Amazon Aurora Performance Assessment

Technical Guide

June 2019

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

About this Guide

This technical guide outlines the steps to assess the performance of Amazon Aurora MySQL 5.6 and Aurora MySQL 5.7 using the sysbench benchmarking tool. A link to an AWS CloudFormation template that launches and configures the AWS services required to deploy this solution using AWS best practices for security and availability.

The guide is intended for database administrators, developers and other technical personnel who are familiar with relational databases and architecting on the AWS Cloud.

Overview

To help customers assess the performance of Amazon Aurora MySQL 5.6 and Aurora MySQL 5.7 (Aurora MySQL 5.6/5.7), AWS is providing this guide, which includes an AWS CloudFormation template to automate the assessment process. The sysbench benchmarking tool is used to conduct the assessment. Although this guide uses sysbench to assess performance, you can adapt the instructions to use your preferred tool by adjusting the bash script referenced in this guide.

Assessing Amazon RDS for Aurora Performance Using Sysbench

Follow the instructions in this section to execute the following activities:

1. Deploy an AWS CloudFormation template to set up the necessary AWS resources that will be used to assess the performance of Amazon Aurora MySQL 5.6/5.7.
2. Run a performance assessment of Amazon Aurora MySQL 5.6/5.7 using the sysbench tool.

The performance assessment lets you simulate two different workloads:

- A simple read-heavy workload (100 percent read)
- A write-heavy workload (100 percent write)

Deploy the AWS CloudFormation Template

Deploying the AWS CloudFormation template sets up the following environment in the AWS Cloud.

