Cloud Developers Certification Training

Section 3.2 - Summarize the Steps in building a Scalable Application

Lab Exercise

Version:	1
Last modification date:	2-Sep-15
Owner:	IBM Ecosystem Development
Source:	IBM Redbooks (red5184)

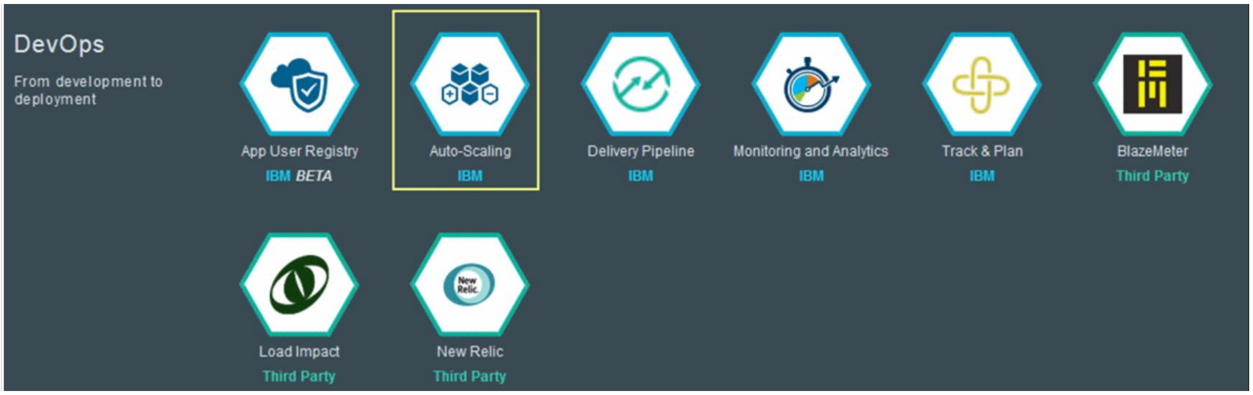
Exercise 3.2 – Lab Prerequisites

- Have a IBM Bluemix account
  - Sing up for Bluemix http://bluemix.net
- A Web Browser supported by Bluemix
  - Chrome, latest version for your OS
  - Firefox, latest version for your OS and ESR 31 or ESR 38
  - Internet Explorer, version 10 and 11
  - Safari, latest version for Mac
- **Build a sample java application to configure autoscaling service**
- **OR Use the existing java application for configuring autoscaling service**
- **OR Use Java Cache Web Starter Boiler Plate to configure autoscaling service**

Exercise 3.2.1 – Add Auto Scaling Service to your application

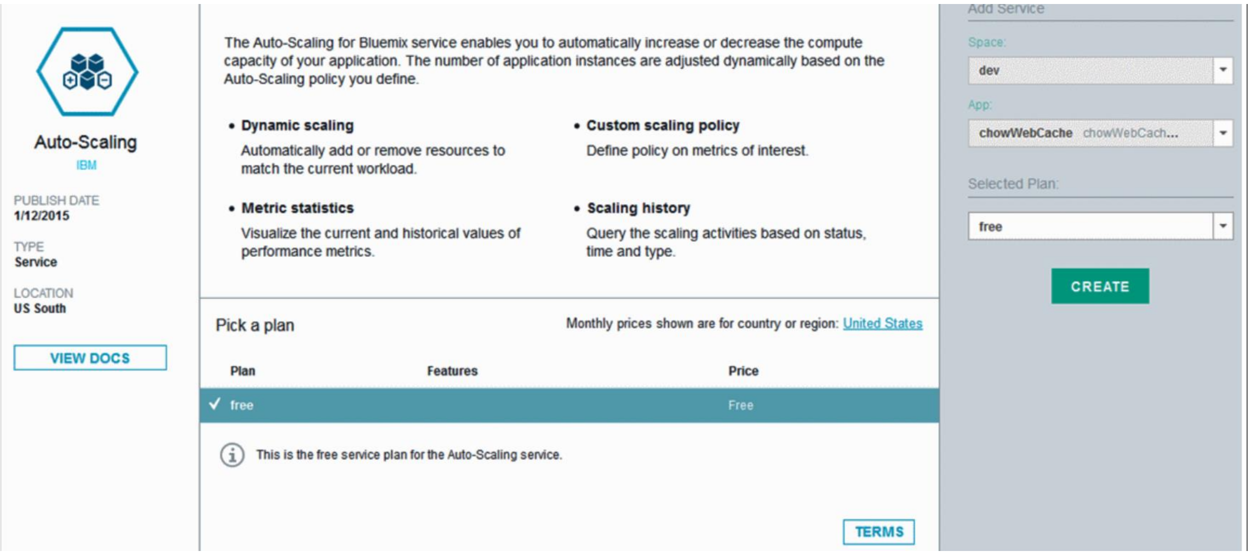
Complete the following steps:

1. Go to Bluemix Dashboard and click the Bluemix-Sample-Java application.
2. Click Add a Service and select the Auto-Scaling service (Figure 3-1).



