MariaDB ash_sampler tool

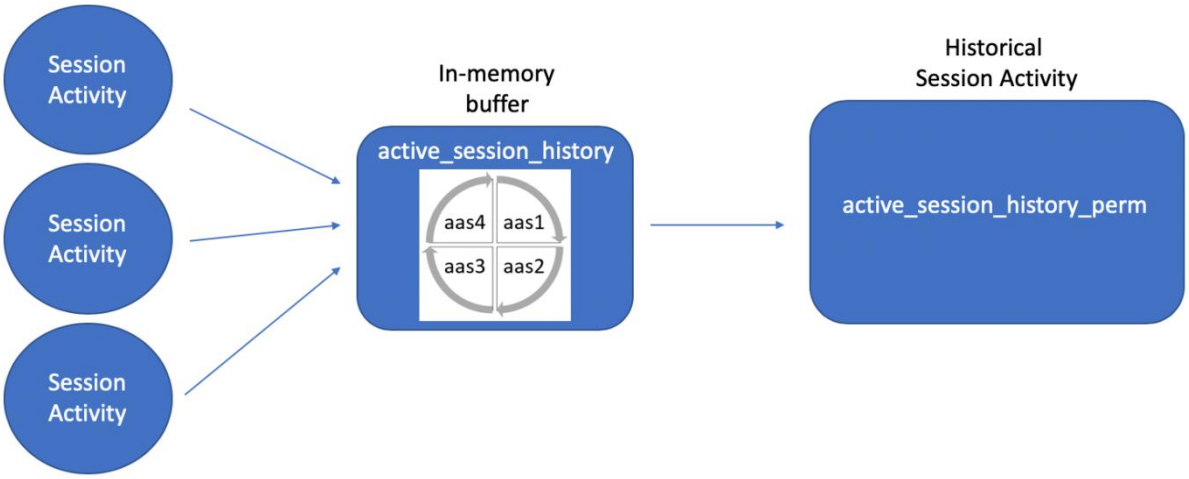
By: Karl Arao

Table of Contents

The ash_sampler tool

This tool takes a snapshot of an active session activity with 1 second granularity. The data is initially gathered in a circular buffer 4 table memory storage engine. And once each buffer gets filled the data is de-staged to a longer-term permanent table for historical reporting.

This concept is similar to Oracle's v$active_session_history (in-memory) and dba_hist_active_sess_history (longer term) performance views and also AWS RDS Performance Insights (only available on RDS MariaDB 10.2.21 and higher) which are used for troubleshooting complex performance problems.



Each session sample is 1 row and the underlying data is divided into two dimensions and measures. This allows for data drilldowns and different ways of aggregating the measures and slicing the dimensions. This detailed data is useful for correlating the database activity across sessions and also with system metrics (CPU, memory, IO, network, etc.)

Time dimension	Session dimension	Measures
• snap_time	• thd_id • conn_id • user • db • command • state • statement_digest • current_statement • full_scan • last_statement_digest • last_statement • last_wait • source • pid • program_name	• time • statement_latency • lock_latency • rows_examined • rows_sent • rows_affected • tmp_tables • tmp_disk_tables • last_statement_latency • last_wait_latency

Tool Requirements

- The tool needs to have the performance_schema turned on. On AWS RDS parameter groups set the performance_schema to 1.
- Execute the command below to verify the performance_schema

```
mysql> SHOW GLOBAL VARIABLES LIKE 'performance_schema';
Variable_name  Value
performance_schema  ON
```

Step 1) Download and Install the tool

- Download the tool on your laptop at https://github.com/karlarao/ash_sampler-mariadb

```
wget https://github.com/karlarao/ash_sampler-mariadb/archive/master.zip

$ unzip master.zip
Archive:  master.zip
2e30d93a7a91c6650ef4f3994dba0773cd7b56b6
   creating: ash_sampler-mariadb-master/
  inflating: ash_sampler-mariadb-master/01_create_objects.sql
  inflating: ash_sampler-mariadb-master/02_ash_sampler.sql
  inflating: ash_sampler-mariadb-master/LICENSE
  inflating: ash_sampler-mariadb-master/README.md
```

- On your text editor replace “mariadb2” with the username that you'll use for monitoring or use the sed commands below

```
sed 's/mariadb2/your_username_here/g' 01_create_objects.sql.bak > 01.sql
sed 's/mariadb2/your_username_here/g' 02_ash_sampler.sql > 02.sql
```

- Open the file `01_create_objects.sql` and execute the commands using the monitoring user
- Download and install DBeaver Community Edition at https://dbeaver.io/download/
- Open the file `02_ash_sampler.sql` using DBeaver and execute the commands using the monitoring user

