



Cloud Profiling Workshop

PATIENT MONITORING

Lei Yin (lei.yin@philips.com)

Jiahua Liu (jiahua.liu@philips.com)

Qiong Wu (sherry.Wu@philips.com)

Kairavi Mehta (kairavi.mehta@philips.com)

Aditya Sirohi (aditya.sirohi@philips.com)

Sankar Gireesan Nair (sankar.gireesannair@philips.com)

Harsh Patel (harsh.patel@philips.com)

Contents

Docker Installation	2
Download, Build JMeter Docker Image.....	2
Getting Started with AWS Console and AWS Services	3
Login in AWS Console.....	3
Find AWS Services in Console	3
Subscribe to the IoT topics.....	4
DynamoDB	5
Elastic Beanstalk Monitoring and Configuration.....	5
CloudWatch Metrics	6
CloudWatch Alarms	7
Run JMeter Docker Image to emulate data to cloud	10
AWS Resources Observation after load test has started	10
MQTT watch for receiving payloads	10
DynamoDB	10
Elastic Beanstalk Monitoring.....	11
CloudWatch memory utilization metrics	11
EC2 autoscaled	11
CloudWatch Alarms	11
DataDog Charts.....	12
Login to DataDog console	12
Use Dashboards and Metrics to observe high workload that has been emulated through JMeter Docker	12
Create your own DataDog Dashboard	13

Docker Installation

1. Download community edition docker based on the OS

<https://www.docker.com/community-edition>

2. Follow the installer to complete installation.

Docker will enable virtualization on Windows at BIOS level.

More instruction in installation: <https://docs.docker.com/docker-for-windows/>

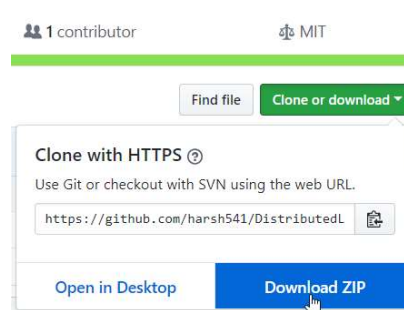
3. To confirm successful installation, give the command > docker --version

```
Administrator: Command Prompt
C:\Users\310243108>docker --version
Docker version 18.03.1-ce, build 9ee9f40
```

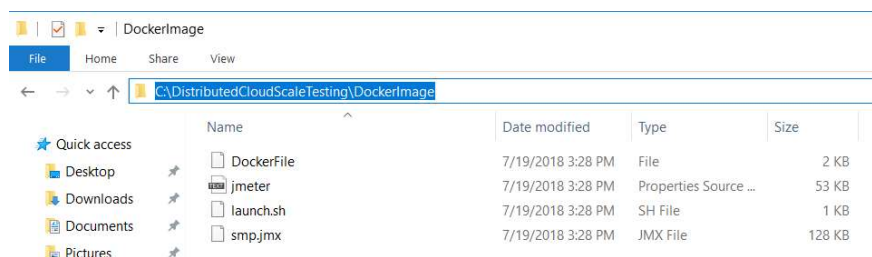
Download, Build JMeter Docker Image

1. Download JMeter Docker Image from <https://tinyurl.com/y9wzvfvb>

2. Clone the GIT repo and download as ZIP



3. Unzip the archive to your local filesystem



4. Open command prompt and direct to the extracted folder, i.e. DockerImage folder

```
Administrator: Command Prompt
C:\Users\310243108>cd C:\DistributedCloudScaleTesting\DockeImage
C:\DistributedCloudScaleTesting\DockeImage>docker build -t jmeterdocker:version1 .
Sending build context to Docker daemon 190kB
Step 1/12 : FROM      hauptmedia/java:oracle-java8
Get https://registry-1.docker.io/v2/: net/http: request canceled while waiting for connection (Client.Timeout e
xceeded while awaiting headers)
```

5. Run the following command to build the docker image

```
C:\DistributedCloudScaleTesting\DockeImage>docker build -t jmeterdocker:version1 .
Sending build context to Docker daemon 190kB
Step 1/12 : FROM      hauptmedia/java:oracle-java8
oracle-java8: Pulling from hauptmedia/java
8b87079b7a06: Pull complete
a3ed95cae02: Pull complete
797bafb726a1: Pull complete
ab4aa99383d6: Pull complete
21ccbf2e52f5: Pull complete
Digest: sha256:780be864c412cd20063901c1259eec6d17eed779137c8bcb68bd1569b02e5391
Status: Downloaded newer image for hauptmedia/java:oracle-java8
```

Following message is seen once docker image is successfully built

```
Step 9/12 : ADD jmeter.properties /opt/apache-jmeter-4.0/
--> ebed68e34388
Step 10/12 : COPY launch.sh /opt/apache-jmeter-4.0/launch.sh
--> f68137ed8531
Step 11/12 : ADD smp.jmx /opt/apache-jmeter-4.0/smp.jmx
--> a273ac3353e0
Step 12/12 : WORKDIR      ${JMeter_HOME}
Removing intermediate container 56dc3aed44a5
--> 29105a4e7edc
Successfully built 29105a4e7edc
Successfully tagged jmeterdocker:version1
```