3. If you have multiple apps, select the Bluemix-Sample-Java app, and click CREATE

4. If you see a pop-up message for Restage, click RESTAGE. Your new Auto-Scaling service will now be created and bind to your application.



Auto-Scaling
IBM

PUBLISH DATE
1/12/2015

TYPE
Service

LOCATION
US South

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.

- **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.

Pick a plan

Monthly prices shown are for country or region: United States

Plan	Features	Price
✓ free		Free

This is the free service plan for the Auto-Scaling service.

Add Service

Space: dev

App: chowWebCache chowWebCach...

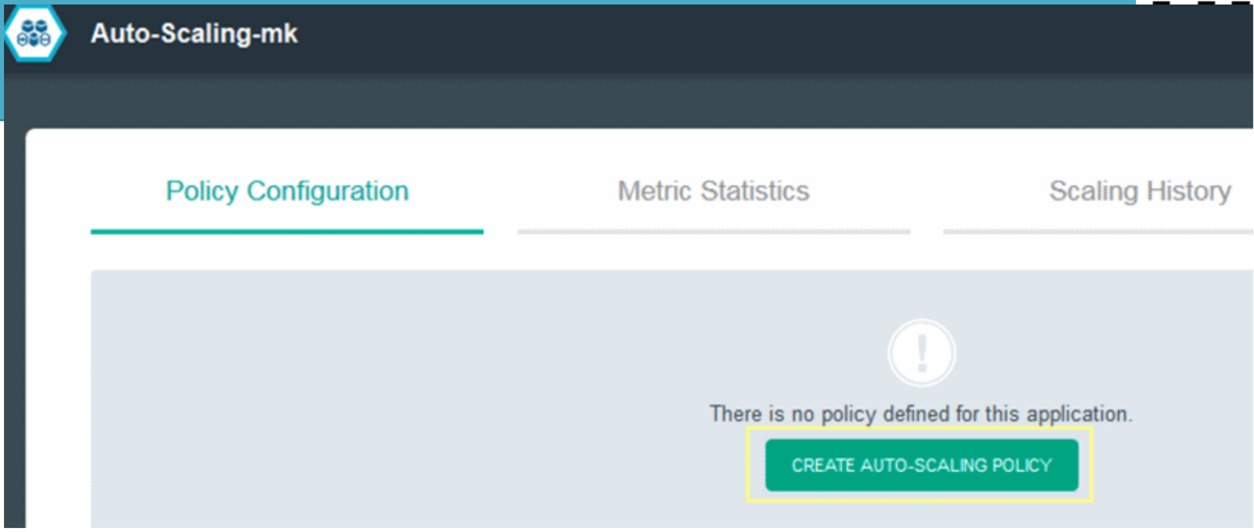
Selected Plan: free

Exercise 3.2.2 -Create and modify Auto-scaling polices

With an Auto-scaling policy, you can create rules that determine when the Auto-Scaling service will increase or decrease the number of instances of your application. In this task, you create and modify rules for your Java Liberty application.

Complete the following steps:

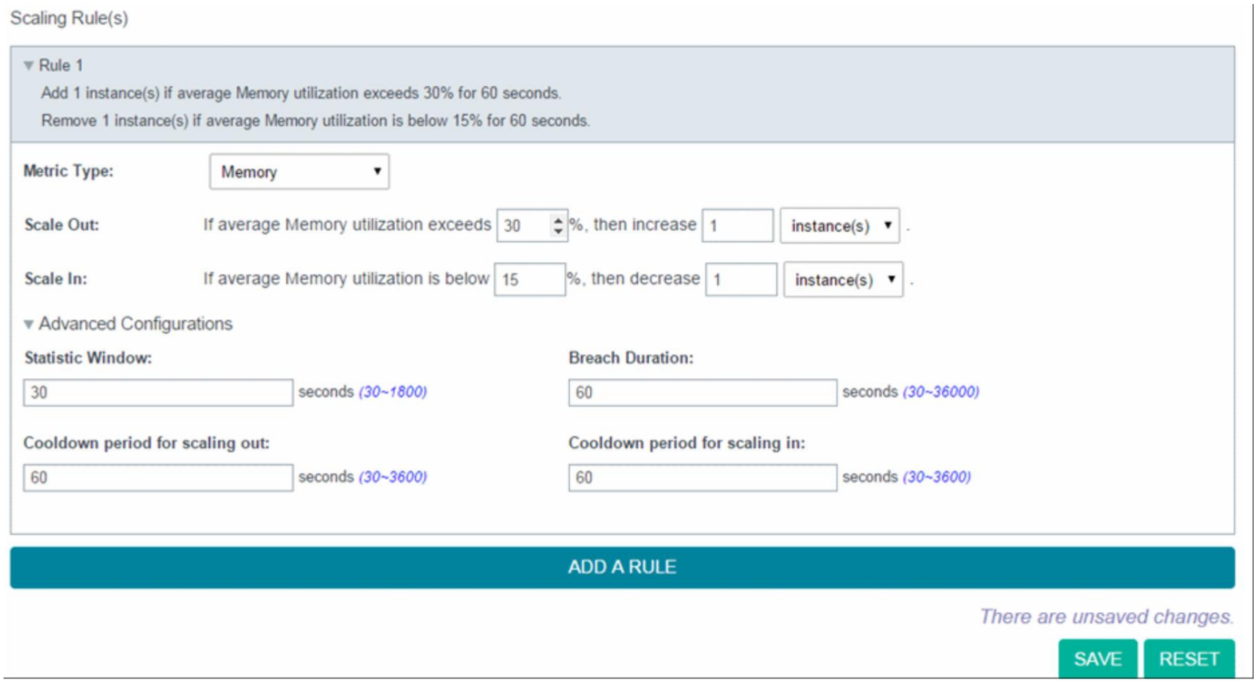
1. Within your application, click the **Auto-Scaling** service.
2. Click **CREATE AUTO-SCALING POLICY**


Auto-Scaling-mk

There is no policy defined for this application.

3. Enter the parameters for the policy as follows:
   a. Enter the name.
   b. Enter minimum and maximum number of application instances:
      - Use the default configuration or configure as mentioned below.
      - Minimum: If the number of the instances equals this value, the Auto-Scaling service will not scale in the application any more.
      - Maximum: If the number of the instances equals this value, the Auto-Scaling service will not scale out the application any more.
   c. Scroll to the Rule1 section and make these changes:
      - Keep the Metric Type set to **Memory**.
      - Change Scale Out percentage to 30, and change Scale In to 15.
      - Expand the **Advanced Configurations** section.
      - Set Statistic Window to 30 seconds.
      - Set the value of 60 seconds for Breach Duration and both Cooldown periods. We use these smaller time frames for testing later.

        Note: These settings are much lower than in a typical production application. The values used are low enough to ensure that the Auto-Scaling service properly scales your application within a shorter test window during progression of this lab.


Scaling Rule(s)

▾ Rule 1

Add 1 instance(s) if average Memory utilization exceeds 30% for 60 seconds.

Remove 1 instance(s) if average Memory utilization is below 15% for 60 seconds.

Metric Type: Memory

Scale Out: If average Memory utilization exceeds 30 %, then increase 1 instance(s).

Scale In: If average Memory utilization is below 15 %, then decrease 1 instance(s).

▾ Advanced Configurations

Statistic Window: 30 seconds (30~1800)

Breach Duration: 60 seconds (30~36000)

