100 Days of DevOps — Day 1(Introduction to CloudWatch Metrics)

*Welcome to Day 1 of 100 Days of DevOps, I would like to start this journey with one of the most critical concepts in DevOps Monitoring and Alerting.*

**Problem Statement**

* *Create a CloudWatch alarm that sends an email using SNS notification when CPU Utilization is more than 70%.*
* *Creating a Status Check Alarm to check System and Instance failure and send an email using SNS notification*

**Solution**

*This can be achieved via in one of the three ways*

* *AWS Console*
* *AWS CLI*
* *Terraform*

*NOTE: This is not the complete list and there are more ways to achieve the same*

What is CloudWatch?

*AWS CloudWatch is a monitoring service to monitor AWS resources, as well as the applications that run on AWS.*

*As per official documentation*

Amazon CloudWatch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real time. You can use CloudWatch to collect and track metrics, which are variables you can measure for your resources and applications.

*Reference:*

[What is Amazon CloudWatch? - Amazon CloudWatch  
Monitor your AWS resources and applications using Amazon CloudWatch to collect and track metrics on performance.docs.aws.amazon.com](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html)

*EC2/Host Level Metrics that CloudWatch monitors by default consist of*

* *CPU*
* *Network*
* *Disk*



* *Status Check*

https://cdn-images-1.medium.com/max/800/1*1Ldt-iquTehzqQ3Rkji2UQ.png

*There are two types of status check*

* ***System status check:****Monitor the AWS System on which your instance runs. It either requires AWS involvement to repair or you can fix it by yourself by just stop/start the instance(in case of EBS volumes).Examples of problems that can cause system status checks to fail*

*\* Loss of network connectivity  
\* Loss of system power  
\* Software issues on the physical host  
\* Hardware issues on the physical host that impact network reachability*

* ***Instance status check:****Monitor the software and network configuration of an individual instance. It checks/detects problems that require your involvement to repair.*

*\* Incorrect networking or startup configuration  
\* Exhausted memory  
\* Corrupted filesystem  
\* Incompatible ketnel*

***NOTE***

* *Memory/RAM utilization is custom metrics.*
* *By default, EC2 monitoring is 5 minutes intervals but we can always enable detailed monitoring(1 minutes interval, but that will cost you some extra $$$)*

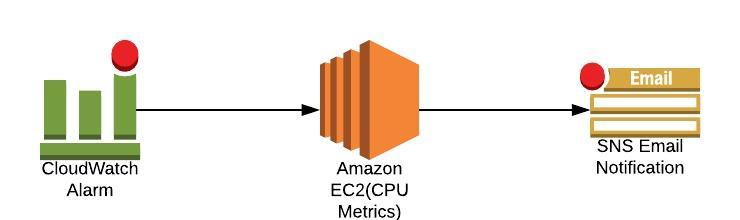
*Reference*

[Amazon CloudWatch Pricing - Amazon Web Services (AWS)  
EC2 Detailed Monitoring is priced at $2.10 per instance per month and goes down to $0.14 per instance at the lowest…aws.amazon.com](https://aws.amazon.com/cloudwatch/pricing/)

***P.S:****CloudWatch can be used on premise too. We just need to install the SSM(System Manager) and CloudWatch agent.*

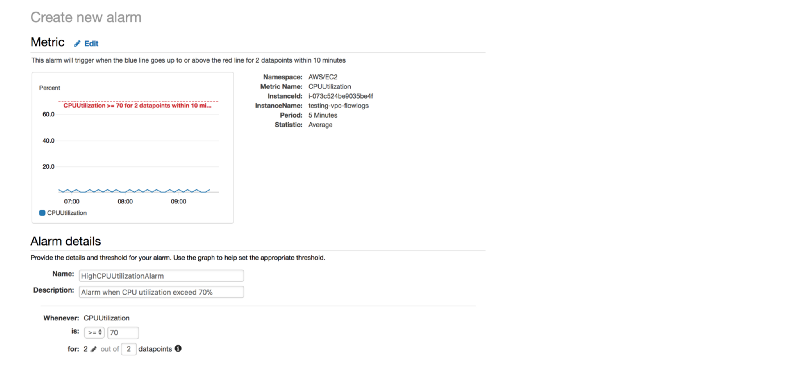
*Enough of the theoretical concept, let setup first CloudWatch alarm*

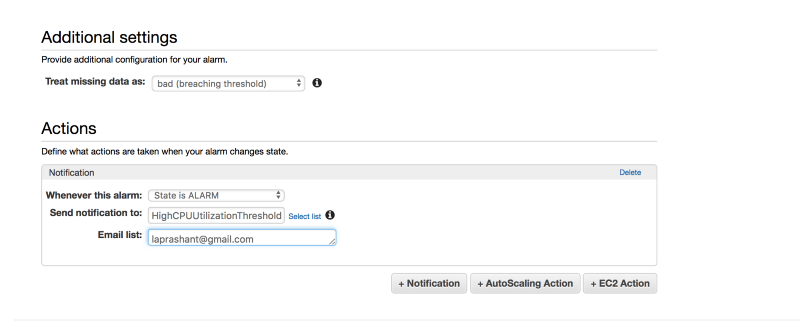
* ***Scenario1:****We want to create a CloudWatch alarm that sends an email using SNS notification when CPU Utilization is more than 70%*



