(5 of 12)

Updated: Jan 9th, 2017

IBM Bluemix
Development & Certification

Summary decks for a course that covers the A to Z of IBM Bluemix.

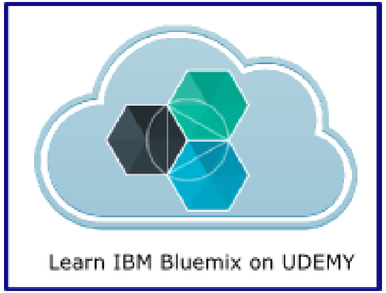
For more information visit:
http://www.acloudfan.com

raj@acloudfan.com

1. Debugging
2. Scaling
3. Logging
4. Testing
5. Monitoring

PS: Certification practice test questions NOT available in the summary decks

Discounted access to the courses:



https://www.udemy.com/ibm-bluemix/?couponCode=BLUE100

Coupon Code = **BLUE100**

https://www.udemy.com/rest-api/?couponCode=REST100

Coupon Code = **REST100**

PS:

- For latest coupons & courses please visit: http://www.acloudfan.com
- Enter to **WIN Free access** – please visit: http://www.acloudfan.com/win-free-access

raj@acloudfan.com
http://www.acloudfan.com

Debugging

Debugging tools

- 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

Scaling

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 or cf commands
2. Scaling through the Manifest file
3. Using the Auto scaling service