Step 2) Start the ash_sampler

- Connect to your MariaDB AWS database and run the ash_sampler
- The recommended input parameters (1,1,16) means infinite gathering, for every 1 sec, and create 16MB in-memory buffers

```
$ mysql -s -h mariadb2.us-east-1.rds.amazonaws.com --port=3306 -u mariadb2
-p<your_password_here> -D mariadb2
mysql: [Warning] Using a password on the command line interface can be
insecure.
mysql>
mysql> call ash_sampler(1,1,16);
```

Step 3) Stop the ash_sampler

- To stop the collection CTRL-C or kill the session running the ash_sampler

```
mysql> show processlist;
Id      User     Host     db       Command  Time     State    Info     Progress
4       rdsadmin localhost:11948  mysql    Sleep    1                 NULL     0.000
121     mariadb2 cpe-69-200-224-180.nyc.res.rr.com:53421    mariadb2 Sleep    16600             NULL     0.000
122     mariadb2 cpe-69-200-224-180.nyc.res.rr.com:53422    mariadb2 Sleep    16707             NULL     0.000
170     mariadb2 cpe-69-200-224-180.nyc.res.rr.com:57005    mariadb2 Sleep    9271              NULL     0.000
200     mariadb2 cpe-69-200-224-180.nyc.res.rr.com:59937    mariadb2 Query    429      User sleep       SELECT SLEEP(
NAME_CONST('in_interval',1.00)) INTO @sleep        0.000
202     mariadb2 cpe-69-200-224-180.nyc.res.rr.com:60092    mariadb2 Query    0        init     show processlist 0.000

mysql> kill 200;
```

Step 4) Export the data

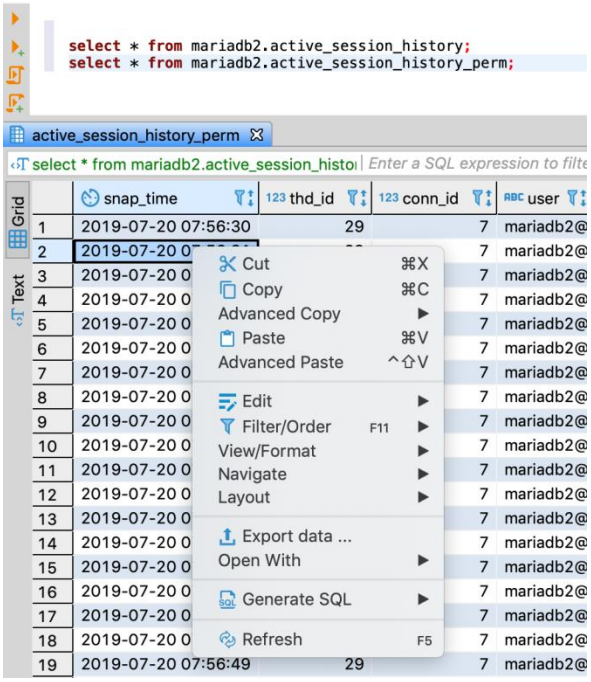
- Using DBeaver Community Edition. Export the data from the following:
    - active_session_history
        - to export the recent in-memory data that is not yet de-staged to permanent storage
    - active_session_history_perm
        - this contains the historical data since ash_sampler is first started

    NOTE: Whenever the ash_sampler is restarted it clears the in-memory data so make sure to export the contents to CSV or put it in a table using CTAS before starting the ash_sampler to not lose any performance data.

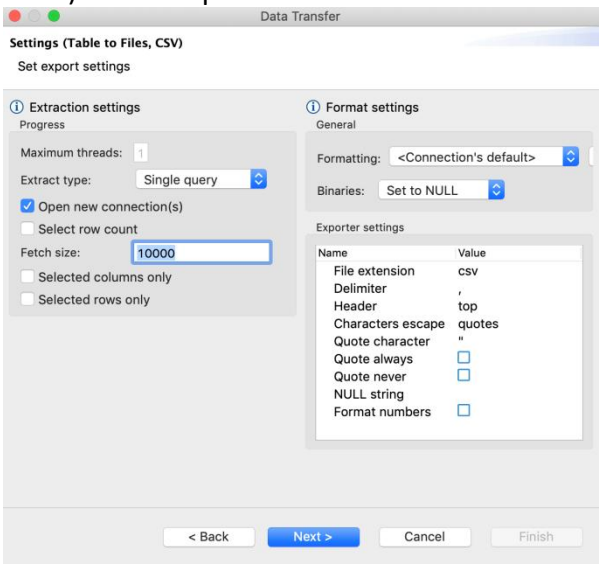
```
-- on DBeaver execute the two SQLs and export to CSV
select * from active_session_history;
select * from active_session_history_perm;
```

Follow the screenshots below:

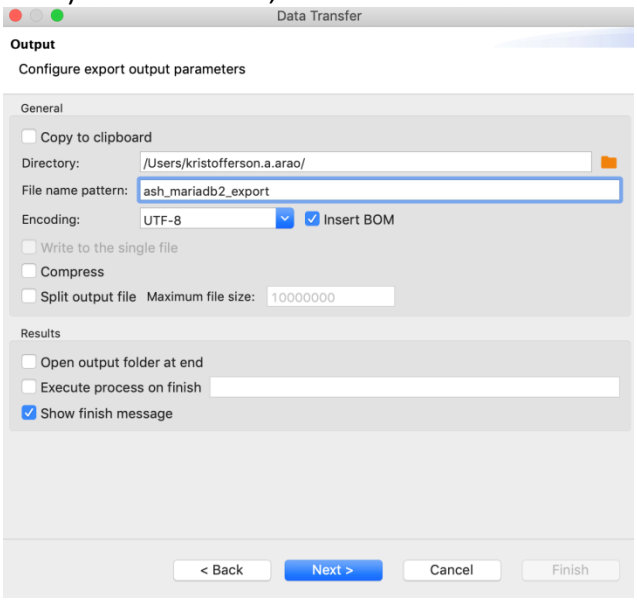
1) Export data



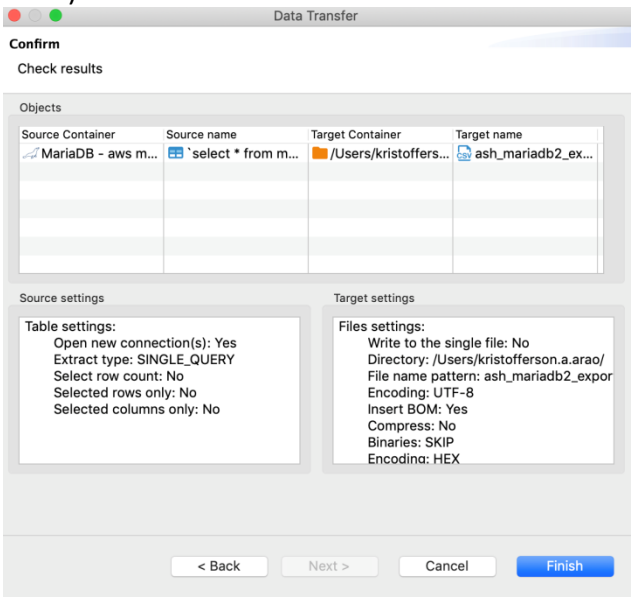
2) Select open new connection



3) Set file name, Click Next



4) Click Finish



Step 5) Visualize and correlate with Cloudwatcher

The result is a CSV data set which can be visualized and analyzed in Tableau

```
$ head ash_mariadb2_export.csv
"snap_time","thd_id","conn_id","user","db","command","state","time","statement_digest","current_statement","statement_latency","lock_latency","rows_examined","rows_sent","
rows_affected","tmp_tables","tmp_disk_tables","full_scan","last_statement_digest","last_statement","last_statement_latency","last_wait","last_wait_latency","source","pid",
"program_name"
2019-07-20 07:56:30,29,7,mariadb2@cpe-69-200-224-180.nyc.res.rr.com,mariadb2,Query,User sleep,0,c527d205346af37ae7d52fb0747e4109,"SELECT a.table_schema, a.table_name, slee
p(1)   FROM information_schema.tables a, information_schema.tables b   where a.table_sc",494296746000,154000000,0,0,0,2,0,YES,,,,,,,"64594",mysql
2019-07-20 07:56:31,29,7,mariadb2@cpe-69-200-224-180.nyc.res.rr.com,mariadb2,Query,User sleep,1,c527d205346af37ae7d52fb0747e4109,"SELECT a.table_schema, a.table_name, slee
p(1)   FROM information_schema.tables a, information_schema.tables b   where a.table_sc",1497894054000,154000000,0,1,0,2,0,YES,,,,,,,"64594",mysql
2019-07-20 07:56:32,29,7,mariadb2@cpe-69-200-224-180.nyc.res.rr.com,mariadb2,Query,User sleep,2,c527d205346af37ae7d52fb0747e4109,"SELECT a.table_schema, a.table_name, slee
p(1)   FROM information_schema.tables a, information_schema.tables b   where a.table_sc",2501633044000,154000000,0,2,0,2,0,YES,,,,,,,"64594",mysql
2019-07-20 07:56:33,29,7,mariadb2@cpe-69-200-224-180.nyc.res.rr.com,mariadb2,Query,User sleep,3,c527d205346af37ae7d52fb0747e4109,"SELECT a.table_schema, a.table_name, slee
p(1)   FROM information_schema.tables a, information_schema.tables b   where a.table_sc",3506193113000,154000000,0,3,0,2,0,YES,,,,,,,"64594",mysql
2019-07-20 07:56:34,29,7,mariadb2@cpe-69-200-224-180.nyc.res.rr.com,mariadb2,Query,User sleep,4,c527d205346af37ae7d52fb0747e4109,"SELECT a.table_schema, a.table_name, slee
p(1)   FROM information_schema.tables a, information_schema.tables b   where a.table_sc",4510484168000,154000000,0,4,0,2,0,YES,,,,,,,"64594",mysql
2019-07-20 07:56:35,29,7,mariadb2@cpe-69-200-224-180.nyc.res.rr.com,mariadb2,Query,User sleep,5,c527d205346af37ae7d52fb0747e4109,"SELECT a.table_schema, a.table_name, slee
p(1)   FROM information_schema.tables a, information_schema.tables b   where a.table_sc",5514364478000,154000000,0,5,0,2,0,YES,,,,,,,"64594",mysql
2019-07-20 07:56:36,29,7,mariadb2@cpe-69-200-224-180.nyc.res.rr.com,mariadb2,Query,User sleep,6,c527d205346af37ae7d52fb0747e4109,"SELECT a.table_schema, a.table_name, slee
p(1)   FROM information_schema.tables a, information_schema.tables b   where a.table_sc",6518152741000,154000000,0,6,0,2,0,YES,,,,,,,"64594",mysql
2019-07-20 07:56:37,29,7,mariadb2@cpe-69-200-224-180.nyc.res.rr.com,mariadb2,Query,User sleep,7,c527d205346af37ae7d52fb0747e4109,"SELECT a.table_schema, a.table_name, slee
p(1)   FROM information_schema.tables a, information_schema.tables b   where a.table_sc",7523094600000,154000000,0,7,0,2,0,YES,,,,,,,"64594",mysql
2019-07-20 07:56:39,29,7,mariadb2@cpe-69-200-224-180.nyc.res.rr.com,mariadb2,Query,User sleep,9,c527d205346af37ae7d52fb0747e4109,"SELECT a.table_schema, a.table_name, slee
p(1)   FROM information_schema.tables a, information_schema.tables b   where a.table_sc",8562704969000,154000000,0,8,0,2,0,YES,,,,,,,"64594",mysql
```

