MQ on AWS: Sending MQ error logs to CloudWatch

[Arthur Barr](javascript:;" \o "Arthur Barr) |Aug 31 2016 Updated

There are several options for centralizing your MQ error logs, when running on Amazon Web Services (AWS).  For example, you could use the AWS ElasticSearch service, and forward logs.You could also forward your logs to a third-party log service, such as [Splunk](http://www.splunk.com/), [Loggly](https://www.loggly.com/), [Elastic Cloud](https://www.elastic.co/cloud), or the IBM Bluemix Logmet service.  Of course, you could use Amazon's own service, [CloudWatch](https://aws.amazon.com/cloudwatch/).  Although CloudWatch is arguably less powerful and feature rich than many of the alternatives, it is easy to set up, and working with CloudWatch is far better than not centralizing your logs at all.  In this blog entry, we'll take you through the simple steps required to centralize your MQ error logs.

**Authorizing your EC2 instance for CloudWatch**

The first thing you need to do is to make sure that your EC2 instances are authorized to talk to the CloudWatch service.  This is done by [creating a policy](http://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_create.html) in the Identity and Access Management (IAM) service, and then [assigning that policy to a role](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-service.html).  Roles can then be assigned to your EC2 instances.  AWS provides certain pre-canned policies, but they offer very coarse-grained access to CloudWatch.  The following is a more fine-grained example policy:

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"logs:CreateLogGroup",

"logs:CreateLogStream",

"logs:PutLogEvents",

"logs:DescribeLogStreams"

],

"Resource": [

"arn:aws:logs:\*:\*:\*"

]

}

]

}

Once you've create a new role with the appropriate policies you then assign it to any EC2 instances you want to use with CloudWatch.

**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 you've [installed the CloudWatch agent](https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/CWL_GettingStarted.html), you can add the necessary configuration for the MQ error logs:

[/var/mqm/errors/AMQERR01.LOG]

datetime\_format = %m/%d/%Y %I:%M:%S %p

file = /var/mqm/errors/AMQERR01.LOG

buffer\_duration = 5000

log\_stream\_name = {instance\_id}

initial\_position = start\_of\_file

log\_group\_name = /var/mqm/errors/AMQERR01.LOG

multi\_line\_start\_pattern = {datetime\_format}

[/var/mqm/qmgrs/QM1/errors/AMQERR01.LOG]

datetime\_format = %m/%d/%Y %I:%M:%S %p

file = /var/mqm/qmgrs/QM1/errors/AMQERR01.LOG

buffer\_duration = 5000

log\_stream\_name = {instance\_id}

initial\_position = start\_of\_file

log\_group\_name = /var/mqm/qmgrs/QM1/errors/AMQERR01.LOG

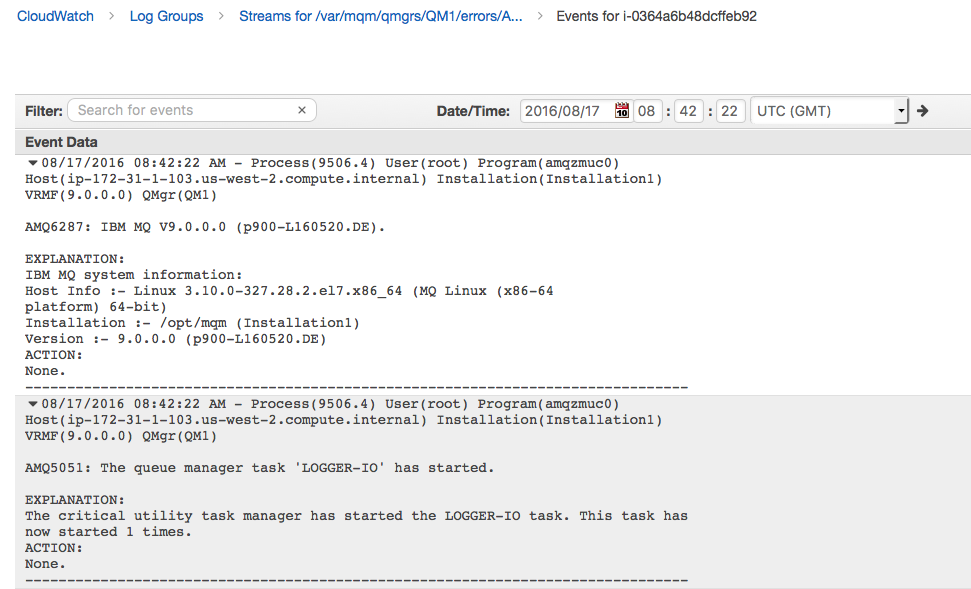
multi\_line\_start\_pattern = {datetime\_format}

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 QM1.  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.

After writing the above configuration, you will need to restart the CloudWatch service:

$ sudo service awslogs restart

You can then view the logs in the AWS management console (under CloudWatch -> Logs):

[](https://www.ibm.com/developerworks/community/blogs/messaging/resource/BLOGS_UPLOADED_IMAGES/ScreenShot2016-08-17at09.48.01.png)

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 (say) 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.