



Scaling Section 3.2

Presented by:

IBM Cloud Ecosystem
Development

IBM **Cloud**

Objective

- Understand scaling concepts for a Cloud application and steps to scale an application in IBM Bluemix PaaS
 - Vertical scaling by increasing resources to an application instance
 - Horizontal scaling by increasing the number of application instance
 - Understand how to manually scale applications through IBM Bluemix PaaS dashboard
 - Automatically scaling applications in IBM Bluemix PaaS using the Auto-Scaling service and scaling policy fields and options such as: available metric types for runtimes, breach duration, cooldown period

What is Scaling?

kind of on demand

Capacity to alter computing resources manually (by executing a command on a command line or through a web interface) or automatically (through predefined changes in capacity or through software that automatically adjusts capacity to meet actual demand) is called scaling.

Cloud empowers you to alter your computing resources to meet your load requirements. The ability to manually adjust capacity is a huge advantage over traditional computing.

Types of Scaling

Vertical Scaling :

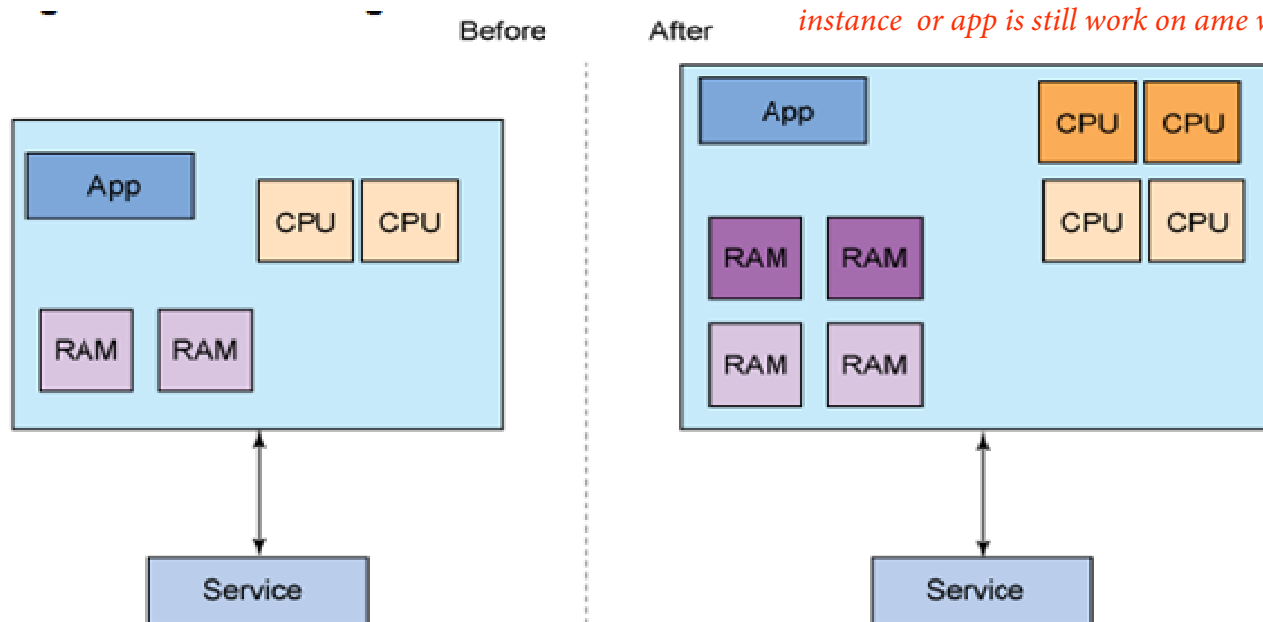
one server when increase vertically memory side is increase cpu is increase but still same instance

more mem and cou but same number of instance

Vertical scaling is often referred to as *scaling up*. Vertical scaling increases the resources available to an application by adding capacity directly to the individual nodes — for example, adding additional memory or increasing the number of CPU cores. Picture below shows the concept of vertical scaling with the addition of both memory and CPU to an application.