The environment used in this example is AWS RDS db.t2.micro with MariaDB 10.1.23 database.

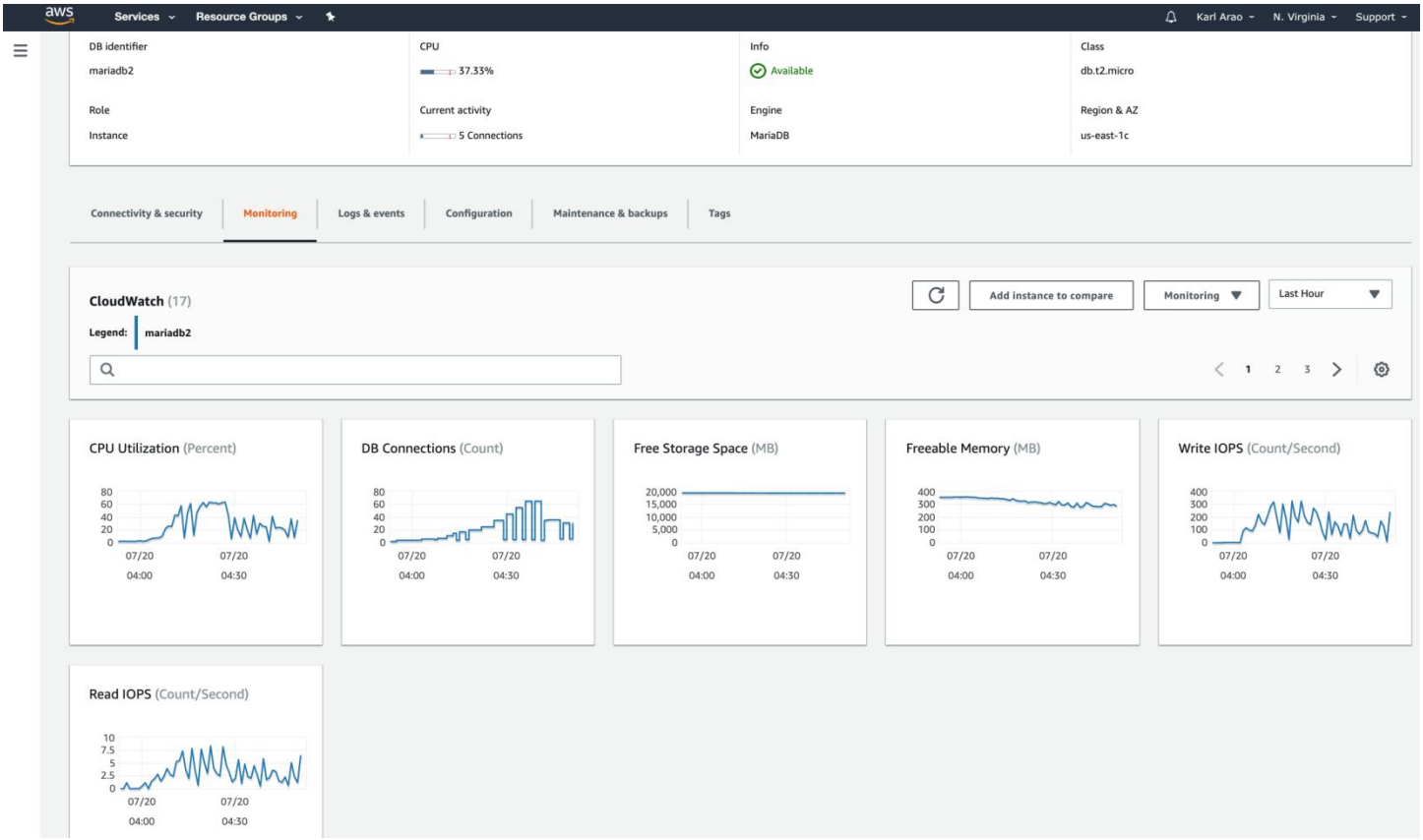
The workload used are the following:

