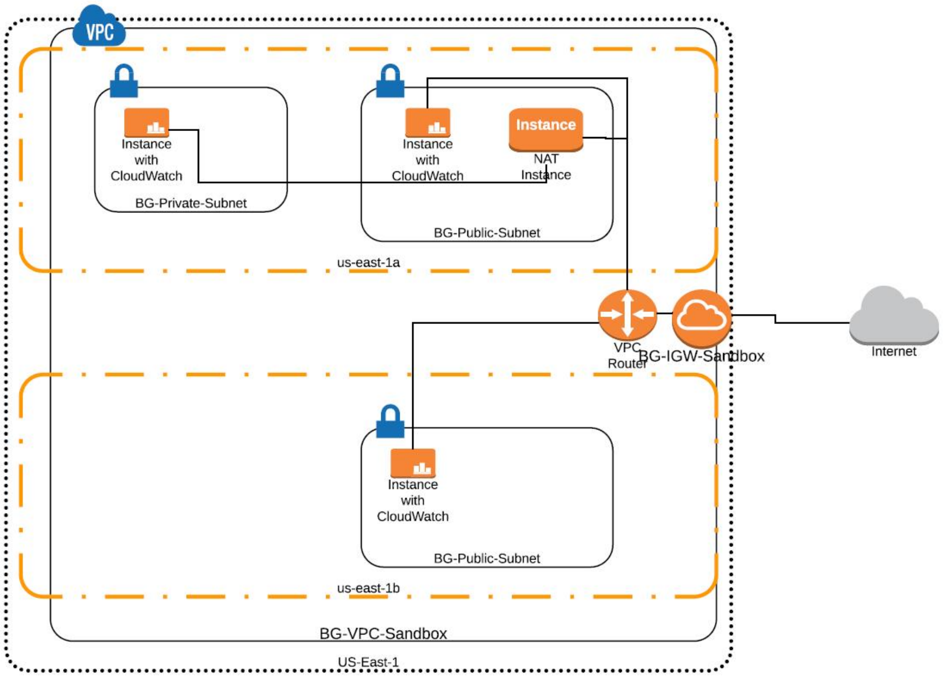
1.) Achieved below objectives using Basic Architect:



- Able to create personal VPC with 2 Public and 2 Private Subnet assigned to it.
- Created IGW (Internet Gateway) and attached to VPC
- Create Public & Private Subnets with associated Routes

- Able to create "Launch Configuration" using ami-c58c1dd3/t2-micro and User data listed below -

User data: (below will install perl, Cloudwatch monitoring script for Memory & Swap space and aws logs on all new EC2 instance)

```
#!/bin/bash
yum update -y
sudo yum install -y perl-Switch perl-DateTime perl-Sys-Syslog perl-LWP-Protocol-https awslogs
httpd24
cd /var/tmp
curl http://aws-cloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-1.2.1.zip
unzip CloudWatchMonitoringScripts-1.2.1.zip
rm CloudWatchMonitoringScripts-1.2.1.zip
chkconfig awslogs on
service awslogs start
service httpd start
chkconfig httpd on
```

- Created "Auto Scaling Group" in Public and Private Subnet using same Launch Configuration above