not need special app or server to balance load that direct on machine

instance or app is still work on ame way but performance is higher



if use same app without reboot this type is better

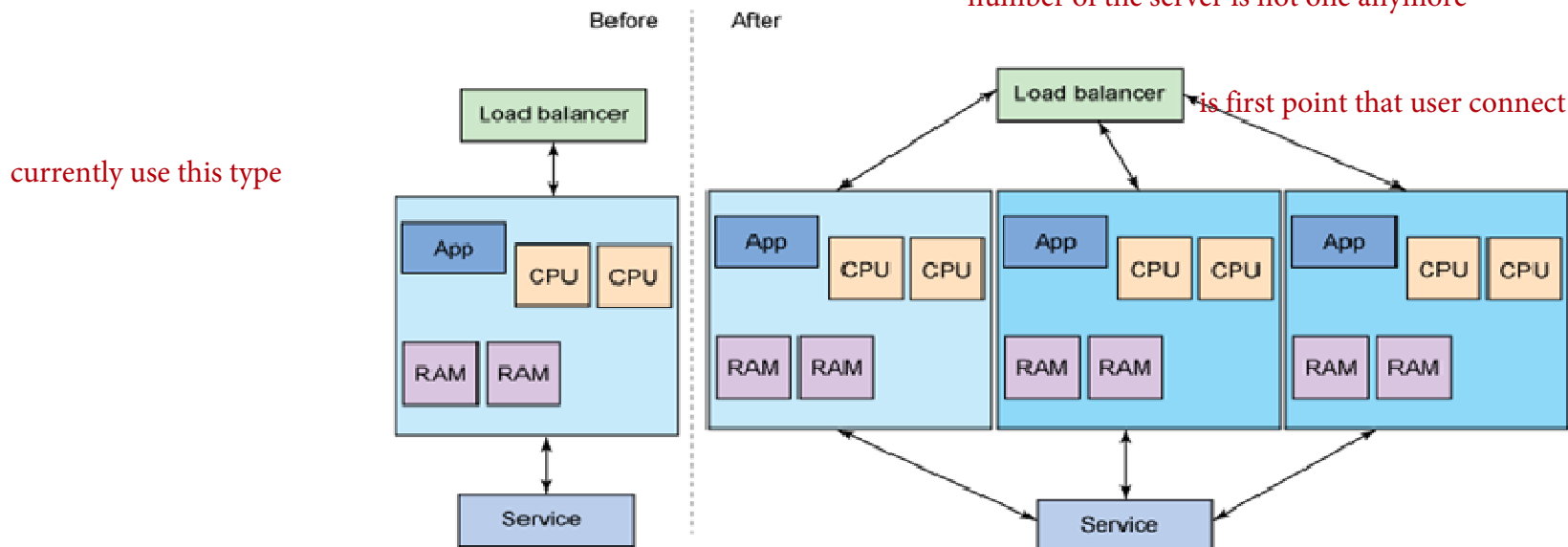
Types of Scaling (Cont'd)

Horizontal Scaling:

scale number of instance

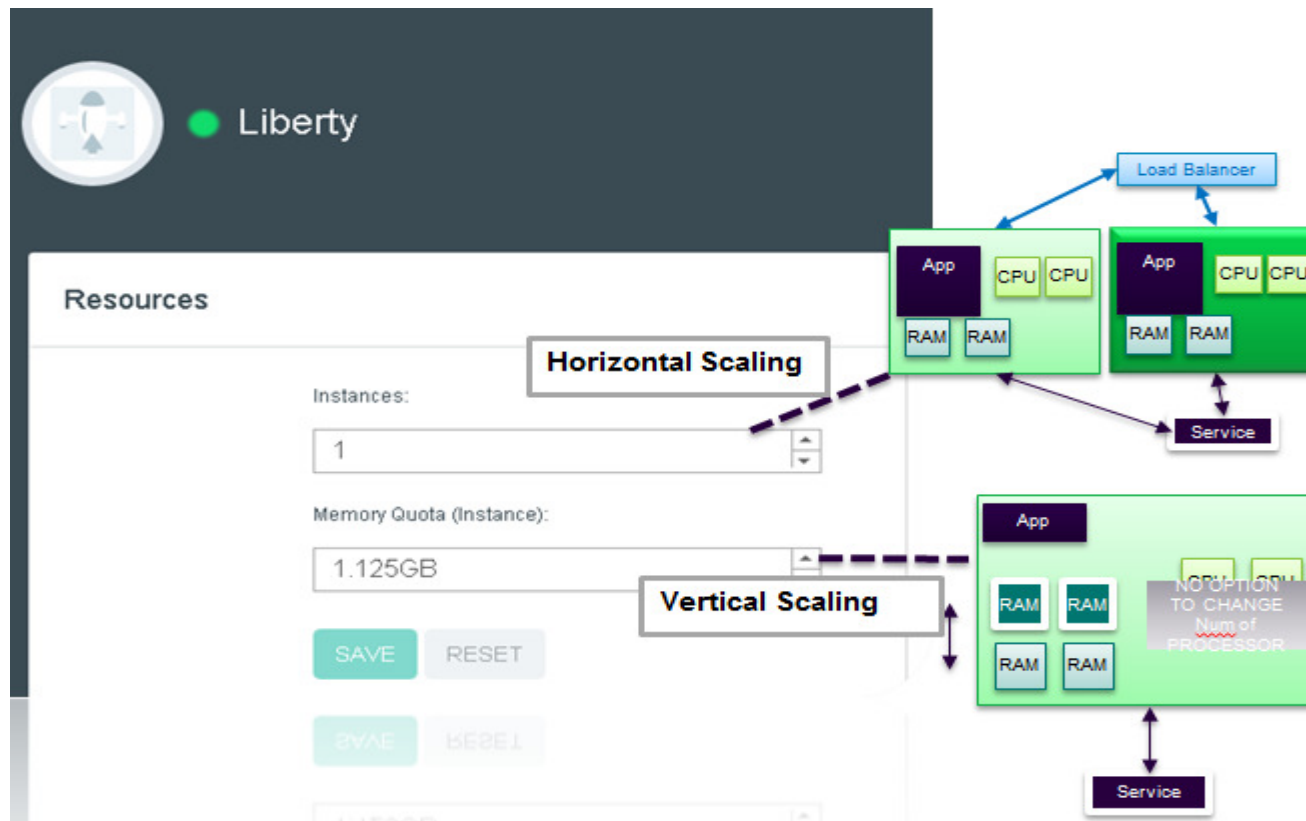
Horizontal scaling is often referred to as *scaling out*. The overall application resource capacity grows through the addition of entire nodes. Each additional node adds equivalent capacity, such as the same amount of memory and the same CPU. Horizontal scaling typically is achievable without downtime. Picture below illustrates the concept of horizontal scaling; you see additional identical nodes added with a load balancer in front of the application nodes.