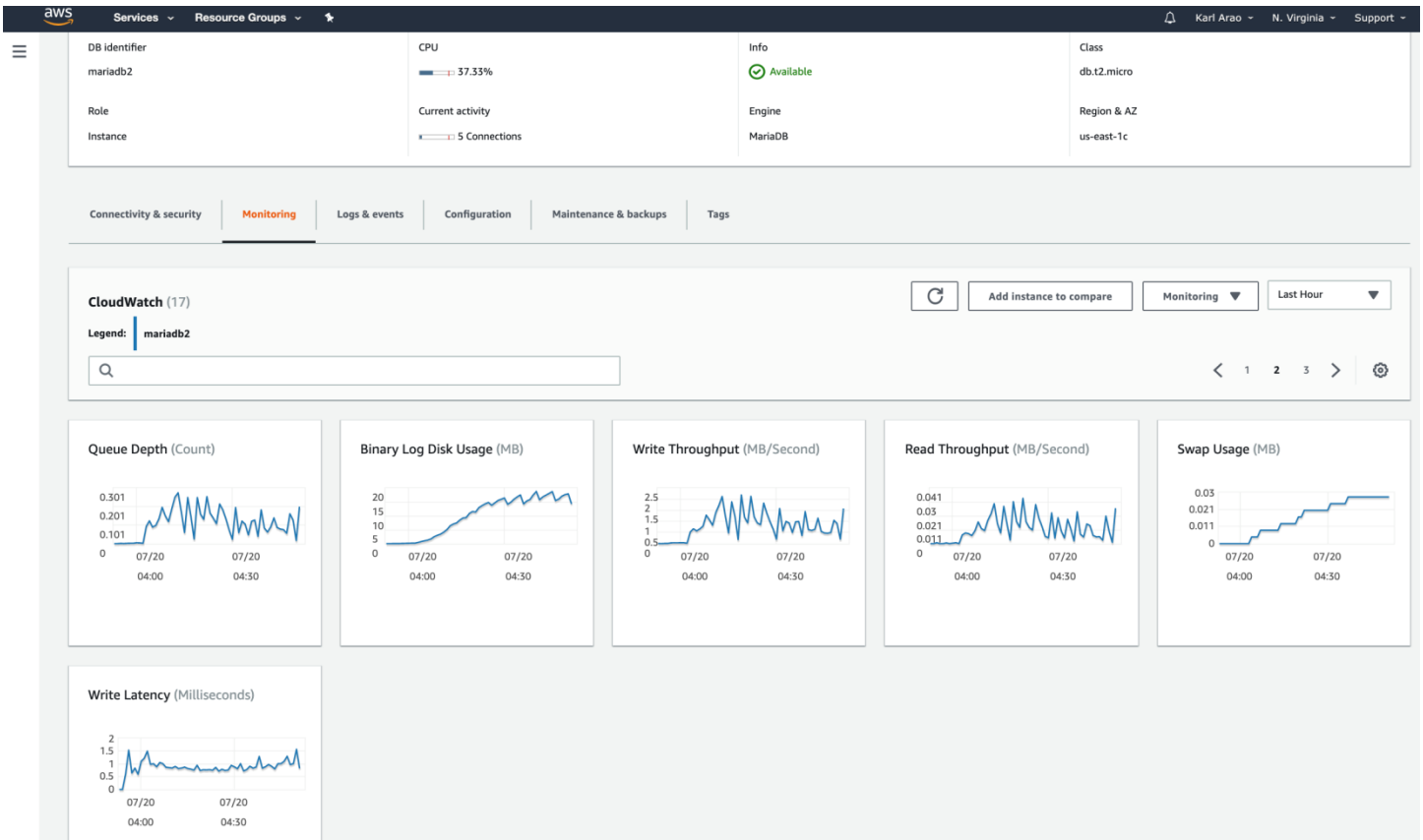
```
1) adhoc SQL that runs for 300 seconds

   SELECT a.table_schema, a.table_name, sleep(1)
   FROM information_schema.tables a, information_schema.tables b
   where a.table_schema <> 'karl'
   limit 300;

2) mysqlslap

   $ mysqlslap -h mariadb2.c3yfiwlyerg0.us-east-1.rds.amazonaws.com --port=3306 -u
   mariadb2 -pwelcome1 --concurrency=25 --iterations=10 --auto-generate-sql --number-of-
   queries=100000

3) deadlock scenario

   create table innodb_deadlock_maker(a int primary key) engine=innodb;
   insert into innodb_deadlock_maker(a) values(0), (1);
   -- connection 0
   set transaction isolation level serializable;
   start transaction;
   select * from innodb_deadlock_maker where a = 0;
   update innodb_deadlock_maker set a = 0 where a <> 0;
   -- connection 1
   set transaction isolation level serializable;
   start transaction;
   select * from innodb_deadlock_maker where a = 1;
   update innodb_deadlock_maker set a = 1 where a <> 1;
   -- session waits are: "Searching rows for update"
```