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

Breach Duration & Cooldown Period

Auto-Scaling

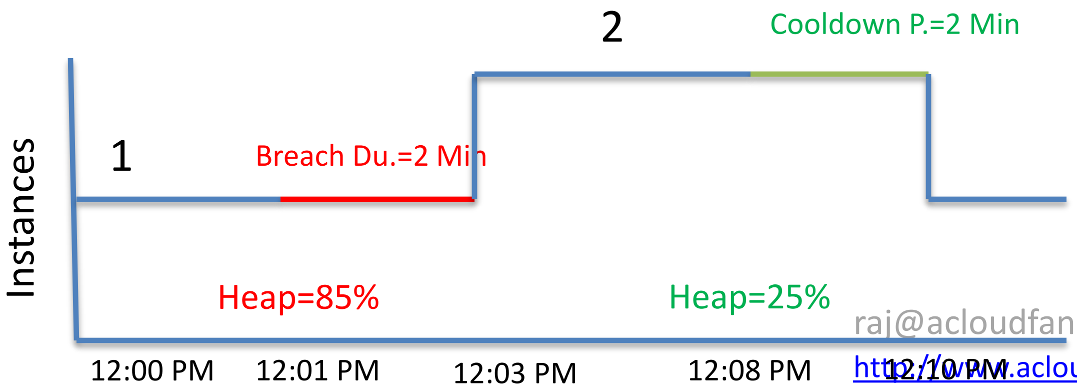
Breach Duration

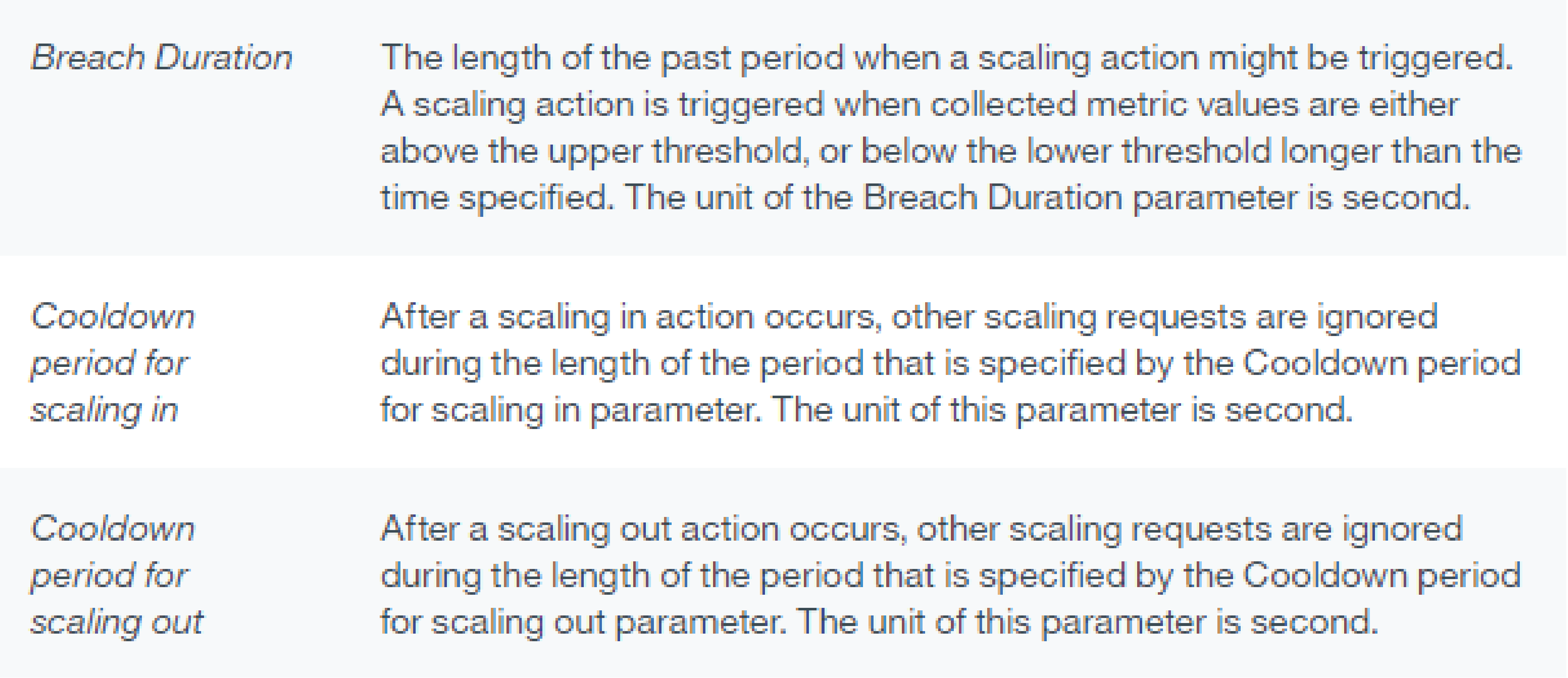
- Wait time before **Scale out** kicks in

Cooldown Period

Reactive Scaling

- Wait time before **Scale in** kicks in





Breach Duration	The length of the past period when a scaling action might be triggered. A scaling action is triggered when collected metric values are either above the upper threshold, or below the lower threshold longer than the time specified. The unit of the Breach Duration parameter is second.
Cooldown period for scaling in	After a scaling in action occurs, other scaling requests are ignored during the length of the period that is specified by the Cooldown period for scaling in parameter. The unit of this parameter is second.
Cooldown period for scaling out	After a scaling out action occurs, other scaling requests are ignored during the length of the period that is specified by the Cooldown period for scaling out parameter. The unit of this parameter is second.

https://console.ng.bluemix.net/docs/services/Auto-Scaling/index.html

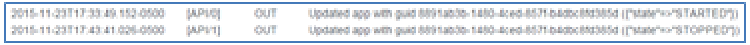
Logging

Log types

- API

  Users make API calls to request changes in application state

  Cloud controller sends these messages

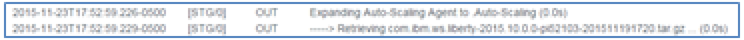


```
2015-11-23T17:33:49.152-0500   [API/0]   OUT   Updated app with guid 8891ab3b-1480-4ced-857f-b4dbc8fd385d ({"state"=>"STARTED"})
2015-11-23T17:43:41.026-0500   [API/1]   OUT   Updated app with guid 8891ab3b-1480-4ced-857f-b4dbc8fd385d ({"state"=>"STOPPED"})
```

- STG

  Emitted during staging and restaging of apps

  DEA emits this message; after staging the DEA emits message with type DEA

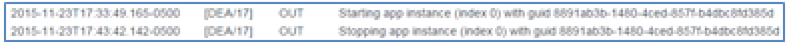


```
2015-11-23T17:52:59.226-0500   [STG/0]   OUT   Expanding Auto-Scaling Agent to .Auto-Scaling (0.0s)
2015-11-23T17:52:59.229-0500   [STG/0]   OUT   -----> Retrieving com.ibm.ws.liberty-2015.10.0.0-pi62103-201511191720.tar.gz ... (0.0s)
```