create more number of machine but when create you can not just let user to connect or choose which one the user have to connect to the server because number of the server is not one anymore



Steps to Manually Scale in IBM Bluemix PaaS

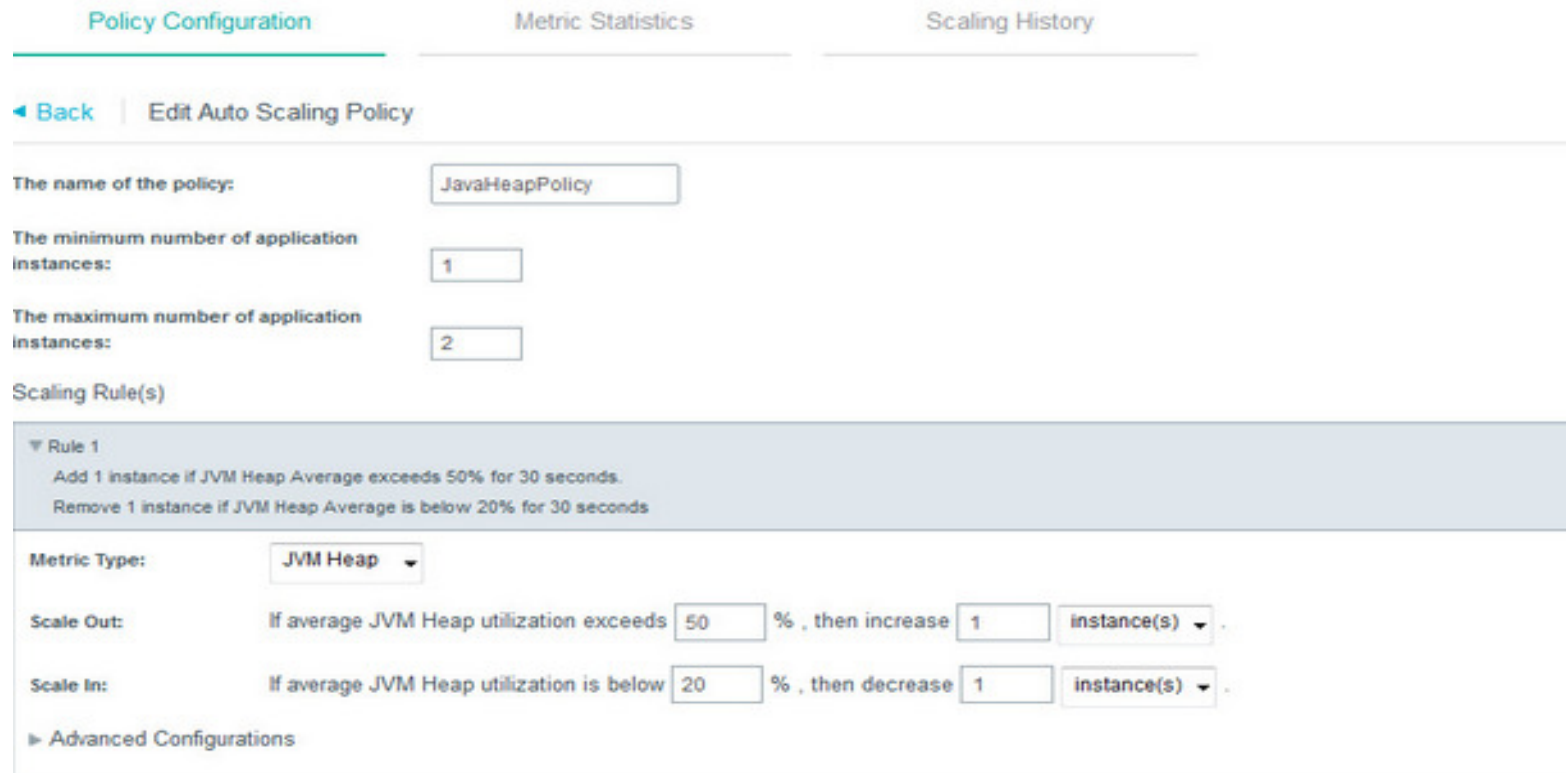
The IBM Bluemix UI Dashboard and command line support both vertical and horizontal scaling through increasing the amount of memory and increasing the number of instances of an application runtime. Both techniques can be applied to the same application.