o Desired Instance: 2
o Min: 1
o Max:2

- Created SNS Topics with email Subscription

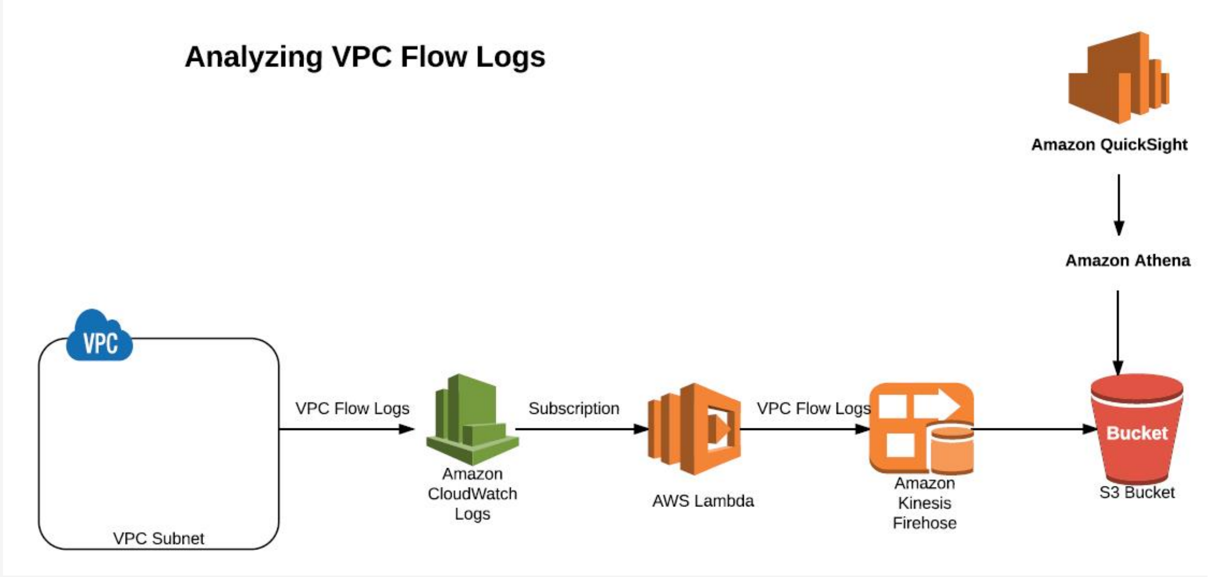
o It can be used with CloudWatch email alerts when Metrics (ex. CPU >80% for 10mins) reach certain threshold

- Generated basic + customized metrics chart from EC2 instance on Dashboard

o Created IAM Role to allow CloudWatchFullAccess to EC2 instance
o Created cron job which run every 5mins to collect memory, swap and disk space utilization data on system and make remote call to Amazon Cloudwatch to report the collected data as Custom metrics (Metrics Name - Linux System).
o http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/mon-scripts.html

• Installed cloudwatch logs agent on EC2 instance as part of Launch Configuration.
o Forwarding /var/log/messages to cloudwatch logs.
o http://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/QuickStartEC2Instance.html

2.) Performed exercise to Analyzing VPC Flow Logs with Amazon Lambda, Amazon Kinesis Firehose, Athena and QuickSight