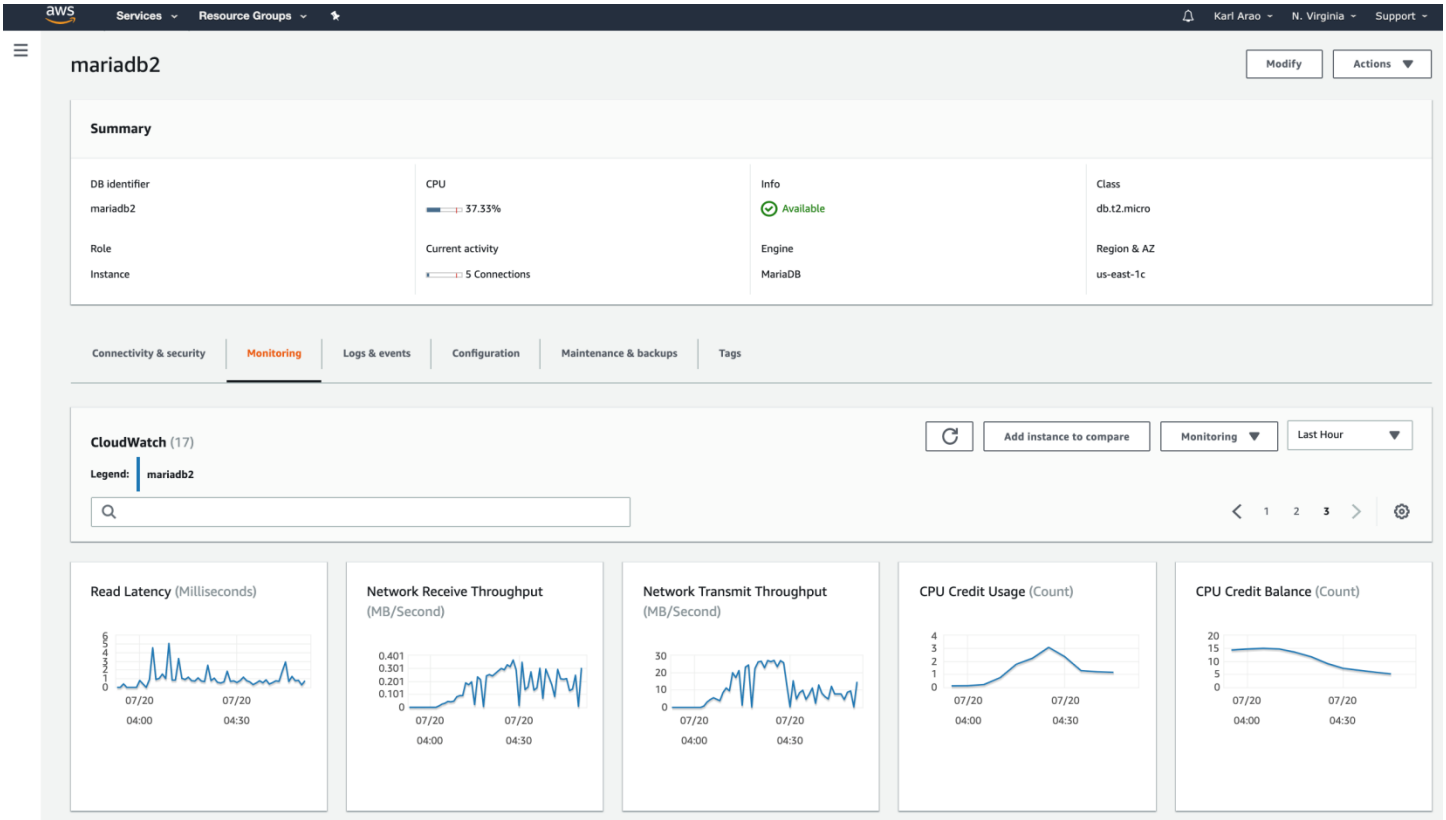
Below are some example screenshots:

