EC2 Monitoring

Friday, June 22, 2018 2:19 PM

# How to set up

* Add a tag to all EC2 Instances to identify which ones will be monitored
* Add all of the files to an S3 bucket
* Make sure you have permissions to create IAM Roles and policies
* Create a Cloudformation stack
  1. Use the URL for the monitoringSystem.yml template in the S3 bucket

○ Fill out parameters

■ The EC2TagKey and the EC2TagValue should be the key and value of the tag that was added to the EC2 instances earlier

■ Make sure the BucketName is the name of the bucket where the template and the code is in

■ If you want to add more SNS endpoints, sign up manually to the monitorNotification SNS topic

■ RestartInstances specifies whether the instance is turned on when instance is accidentally turned off ■ RecoverStatusCheck specifies whether to recover the instance after a status check failure

□ Turns the instance on and off again

□ Will get a notification that the instance was accidentally turned off, disregard □ For this to work, RestartInstances must be turned on

If not, the instance will just be stopped

○ Can ignore options page

○ In the review screen make sure to check the Capabilities box

■ "I acknowledge that AWS CloudFormation might create IAM resources with custom names."

* Done and done
  1. It takes a little bit for the alarms to be attached

○ To change resource names, you have to edit the maps in the YAML template ○ Remember to confirm email

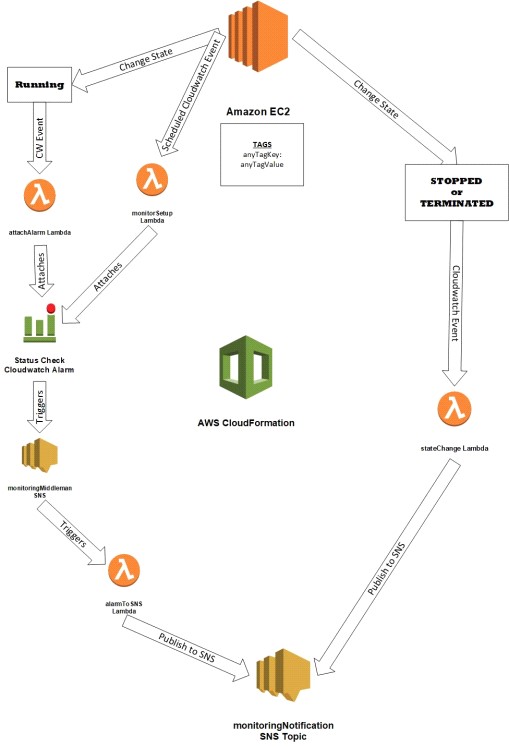
# Goal

Set up and Up-Down monitoring system that sends a notification if an instance turns unhealthy or if it is accidentally turned off. Does not add termination protection.

Two parts:

* Health check of EC2 instance
* Instance state monitoring

Here is an incredibly crude visualization of the system



# Overview

All names are as in CFN template

* All lambda functions use the boto3 python SDK
* IAM Role

## EC2 Instance

- Needs a specified tag for system to work

Needs a specified tag for system to work

## setUpRule

* Scheduled cloudwatch rule that triggers every minute (doesn't matter because rule will be disabled) - Triggers **monitorSetupLambda**
* Exists so when the CFN template is created, a lambda function can attach an alarm to existing instances
* <https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/WhatIsCloudWatchEvents.html>
* It's probably better practice to use custom resources or an SNS topic that activates on creation, but I wasn't able to automa te those processes

## monitorSetupLambda

* Should be triggered a minute after CFN creation because of the **setUpRule**
* Function attaches a status\_check alarm to all *running* instances with the identifying tag
  1. Checks to see if instance already has alarm of the same name

■ Will overwrite an alarm of the same name

○ More info about alarm can be found in **Cloudwatch Alarm** heading

* Updates the Cloudwatch dashboard when attaching an alarm if defined in CFN parameters
  1. Can be turned on/off on CFN creation

○ Adds a widget looking at the instance status check metric

* Function then disables the scheduled cloudwatch rule ○ So it doesn't keep getting invoked

## attachAlarmRule

* Cloudwatch rule that is triggered when an EC2 instance changes state to running - Invokes attachAlarmLambda
* <https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/WhatIsCloudWatchEvents.html>

## attachAlarmLambda

* When EC2 changes state to running, it triggers a Cloudwatch rule, which then triggers the attachAlarmLambda function - Attaches status\_check alarm to instance
  1. Will override an alarm with the same name

○ More info about alarm can be found in **Cloudwatch Alarm** heading

* Updates the Cloudwatch dashboard when attaching an alarm if defined in CFN parameters
  1. Can be turned on/off on CFN creation