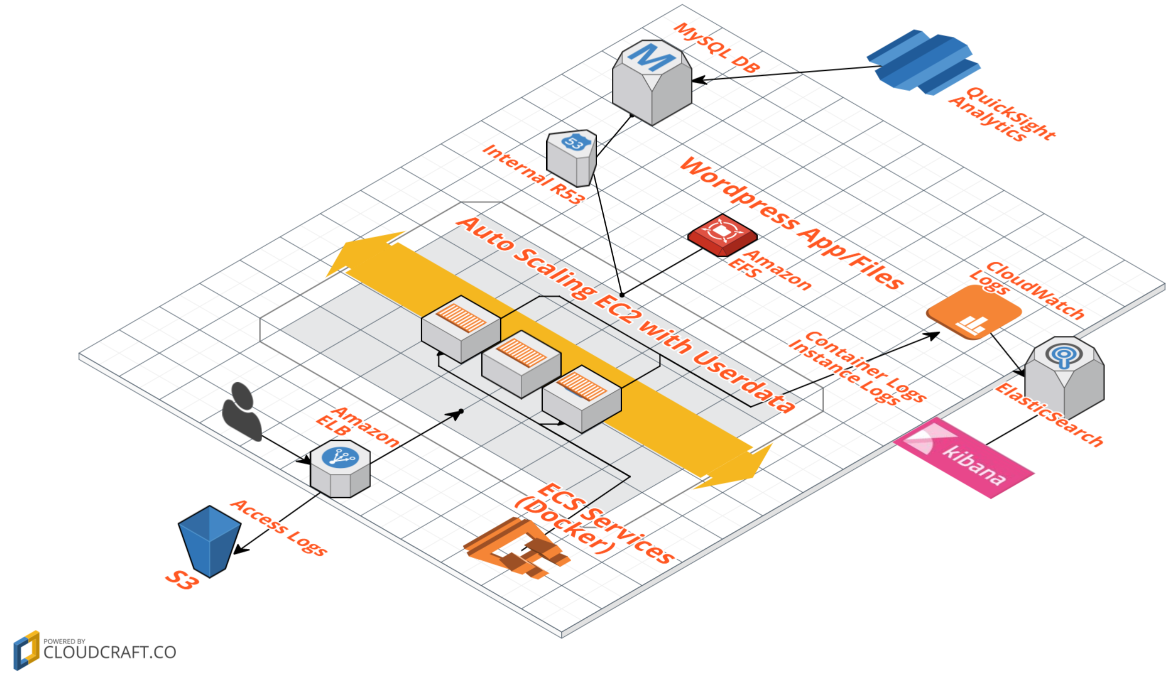


https://aws.amazon.com/blogs/big-data/analyzing-vpc-flow-logs-with-amazon-kinesis-firehose-amazon-athena-and-amazon-quicksight/

- VPC Subnet > Amazon CloudWatch Logs->(Lambda Subscription)>(VPC Flow Logs)>S3
- Athena > Create DB Table and dump logs data into table.
- Athena -> Select Query to Analyze Logs
  QuickSight -> Connects to Athena DB Tables to get Analyzed data.

3.) Able to achieve below objective using key AWS component,

• Created basic setup of Elastic Container services to host wordpress site using Elastic File System
• Manage container using ECS Services
• Utilize functional Elastic Load Balancer with configuration of access logs
• Created Bootstrap script to automate cloud watch logs for Container and hosting Instance at boot time.
• Automated mounting of Elastic File System at boot time.
• Utilized RDS DB (MySQL)
• Created internal Route 53 DNS record to send all traffic to DB instead of DB endpoint.

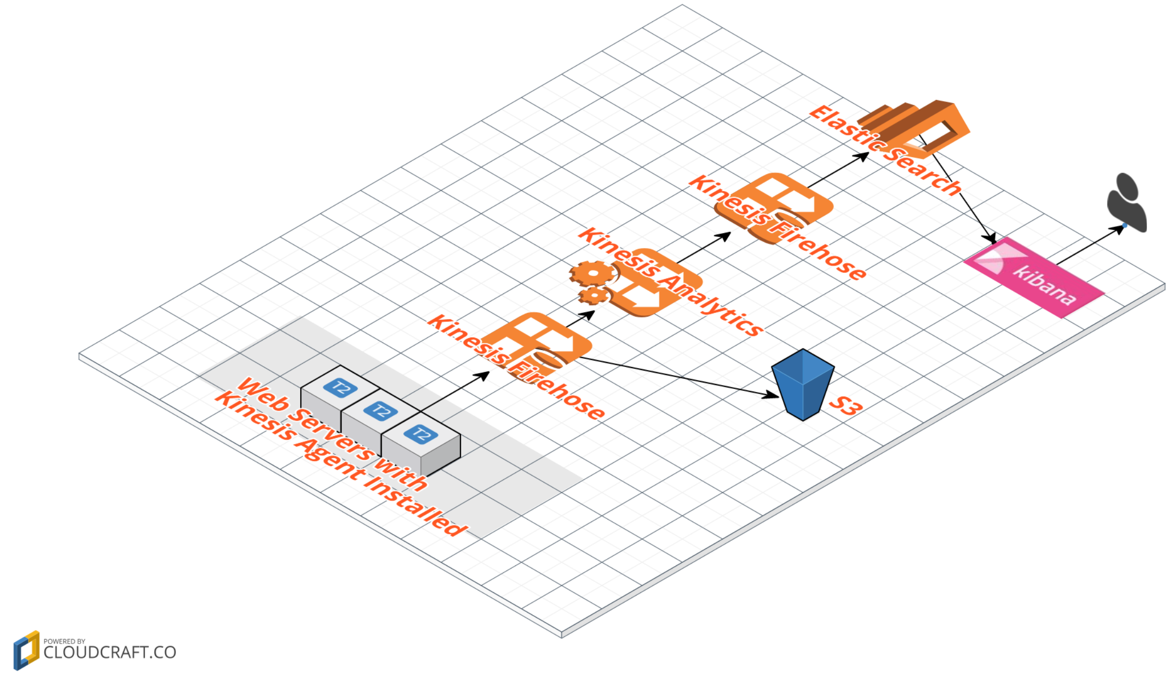


¬ Step by Step instruction on how we achieve this

¬ Logs Analysis - Kibana, QuickSight, Athena – https://s3.amazonaws.com/portfolio.bhavikgandhi.info/Logs_Analysis_Kibana_QuickSight_Athena.pdf