***Solution1:****Setup a CPU Usage Alarm using the AWS Management Console*

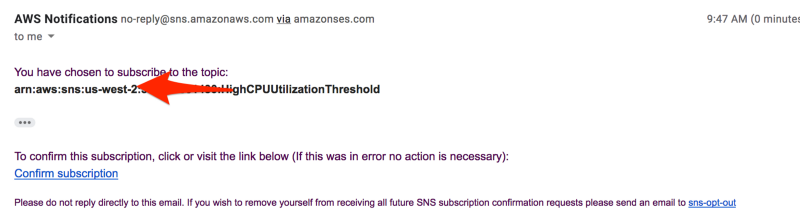
* *Open the CloudWatch console at*[*https://console.aws.amazon.com/cloudwatch/*](https://console.aws.amazon.com/cloudwatch/)*.*
* *In the navigation pane, choose****Alarms****,****Create Alarm****.*
* *Go to Metric → Select metric → EC2 → Per-Instance-Metrics → CPU Utilization → Select metric*





*Define the Alarm as follows*

*\* Type the unique name for the alarm(eg: HighCPUUtilizationAlarm)  
\* Description of the alarm  
\* Under whenever,choose >= and type 70, for type 2. This specify that the alarm is triggered if the CPU usage is above 70% for two consecutive sampling period  
\* Under Additional settings, for treat missing data as, choose bad(breaching threshold), as missing data points may indicate that the instance is down  
\* Under Actions, for whenever this alarm, choose state is alarm. For Send notification to, select an exisiting SNS topic or create a new one   
\* To create a new SNS topic, choose new list, for send notification to, type a name of SNS topic(for eg: HighCPUUtilizationThreshold) and for Email list type a comma-seperated list of email addresses to be notified when the alarm changes to the ALARM state.  
\* Each email address is sent to a topic subscription confirmation email. You must confirm the subscription before notifications can be sent.  
\* Click on Create Alarm*



***Solution2: Setup CPU Usage Alarm using the AWS CLI***

* *Create an alarm using the put-metric-alarm command*

*aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm when CPU exceeds 70 percent" --metric-name CPUUtilization --namespace AWS/EC2 --statistic Average --period 300 --threshold 70 --comparison-operator GreaterThanThreshold --dimensions "Name=InstanceId,Value=i-12345678" --evaluation-periods 2 --alarm-actions arn:aws:sns:us-east-1:111122223333:MyTopic --unit Percent*

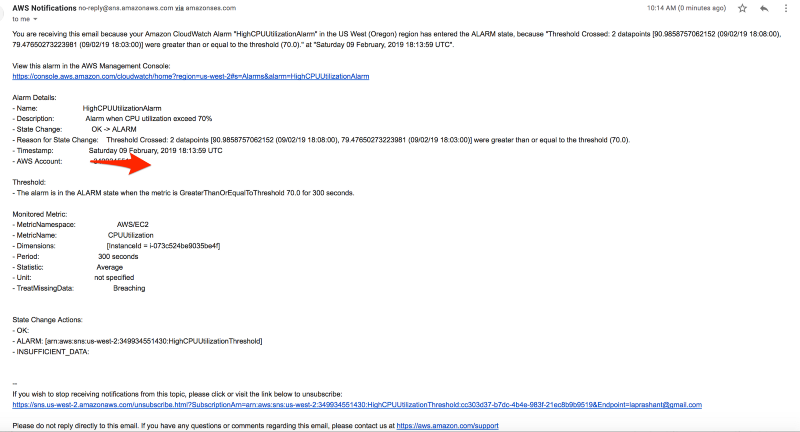
* *Using the command line, we can test the Alarm by forcing an alarm state change using a set-alarm-state command*
* *Change the alarm-state from INSUFFICIENT\_DATA to OK*

*# aws cloudwatch set-alarm-state --alarm-name "cpu-monitoring" --state-reason "initializing" --state-value OK*

* *Change the alarm-state from OK to ALARM*

*# aws cloudwatch set-alarm-state --alarm-name "cpu-monitoring" --state-reason "initializing" --state-value ALARM*

* *Check if you have received an email notification about the alarm*



*GITHUB link:*

[100daysofdevops/100daysofdevops  
Contribute to 100daysofdevops/100daysofdevops development by creating an account on GitHub.github.com](https://github.com/100daysofdevops/100daysofdevops/blob/master/cloudwatch_cpu_alarm_via_aws_cli)

*Solution3:****Setup CPU Usage Alarm using the Terraform***

*#cloudwatch.tf*

*resource "aws\_cloudwatch\_metric\_alarm" "cpu-utilization" {  
 alarm\_name = "high-cpu-utilization-alarm"  
 comparison\_operator = "GreaterThanOrEqualToThreshold"  
 evaluation\_periods = "2"  
 metric\_name = "CPUUtilization"  
 namespace = "AWS/EC2"  
 period = "120"  
 statistic = "Average"  
 threshold = "80"  
 alarm\_description = "This metric monitors ec2 cpu utilization"  
 alarm\_actions = [ "${aws\_sns\_topic.alarm.arn}" ]*