System metrics using Cloudwatcher

- CPU Utilization peaked at 70%
- The example workload is heavy on Write IOPS and peaked at around 350-400 IOPS


Services
Resource Groups
Karl Arao
N. Virginia
Support
DB identifier
mariadb2
CPU
37.33%
Info
Available
Class
db.t2.micro
Role
Instance
Current activity
5 Connections
Engine
MariaDB
Region & AZ
us-east-1c
Connectivity & security
Monitoring
Logs & events
Configuration
Maintenance & backups
Tags
CloudWatch (17)
Legend:
mariadb2
Add instance to compare
Monitoring
Last Hour
CPU Utilization (Percent)
DB Connections (Count)
Free Storage Space (MB)
Freeable Memory (MB)
Write IOPS (Count/Second)
Read IOPS (Count/Second)


Services
Resource Groups
Karl Arao
N. Virginia
Support
DB identifier
mariadb2
CPU
37.33%
Info
Available
Class
db.t2.micro
Role
Instance
Current activity
5 Connections
Engine
MariaDB
Region & AZ
us-east-1c
Connectivity & security
Monitoring
Logs & events
Configuration
Maintenance & backups
Tags
CloudWatch (17)
Legend:
mariadb2
Add instance to compare
Monitoring
Last Hour
Queue Depth (Count)
Binary Log Disk Usage (MB)
Write Throughput (MB/Second)
Read Throughput (MB/Second)
Swap Usage (MB)
Write Latency (Milliseconds)