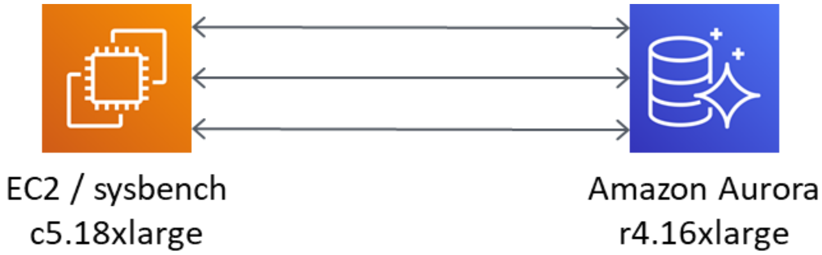


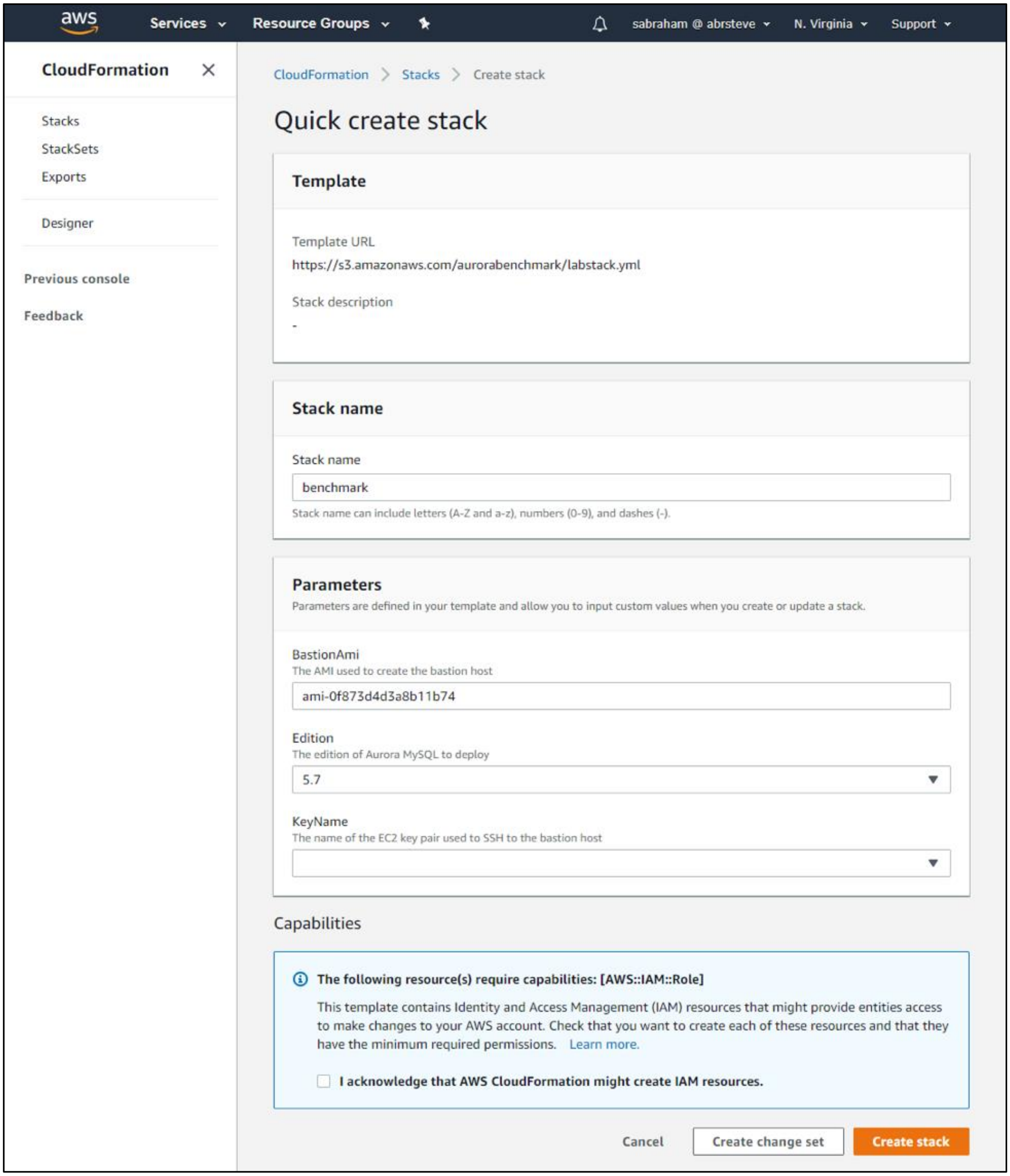
Figure 1: AWS services deployed by the AWS CloudFormation template

Launch Template

The AWS CloudFormation template creates an Amazon Virtual Private Cloud (Amazon VPC) containing:

- An Amazon Elastic Compute Cloud (Amazon EC2) c5.18xlarge Amazon Linux instance running the sysbench tool and serving as a bastion host
- An Aurora MySQL 5.6/5.7 database (DB) instance, configured as an r4.16xlarge Amazon Linux instance
- An Amazon S3 bucket to place bash or other scripts that will be downloaded to the EC2 instance to execute

The instance is created in an Amazon VPC with enhanced networking enabled to ensure that throughput numbers are not constrained by network bandwidth. The Amazon EC2 instance queries the Aurora MySQL 5.6/5.7 database (DB) instance. Refer to Figure 2 for a screenshot of the AWS CloudFormation template that is deployed.



Quick create stack

Template

Template URL
https://s3.amazonaws.com/aurorabenchmark/labstack.yml

Stack description
-

Stack name

Stack name
benchmark
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

BastionAmi
The AMI used to create the bastion host
ami-0f873d4d3a8b11b74

Edition
The edition of Aurora MySQL to deploy
5.7 ▼

KeyName
The name of the EC2 key pair used to SSH to the bastion host
▼

Capabilities

ⓘ **The following resource(s) require capabilities: [AWS::IAM::Role]**

This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. Learn more.

□ **I acknowledge that AWS CloudFormation might create IAM resources.**

Figure 2: AWS CloudFormation stack

Creating the Stack

The stack is launched in the US East (N. Virginia) Region by default. Refer to Figure 2 and review the following details. Note that before you can create the stack, you need to review the acknowledgement and check the box.

