics.

Custom scaling policy

Define policy on metrics of interest.

Scaling history

Query the scaling activities based on status, time and type.

Pricing Plans

Monthly prices shown are for country or region: [United States](#)

Plan

Features

Pricing



Application Details - IBM

Auto-Scaling - IBM Bluemix

IBM Bluemix - Next-Gen

Getting started with the

https://new-console.ng.bluemix.net/catalog/services/auto-scaling/?context=services&app=ebc74b1d-7fb1-41eb-a2c3-af1afdff78e6

IBM Bluemix

Console

Go to Classic Experience

Docs

rajeev s  
Cloud Rocket | Development

Connect to:  

CloudRocketApp

[View Docs](#)

AUTHORIBM

PUBLISHED01/22/2016

TYPEService

LOCATIONUS South

Features

- Dynamic scaling**  
Automatically add or remove resources to match the current workload.
- Custom scaling policy**  
Define policy on metrics of interest.
- Metric statistics**  
Visualize the current and historical values of performance metrics.
- Scaling history**  
Query the scaling activities based on status, time and type.

Pricing Plans

Monthly prices shown are for country or region: [United States](#)

Plan	Features	Pricing
✓ free		Free
<div>This is the free service plan for the Auto-Scaling service.</div>		

[Terms](#)

Need Help?  
Contact Bluemix Sales

Estimate Monthly Cost  
[Cost Calculator](#)

Create

?





# CloudRocketApp

Status: ● Your app is stopped

View App

Getting Started

Overview

Runtime

Connections

Logs

Monitoring

## Restage Application

Your 'CloudRocketApp' app must be restaged to use the new 'Auto-Scaling-Cloud-Rocket' service. Restaging makes this service available for use. Do you want to restage it now?

Cancel

Restage

View Credentials

Docs



# CloudRocketApp

Status: ● Your app is stopped

View App

Getting Started

Overview

Runtime

Connections

Logs

Monitoring

## Restage Application

Your 'CloudRocketApp' app must be restaged to use the new 'Auto-Scaling-Cloud-Rocket' service. Restaging makes this service available for use. Do you want to restage it now?

Cancel

Restage

View Credentials

Docs

← CF Applications



# CloudRocketApp

Status: ● Your app is stopped

View App

Getting Started Overview Runtime Connections Logs Monitoring

Connect Existing

Connect New



Auto-Scaling-Cloud-Ro...


Auto-Scaling  
free

View Credentials

[Docs](#)



← CF Applications


 CloudRocketApp

Status: ● Your app is stopped

[View App](#)

Getting Started Overview Runtime **Connections** Logs Monitoring

[Connect Existing](#) [Connect New](#)



Auto-Scaling-Cloud-Ro...

Auto-Scaling  
free

[View Credentials](#) [Docs](#)

Policy Configuration

Metric Statistics

Scaling History



No policy is defined for this application.

CREATE AUTO-SCALING POLICY



[◀ Back](#) | [Edit Auto-Scaling Policy](#)

## Default Instance Limits

Allowable maximum instance count

Default Minimum Instance Count

The default minimum number of application instances.

## Scaling Rule(s)

## ▼ Rule 1

Add 1 instance(s) if average Memory utilization exceeds 80% for 600 seconds.

Remove 1 instance(s) if average Memory utilization is below 30% for 600 seconds.

Metric Type:

Scale Out:

If average Memory utilization exceeds %, then increase  

Scale In:

If average Memory utilization is below %, then decrease  

► Advanced Configurations

ADD A RULE



## Default Instance Limits

Allowable maximum instance count

Default Minimum Instance Count

## Scaling Rule(s)

### ▼ Rule 1

Add 1 instance(s) if average Memory utilization exceeds 80% for 600 seconds.

Remove 1 instance(s) if average Memory utilization is below 30% for 600 seconds.