Auto-Scale in IBM Bluemix

▪ Auto-Scale in Bluemix - Policy Configuration

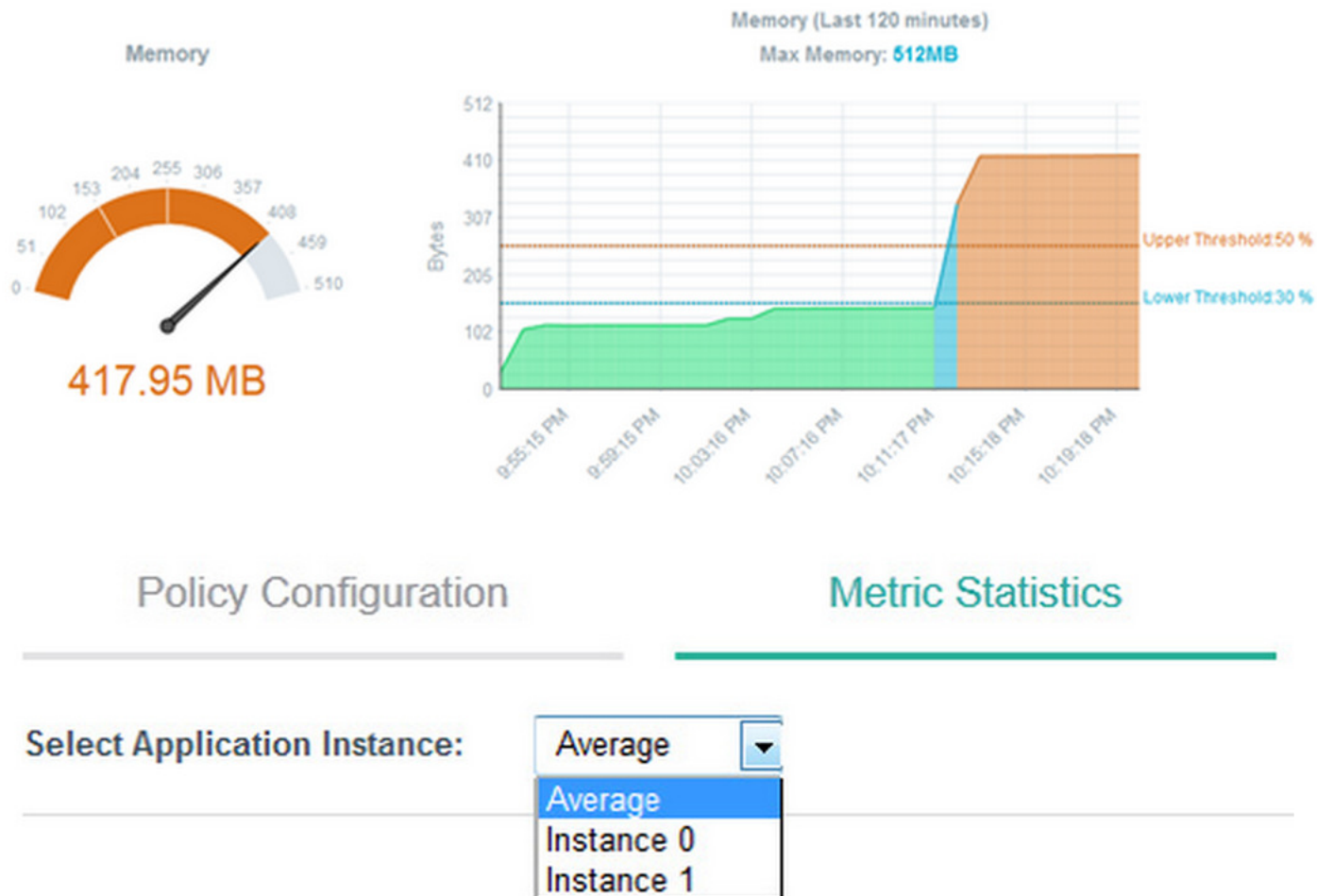
- Where you can create or edit the scaling policy that contains instructions on how to trigger the scaling activities. For Liberty for Java™ applications, you can define scaling rules for Throughput, JVM Heap, Memory, and Throughput.
- For Node.js applications, you can define scaling rules for Throughput and Memory.
- For Ruby applications, you can define scaling rules for Memory.



