MQ Monitoring on Cloud

Suganya Rane
Digital Automation, Integration & Cloud Solutions
Agenda

- Metrics & Monitoring
- Monitoring Options
- AWS ElasticSearch
- Kibana
- MQ CloudWatch on AWS
- Prometheus
- Grafana
- MQ on AWS Options
Metrics & Monitoring

- MQ V9 makes many statistics available through a pub/sub interface

- Option to remotely subscribe to topics under $SYS/MQ for information on:
  - CPU usage
  - Disk usage
  - Connections and disconnections
  - Opening and closing of queues
  - Pub/sub and put/get
  - Syncpoint calls
  - Changes to MQ objects (MQSET and MQINQ)
Metrics & Monitoring

- Publish to remote metrics servers like Graphite, Prometheus
  - Visualize using Grafana
What can we monitor?

IBM MQ contains seven key elements that are critical to the overall functionality and availability of the application:

- Queue Managers
- Queues
- Cluster Queue Managers
- Channels
- Listeners
- Service Objects
- Events
Monitoring Options on the Cloud

- Use the AWS ElasticSearch service (forward logs).

- You could also forward your logs to a third-party log service, such as Splunk, Loggly, Elastic Cloud, or the IBM Bluemix Logmet service.

- Use Amazon's own service, CloudWatch.
AWS ElasticSearch

The ElasticSearch stack is a useful way to store, search and visualize data.

The ELK stack comprises 3 components:

- ElasticSearch - A distributed search engine
- Logstash - a log collection & processing tool
- Kibana - a browser based visualization tool for displaying and searching log data
AWS ElasticSearch

1. Install your own ElasticSearch
2. Install your Kibana server (The above two have to be installed & ready to send data to them).
3. Configure Logstash to store & process MQ logs (i.e. AMQERR*.LOG)
4. Use Kibana to view & search through those logs.
Sending logs to ElasticSearch

• Logs are written into the Elasticsearch engine by Logstash.
• Consolidate logs from multiple machines producing log records.
• Logstash needs to get the log records from those machines, process them to filter out unwanted data, and send them to Elasticsearch.
Sending logs to Elasticsearch

Getting logs from each of your machines into Elasticsearch is done in one of two ways:

1. Sending the complete log to a central Logstash agent, where it is processed
2. Running a Logstash agent on each machine and processing them locally

**Refer to Matt Whitehead’s blog on details of how to do this.**
Kibana Dashboards

After having the MQ records being written to Elasticsearch we can use Kibana to view and search through them.
Kibana Dashboards
Kibana Dashboards

The image shows a screen capture of Kibana dashboards with a focus on MQ Technical Conference v2.0.1.8. The dashboard includes tables and JSON data, indicating a detailed view of MQ logs.
## Kibana Dashboards

![Graph showing timestamp per second with data points]

<table>
<thead>
<tr>
<th>Time</th>
<th>errDesc</th>
<th>errCode</th>
<th>program</th>
<th>qmgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 12th 2016, 17:08:10.678</td>
<td>Queued Publish/Subscribe Daemon started for queue manager QM1.</td>
<td>AM05806</td>
<td>anqfpub</td>
<td>QM1</td>
</tr>
<tr>
<td>August 12th 2016, 17:08:10.673</td>
<td>IBM MQ channel initiator started.</td>
<td>AM00024</td>
<td>runmqchi</td>
<td>QM1</td>
</tr>
<tr>
<td>August 12th 2016, 17:08:10.668</td>
<td>The command server has started. ProcessId(24621).</td>
<td>AM05824</td>
<td>anqzmgr0</td>
<td>QM1</td>
</tr>
<tr>
<td>August 12th 2016, 17:08:10.666</td>
<td>The channel initiator has started. ProcessId(24619).</td>
<td>AM05822</td>
<td>anqzmgr0</td>
<td>QM1</td>
</tr>
<tr>
<td>August 12th 2016, 17:00:10.663</td>
<td>'IBM MQ Distributed Pub/Sub Publish Task' has started.</td>
<td>AM05975</td>
<td>anqzmufo</td>
<td>QM1</td>
</tr>
<tr>
<td>August 12th 2016, 17:00:10.660</td>
<td>'IBM MQ Distributed Pub/Sub Fan Out Task' has started.</td>
<td>AM05975</td>
<td>anqzmufo</td>
<td>QM1</td>
</tr>
<tr>
<td>August 12th 2016, 17:08:10.657</td>
<td>'IBM MQ Distributed Pub/Sub Command Task' has started.</td>
<td>AM05975</td>
<td>anqzmufo</td>
<td>QM1</td>
</tr>
<tr>
<td>August 12th 2016, 17:00:10.650</td>
<td>The queue manager task 'MULTICAST' has started.</td>
<td>AM05051</td>
<td>anqzmu0</td>
<td>QM1</td>
</tr>
<tr>
<td>August 12th 2016, 17:00:10.545</td>
<td>The queue manager task 'PRESERVED-Q' has started.</td>
<td>AM05051</td>
<td>anqzmu0</td>
<td>QM1</td>
</tr>
<tr>
<td>August 12th 2016, 17:00:10.543</td>
<td>The queue manager task 'Q-DELETION' has started.</td>
<td>AM05051</td>
<td>anqzmu0</td>
<td>QM1</td>
</tr>
<tr>
<td>August 12th 2016, 17:08:10.541</td>
<td>The queue manager task 'RESOURCE_MONITOR' has started.</td>
<td>AM05051</td>
<td>anqzmu0</td>
<td>QM1</td>
</tr>
<tr>
<td>August 12th 2016, 17:08:10.539</td>
<td>The queue manager task 'TOPIC_TREE' has started.</td>
<td>AM05051</td>
<td>anqzmu0</td>
<td>QM1</td>
</tr>
</tbody>
</table>
Kibana Dashboards
Kibana - Summary