¬ Also performed Apache Log Analytics - Kinesis Firehose, Analytics and Elastic Search – https://s3.amazonaws.com/portfolio.bhavikgandhi.info/ApacheLogAnalytics_Kinesis_Firehose_Analytics_and_Elastic_Search.pdf

Build Log analytics solution using Kinesis Firehose, Kinesis Analytics, Elastic Search and Kibana.

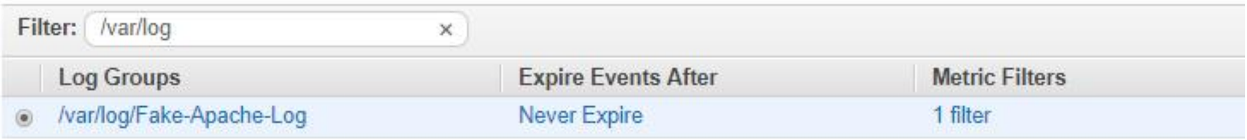


4.) Generating Compass alerts from Clouldwatch Logs - Metrics Filter

This is how it was done,

- Running Fake apache logs generator script and awslogs driver on EC2 instance, reporting all the logs to CloudWatch Log Group "/var/log/Fake-Apache-Log".
- https://github.com/kiritbasu/Fake-Apache-Log-Generator

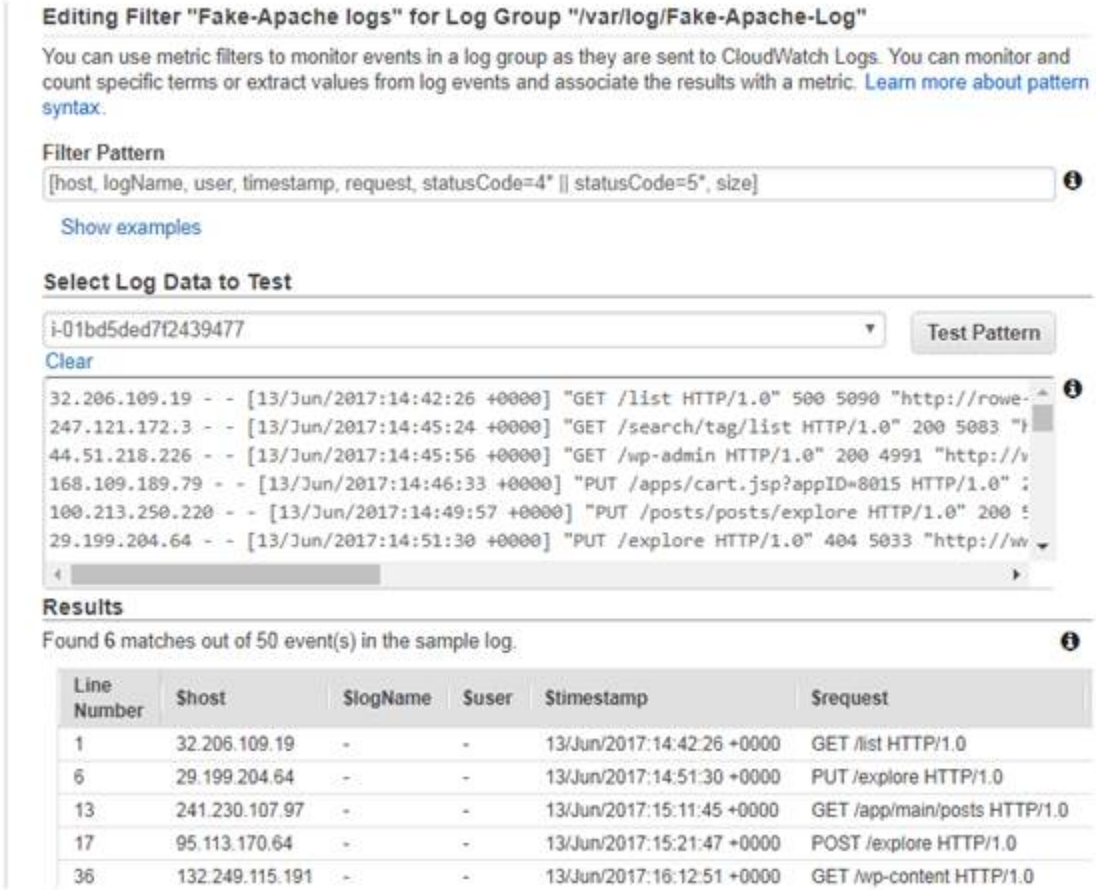
- Create the Metrics Filter from CloudWatch Log Group "/var/log/Fake-Apache-Log"