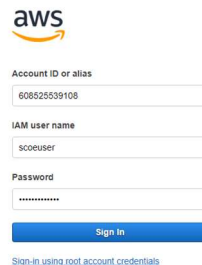
Getting Started with AWS Console and AWS Services

Login in AWS Console

Open the AWS Console login:

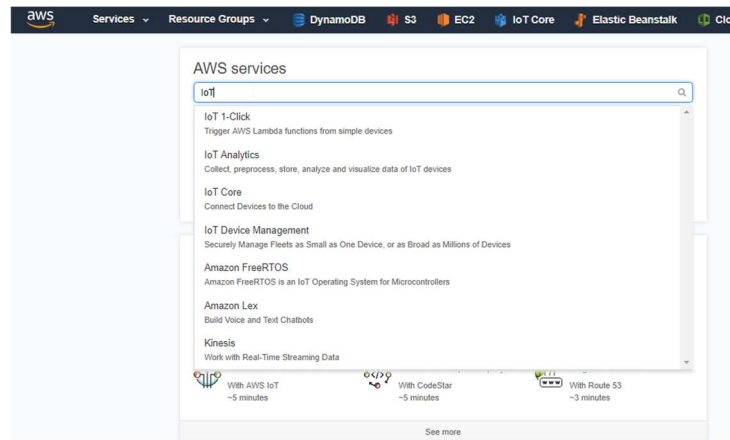
<https://608525539108.signin.aws.amazon.com/console>

Using the 'IAM user name' and 'Password' login provided in the workshop. Click 'Sign in'.

The image shows the AWS sign-in page. At the top is the AWS logo. Below it are four input fields: 'Account ID or alias' with the value '608525539108', 'IAM user name' with the value 'scoeuser', and 'Password' with masked characters. A blue 'Sign in' button is at the bottom. A small link 'Sign in using root account credentials' is at the very bottom.

Find AWS Services in Console

To find other AWS services, type in a service name in search box below. For example, type 'IoT'; in search box, and click on 'IoT Core':

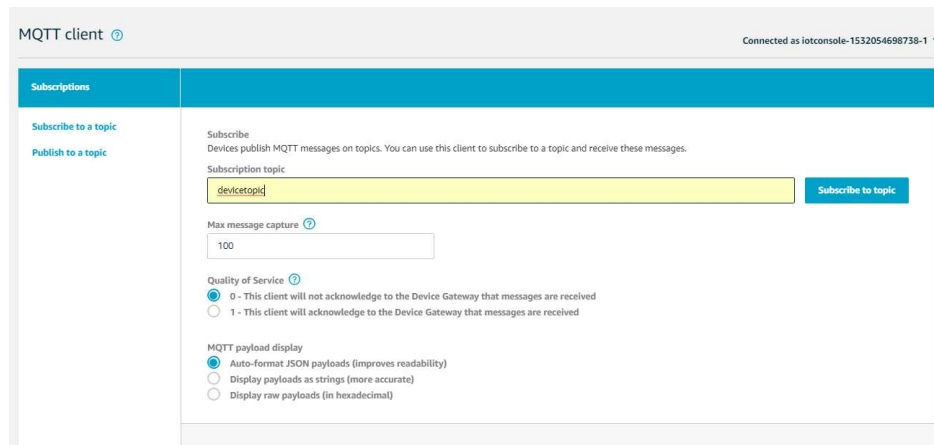


Subscribe to the IoT topics

Go to “IoT Core->Test” and see ‘MQTT client’ options:



Click ‘Subscribe to a topic’ and type ‘devicetopic’

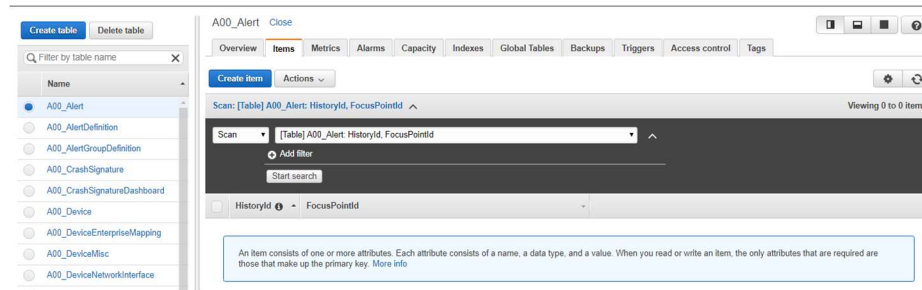


Click ‘Subscribe to topic’ button:



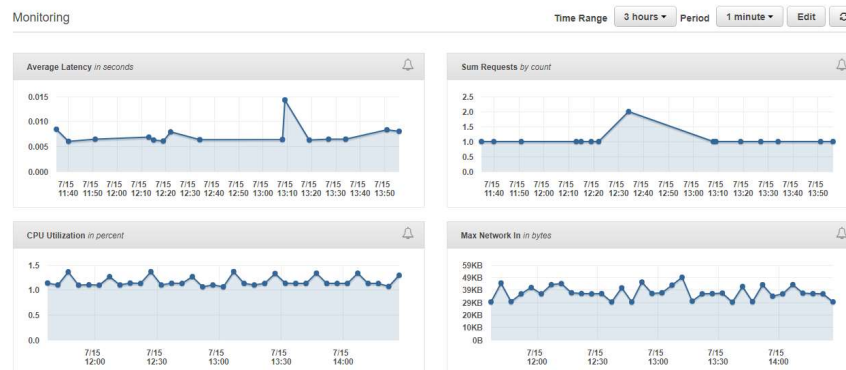
DynamoDB

Open another browser tab, go to “DynamoDB->Tables” and select one Device table, the tables are empty:

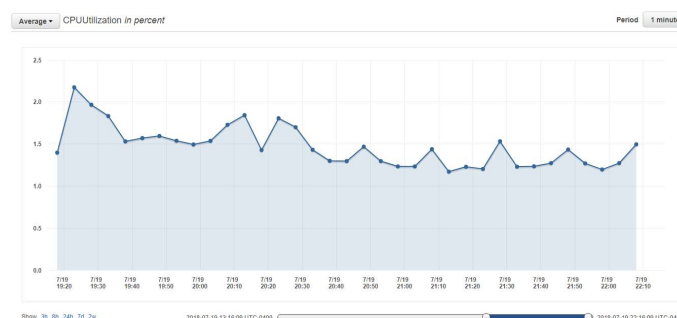


Elastic Beanstalk Monitoring and Configuration

Open another browser tab, go to “Elastic Beanstalk->Monitoring” and see all the default monitoring charts for the baseline before sending data, change on the ‘Time Range’ ‘Period’ controls and refresh button to see charts get updated:



Click on ‘CPU Utilization’ and see the details. Change “Average” to other options, change ‘Period’ and slider on the chart



Go to “Elastic Beanstalk->Configuration->Capacity” and see the autoscaling settings and show all the default trigger options, ‘Statistics’, ‘Upper threshold’, ‘Lower threshold’ etc.

Modify capacity

Auto Scaling Group

Configure the compute capacity of your environment and Auto Scaling settings to optimize the number of instances used.

Environment type: Load balanced

Instances: Min 1 Max 5

Availability Zones: Any 1

Number of Availability Zones (AZs) to use:

Placement: us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f

Specify Availability Zones (AZs) to use:

Scaling cooldown: 360 seconds

Scaling triggers

Metric: CPUUtilization

Statistic: NetworkIn, NetworkOut, DiskWriteOps, DiskReadOps, DiskWriteBytes, DiskReadBytes, Latency, RequestCount, HealthyHostCount, UnHealthyHostCount

Unit: DiskWriteOps, DiskReadOps, DiskWriteBytes, DiskReadBytes, Latency, RequestCount, HealthyHostCount, UnHealthyHostCount

Period: DiskWriteOps, DiskReadOps, DiskWriteBytes, DiskReadBytes, Latency, RequestCount, HealthyHostCount, UnHealthyHostCount

Breach duration: Latency, RequestCount, HealthyHostCount, UnHealthyHostCount

Upper threshold: RequestCount, HealthyHostCount, UnHealthyHostCount

Scale up increment: UnHealthyHostCount

Lower threshold: 0.01 Percent

And ‘Time-based Scaling’ at the end.

Time-based Scaling

Use the following settings to control time-based scaling actions. [Learn more](#)

Current status: 1 instance(s) in service, Min: 1, Max: 5

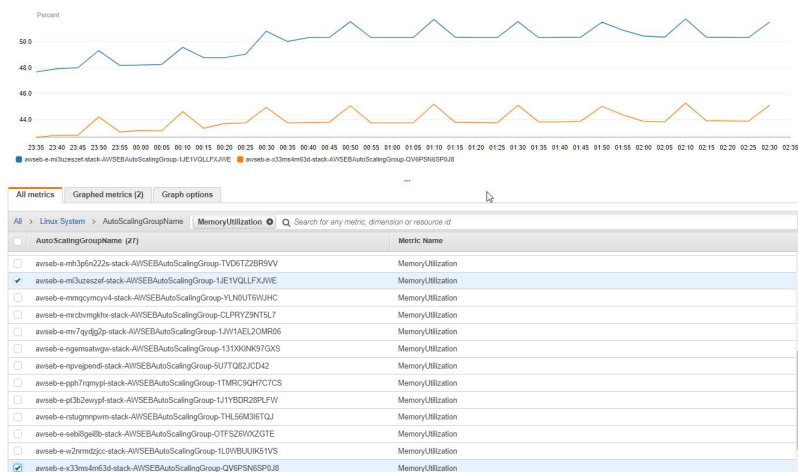
Time zone: UTC Local

Actions Add scheduled action

Name	Min	Max	Desired	Next occurrence (UTC)
No scheduled actions				

CloudWatch Metrics

Open another browser tab, go to “CloudWatch->metrics->Linux Systems->AutoScalingGroupName”, search ‘MemoryUtilization’, select ‘MemoryUtilization’ metric’s with data:



Click on 'Graphed metrics' and try with details:

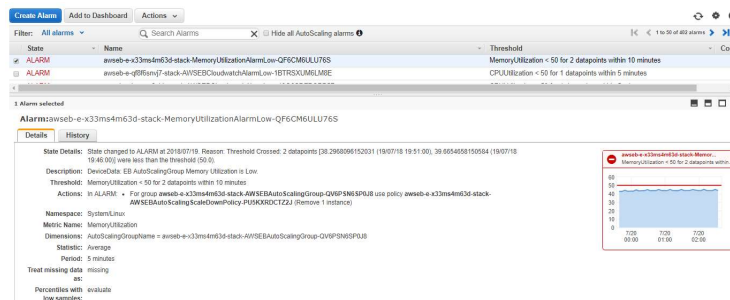


CloudWatch Alarms

Log in to AWS Console, go to "CloudWatch->Alarms", see 'All alarms':

Create Alarm		Add to Dashboard	Actions			
Filter: All alarms		Search Alarms		Hide all AutoScaling alarms		
State	Name	Threshold				
ALARM	TargetTracking-table/A00_DeviceNetworkInterface/Index/ExtractionIndex-AlarmLow-da68238-e3c2-4973-8460-a51c9eb36b6	ConsumedReadCapacityUnits < 3,600 for 15 datapoints within 15 minutes				
ALARM	TargetTracking-table/A00_DevicePlatform/Index/ExtractionIndex-AlarmLow-9a24878c-fac1-4aa1-8262-7b1d7f5481f1	ConsumedReadCapacityUnits < 3,600 for 15 datapoints within 15 minutes				
ALARM	TargetTracking-table/A00_DeviceSoftware/Index/ExtractionIndex-AlarmLow-edcb36d0-466c-4fa4-9a55-e88a06ba5d12	ConsumedReadCapacityUnits < 3,600 for 15 datapoints within 15 minutes				
ALARM	TargetTracking-table/A00_Device/Index/ExtractionIndex-AlarmLow-921de0d5-3ca8-469a-a004-ac16757b0c4	ConsumedReadCapacityUnits < 3,600 for 15 datapoints within 15 minutes				
ALARM	awseb-e-x33ms4m63d-stack-MemoryUtilizationAlarmLow-QF6CM6ULU76S	MemoryUtilization < 50 for 2 datapoints within 10 minutes				
ALARM	awseb-e-q8ffsmfj7-stack-AWSEBCloudwatchAlarmLow-1BTR5XUMLM8E	CPUUtilization < 50 for 1 datapoints within 5 minutes				
ALARM	awseb-e-btapaq3abb-stack-AWSEBCloudwatchAlarmLow-10Q8SDZPOPP5D	CPUUtilization < 50 for 1 datapoints within 5 minutes				
ALARM	awseb-e-49779ybjps-stack-AWSEBCloudwatchAlarmLow-87VB3I10I0G	CPUUtilization < 50 for 1 datapoints within 5 minutes				
ALARM	awseb-e-m332cnuz-stack-AWSEBCloudwatchAlarmLow-153FLV7H6BSXS	CPUUtilization < 50 for 1 datapoints within 5 minutes				
ALARM	awseb-e-jgpyhuxuik-stack-AWSEBCloudwatchAlarmLow-FW3NLSLUM3	CPUUtilization < 50 for 1 datapoints within 5 minutes				
INSUFFICIENT_DATA	TargetTracking-table/A00_Inventory-AlarmHigh-962c5c11-7747-43aa-ab72-af64fb8de58a	ConsumedReadCapacityUnits >= 4,800 for 5 datapoints within 5 minutes				
INSUFFICIENT_DATA	A00_Inventory-ReadCapacityUnitsLimit-BasicAlarm	ConsumedReadCapacityUnits >= 4,800 for 5 datapoints within 5 minutes				
INSUFFICIENT_DATA	TargetTracking-table/A00_DevicePlatform/Index/ExtractionIndex-AlarmHigh-32c86a2-e909-4c57-808b-c0d24cc453dd	ConsumedReadCapacityUnits >= 4,800 for 5 datapoints within 5 minutes				
INSUFFICIENT_DATA	TargetTracking-table/A00_DeviceSoftware/Index/ExtractionIndex-AlarmHigh-7547eaa7-03fe-431d-b356-041608067c2c	ConsumedReadCapacityUnits >= 4,800 for 5 datapoints within 5 minutes				
INSUFFICIENT_DATA	TargetTracking-table/A00_DeviceNetworkInterface/Index/ExtractionIndex-AlarmHigh-3ad60ff-21cf-4c41-8d27-2a9011295172	ConsumedReadCapacityUnits >= 4,800 for 5 datapoints within 5 minutes				

Then select our customized MemoryUtilization Alarm and show 'Details' and 'History' tabs.



Click on 'Action->Modify' to try, and show the 'AutoScaling Action':

Modify Alarm

1. Select Metric 2. Define Alarm

Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name:

Description:

Whenever: MemoryUtilization

Is:

for: 2 out of 2 datapoints

Additional settings

Provide additional configuration for your alarm.

Treat missing data as:

Actions

Define what actions are taken when your alarm changes state.

AutoScaling Action

Whenever this alarm:

From resource type:

From the:

Take this action:

Alarm Preview

This alarm will trigger when the blue line goes below the red line for 2 datapoints within 10 minutes

awsseb-e-x33ms4m63d-stack-MemoryUtilization < 50 for 2 datapoints within 10 min...