The screenshot shows the 'Policy Configuration' tab of the IBM Bluemix Auto-Scale interface. At the top, there are three tabs: 'Policy Configuration' (active), 'Metric Statistics', and 'Scaling History'. Below the tabs, there is a navigation bar with a 'Back' button and the title 'Edit Auto Scaling Policy'. The main configuration area includes the following fields:

- The name of the policy:** A text input field containing 'JavaHeapPolicy'.
- The minimum number of application instances:** A numeric input field containing '1'.
- The maximum number of application instances:** A numeric input field containing '2'.
- Scaling Rule(s):** A section with a dropdown arrow and the text 'Rule 1'. Below it, the rule description reads: 'Add 1 instance if JVM Heap Average exceeds 50% for 30 seconds. Remove 1 instance if JVM Heap Average is below 20% for 30 seconds'.
- Metric Type:** A dropdown menu with 'JVM Heap' selected.
- Scale Out:** A configuration line: 'If average JVM Heap utilization exceeds 50 % , then increase 1 instance(s)'.
- Scale In:** A configuration line: 'If average JVM Heap utilization is below 20 % , then decrease 1 instance(s)'.
- Advanced Configurations:** A link with a right-pointing arrow.

Viewing Metrics in Auto-Scaling Service



Viewing History of Scaling in Auto-Scaling Service

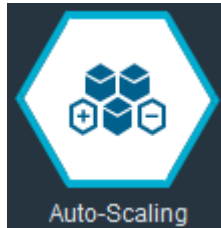
Policy Configuration		Metric Statistics		Scaling History		
Past Week		Past Month		Custom Range		
Time Zone: (GMT +08:00) Asia/Shanghai		Scaling Status: Any		Scaling In/Out: Any		Refresh
Status	Start Time	Duration	Event	Description	Instance number after scaling	
Completed	Jul 1, 2015, 10:19:36 PM	18 seconds 319 milliseconds	Memory usage exceeds 50% for 120 seconds.	Add 1 new instance.	3	
Completed	Jul 1, 2015, 10:15:18 PM	15 seconds 242 milliseconds	Memory usage exceeds 50% for 120 seconds.	Add 1 new instance.	2	

Bookmarks

- <https://www.ng.bluemix.net/docs/services/Auto-Scaling/index.html>
- <http://www.ibm.com/developerworks/cloud/library/cl-bluemix-autoscale/>
- <http://www.ibm.com/developerworks/cloud/library/cl-autoscale-app/index.html>

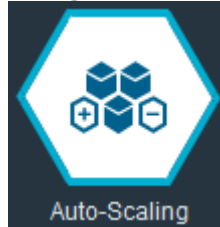
Backup – script for demonstration and lab exercise

Using IBM Bluemix Auto-Scaling Lab Exercise



1. Adding an Auto-Scaling service to your application
2. Creating and modifying auto-scaling policies
3. Adding Load Impact service and account creation
4. Creating a user scenario
5. Create and run a test configuration
6. Load Impact test results and adding graphs
7. Auto-Scaling metric statistics and history
8. Monitoring and analytics performance

Using IBM Bluemix Auto-Scaling Demonstration Video



<https://www.youtube.com/watch?v=boqAWY99CVE>

Bookmarks

<https://www.12factor.net>