Log types

- DEA

  The Droplet Execution Agent emits DEA logs when starts/stops apps or when the app crashes

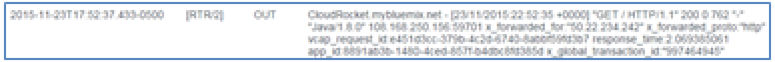


```
2015-11-23T17:33:49.165-0500   [DEA/17]   OUT   Starting app instance (index 0) with guid 8891ab3b-1480-4ced-857f-b4dbc8fd385d
2015-11-23T17:43:42.142-0500   [DEA/17]   OUT   Stopping app instance (index 0) with guid 8891ab3b-1480-4ced-857f-b4dbc8fd385d
```

- RTR

  The Router emits RTR logs when it routes HTTP requests to the application

  App name followed by Router timestamp and selections from the HTTP request



```
2015-11-23T17:52:37.433-0500   [RTR/2]   OUT   CloudRocket.mybluemix.net - [23/11/2015:22:52:35 +0000] "GET / HTTP/1.1" 200 0 762 "-"
                                               "Java/1.8.0" 108.168.250.156:59701 x_forwarded_for:"50.22.234.242" x_forwarded_proto:"http"
                                               vcap_request_id:e451d3cc-379b-4c2d-6740-8abbf59fd3b7 response_time:2.069385061
                                               app_id:8891ab3b-1480-4ced-857f-b4dbc8fd385d x_global_transaction_id:"997464945"
```

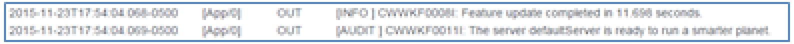
Log types

- LGR

  Loggregator emits message to indicate any problem with the logging
  E.g., "Dropped log messages due to high rate"

- APP

  Emitted by application



```
2015-11-23T17:54:04.068-0500   [App/0]   OUT   [INFO ] CWWKF0008I: Feature update completed in 11.696 seconds.
2015-11-23T17:54:04.069-0500   [App/0]   OUT   [AUDIT ] CWWKF0011I: The server defaultServer is ready to run a smarter planet.
```

Loggeregator

- Loggregator
  - Cloud Foundry component responsible for logging
  - By default streams logs to the terminal
  - If the consumer is unable to consume logs they may be lost
  - For persisting the logs stream the logs to a **third party log management**

Stream to Papertrail

- Papertrail is third party service for log management

  Visit Http://papertrail.com for information and to open an account (free)
- Other third party log management tools supported are:

  Splunk, SplunkStorm, SumoLogic
- At a high level the setup process involves:
  1. Setting up the 3rd party service for receiving logs
  2. Creation of User Provided Service with –l option on Bluemix/CF
  3. Binding of App to UPS
  4. Re-stage App

Summary

- All log messages have a log type that identifies the component of Cloud Foundry that emitted the message
  - API, STG, DEA, RTR, LGR, APP
  - In case of issues the log type will help you understand where the issue occurred
- Third party log management system allows
  - Persistence of log messages
  - Run analytics on historical information
  - Generate alert notifications e.g., Emails

Testing

Types of Tests

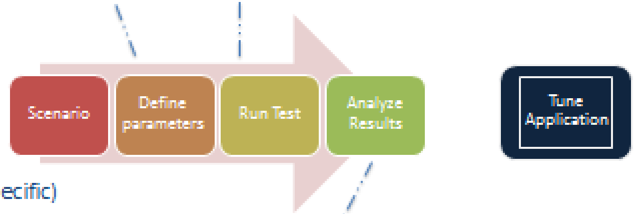
1. Functional testing
   - Links on webpages, Data from DB, Cookies, Form validations
2. Usability testing
   - Navigation, Content should be logical, Sitemap
3. Interface testing
   - Interaction between web servers and app servers is good

Types of Tests

4. Compatibility testing
   - Browser, Mobile
5. Performance testing
   - Load testing, Stress testing, Endurance testing
6. Security testing
   - Direct URL access, Invalid data in forms

