Kops Nodes Alerts

find a solution to automatic alert (by slack or email) when our cluster is down (either master is all down or nodes is all down) you can use cloudwatch, or other free/open-source solution.
so the issue here is "we want to be alerted so we can take manual action when it's completely broken", you can think freely how we can solve the issue

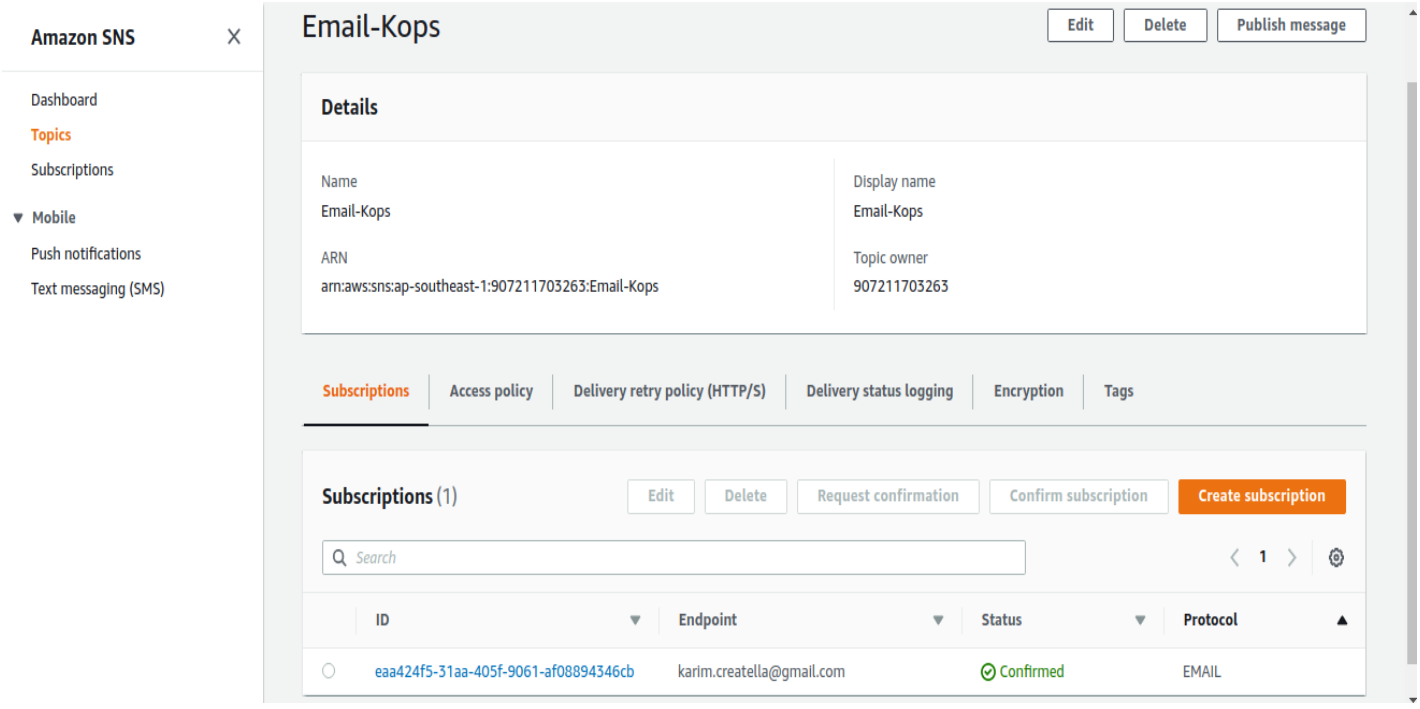
Solution

1. Using Cloudwatch to send Email Alert.
2. Using Cloudwatch with Slack to send Notification in Channel.

In Two options we will use “SNS” aws notification service to create topic and subscription, when cloudwatch detect alert “Terminate - Launch instances in autoscaling group of kops Cluster”.

1.Using Cloudwatch to send Email Alert.

1. Access AWS console and Create topic called “Email-Kops” and subscription type “Email” to send mail for example to my Email “karim.creatella@gmail.com”



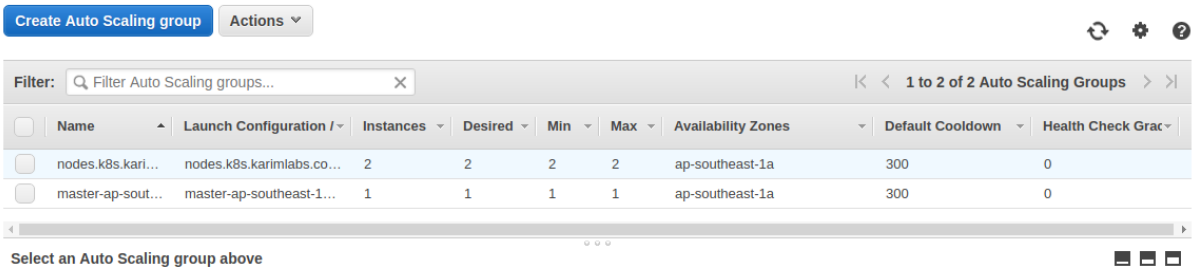
Note : you will ask for confirmation in your mail to allow aws send mails “Notifications” to your mail, so you need to confirm that.

2.Create Notification for AWS Kops cluster Instances groups.