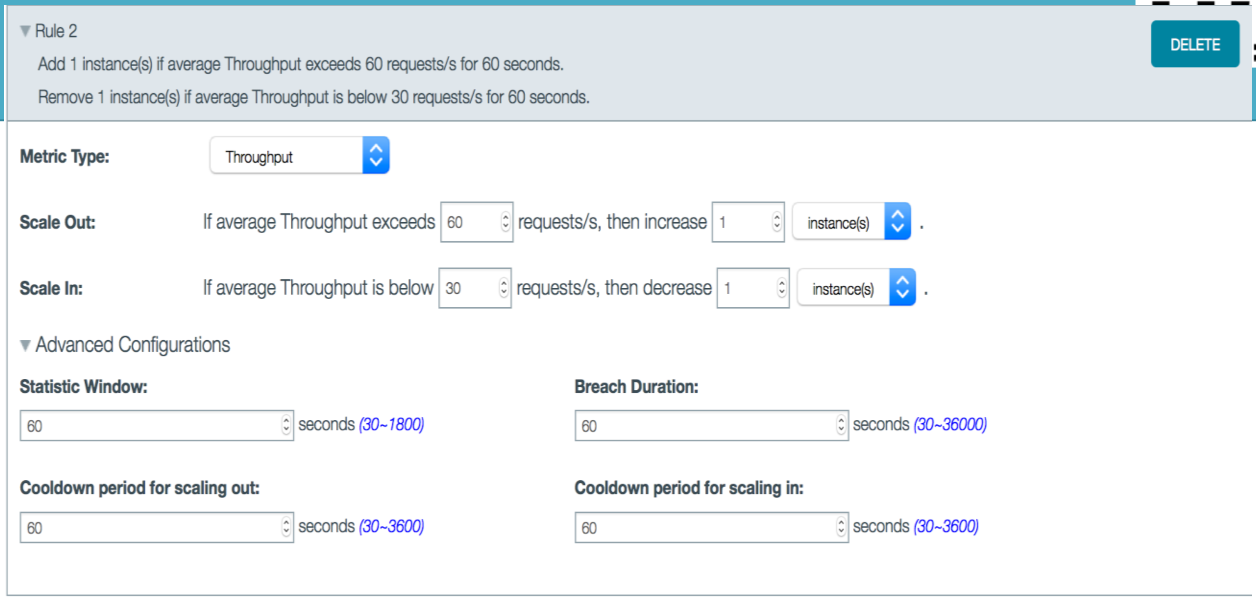
Cooldown period for scaling out: 60 seconds (30~3600)

Cooldown period for scaling in: 60 seconds (30~3600)

There are unsaved changes.

4. Click **ADD A RULE** to specify another rule.
5. This time, for Rule2 set the Metric Type to **throughput**. Keep the other values as

   they were for the previous rule, Rule 1.



▾ Rule 2

Add 1 instance(s) if average Throughput exceeds 60 requests/s for 60 seconds.

Remove 1 instance(s) if average Throughput is below 30 requests/s for 60 seconds.

Metric Type: Throughput

Scale Out: If average Throughput exceeds 60 requests/s, then increase 1 instance(s) .

Scale In: If average Throughput is below 30 requests/s, then decrease 1 instance(s) .

▾ Advanced Configurations

Statistic Window: 60 seconds *(30~1800)*

Breach Duration: 60 seconds *(30~36000)*

Cooldown period for scaling out: 60 seconds *(30~3600)*

Cooldown period for scaling in: 60 seconds *(30~3600)*

Your application can now automatically scale based on the policies you defined.

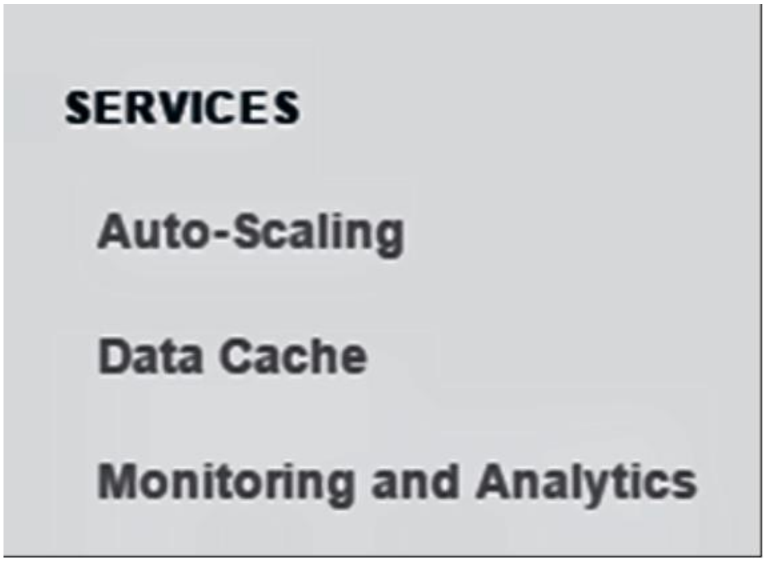
Exercise 3.2.3: View Auto-Scaling metric statistics and history

You can examine more closely, the effects of the test on your Auto-Scaling policy rules (set in 3.2.2; Create and modify Auto-scaling polices" The metrics in this task show how effectively your Auto-Scaling policy handled.