Filter: /var/log

	Log Groups	Expire Events After	Metric Filters
●	/var/log/Fake-Apache-Log	Never Expire	1 filter

- Create the Filter Patter for Status Code = 4* (Client Error) & 5* (Server Error).



Editing Filter "Fake-Apache logs" for Log Group "/var/log/Fake-Apache-Log"

You can use metric filters to monitor events in a log group as they are sent to CloudWatch Logs. You can monitor and count specific terms or extract values from log events and associate the results with a metric. Learn more about pattern syntax.

Filter Pattern

[host, logName, user, timestamp, request, statusCode=4* || statusCode=5*, size]

Select Log Data to Test

i-01bd5ded7f2439477

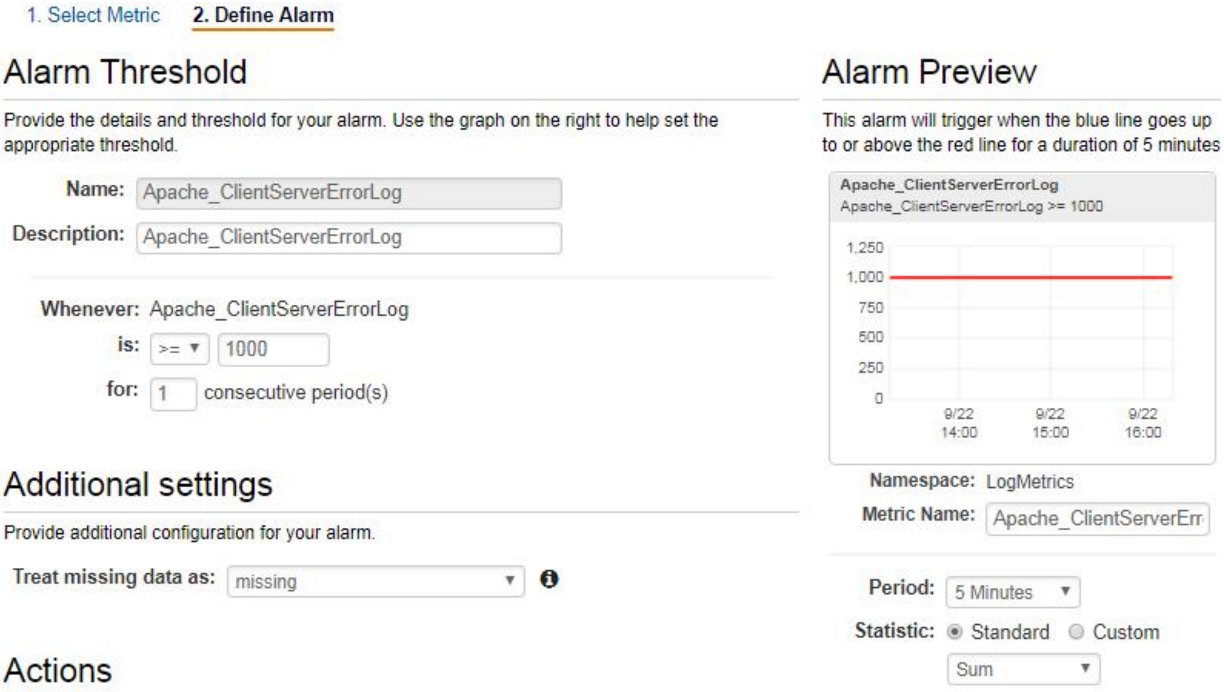
```
32.206.109.19 - - [13/Jun/2017:14:42:26 +0000] "GET /list HTTP/1.0" 500 5090 "http://rowe-
247.121.172.3 - - [13/Jun/2017:14:45:24 +0000] "GET /search/tag/list HTTP/1.0" 200 5083 "h
44.51.218.226 - - [13/Jun/2017:14:45:56 +0000] "GET /wp-admin HTTP/1.0" 200 4991 "http://w
168.109.189.79 - - [13/Jun/2017:14:46:33 +0000] "PUT /apps/cart.jsp?appID=8015 HTTP/1.0" 2
100.213.250.220 - - [13/Jun/2017:14:49:57 +0000] "PUT /posts/posts/explore HTTP/1.0" 200 5
29.199.204.64 - - [13/Jun/2017:14:51:30 +0000] "PUT /explore HTTP/1.0" 404 5033 "http://ww
```