• You can add and remove different components to the dashboard

• You can save dashboards once you're happy with their layout

• Create various charts and graphs of your data.
AWS CloudWatch

- AWS CloudWatch is a monitoring & management service
- Provides actionable insights to monitor applications
- Respond to system-wide performance changes
- Optimize resource utilization
- Collects monitoring & operational data
AWS CloudWatch

Amazon CloudWatch
Complete visibility into your cloud resources and applications

Collect
Metrics and logs from all your AWS resources, applications, and services that run on AWS and on-premises servers

Monitor
Visualize applications and infrastructure with CloudWatch dashboards; correlate logs and metrics side by side to troubleshoot and set alerts with CloudWatch Alarms

Act
Automate response to operational changes with CloudWatch Events and Auto Scaling

Analyze
Up to 1-second metrics, extended data retention (15 months), and real-time analysis with CloudWatch Metric Math

Application Monitoring

Resource Optimization

System-wide Visibility

Unified Operational Health
<table>
<thead>
<tr>
<th>PROS</th>
<th>CONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature rich</td>
<td>Less powerful for non AWS resources</td>
</tr>
<tr>
<td>Easy to setup</td>
<td>Need work for customization</td>
</tr>
<tr>
<td>Can search individual log streams</td>
<td>Not as advanced as ElasticSearch</td>
</tr>
<tr>
<td>Can setup filters and alarms</td>
<td></td>
</tr>
</tbody>
</table>
AWS CloudWatch

• Make sure that your EC2 instances are authorized to talk to the CloudWatch service.
  o Create a policy in the Identity and Access Management (IAM) service, and then assigning that policy to a role.
• Roles can then be assigned to your EC2 instances.
Sending error logs to CloudWatch

- CloudWatch requires a CloudWatch agent to be installed on your EC2 instance.
- This is a small Python program which will monitor log files, identify separate log entries, and then send those entries to CloudWatch.
- After installing the CloudWatch agent, you can add the necessary configuration for the MQ error logs.
Sending error logs to CloudWatch

• This configuration can either be written to the main CloudWatch configuration file in /var/awslogs/etc/awslogs.conf, or to a standalone file in /var/awslogs/etc/config/mq.conf.

• The configuration identifies two files to watch: the main MQ error log, and the error log for queue manager.

• It configures the date/time format, tells the agent to send any pending log messages to CloudWatch at least every five seconds, and write them to a "log stream" based on the EC2 instance ID.

• Restart CloudWatch service and view the logs in the AWS management console (under CloudWatch -> Logs)
CloudWatch – Summary

• CloudWatch allows you can search individual log streams (per instance), and set up filters and alarms for certain events.

• The search facilities offered aren't anywhere near as advanced as ElasticSearch, but you can choose to forward log events on to ElasticSearch if you want to.

• Importantly, you can also trigger AWS Lambda functions based on certain messages being logged, which will allow you to take programmatic action.

• You can use either solutions and also add email functionality to automatically indicate when specific errors occurs.
Prometheus

• An open-source monitoring and alerting solution, whose particular strengths:
  
  o the collection of time series data, with the ability to easily query that data.

  o it provides libraries in several languages to enable products to export data to it.

  o it works by pulling information from exporters such as this MQ program at configured intervals over an HTTP connection.
Prometheus

• The Prometheus documentation has information on more complex configuration options.

• Once the Prometheus server has picked up the MQ configuration, the metrics can be seen under the specified jobname.
Grafana

• Grafana provides a way to create dashboards and visualize data held in time series databases.

• It has Prometheus as a built-in data source.

• For example, the number of MQPUTs to a queue may be of interest, and this kind of database makes it easy to see how many operations occurred in an interval, or calculate averages.
Configuring Grafana

• Use Prometheus (only a hostname and portnumber is required when adding this type of datasource.)

