**CLOUD WATCH**

* To check the server status

**Need of Monitoring Tool**

* How Cloud Engineer (CE) know which server is up and which server is down for this he/her has to login and particular EC2 instance and verify the server status
* If one or two server we can monitor manually, if more than one server then manual intervention is not possible, we need tool for checking server status frequently
* Amazon CloudWatch is monitoring service for AWS cloud resources and the applications you run on AWS. It is used to collect and track metrics, collect and monitor log files, and set alarms
* CloudWatch provides metrics for every service in AWS
* Metrics: Metrics is a variable to monitor (CPU Utilization, NetworkIn,etc)

**Few monitoring tools in market**

1.Nagios

2.Splunk

3.AppDynamics

4.DynaTrace

5.Metric Fire

To use this tool, we need to buy license

* CloudWatch is used to monitor all AWS Services (EC2, ELB, RDS, ASG, R53, SG, CF)
* CloudWatch, we don’t have to buy or license
* CloudWatch can also monitor applications with EC2
* CloudWatch monitor the performance
* CloudWatch is monitor using alaram
* CloudWatch is all about Alaram, Events and logs
* Host Level metrics is including (**CPU, Network, Disk, Status checks are AWS provided default metrics)**
* CloudWatch can monitor only HOST level metrics (Default Metrics)

**Two Types of Monitoring**

* **Basic Monitoring (default):** Metrics are collected at a 5min interval, **FREE**

Ten pre-selected metrics at five-minute frequency, free of charge

* **Detailed Monitoring:**  Metrics are collected at a 1min interval, **Billable**

Seven pre-selected metrics at one-minute frequency, for an additional charge

Includes CPU, Network, Disk and Status Check Metrics

* **Amazon EBS volumes:** Ten pre-selected metrics at five-minute frequency, free of charge
* **Elastic Load Balancers:** Ten pre-selected metrics at one-minute frequency, free of charge
* **Auto Scaling groups:** Seven pre-selected metrics at one-minute frequency, optional and charged at standard pricing
* **Amazon RDS DB instances:** Thirteen pre-selected metrics at one-minute frequency,

**Custom Metrics (Yours to push)**

Basic Resolution: 1 minute resolution

High Resolution: all the way to 1 second resolution

Include RAM, application-level metrics

Make sure the IAM permissions on the EC2 instance role are correct

RAM is not included in the AWS EC2 metrics

**With Amazon CloudWatch, the user can get:**

Up-to-minute statistics (Basic and Detailed Monitoring)

View graphs

Set alarms for your metrics data

Use Auto Scaling to add/remove resources based on CloudWatch Metrics

**AWS Monitoring Tools**

**CLOUDTRAIL**

Monitor entire AWS environment

Records, monitor, track, audit, logs

**CONFIG**

Monitors the changes in AWS resources

**METRICS**

Metrics = CPU + Network + Memory

**ALARAM**

Alaram can do some actions based on metrics of CPU usage,

If CPU usage less than 2, it will terminate EC2 instance or stop, reboot

An Alarm basically watches over a particular metric for stipulated period of time and performs some actions based on its trigger. These actions can be anything from sending a notification to the user using SNS.

**Namespaces:** A container for CloudWatch metrics. It is a grouping to know what is metrics belongs to.

Example: AWS/EC2, AWS/EBS, AWS/ASG

**Dimensions:** Dimension is a name/value pair that you uniquely identify a metric.

Example: AutoscalingGroup, Imaged, Instanced, Instance Type, Volume ID

**Timestamps:** To know what timestamp it had captured.

**Units:** Unit represents the statistic’s unit of measure. For example: EC2 Network In metric in bytes.

**Metrics Retention Period**

CloudWatch Metrics now supports the following three retention schedules

1-minute datapoints are available for 15 days

5-minute datapoints are available for 63 days

1-hours datapoints are available for 455 days

**Free Tier Eligible of CloudWatch Alaram**

If 100 instance each instance one alaram means 100 alaram we need to create

Per account 10 alaram free including all region and interval should be 5min

**Alaram has 3 states**

1.OK

2.in Alaram

mater of concern

3.Insufficent

If somebody Terminated instance accidentally then “CloudWatch” will continuously wait for the information from the server that server some body deleted which is not there,

then CloudWatch not having any information either go to OK status or go to ALARAM status because it is having insufficient info

Therefore, alaram status will be Insufficient

**EVENTS**

If some some one stop EC2 instance, we need to get notification

Stop event should be triggered

Pending/running, stopping/stop shutting down/Terminated

EC2 instance status change notification

Amazon CloudWatch Events delivers a near real-time stream of

system events that describe changes in Amazon Web Services

(AWS) resources.