Namespace: SystemLinux

AutoScaling Group Name:

Metric Name:

Period:

Statistic:

Average

Cancel Previous Next Save Changes

In the AutoScaling 'Action' section see the option in 'Take this action':

Modify Alarm

1. Select Metric 2. Define Alarm

Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name:

Description:

Whenever: MemoryUtilization

Is:

for: 2 out of 2 datapoints

Additional settings

Provide additional configuration for your alarm.

Treat missing data as:

Actions

Define what actions are taken when your alarm changes state.

AutoScaling Action

Whenever this alarm:

From resource type:

From the:

Take this action:

Please select a policy...

awsseb-e-x33ms4m63d-stack-AWSEBAutoScalingScaleDownPolicy-1458CAURVUR3F - Remove 1 instance

awsseb-e-x33ms4m63d-stack-AWSEBAutoScalingScaleUpPolicy-1458CAURVUR3F - Add 1 instance

Alarm Preview

This alarm will trigger when the blue line goes below the red line for 2 datapoints within 10 minutes

awsseb-e-x33ms4m63d-stack-MemoryUtilization < 50 for 2 datapoints within 10 min...

Namespace: SystemLinux

AutoScaling Group Name:

Metric Name:

Period:

Statistic:

Average

Cancel Previous Next Save Changes

Open another browser tab, go to "EC2->Auto Scaling Groups", it should match one of the AutoScaling Policies:

One of the ‘Scaling Policies’ should match the above “AutoScaling action”:

Try adding ‘Notification’ action on ‘Modify Alarm’:

Run JMeter Docker Image to emulate data to cloud

Run the docker image giving following command:

```
docker run -d=false -i -t jmeterdocker:version1 sh launch.sh <tenant-name> <access-key>
```

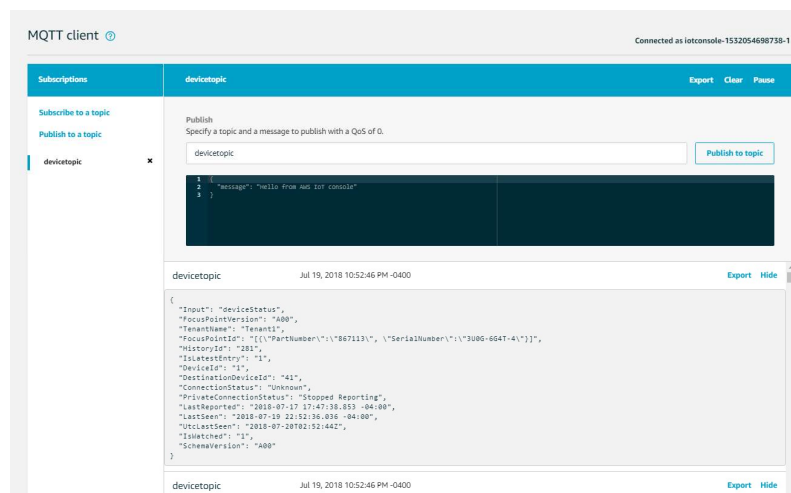
tenant-name = will be provided in the workshop

access-key = will be provided in the workshop

AWS Resources Observation after load test has started

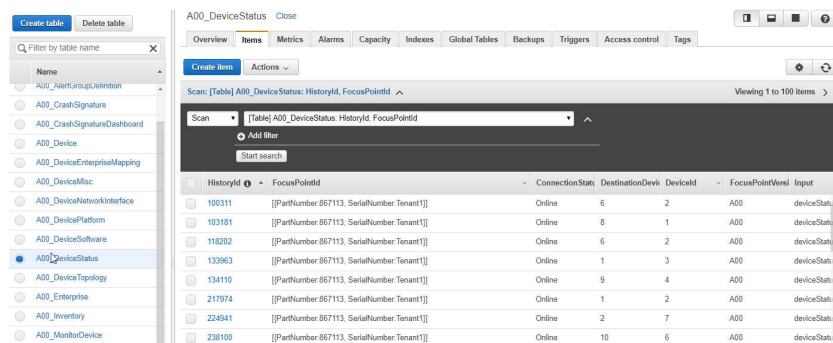
MQTT watch for receiving payloads

Go to tab for “IoT Core->Test”, watch receiving JMeter test payloads:



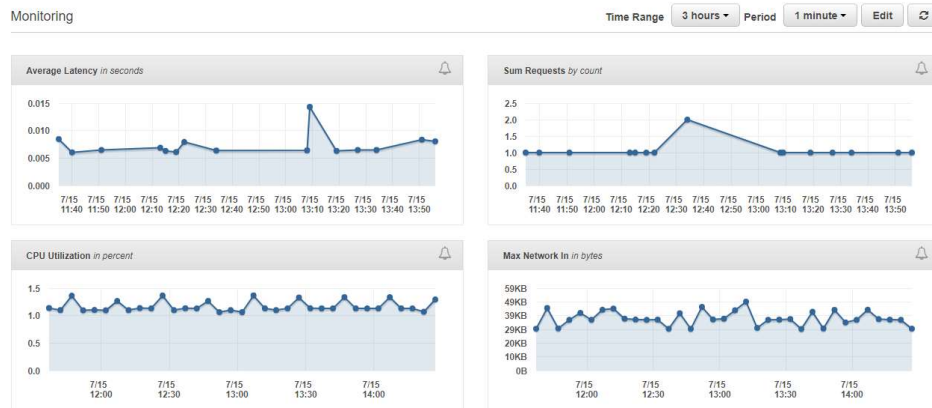
DynamoDB

Go to tab for “DynamoDB->Tables” and select one Device table, watch for new items inserted into table as JMeter sending payloads:



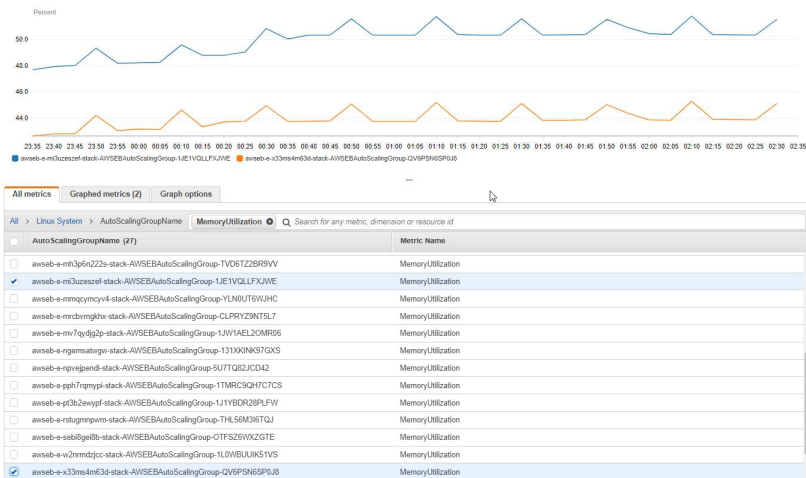
Elastic Beanstalk Monitoring

Go to the tab for “Elastic Beanstalk->Monitoring” and see all the “CPU Utilization” goes up upon receiving data:



CloudWatch memory utilization metrics

Go to the tab for “CloudWatch->metrics->Linux Systems->AutoScalingGroupName”, search ‘MemoryUtilization’, watch for ‘MemoryUtilization’ metrics’ increase:



EC2 autoscaled

Log in to AWS Console, go to “EC2->All Instances”, watch autoscaled instances, e.g. 2 ‘DeviceData’ instances.

The screenshot shows the AWS Management Console EC2 Dashboard. The 'Instances' tab is selected, and a list of instances is displayed. Two instances are highlighted: 'DeviceData' and 'DeviceData2'. The 'DeviceData' instance is of type 't2.micro' and is in the 'running' state. The 'DeviceData2' instance is also of type 't2.micro' and is in the 'running' state.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
StatisticsData	i-0701ca8364e04241	t1.micro	us-east-1b	running	2/2 checks ...	None	ec2-34-224-190-214.co.	34.224.190.214	-
StatisticsData	i-0046f1d4b30e4e689	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-52-201-255-131.co.	52.201.255.131	-
DeviceData	i-073ba6381bf3938b8	t2.medium	us-east-1b	running	2/2 checks ...	None	ec2-54-174-177-224.co.	54.174.177.224	-
DeviceData2	i-08b45034178b48c	t2.medium	us-east-1b	running	2/2 checks ...	None	ec2-34-226-192-216.co.	34.226.192.216	-
KeyPerformanceIndicator	i-08b6c2137e9d98ea	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-54-144-164.co.	54.144.164.164	-
Inventory	i-0c0a0c3b07a9b6b	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-54-175-192-129.co.	54.175.192.129	-
IntegrationService	i-04e4e089d8a12ca	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-54-175-192-129.co.	54.175.192.129	-
DeviceData2	i-041185c03b4a862	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-184-72-64-174.co.	184.72.64.174	-
DeviceData	i-0cdee18ba3b67c4	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-34-229-88-51.co.	34.229.88.51	-

CloudWatch Alarms

Go to the tab “CloudWatch->Alarms”, watch “MemoryUtilization’ metrics’ in ‘Alarm’ state:

Create Alarm Add to Dashboard Actions			
Filter:	All alarms	Search Alarms	Hide all AutoScaling alarms
State	Name	Threshold	Conf
ALARM	TargetTracking-table/A00_DeviceNetworkInterface/Index/ExtractionIndex-AlarmLow-da68238-e3c2-4973-8469-a51c9eb3066	ConsumedReadCapacityUnits < 3,600 for 15 datapoints within 15 minutes	
ALARM	TargetTracking-table/A00_DevicePlatform/Index/ExtractionIndex-AlarmLow-9a24878c-fac4-4ae1-8252-7b1d7f5d81f1	ConsumedReadCapacityUnits < 3,600 for 15 datapoints within 15 minutes	
ALARM	TargetTracking-table/A00_DeviceSoftware/Index/ExtractionIndex-AlarmLow-ed3c36d8-466c-4fa4-9a55-a86e08ba5d12	ConsumedReadCapacityUnits < 3,600 for 15 datapoints within 15 minutes	
ALARM	TargetTracking-table/A00_DeviceIndex/ExtractionIndex-AlarmLow-921d6d45-3ca8-4a9a-a004-ac16757bbcc4	ConsumedReadCapacityUnits < 3,600 for 15 datapoints within 15 minutes	
ALARM	awsab-e-333ms4mE3d-stack-MemoryUtilizationAlarmLow-QFECMLULUT6S	MemoryUtilization < 50 for 2 datapoints within 10 minutes	
ALARM	awsab-e-qf8cm7f-stack-AWSEBCloudwatchAlarmLow-1BTRSCXUMLME	CPUUtilization < 50 for 1 datapoints within 5 minutes	
ALARM	awsab-e-tpagp3abb-stack-AWSEBCloudwatchAlarmLow-10Q8SOZPOPP5D	CPUUtilization < 50 for 1 datapoints within 5 minutes	
ALARM	awsab-e-F977yfgs-stack-AWSEBCloudwatchAlarmLow-87V631Y0RG	CPUUtilization < 50 for 1 datapoints within 5 minutes	
ALARM	awsab-e-m32czunz-stack-AWSEBCloudwatchAlarmLow-153FLV7H6K5XS	CPUUtilization < 50 for 1 datapoints within 5 minutes	
ALARM	awsab-e-jggyhuuwx-stack-AWSEBCloudwatchAlarmLow-FV3N0L5LM3	CPUUtilization < 50 for 1 datapoints within 5 minutes	
INSUFFICIENT_DATA	TargetTracking-table/A00_Inventory-AlarmHigh-962c5c11-7747-43aa-ab72-a84b8de68a	ConsumedReadCapacityUnits > 4,800 for 5 datapoints within 5 minutes	
INSUFFICIENT_DATA	A00_Inventory-ReadCapacityUnitsLimit-BasicAlarm	ConsumedReadCapacityUnits >= 480 for 5 datapoints within 5 minutes	No no
INSUFFICIENT_DATA	TargetTracking-table/A00_DevicePlatform/Index/ExtractionIndex-AlarmHigh-32d86a2-4909-4c57-808b-c024cc453dd	ConsumedReadCapacityUnits > 4,800 for 5 datapoints within 5 minutes	
INSUFFICIENT_DATA	TargetTracking-table/A00_DeviceSoftware/Index/ExtractionIndex-AlarmHigh-7547aea7-036-431d-b356-8416086b72c	ConsumedReadCapacityUnits > 4,800 for 5 datapoints within 5 minutes	
INSUFFICIENT_DATA	TargetTracking-table/A00_DeviceNetworkInterface/Index/ExtractionIndex-AlarmHigh-3ad68ff-2fd4-4c41-8271-2a5011295172	ConsumedReadCapacityUnits > 4,800 for 5 datapoints within 5 minutes	

And if configured notification, should expect an email on this alarm.

DataDog Charts

Login to DataDog console

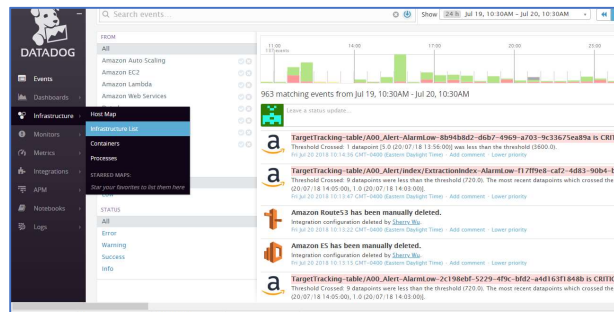
<https://www.datadoghq.com/>

Credentials will be provided in the handout during the workshop

Use Dashboards and Metrics to observe high workload that has been emulated through JMeter Docker

1. Network Spikes on Elastic Load Balancer

On Home page, choose “Infrastructure” from menu bar -> choose “Infrastructure List”

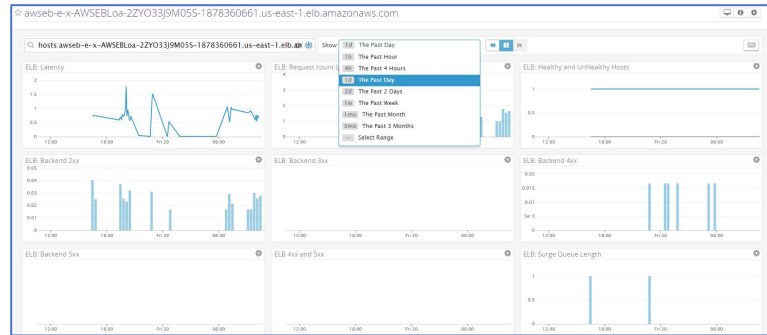


Click on “devicedata” which has “elb” listed under its Apps

21 Hosts up / 21 total					
Q devicedata Group by tags: Showing all 4 matching entries.					
AWS Name	Status	CPU i	IOWait	Load 15	Apps
scoedevicedata	UP				aws
devicedatasherryenvir	UP				aws
devicedata	UP				elb
devicedata	UP				aws

You will be redirected to this ELB’s dashboard and can observe network traffic spikes through this load gateway created by JMeter load simulator.

Adjust the time interval to be “Past 4 Hours”, “Past Day”, etc. to monitor the traffic trends.



2. High Disk I/O on EC2

Follow the same steps as in 2.1, but instead of choosing the “devicedata” that has “elb” under its Apps, choose the other one that has “aws” in the Apps column.

