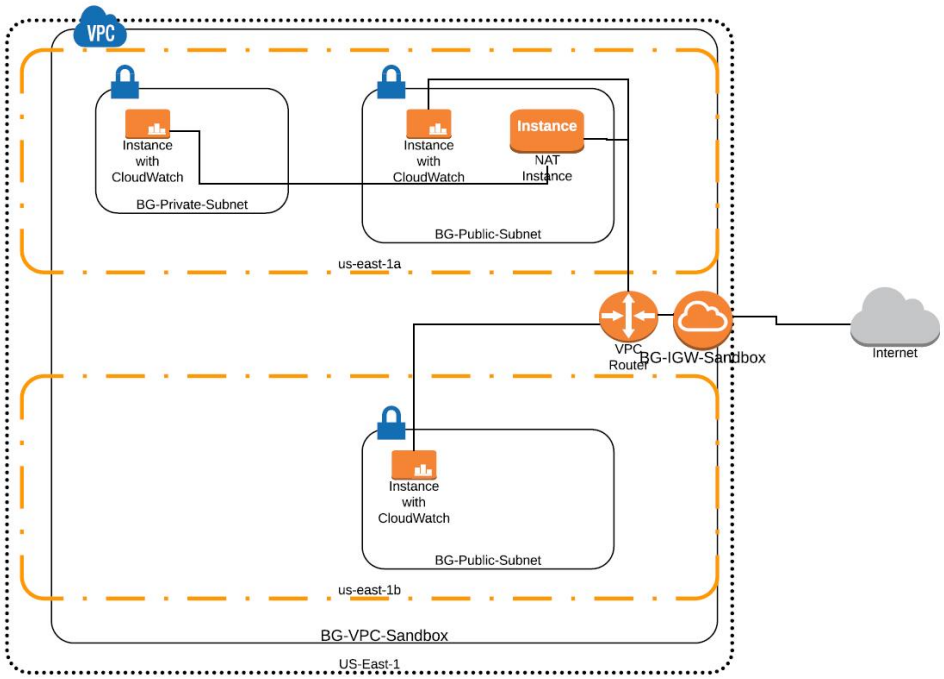
1.) Achieved below objectives using Basic Architect:



• 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

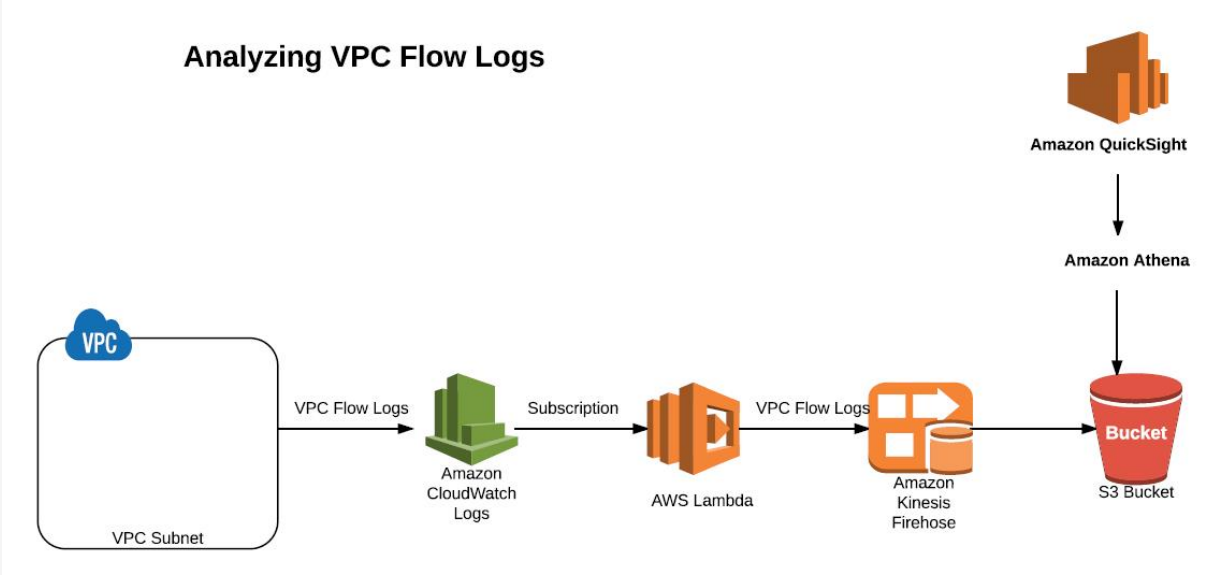
• Created "Auto Scaling Group" in Public and Private Subnet using same Launch Configuration above
o Desired Instance: 2
o Min: 1
o Max:2