Results

Found 6 matches out of 50 event(s) in the sample log.

Line Number	$host	$logName	$user	$timestamp	$request
1	32.206.109.19	-	-	13/Jun/2017:14:42:26 +0000	GET /list HTTP/1.0
6	29.199.204.64	-	-	13/Jun/2017:14:51:30 +0000	PUT /explore HTTP/1.0
13	241.230.107.97	-	-	13/Jun/2017:15:11:45 +0000	GET /app/main/posts HTTP/1.0
17	95.113.170.64	-	-	13/Jun/2017:15:21:47 +0000	POST /explore HTTP/1.0
36	132.249.115.191	-	-	13/Jun/2017:16:12:51 +0000	GET /wp-content HTTP/1.0

- Create an alarm with certain Threshold and Actions.



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: Apache_ClientServerErrorLog

Description: Apache_ClientServerErrorLog

Whenever: Apache_ClientServerErrorLog

is: >= 1000

for: 1 consecutive period(s)

Additional settings

Provide additional configuration for your alarm.

Treat missing data as: missing

Actions

Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

Apache_ClientServerErrorLog
Apache_ClientServerErrorLog >= 1000
1,250
1,000
750
500
250
0
9/22 14:00
9/22 15:00
9/22 16:00

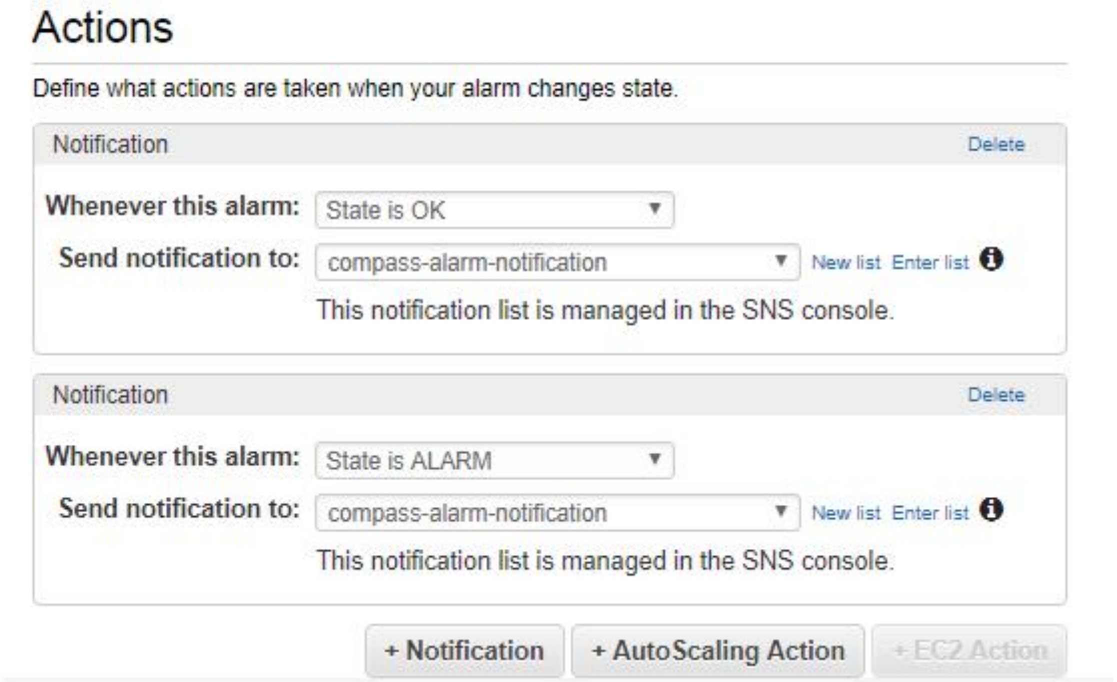
Namespace: LogMetrics

Metric Name: Apache_ClientServerErr

Period: 5 Minutes

Statistic: ● Standard ○ Custom

Sum



Actions

Define what actions are taken when your alarm changes state.

Notification   Delete

Whenever this alarm: State is OK

Send notification to: compass-alarm-notification   New list Enter list

This notification list is managed in the SNS console.

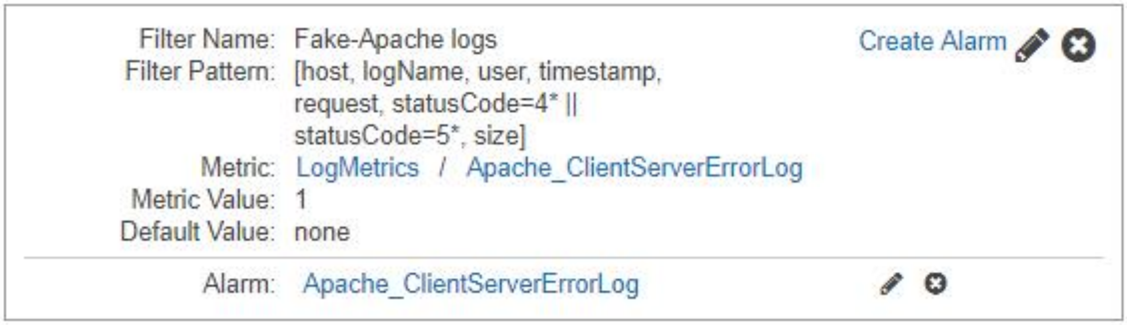
Notification   Delete

Whenever this alarm: State is ALARM

Send notification to: compass-alarm-notification   New list Enter list

This notification list is managed in the SNS console.

- Filter Metrics for CloudWatch Log Group



Filter Name: Fake-Apache logs

Filter Pattern: [host, logName, user, timestamp, request, statusCode=4* || statusCode=5*, size]

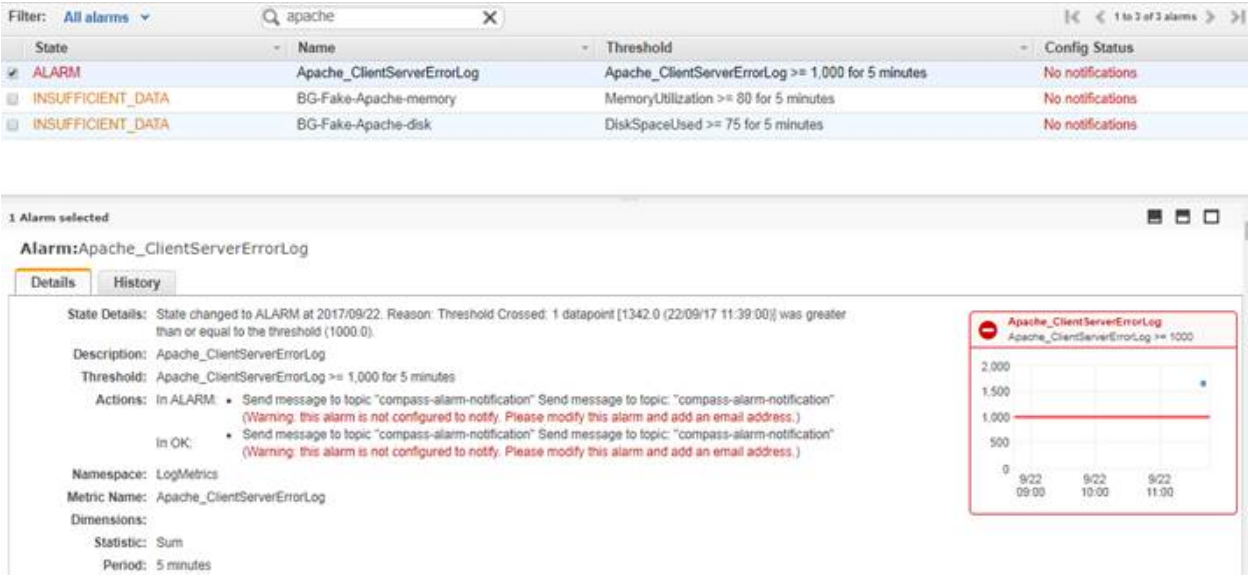
Metric: LogMetrics / Apache_ClientServerErrorLog