- The default name of the stack is *benchmark*.
- Parameters:
    - The **BastionAmi** parameter points to a baseline Amazon Linux 2 AMI in the US East (N. Virginia) Region.
    - The KeyName parameter references the EC2 key pair that you can use to SSH into the bastion host.

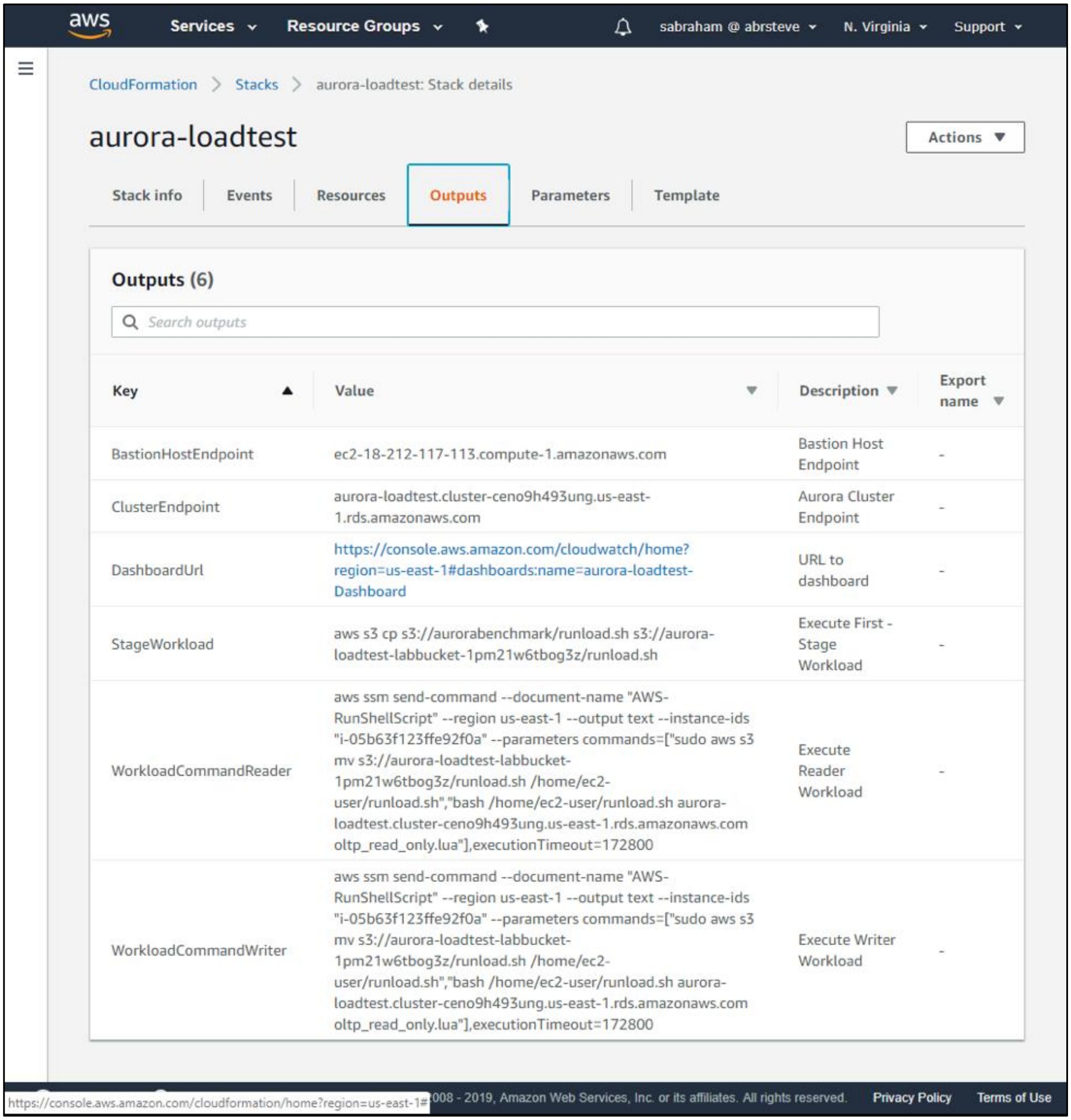
Steps to create the stack

1. **Optional:** Change the Stack name from the default and ensure that the name is entered in lowercase letters only.
   You may keep the default stack name, but ensure you do not have another stack named *benchmark*.
2. Under **Capabilities**, check the acknowledgement box indicating that AWS CloudFormation might create AWS Identity and Access Management (IAM) resources.
3. Select **Create stack**.

Once the AWS CloudFormation stack is complete, the **Outputs** section of the stack will list the following **Keys** (shown in Table 1 and Figure 3).

Table 1: AWS CloudFormation Outputs

BastionHostEndpoint	The DNS endpoint of the bastion host
ClusterEndpoint	The DNS endpoint of the Aurora cluster
DashboardUrl	The URL to the dashboard that will show metrics from the load test
StageWorkload	Command to copy the workload bash script to the bastion host
WorkloadCommandReader	Command to execute the read-only workload
WorkloadCommandWriter	Command to execute the write-only workload


aws
Services
Resource Groups
sabraham @ abrsteve
N. Virginia
Support
CloudFormation > Stacks > aurora-loadtest: Stack details
Actions ▼
Stack info
Events
Resources
Outputs
Parameters
Template
Search outputs
https://console.aws.amazon.com/cloudformation/home?region=us-east-1#
2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

aurora-loadtest

Outputs (6)

Key ▲	Value ▼	Description ▼	Export name ▼