• 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

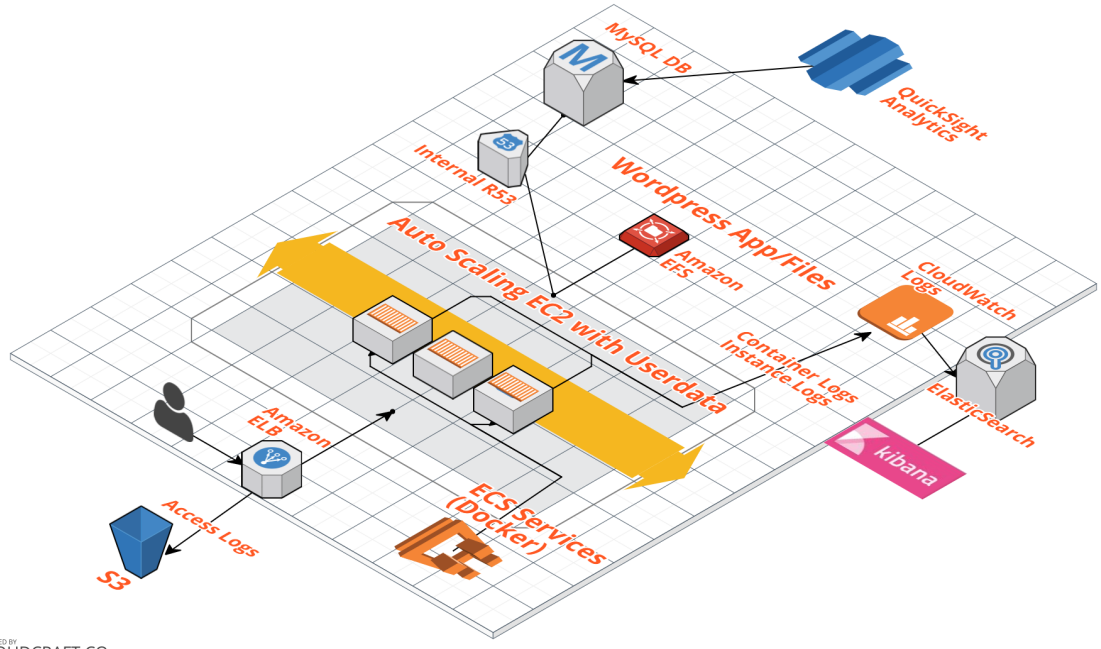


https://aws.amazon.com/blogs/big-data/analyzing-vpc-flow-logs-with-amazon-kinesis-firehose-amazon-athena-and-amazon-quicksight/

- 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,

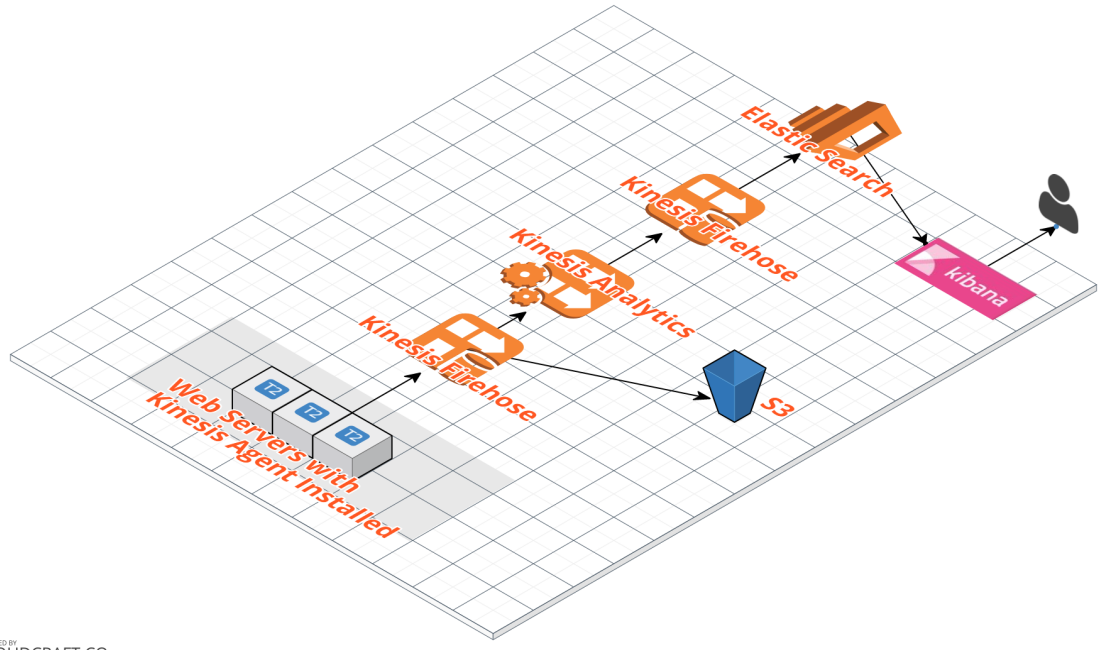
• Created basic setup of Elastic Container services to host wordpress site using Elastic File System
• Manage container using ECS Services
• Utilize functional Elastic Load Balancer with configuration of access logs
• Created Bootstrap script to automate cloud watch logs for Container and hosting Instance 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.