○ Adds a widget looking at the instance status check metric

## Cloudwatch Alarm

* Alarms are named the instance id
* Status\_check alarms
* Two types of status checks
  1. StatusCheckFailed\_System

■ Cannot be simulated

○ StatusCheckFailed\_Instance

■ Can be simulated by killing eth0

○ <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-system-instance-status-check.html>

* For other services, alarms can be configured for other metrics
* Alarms can send notifications when in a defined mode {'alert', 'insufficient data', 'ok'}
* Can be configured to take EC2 action
  1. Recover

○ Stop

○ Terminate

○ Reboot

* When to recover an instance in this system, the alarm stops the instance and the stateChangeLambda restarts it
  1. The alarm doesn't use the recover option because that only fixes system status failures

○ If the restart parameter is not set to yes, the instance will just be stopped

* Lambda function will not overwrite an alarm of the same name
* Tested by entering (sudo ifconfig eth0 down) in a Linux instance or killing network connection in a windows

## monitoringMiddleman (SNS)

* Alarm triggers this when instance fails health check

Alarm triggers this when instance fails health check

* Passes the alarm message to alarmToSNSLambda
* The default message that the alarm gives doesn't have the right amount of data and the formatting cannot be customized ○ So the info is passed to a Lambda to do the rest

## alarmToSNSLambda

* Formats event data from monitoringMiddleman SNS topic
* Retrieves more data about the instance
* Sends a message to **monitoringNotification SNS**

## stateChangeRule

* Invokes stateChangeLambda
* <https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/WhatIsCloudWatchEvents.html>

**stateChangeLambda**

* Retrieves info about shut off instance
* Sends message to **monitoringNotification SNS**
* Can be configured to restart instance if stopped

**monitoringNotification SNS**

* Main SNS topic that sends out monitoring alerts

## Dashboard

* CFN Parameter to create a dashboard
* A widget is a graph
* On creation, the dashboard has widgets for
  1. Number of invocations for all of the Lambda functions

○ Total number of publishes for both SNS topics

* Default refresh time is 5 min
* Default history is 3 hours
  1. How far back the widgets go
* When a lambda function attaches an alarm to an EC2 instance
  1. It can update the dashboard to include a status\_check data widget

## CloudFormation Template

* To update code, upload a zip folder to S3 with the specified name
* Creates an IAM Role for Lambda functions
  1. Make sure you have permissions
* Parameters are grouped with CloudFormation::Interface
* CloudFormation passes data to Lambda with environment variables
  1. Passes SNS Arn

○ Passes identifying tag(asked for in the parameters section)

■ Tag is needed to attach an alarm or for state change to send a message

○ And a lot of parameters

* Everything needed for the system is in connorsbucketbethyname S3 bucket
  1. If bucket name changes then things break

○ Good thing its part of a map in the template so only have to change on thing

* On CloudFormation creation
  1. A scheduled cloudwatch event is created and sends a message to a lambda function

○ Function

■ Attaches alarm to all marked instances

■ Deletes CW event and itself

* CFN EC2Tag to identify instances should be pre-created
* Setup lambda has to have its set up event name hardcoded to remove circular dependencies
* Link for YAML template when creating stack <https://s3.amazonaws.com/connorsbucketbethyname/monitoringSystem.yml>

## IAM

* CFN Template creates an IAM policy and role
* One role to be used by all of the lambda functions
* Named (region)\_(iamName)

Named (region)\_(iamName)

○ (region) is the region of the system

○ iamName can be changed within the CFN template

# Problems Encountered

* I tried to test many combinations of parameters that I could but there is almost certainly more bugs
* SNS to lambda event is weird
  1. Sends a mixed event of JSON and Unicode formatting

○ Solved by changing the python runtime from 2.7 to 3.6 which make the Unicode formatting a string

■ The string could then be turned into a dictionary

* Circular dependencies in CFN template
  1. Solved by putting all resource names into a map then calling the map
* The dashboard occasionally doesn’t delete

# !Use cases

* Security logging/monitoring
* Billing
* Termination protection

# Limitations

* SNS - can only send a certain amount of SMS messages per month b/c dollar limit
* Can't have more than one monitoring system per region per account

# Pricing / Costs

- Cloudwatch

○ <https://aws.amazon.com/cloudwatch/pricing/>

○ $0.10 per alarm per month

○ Basic metrics are free at the 5 min period

# Other

* How do we expand to other services

○ Alarms can be set up with different metrics for other services

* Email me at caicon18@gmail.com if there are any game breaking problems or if nothing makes sense