BastionHostEndpoint	ec2-18-212-117-113.compute-1.amazonaws.com	Bastion Host Endpoint	-
ClusterEndpoint	aurora-loadtest.cluster-ceno9h493ung.us-east-1.rds.amazonaws.com	Aurora Cluster Endpoint	-
DashboardUrl	https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#dashboards:name=aurora-loadtest-Dashboard	URL to dashboard	-
StageWorkload	aws s3 cp s3://aurorabenchmark/runload.sh s3://aurora-loadtest-labbucket-1pm21w6tbog3z/runload.sh	Execute First - Stage Workload	-
WorkloadCommandReader	aws ssm send-command --document-name "AWS-RunShellScript" --region us-east-1 --output text --instance-ids "i-05b63f123ffe92f0a" --parameters commands=["sudo aws s3 mv s3://aurora-loadtest-labbucket-1pm21w6tbog3z/runload.sh /home/ec2-user/runload.sh","bash /home/ec2-user/runload.sh aurora-loadtest.cluster-ceno9h493ung.us-east-1.rds.amazonaws.com oltp_read_only.lua"],executionTimeout=172800	Execute Reader Workload	-
WorkloadCommandWriter	aws ssm send-command --document-name "AWS-RunShellScript" --region us-east-1 --output text --instance-ids "i-05b63f123ffe92f0a" --parameters commands=["sudo aws s3 mv s3://aurora-loadtest-labbucket-1pm21w6tbog3z/runload.sh /home/ec2-user/runload.sh","bash /home/ec2-user/runload.sh aurora-loadtest.cluster-ceno9h493ung.us-east-1.rds.amazonaws.com oltp_read_only.lua"],executionTimeout=172800	Execute Writer Workload	-

Figure 3: AWS CloudFormation example stack Outputs tab

Running the StageWorkload Bash Script

After the stack is created, the message **CREATE_COMPLETE** displays in your console.

Follow the steps below to run the bash script to set up the performance assessment tool.

Steps to run the StageWorkload bash script

NOTE: In order to run any command on your computer's command line, ensure that AWS CLI is installed.