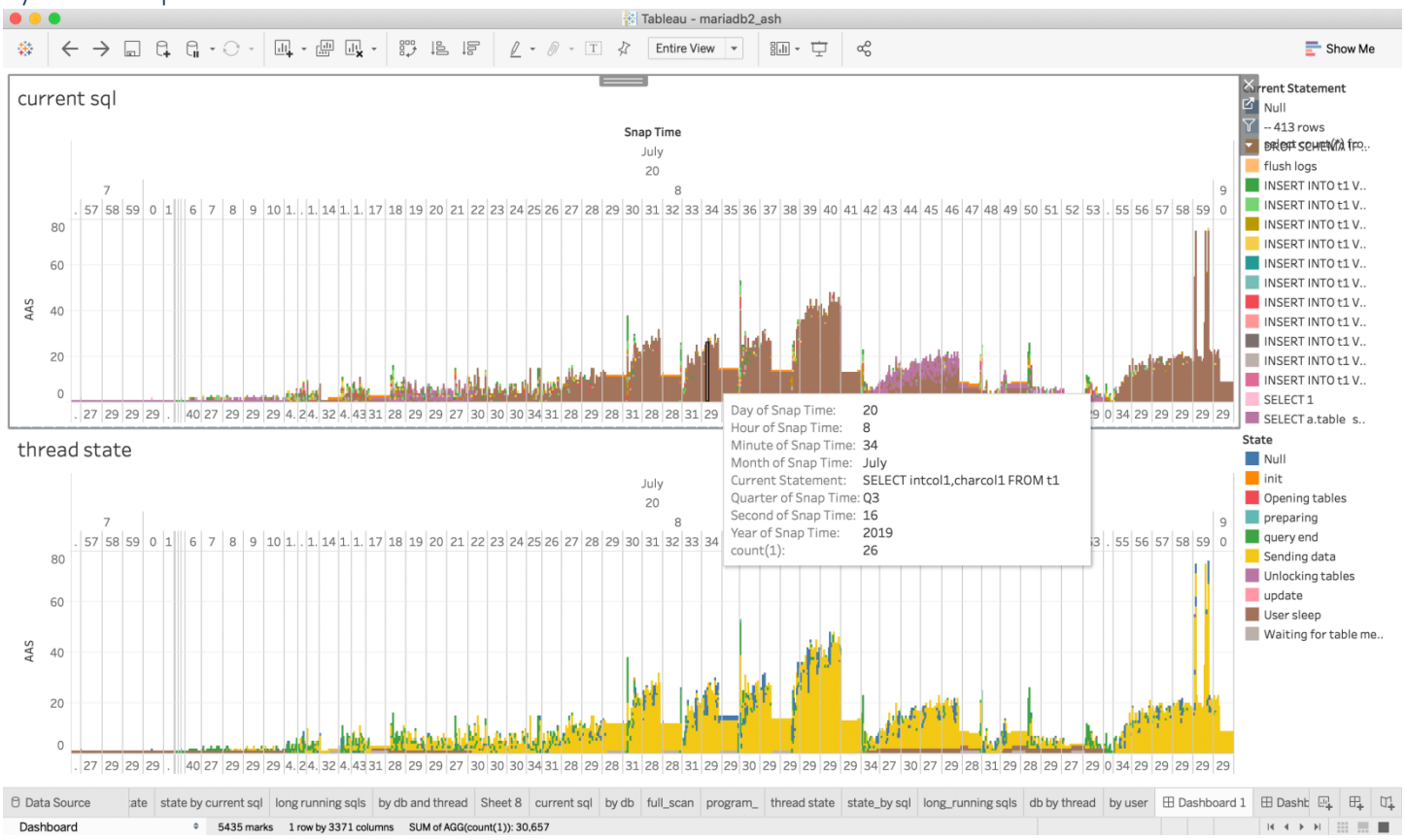


Workload drill down using ash_sampler data

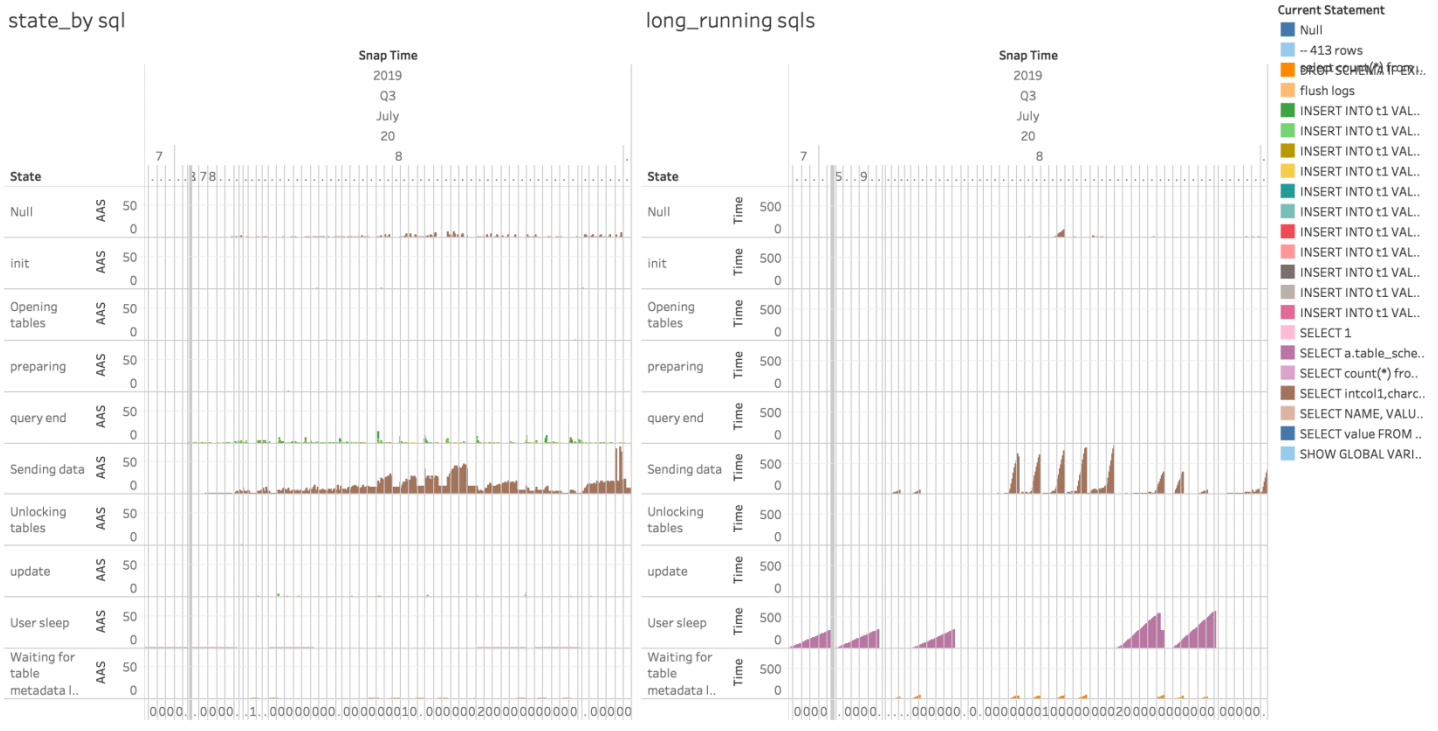
- Cloudwatcher system metrics correlated with ash_sampler session data provides a lot of insights on solving complex performance problems.

- The ash_sampler data can be sliced and diced using the available dimensions. And the five Ws can be asked/answered:
    - who is executing the SQL
    - what the application is doing
    - when is it doing that slow process
    - where most of the time is going
    - why are the sessions locking and causing deadlocks

by current sql and thread state



state by sql and long running sqls



deadlock sessions