• All of the MQ metrics can be accessed and added to dashboards.
Configuring Grafana Dashboards

As an example, this dashboard is looking at several items including the same queues as above, and CPU and logging information:
Grafana Dashboards

This picture shows how the top panel was configured, to select several metrics and show the object name in the legend.
MQ Recover times (*estimates*)

This could be much longer, depending on number of clients, and other application logic.

Single resilient queue manager (VM)
- Reconnect 1000 clients
- Detect failure
- Replay transaction logs
- Start queue manager
- Boot VM
- Download image to new host

This could be much longer, depending on transacted message load.

Single resilient queue manager (container)

Multi-instance queue manager

Replicated data queue manager

Images can be pre-pulled to remove this.
MQ on AWS Options

- Quick Start Guide for AWS (MQV9)
- Partner World Licensing (non-prod environments)
- BYOL from your current license stack
- Purchase of MQ from AWS Marketplace (MQV8)
- MQ Clients on AWS (connecting back to your on-prem qmqr) – no software license costs!
MQ on AWS (the quick start with trials)
MQ from the AWS Marketplace

<table>
<thead>
<tr>
<th>MQ from the AWS Marketplace</th>
<th>COSTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>This AMI has WebSphere MQ v 8.0 installed on Red Hat Enterprise Linux 7.5 base.</td>
<td>Hourly Cost:</td>
</tr>
<tr>
<td>Highlights:</td>
<td></td>
</tr>
<tr>
<td>• Install WebSphere MQ in just a few minutes</td>
<td><strong>EC2 Instance type</strong></td>
</tr>
<tr>
<td></td>
<td>m3.medium</td>
</tr>
<tr>
<td>• Ideal for Development, Test and peak production workloads</td>
<td></td>
</tr>
<tr>
<td>• If the version you require is not listed, you can request it.</td>
<td></td>
</tr>
</tbody>
</table>
## AWS Free Tier

12 months free and always free products

AWS Free Tier includes offers that expire 12 months following sign up and others that never expire.

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Amazon EC2</strong></td>
<td>- 750 hours per month of Linux, RHEL, or SLES t2.micro instance usage</td>
</tr>
<tr>
<td></td>
<td>- 750 hours per month of Windows t2.micro instance usage</td>
</tr>
<tr>
<td><strong>Amazon QuickSight</strong></td>
<td>- 1 GB of SPICE capacity 1 user perpetual free tier</td>
</tr>
<tr>
<td></td>
<td>- 10 GB of SPICE capacity the first 2 months for free for a total of 4 users</td>
</tr>
<tr>
<td><strong>Amazon CloudWatch</strong></td>
<td>- 10 Custom Metrics and 10 Alarms</td>
</tr>
<tr>
<td></td>
<td>- 1,000,000 API Requests</td>
</tr>
<tr>
<td></td>
<td>- 5 GB of Log Data Ingestion and 5 GB of Log Data Archive</td>
</tr>
<tr>
<td></td>
<td>- 3 Dashboards with up to 50 Metrics Each per Month</td>
</tr>
<tr>
<td><strong>Amazon RDS</strong></td>
<td>- 750 Hours per month of db or t2.micro database usage (applicable DB-engines)</td>
</tr>
<tr>
<td></td>
<td>- 20 GB of General Purpose (SSD) database storage</td>
</tr>
<tr>
<td></td>
<td>- 20 GB of storage for database backups and DB Snapshots</td>
</tr>
<tr>
<td><strong>Amazon S3</strong></td>
<td>- 5 GB of Standard Storage</td>
</tr>
<tr>
<td></td>
<td>- 20,000 Get Requests</td>
</tr>
<tr>
<td></td>
<td>- 2,000 Put Requests</td>
</tr>
<tr>
<td><strong>AWS Lambda</strong></td>
<td>- 1,000,000 free requests per month</td>
</tr>
<tr>
<td></td>
<td>- Up to 3.2 million seconds of compute time per month</td>
</tr>
</tbody>
</table>
Summary

• These slides and presented material show how the statistics generated by MQ can easily be used in some of the monitoring packages that are commonly used with various cloud and container-based systems.

• The MQ data can be integrated with other metrics to give a complete view of your environment.
Useful Links

• MQ on Cloud – IBM Messaging
  https://developer.ibm.com/messaging/mq-on-cloud/

• GitHub Repositories
  https://github.com/ibm-messaging/mq-aws

• IBM MQ - Using AWS CloudWatch to monitor queue managers

• IBM MQ - Using Prometheus & Grafana to monitor qmgs
Useful Links

• Storing and searching MQ error logs in Elasticsearch

• MQ on AWS: Sending MQ error logs to CloudWatch

Grafana Support For Prometheus
https://prometheus.io/docs/visualization/grafana/
Questions & Answers