Complete the following steps:

1. Return to the Bluemix Dashboard, and click an application.

2. Services are listed in the left navigation bar. Click **Auto-Scaling** as shown in the figure below



3. Click the **Metric Statistics** tab to see graphs of usage statistics overtime for your Bluemix application. Scroll through to see all the graphs. If you set up any scaling rules for a specific property, upper and lower threshold indicators show when your application will create or remove additional instances.

   Figure below shows Throughput Metrics

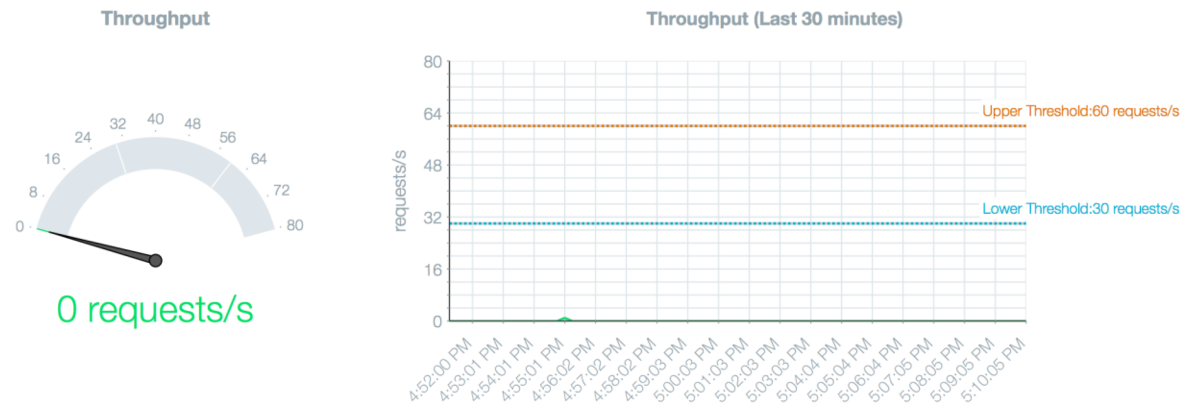
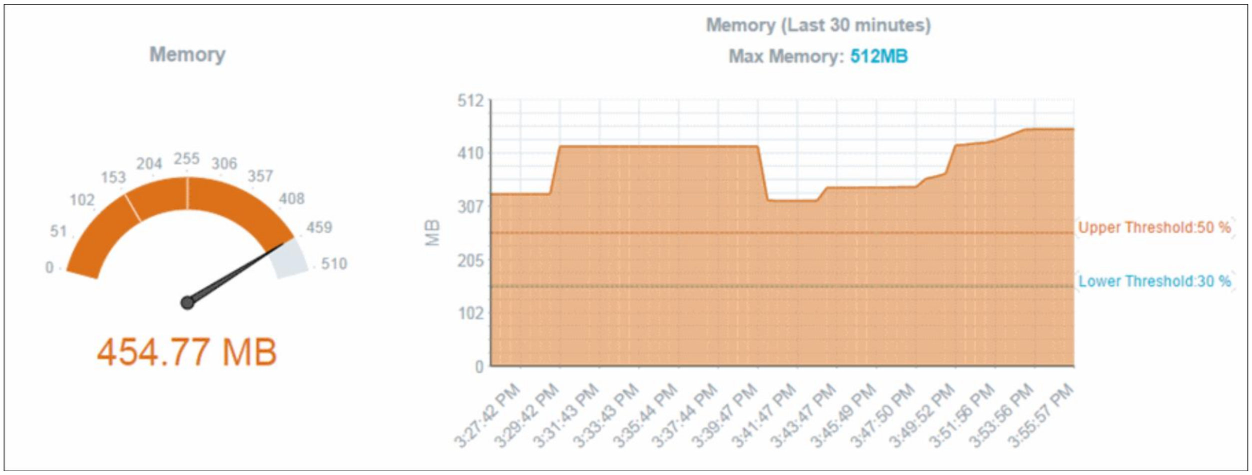
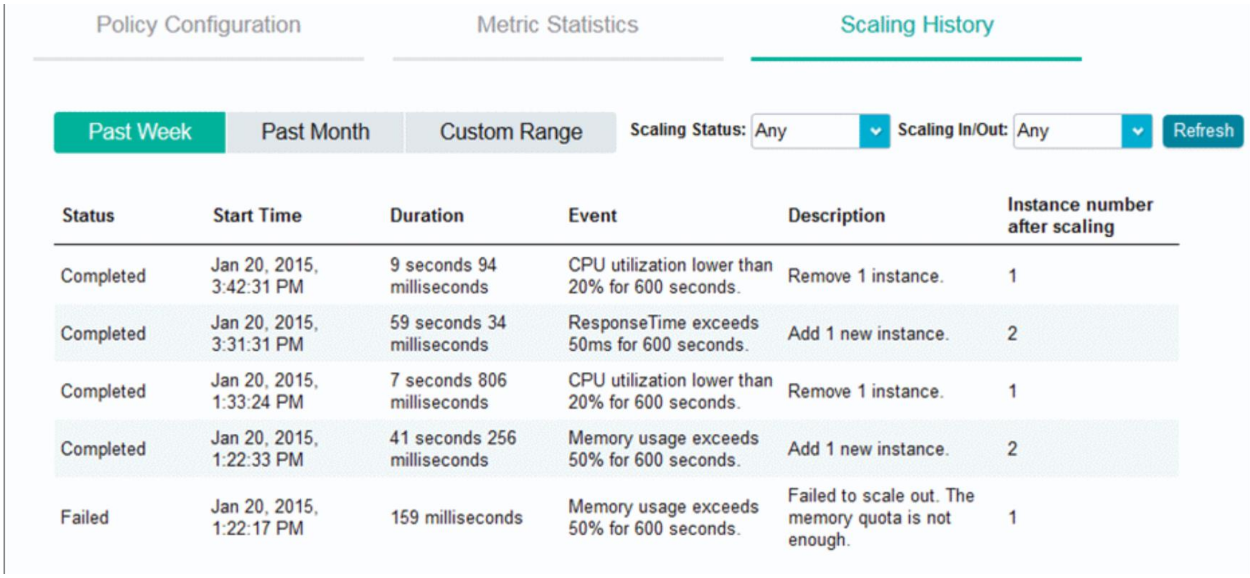