1. On the resulting screen, select the **Outputs** tab.
2. Under the **Key** column, locate the command labeled **StageWorkload**.
3. Copy the **Value** for this command and paste into your operating system's command line.
4. Run the command.

A bash script is copied into the Amazon S3 bucket that is a part of this stack. The bash script contains the commands that will be executed to run the workload. As of the publication date, the bash script is configured and contains commands to run a sysbench workload. However, this script can be modified to run any workload.

Performing an Assessment

Once the **StageWorkload** command completes successfully (normally takes one to two seconds), you can run performance assessments for your Amazon Aurora database. You have two options for the performance assessment: (1) assess the reader workload or (2) assess the writer workload.

Steps to run a performance assessment

1. Determine whether you want to assess the reader workload or the writer workload.
2. Locate the appropriate **Key** for the assessment you want to run: either WorkloadCommandReader or WorkloadCommandWriter.
3. Copy the **Value** for the command and paste into your operating system's command line.
4. Run the command.

The command performs the following actions (in sequence; for your reference):

1. Installs multiple tools (including automake, libtool, mysql-devel, mysql, and git).
2. Downloads the sysbench source code.
3. Builds sysbench using the tools installed in step 1.
4. Deletes the test database, then creates the sysbench database on the ClusterEndpoint.
5. Uses sysbench to load the sysbench database with data.

Once the sysbench database is created, the script enters into a loop iterating 250 times. In each iteration, a new instance of sysbench is launched simulating 20 user connections. After a one second pause, the next iteration launches the next 20 connections. In just over 4 minutes, the connection count will reach 5,000 and a full load capacity.

Accessing the Performance Dashboard

Once the load is underway, you can view the performance dashboard for your Amazon Aurora cluster using the **DashboardUrl** key value.

Steps to view the performance dashboard

1. Locate the **DashboardUrl** key.

2. Select **Value** to view the dashboard.

Figure 4 is an example run on the read-only workload.