POWERED BY CLOUDCRAFT.CO

¬ Step by Step instruction on how we achieve this
¬ Logs Analysis - Kibana, QuickSight, Athena –
https://s3.amazonaws.com/portfolio.bhavikgandhi.info/Logs+Analysis+-+Kibana%2C+QuickSight%2C+Athena.pdf
¬ Also performed Apache Log Analytics - Kinesis Firehose, Analytics and Elastic Search –
https://s3.amazonaws.com/portfolio.bhavikgandhi.info/Apache+Log+Analytics+-+Kinesis+Firehose%2C+Analytics+and+Elastic+Search.pdf

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



POWERED BY CLOUDCRAFT.CO

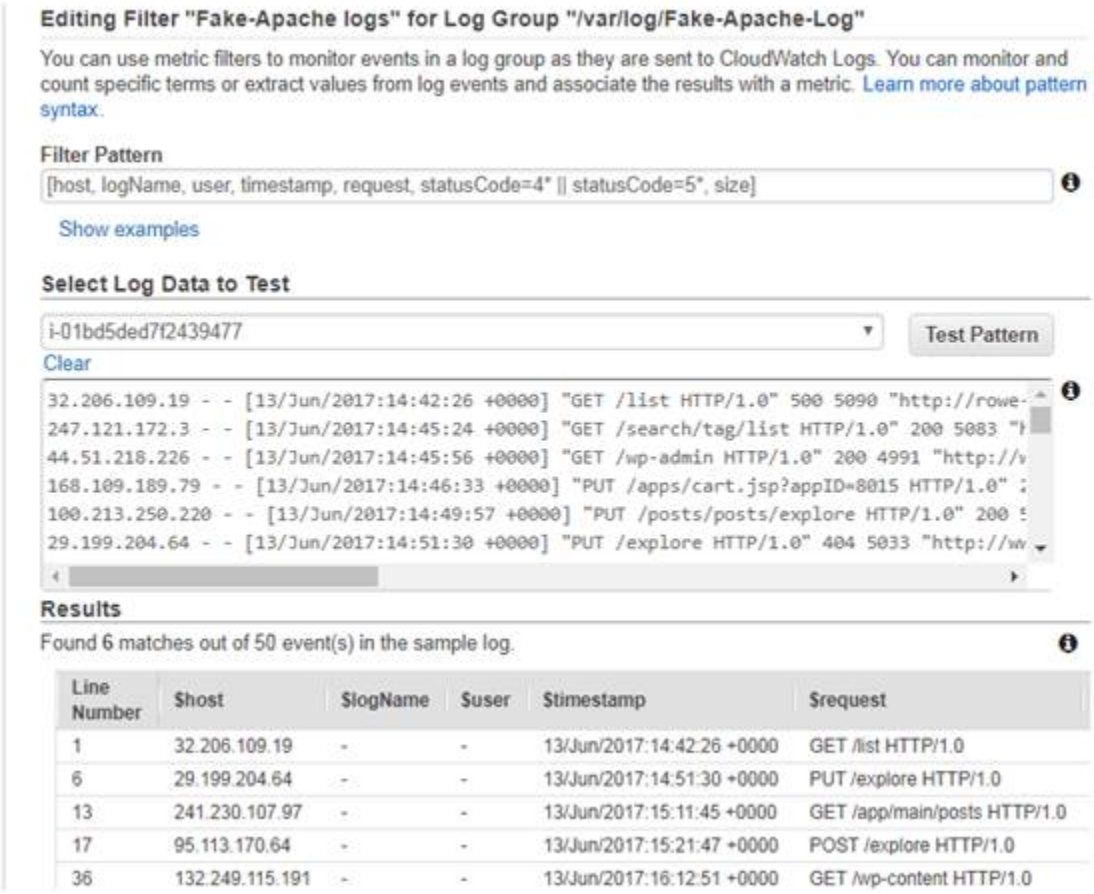
4.) Generating Compass alerts from Clouldwatch Logs - Metrics Filter