Determine events of interest

Create rules to match events

Route to targets

**CLOUDWATCH LOGS**

You can Analyze your logs with CloudWatch Logs Insights. No Infrastructure or setup

needed.

CloudWatch Logs helps you to aggregate, monitor, and store logs. For example, you can:

▪ Monitor HTTP response codes in Apache logs

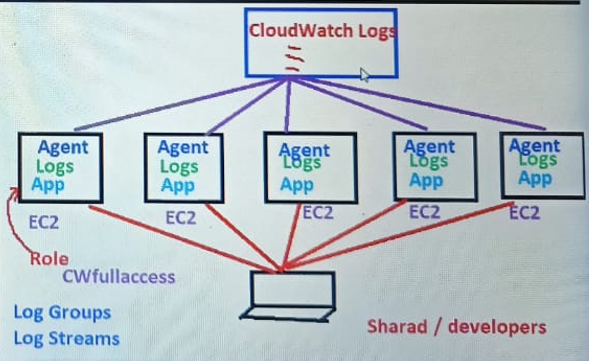
▪ Receive alarms for errors in kernel logs

▪ Count exceptions in application logs

Every EC2 instance will have own application, also have its own application logs

Agent software has to install all EC2 instances

Agent will collect the logs push that logs to central repo logs

****

**Configure CloudWatch logs step by steps**

1. Create IAM role (i.e CloudWatchFullAccess)

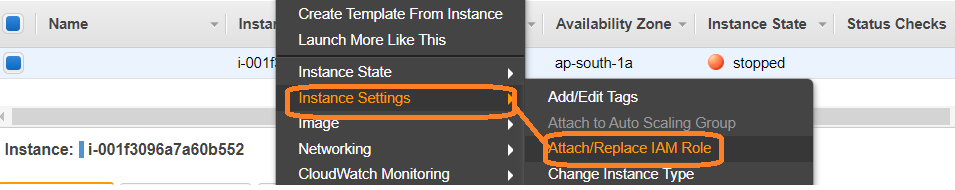
****

1. Launch EC2 instance with root user and attach IAM role which launch EC2 instance

(OR)

We can attach IAM role running instance as well

**EC2 instance 🡪 Actions 🡪 Instance Settings 🡪 Attach/Replace IAM Role**



1. Install CloudWatch agent into EC2 instance

$ sudo yum update -y

$ sudo yum install -y awslogs

1. Configure CloudWatch agent configuration file
   1. Edit /etc/awslogs/awslogs.conf to set as per logs configuration

$ sudo cd /etc/awslogs

$ sudo vi awslogs.conf

[/var/log/messages]

datetime\_format = %b %d %H:%M:%S 🡪 *Date format*

file = /var/log/messages 🡪 *default file path we can change*

buffer\_duration = 5000

log\_stream\_name = {**i-001f3096a7a60b552**} 🡪 *we have to update EC2 instance id*

initial\_position = start\_of\_file

log\_group\_name = /var/log/messages

**[Save and Exit]**

* 1. Edit /etc/awslogs/awscli.conf to set as per aws region ( ap-south-1)