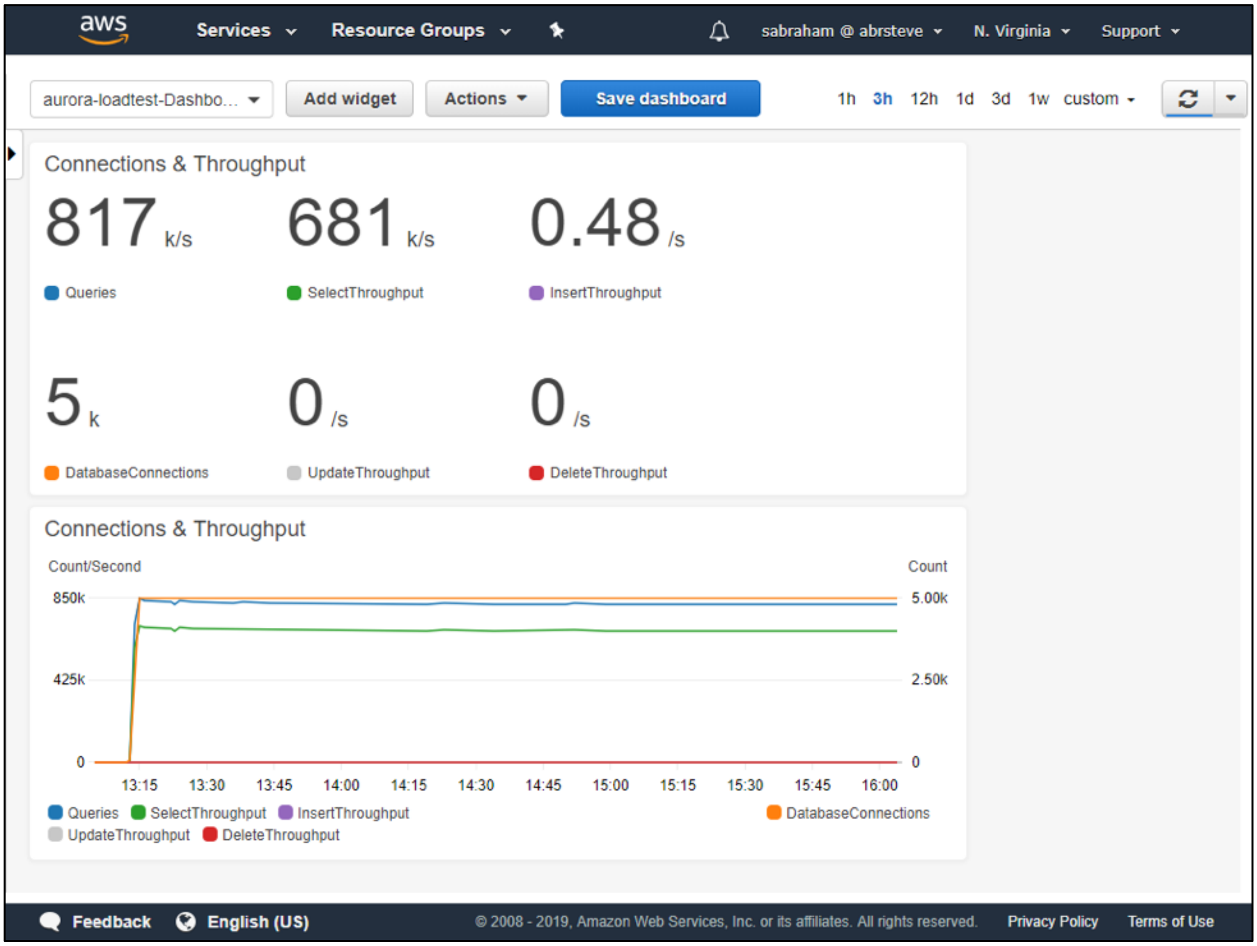


Figure 4: Sample Amazon Aurora performance dashboard

Figure 4 shows that there has been a steady state of more than 800,000 queries per second and close to 700,000 SELECTs per second with exactly 5,000 connections.