This is how it was done,

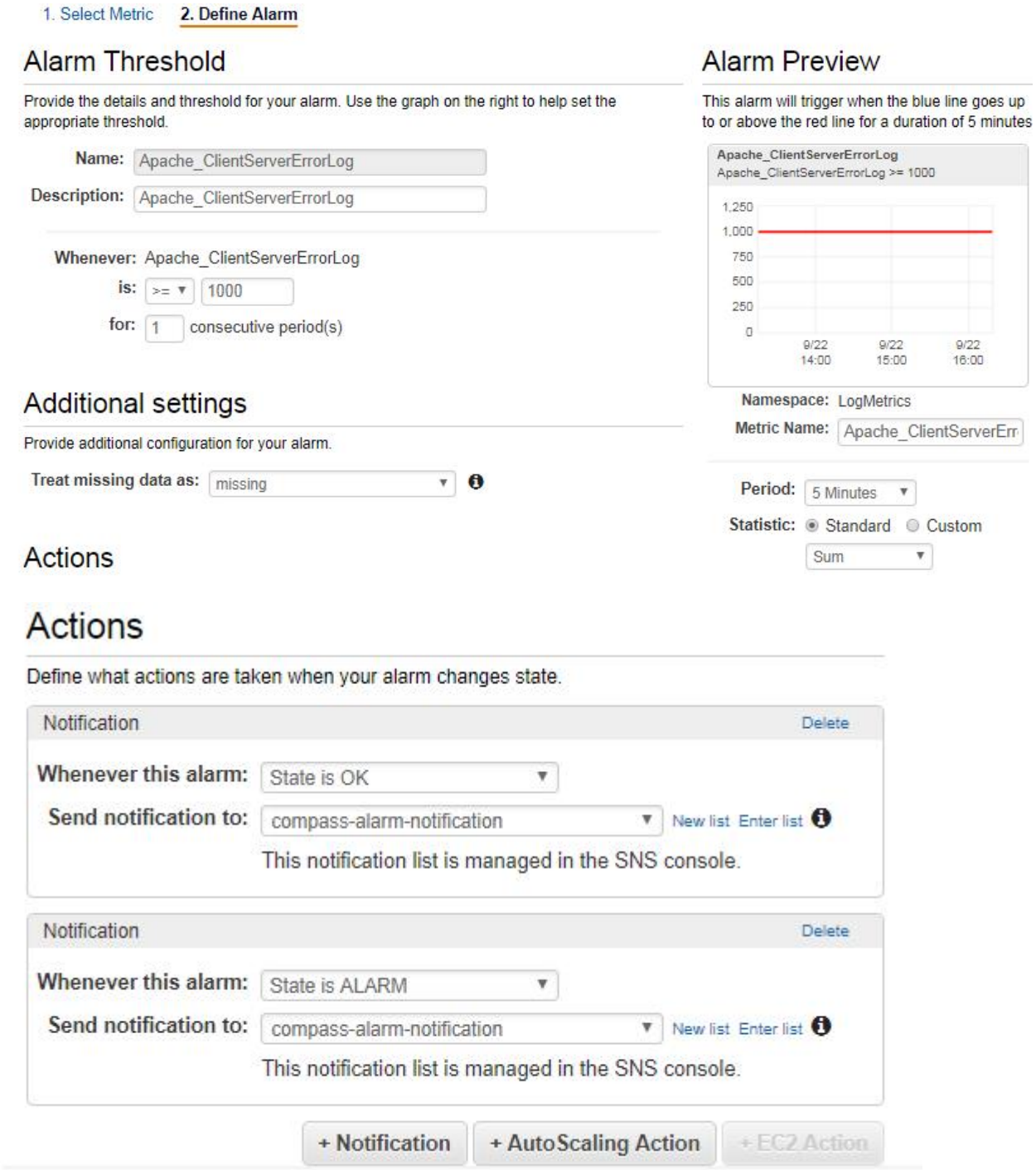
- Running Fake apache logs generator script and awslogs driver on EC2 instance, reporting all the logs to CloudWatch Log Group "/var/log/Fake-Apache-Log".
- https://github.com/kiritbasu/Fake-Apache-Log-Generator