Figure below shows Memory Metrics



If your application is stressed enough, you see that the scaling polices you defined will take effect. Depending on the conditions, you see that the applications scale out and scale in. To view the scaling history, click the **Scaling History** tab as shown below

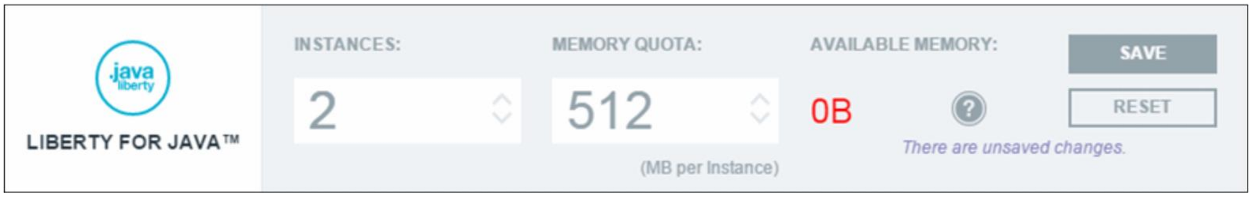


Scaling Status: Any

Scaling In/Out: Any

Status	Start Time	Duration	Event	Description	Instance number after scaling
Completed	Jan 20, 2015, 3:42:31 PM	9 seconds 94 milliseconds	CPU utilization lower than 20% for 600 seconds.	Remove 1 instance.	1
Completed	Jan 20, 2015, 3:31:31 PM	59 seconds 34 milliseconds	ResponseTime exceeds 50ms for 600 seconds.	Add 1 new instance.	2
Completed	Jan 20, 2015, 1:33:24 PM	7 seconds 806 milliseconds	CPU utilization lower than 20% for 600 seconds.	Remove 1 instance.	1
Completed	Jan 20, 2015, 1:22:33 PM	41 seconds 256 milliseconds	Memory usage exceeds 50% for 600 seconds.	Add 1 new instance.	2
Failed	Jan 20, 2015, 1:22:17 PM	159 milliseconds	Memory usage exceeds 50% for 600 seconds.	Failed to scale out. The memory quota is not enough.	1

As Auto-Scaling creates or removes instances of your application, you can see these changes reflected through your apps page from the Bluemix dashboard



INSTANCES: 2

MEMORY QUOTA: 512 (MB per Instance)

AVAILABLE MEMORY: 0B

There are unsaved changes.