- Number of virtual users
- Ramping, Cooldown
- Peak load – sustained load

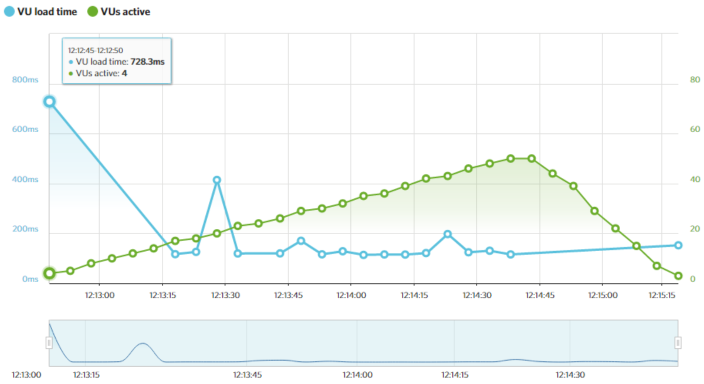
- Specific to the tool
- Gather reports



- Scripting (tools specific)
- Browser actions
- Representative usage

- Error rates
- Response/Throughput issues
- Correlate with information gathered from app

Tune Application



- Response time
- Throughput
- Errors
- Spikes

- Look at the application logs to identify health issues, resource constraints
- Auto scaling metrics, history
- Behavior of the backend system e.g., database
- Runtime specific aspects e.g., could Java garbage collection be the cause of spikes

Bluemix Performance Testing Services

BlazeMeter
Third Party

“A cloud based performance testing platform for web and mobile apps made for developers, operations, devops and QA professionals”

- JMeter on the cloud

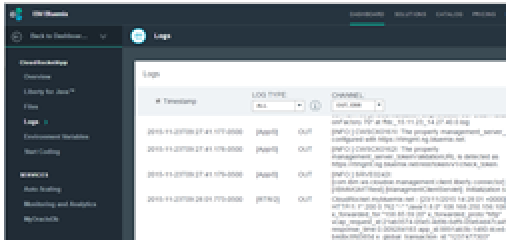
Load Impact
Third Party

“Automated and on-demand performance testing for DevOps. Load test your website, web app, mobile app or API instantly with up to 1.2 million concurrent users”

Monitoring

1. Monitor using log tools

- Monitor using log tools
  - cf logs commands
  - Bluemix console

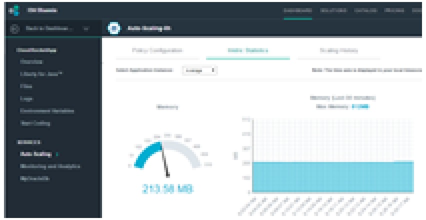


- Monitor using external logging services
  - E.g., Papertrail, Splunk

2. Metrics from Auto scaling

- Metrics tab for auto scaling application shows:
  - Scaling history
  - Value of metrics that triggered the scaling policy

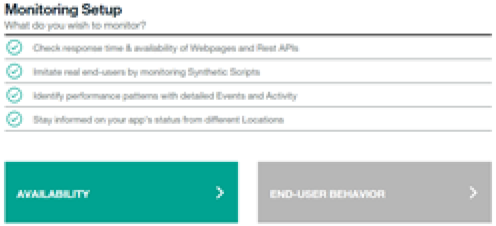
Memory,
JVM Heap,
Throughput
& Response



3. Monitoring panel on dashboard

- Bluemix console >> App overview >> Monitoring

1. Availability monitoring
2. Performance monitoring



Monitoring Setup

What do you wish to monitor?

- ✓ Check response time & availability of Webpages and Rest APIs
- ✓ Imitate real end-users by monitoring Synthetic Scripts
- ✓ Identify performance patterns with detailed Events and Activity
- ✓ Stay informed on your app's status from different Locations

4. Bluemix *Monitoring and Analytics service*

- Instant visibility and transparency into your applications performance and health without the need to learn or deploy any other tools
- Plans

  Free & Diagnostic
- How to use?
  1. Create *Monitoring & Analytics* service instance
  2. Bind the application
  3. Restage the application

Monitoring and Analytics
IBM

5. Monitor apps using third party monitoring services

- Use third party monitoring services

E.g., *New Relic*

Summary

1. There are various monitoring tools/options available on Bluemix

2. Bluemix *Monitoring & Analytics* service bound to a app gives visibility into the runtime specific metrics, shows the logs, send event/alarm notifications