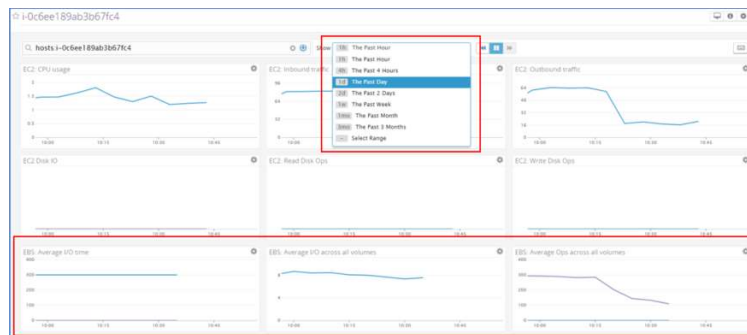
21 Hosts up / 21 total

Search: devicedata Group by tags: Showing all 4 matching entries.

AWS Name	Status	CPU	IOWait	Load 15	Apps
scoedevicedata	UP				aws
devicedatasherryenvir	UP				aws
devicedata	UP				aws
devicedata	UP				aws

You will be redirected to this EC2 instance’s dashboard and can observe high disk I/O caused by load pressure added by JMeter load simulator

Note that disk I/O is against EBS (elastic block storage) instead of EC2, this is because we attached EBS to EC2 as storage.

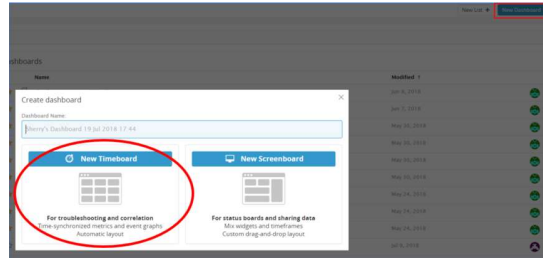


Adjust time interval to “Past 4 Hours”, “Past Day”, etc. to monitor traffic trends.

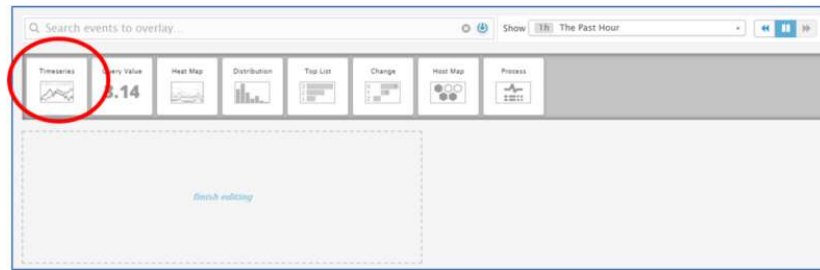
Create your own DataDog Dashboard to observe spikes in other resources

1. Create your own dashboard and metric to observe increasing DynamoDB write throughput

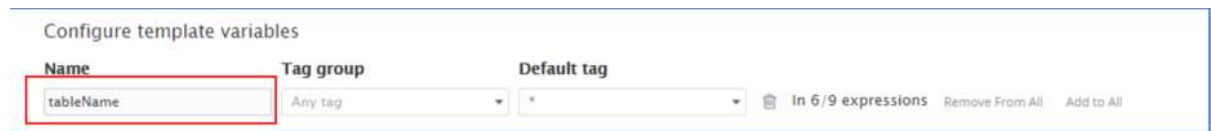
Click “New Dashboard +”



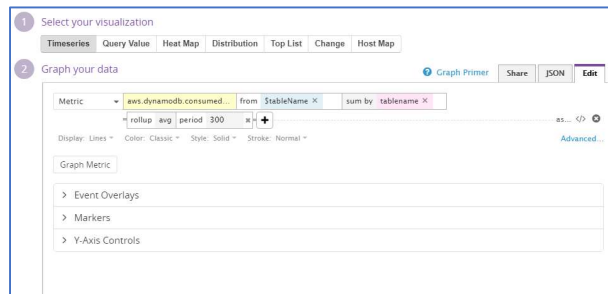
Choose "New Timeboard" -> "Timeseries" since we want to see the throughput changes against timeline:



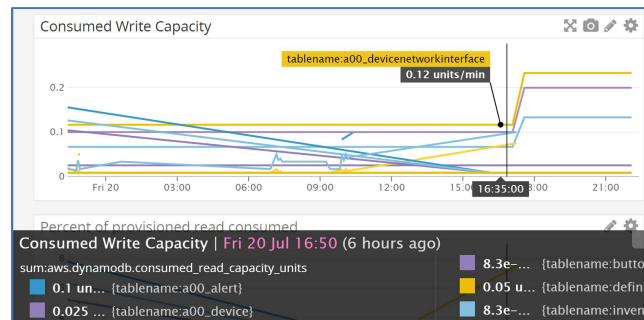
Configure template variables as follows so that statistics can be aggregated per table:



Choose "aws.dynamodb.provisioned_write_capacity_units" from "\$tableName" sum by "tablename"



You should be able to observe the increasing write throughput in the metric just created:



- Question: Can you please create a metric that allows us to observe percentage of consumed read capacity of each dynamo table?