$ sudo cd /etc/awslogs

$ sudo vi awscli.conf

[plugins]

cwlogs = cwlogs

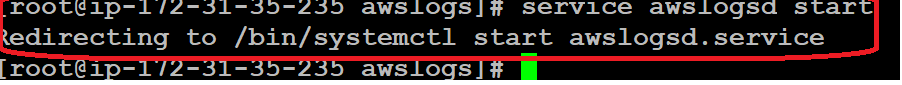
[default]

region = ***ap-south-1***

**[Save and Exit]**

1. Start the awslogs agent

$ sudo service awslogsd start



1. Install any software in EC2 instance

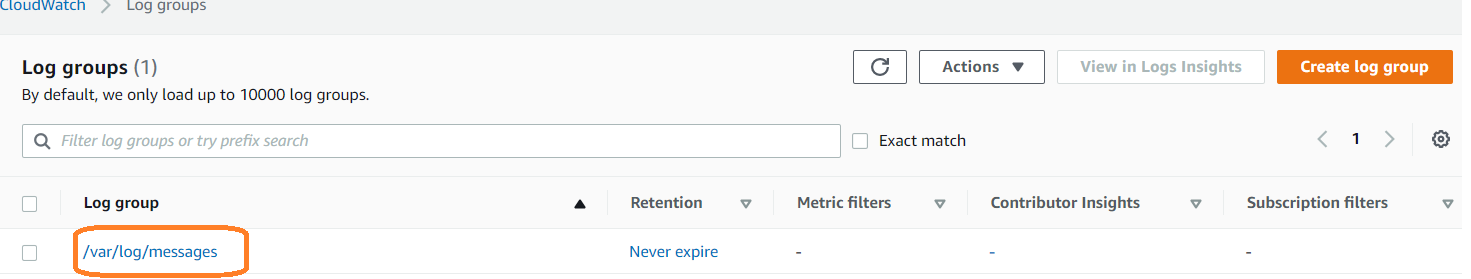
For example

$sudo yum install git

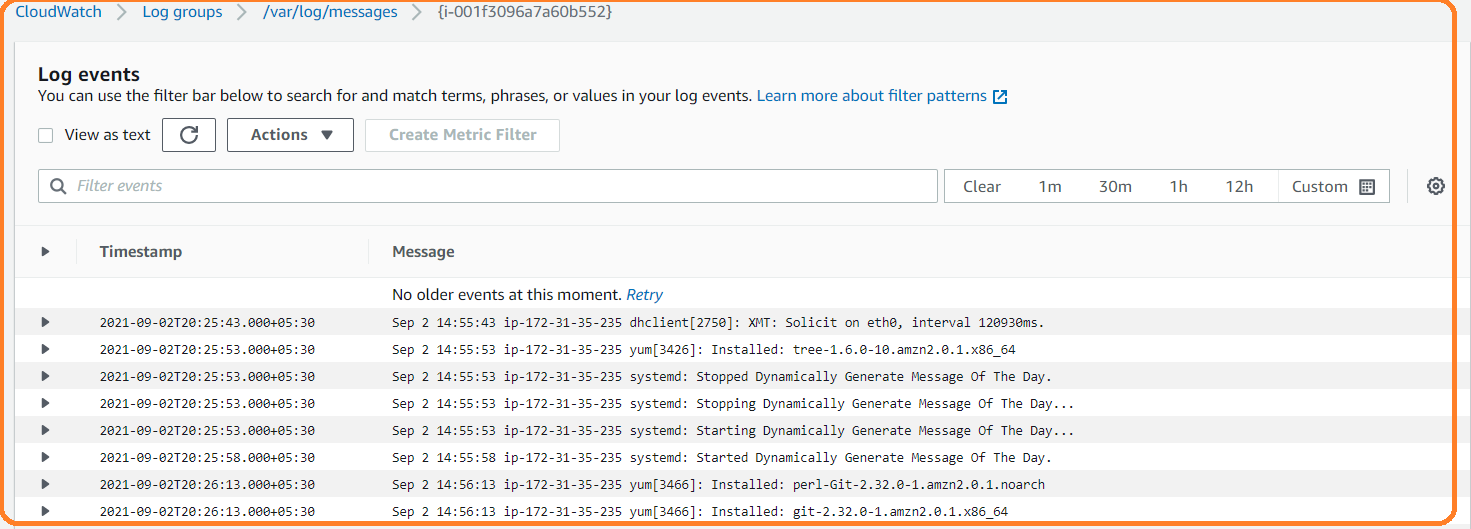
$sudo yum install httpd

And verify logs in CloudWatch Dash board under logs section

1. **CloudWatch 🡪 Log group**



Verify that you see logs into CloudWatch logs console



We can also create dummy log file at the location we configured in the file and keep appending some data(need to check)