Metric Value: 1

Default Value: none

Alarm: Apache_ClientServerErrorLog

- Run Fake apache logs generator script to generate thousands of Apache logs


Filter: All alarms   apache   1 to 3 of 3 alarms

	State	Name	Threshold	Config Status
☑	ALARM	Apache_ClientServerErrorLog	Apache_ClientServerErrorLog >= 1,000 for 5 minutes	No notifications
□	INSUFFICIENT_DATA	BG-Fake-Apache-memory	MemoryUtilization >= 80 for 5 minutes	No notifications
□	INSUFFICIENT_DATA	BG-Fake-Apache-disk	DiskSpaceUsed >= 75 for 5 minutes	No notifications

1 Alarm selected

Alarm: Apache_ClientServerErrorLog

Details | History

State Details: State changed to ALARM at 2017/09/22. Reason: Threshold Crossed: 1 datapoint [1342.0 (22/09/17 11:39:00)] was greater than or equal to the threshold (1000.0).

Description: Apache_ClientServerErrorLog

Threshold: Apache_ClientServerErrorLog >= 1,000 for 5 minutes

Actions: In ALARM:
- Send message to topic "compass-alarm-notification" Send message to topic: "compass-alarm-notification" (Warning: this alarm is not configured to notify. Please modify this alarm and add an email address.)

In OK:
- Send message to topic "compass-alarm-notification" Send message to topic: "compass-alarm-notification" (Warning: this alarm is not configured to notify. Please modify this alarm and add an email address.)

Namespace: LogMetrics

Metric Name: Apache_ClientServerErrorLog

Dimensions:

Statistic: Sum

Period: 5 minutes

Apache_ClientServerErrorLog
Apache_ClientServerErrorLog >= 1000
2,000
1,500
1,000
500
0
9/22 09:00
9/22 10:00
9/22 11:00

- And Alert will appear on CAM as well

How to TAG CloudWatch Logs Group:

- Use AWS CLI 1.11

```
bash-3.2$ aws --version
aws-cli/1.11.129 Python/2.7.10 Darwin/16.7.0 botocore/1.5.92
```

- Connect to cloudtool using
  - cloud-tool login (Use M account)
  - Use appropriate profile and log group name to check if tags exist,

```
$ aws --profile xx-xx-sandbox logs list-tags-log-group --log-group-name /var/log/Fake-Apache-Log
{
    "tags": {}
}
```

  - Tag using below command,

```
$ aws --profile xx-xx-sandbox logs tag-log-group --log-group-name /var/log/Fake-Apache-Log --tags Name=bg-Fake-Apache-Log-Metrics
```

  - Use same --tags command for other required Standards.
  - Verify using "logs list-tags-log-group" command

```
$ aws --profile xx-xx-sandbox logs list-tags-log-group --log-group-name /var/log/Fake-Apache-Log
{
    "tags": {
        "application-asset-insight-id": "2XXXX6",
        "environment-type": "Sandbox",
        "resource-owner": "BhavikGandhi",
        "Name": "bg-Fake-Apache-Log-Metrics",
        "identifier": "6XXXXXX02"
    }
}
```

  - After this Tagging Details will appear on CAM

5.) **DataDog** - (Third party SaaS vendor for **Infrastructure monitoring** in AWS)

- Forward metrics alerts to DataDog

6.) Splunk - (Third party SaaS vendor for log management in AWS)

- Install Splunk Forwarder

- Using Splunk Forwarder to forward EC2 and ECS logging to SPLUNK
- Forward CloudWatch Log to SPLUNK