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

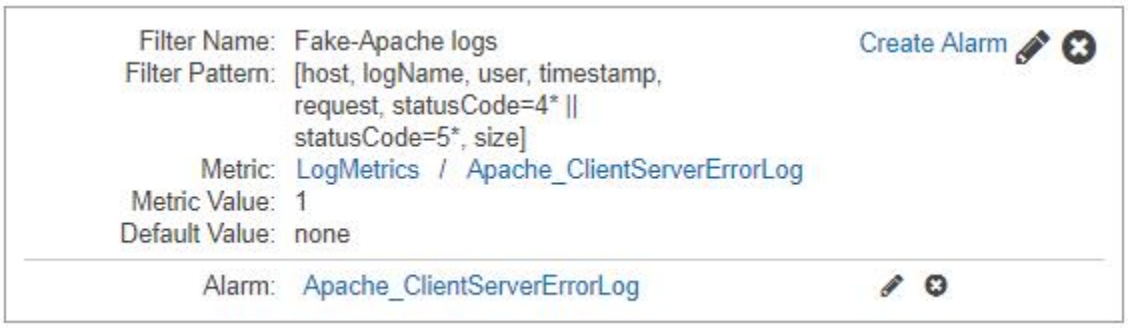
- Create the Filter Patter for Status Code = 4* (Client Error) & 5* (Server Error).



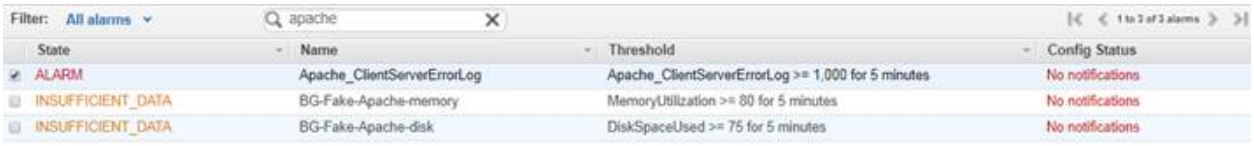
- Create an alarm with certain Threshold and Actions.

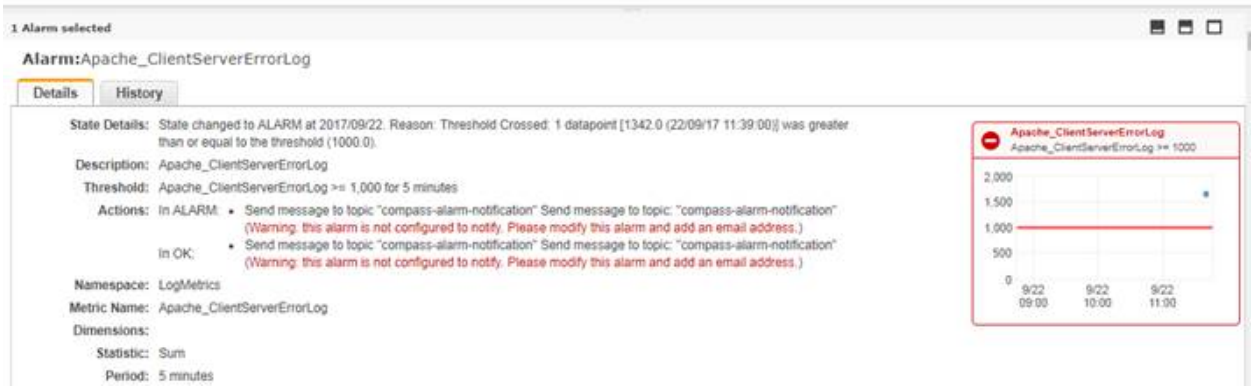


- Filter Metrics for CloudWatch Log Group



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





- 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