Metric Type:

Memory ▼

Scale Out:

JVM Heap

Memory

Throughput

ResponseTime

utilization exceeds  %, then increase  instance(s) ▼

Scale In:

If average memory utilization is below  %, then decrease  instance(s) ▼

► Advanced Configurations

ADD A RULE

► Schedule(s)

*There are no unsaved changes.*

SAVE

RESET

### Default Instance Limits

Allowable maximum instance count

Default Minimum Instance Count

### Scaling Rule(s)

#### ▼ Rule 1

Add 1 instance(s) if average Response Time exceeds 80ms for 600 seconds.

Remove 1 instance(s) if average Response Time is below 30ms for 600 seconds.

Metric Type:

Scale Out: If average Response Time exceeds  ms, then increase

Scale In: If average Response Time is below  ms, then decrease

► Advanced Configurations

ADD A RULE

► Schedule(s)

*There are unsaved changes.*

SAVE

RESET

## Default Instance Limits

Allowable maximum instance count

Default Minimum Instance Count

## Scaling Rule(s)

### ▼ Rule 1

Add 1 instance(s) if average JVM Heap utilization exceeds 80% for 600 seconds.

Remove 1 instance(s) if average JVM Heap utilization is below 30% for 600 seconds.

Metric Type:

Scale Out: If average JVM Heap utilization exceeds %, then increase

Scale In: If average JVM Heap utilization is below %, then decrease

► Advanced Configurations

ADD A RULE

► Schedule(s)

*There are unsaved changes.*

SAVE

RESET



## Default Instance Limits

Allowable maximum instance count

Default Minimum Instance Count

## Scaling Rule(s)

### ▼ Rule 1

Add 1 instance(s) if average JVM Heap utilization exceeds 80% for 600 seconds.

Remove 1 instance(s) if average JVM Heap utilization is below 30% for 600 seconds.

Metric Type:

Scale Out: If average JVM Heap utilization exceeds %, then increase

Scale In: If average JVM Heap utilization is below %, then decrease

► Advanced Configurations

ADD A RULE

► Schedule(s)

✓ Scaling policy is created.