Subsequent runs

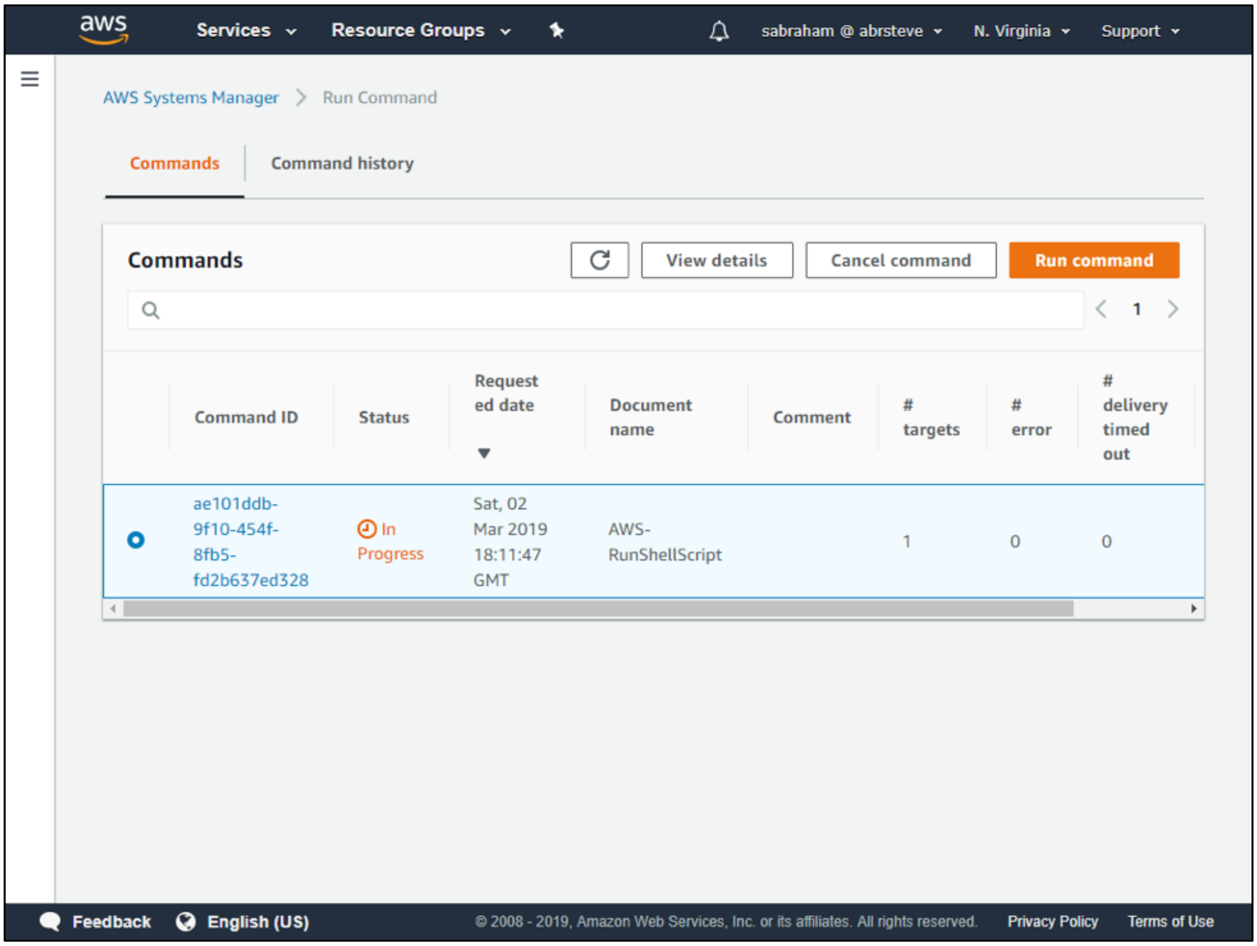
AWS Systems Manager is the tool that invokes the workload script and is part of the AWS platform. AWS Systems Manager will, among other things, allow users to execute commands on remotely configured systems. In this case, when we issue the **WorkLoadCommandReader** command line, the command starts with **aws ssm send-command**. This is the command line interface for sending a command to an instance or multiple instances. In our case, we are issuing two commands. The first command downloads the workload bash script from Amazon S3 and the second command executes the bash script. Once you issue **send-command**, you can view the progress of the command as it runs on the AWS Systems Manager console.

You can stop the current benchmark workload and start another one.

Steps to stop the current benchmark workload and start another one

1. Navigate to the AWS Systems Manager console.
2. Select the **Command ID** and then select **Cancel Command**.

Cancelling the command will stop the current running benchmark test.



	Command ID	Status	Requested date	Document name	Comment	# targets	# error	# delivery timed out
●	ae101ddb-9f10-454f-8fb5-fd2b637ed328	In Progress	Sat, 02 Mar 2019 18:11:47 GMT	AWS-RunShellScript		1	0	0

Figure 4: Sample commands screen on the AWS Systems Manager console

While this specific implementation demonstrates a use case where more than 800,000 queries per second are executed against a single Amazon Aurora instance, you may use this framework to run your own custom bash script or other load testing framework against Amazon Aurora as a part of your own proof of concept or benchmarking tests.

Contributors

Contributors to this document include:

- Steve Abraham, Principal Lead Data Architect, Amazon Web Services

Additional Resources

For additional information, see:

- Amazon Aurora MySQL 5.6/5.7
- AWS CloudFormation
- Amazon VPC
- Amazon EC2
- Amazon S3
- AWS IAM
- AWS CLI
- AWS Systems Manager

Document Revisions

Date	Description
June 2019	Updated the AWS CloudFormation template and deployment details.
July 2015	First publication