*dimensions {  
 InstanceId = "${aws\_instance.my\_instance.id}"  
 }  
}*

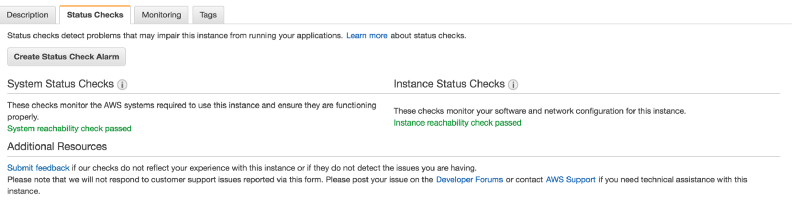
*GitHub link:*

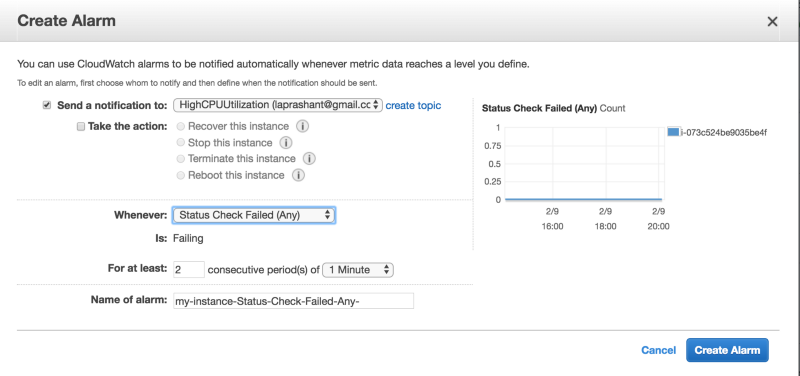
[100daysofdevops/100daysofdevops  
Contribute to 100daysofdevops/100daysofdevops development by creating an account on GitHub.github.com](https://github.com/100daysofdevops/100daysofdevops/blob/master/cloudwatch_cpu_alarm_via_terraform)

**Scenario2: Create a status check alarm to notify when an instance has failed a status check**

**Solution1: Creating a Status Check Alarm Using the AWS Console**

1. *Open the Amazon EC2 console at*[*https://console.aws.amazon.com/ec2/*](https://console.aws.amazon.com/ec2/)*.*
2. *In the navigation pane, choose****Instances****.*
3. *Select the instance, choose the****Status Checks****tab, and choose to****Create Status Check Alarm****.*





*\* You can create new SNS notification or use the exisiting one(I am using the existing one create in earlier example of high CPU utilization)  
\* In* ***Whenever****, select the status check that you want to be notified about(options Status Check Failed(Any), Status Check Failed(Instance) and Status Check Failed(System)  
\* In* ***For at least****, set the number of periods you want to evaluate and in* ***consecutive periods****, select the evaluation period duration before triggering the alarm and sending an email.  
\* In* ***Name of alarm****, replace the default name with another name for the alarm.  
\* Choose* ***Create Alarm****.*

**Solution2: To create a status check alarm via AWS CLI**

* *Use the put-metric-alarm command to create the alarm*

*aws cloudwatch put-metric-alarm --alarm-name StatusCheckFailed-Alarm-for-test-instance --metric-name StatusCheckFailed --namespace AWS/EC2 --statistic Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 --unit Count --period 300 --evaluation-periods 2 --threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --alarm-actions arn:aws:sns:us-west-2:111122223333:my-sns-topic*

GitHub link:

[100daysofdevops/100daysofdevops  
Contribute to 100daysofdevops/100daysofdevops development by creating an account on GitHub.github.com](https://github.com/100daysofdevops/100daysofdevops/blob/master/cloudwatch_status_check_via_aws_cli)

**Solution3:To create a status check alarm via Terraform**

*resource "aws\_cloudwatch\_metric\_alarm" "instance-health-check" {  
 alarm\_name = "instance-health-check"  
 comparison\_operator = "GreaterThanOrEqualToThreshold"  
 evaluation\_periods = "1"  
 metric\_name = "StatusCheckFailed"  
 namespace = "AWS/EC2"  
 period = "120"  
 statistic = "Average"  
 threshold = "1"  
 alarm\_description = "This metric monitors ec2 health status"  
 alarm\_actions = [ "${aws\_sns\_topic.alarm.arn}" ]*

*dimensions {  
 InstanceId = "${aws\_instance.my\_instance.id}"  
 }  
}*

*GitHub link:*

[100daysofdevops/100daysofdevops  
Contribute to 100daysofdevops/100daysofdevops development by creating an account on GitHub.github.com](https://github.com/100daysofdevops/100daysofdevops/blob/master/cloudwatch_status_check_via_terraform)

*Looking forward from you guys to join this journey and spend a minimum an hour every day for the next 100 days on DevOps work and post your progress using any of the below medium.*