✕

# Breach Duration & Cooldown Period



Auto-Scaling

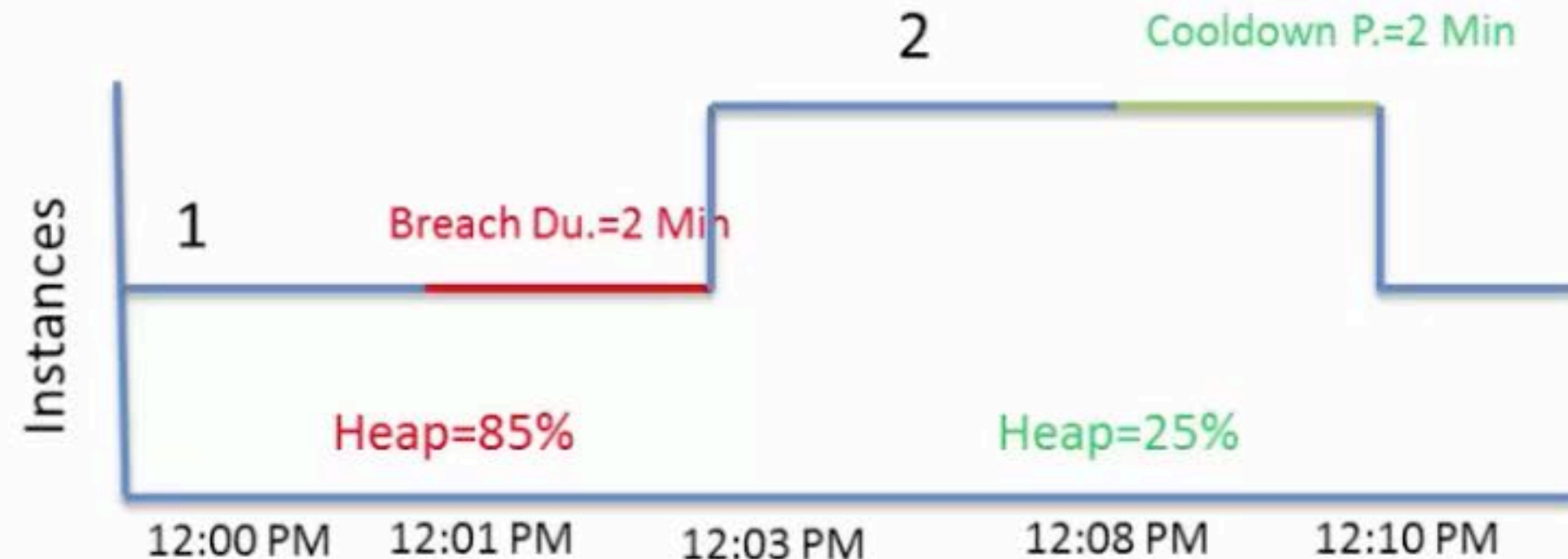
## Breach Duration

- Wait time before Scale out kicks in

## Cooldown Period

- Wait time before Scale in kicks in

Reactive Scaling



## Default Instance Limits

Minimum Instance Count

Maximum Instance Count

## Scaling Rule(s)

### ▼ Rule 1

Add 1 instance(s) if average JVM Heap utilization exceeds 80% for 600 seconds.

Remove 1 instance(s) if average JVM Heap utilization is below 30% for 600 seconds.

Metric Type:

JVM Heap ▼

Scale Out:

If average JVM Heap utilization exceeds  %, then increase 

instance(s) ▼

Scale In:

If average JVM Heap utilization is below  %, then decrease 

instance(s) ▼

Advanced Configurations

ADD A RULE

## ► Schedule(s)

✓ Scaling policy is created.

×

*There are unsaved changes.*

SAVE

RESET



## Default Instance Limits

Minimum Instance Count

Maximum Instance Count

## Scaling Rule(s)

### ▼ Rule 1

Add 1 instance(s) if average JVM Heap utilization exceeds 80% for 600 seconds.

Remove 1 instance(s) if average JVM Heap utilization is below 30% for 600 seconds.

Metric Type:

JVM Heap ▼

Scale Out:

If average JVM Heap utilization exceeds  %, then increase  instance(s) ▼ .

Scale In:

If average JVM Heap utilization is below  %, then decrease  instance(s) ▼ .

### ▼ Advanced Configurations

Statistic Window:

seconds (30~1800)

Breach Duration:

seconds (30~36000)

### Proactive Scaling

Cooldown period for scaling out:

seconds (30~3600)

Cooldown period for scaling in:

seconds (30~3600)

The time period after a scaling in activity ends and before another scaling activity can start.

ADD A RULE

► Schedule(s)

Rule 1  
Add 1 instance(s) if average JVM Heap utilization exceeds 80% for 600 seconds.  
Remove 1 instance(s) if average JVM Heap utilization is below 30% for 600 seconds.

Metric Type:

JVM Heap ▼

Scale Out:

If average JVM Heap utilization exceeds 80 %, then increase 1 instance(s) .

Scale In:

If average JVM Heap utilization is below 30 %, then decrease 1 instance(s) .

► Advanced Configurations

ADD A RULE

▼ Schedule(s)

Time Zone (GMT -12:00) Etc/GMT+12 ▼

Recurring Schedules

Start Time	End Time	Repeat On	Minimum Instance Count	Maximum Instance Count	Action
00:00 ▼	23:59 ▼	▶ Everyday	1	1	×

ADD A RECURRING SCHEDULE

Specific Date(s)

The specific date schedule takes precedence of the recurring schedule if there is overlap.

Start Date & Time	End Date & Time	Minimum Instance Count	Maximum Instance Count	Action
4/5/2016 ▼ 00:00 ▼	4/5/2016 ▼ 23:59 ▼	1	1	×

# Summary

1. Bluemix supports vertical and horizontal scaling of applications
2. Auto scaling allows policies to be setup for automatic scaling out/in
  - Breach duration – wait time before scale out occurs
  - Cooldown period – wait time before scale in occurs
3. Scheduled scaling is an example of Pro-active scaling