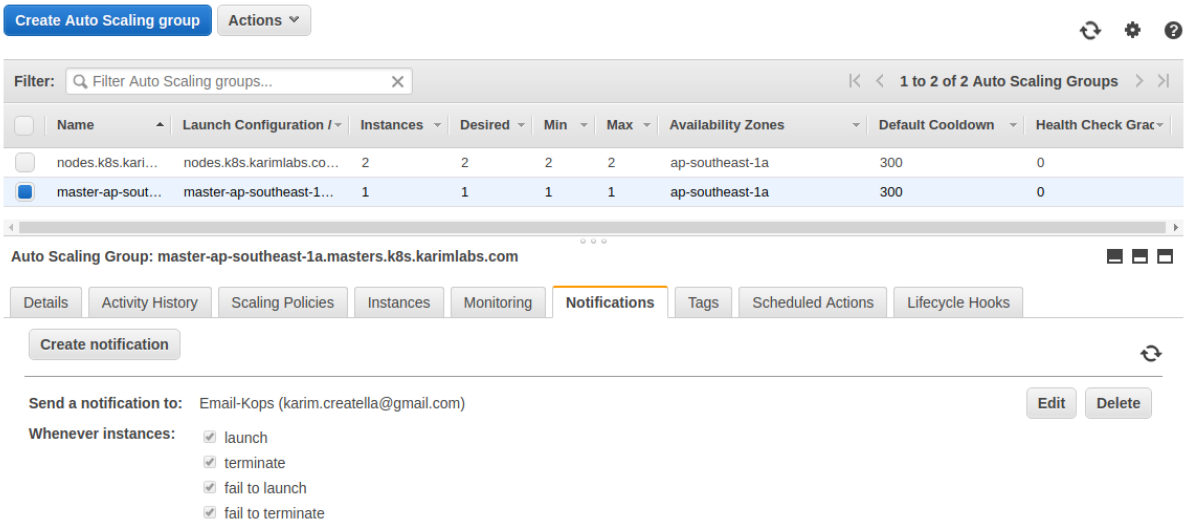
In AWS Console Open Autoscaling group and check the instances groups.

In my case i used this Repo to deploy Kops https://github.com/karimfadl/vagrant-kops-ansible that consist of two instances groups “1 Master have one instances” “1 Nodes have 2 Nodes”

If you use the same example you will find the instances group as the following:



In Both of them “Master and Nodes” you will click to instances group, chose “Notifications” and mark to all options.



Note : you will receive Email that notify this action will send to you in the future ignore it.

Now we will do a simple test to see if it's working or not.

We will terminate one of Kops EC2 nodes and see what will happen.

3.Test Cloudwatch Email Notification in case any Node Terminated or Launched.

When i terminated the instance i received mail like the following:

Auto Scaling: termination for group "nodes.k8s.karimlabs.com"

Inbox ×

Email-Kops no-reply@sns.amazonaws.com via amazonses.com 1:24 AM (47 minutes ago)

to me

Service: AWS Auto Scaling
Time: 2019-07-05T23:24:39.893Z
RequestId: cbc5af11-d0ca-431f-d94f-c0f5b49b5e30
Event: autoscaling:EC2_INSTANCE_TERMINATE
AccountId: 907211703263
AutoScalingGroupName: nodes.k8s.karimlabs.com
AutoScalingGroupARN: arn:aws:autoscaling:ap-southeast-1:907211703263:autoScalingGroup:fcd6e8d5-5f9a-46d9-83b4-ec759ddb21e2:autoScalingGroupName/nodes.k8s.karimlabs.com
ActivityId: cbc5af11-d0ca-431f-d94f-c0f5b49b5e30
Description: Terminating EC2 instance: i-0d7b0bb5864e44443
Cause: At 2019-07-05T23:24:37Z an instance was taken out of service in response to a EC2 health check indicating it has been terminated or stopped.
StartTime: 2019-07-05T23:24:37.904Z
EndTime: 2019-07-05T23:24:39.893Z
StatusCode: InProgress
StatusMessage:
Progress: 50
EC2InstanceId: i-0d7b0bb5864e44443
Details: {"Subnet ID":"subnet-01e46cfa124ccb896","Availability Zone":"ap-southeast-1a"}

Kops detected that and start to launch a new instance and i received a new mail for launching the new instances like the following:

Auto Scaling: launch for group "nodes.k8s.karimlabs.com"

Inbox ×

Email-Kops no-reply@sns.amazonaws.com via amazonses.com 1:26 AM (46 minutes ago)

to me

Service: AWS Auto Scaling
Time: 2019-07-05T23:26:08.665Z
RequestId: 68e5af11-d296-c459-7a1b-b73266169971
Event: autoscaling:EC2_INSTANCE_LAUNCH
AccountId: 907211703263
AutoScalingGroupName: nodes.k8s.karimlabs.com
AutoScalingGroupARN: arn:aws:autoscaling:ap-southeast-1:907211703263:autoScalingGroup:fcd6e8d5-5f9a-46d9-83b4-ec759ddb21e2:autoScalingGroupName/nodes.k8s.karimlabs.com
ActivityId: 68e5af11-d296-c459-7a1b-b73266169971
Description: Launching a new EC2 instance: i-08a0062604a723721
Cause: At 2019-07-05T23:25:07Z a difference between desired and actual capacity changing the desired capacity, increasing the capacity from 1 to 2.
StartTime: 2019-07-05T23:25:07.377Z
EndTime: 2019-07-05T23:26:08.665Z
StatusCode: InProgress
StatusMessage:
Progress: 50
EC2InstanceId: i-08a0062604a723721
Details: {"Subnet ID":"subnet-01e46cfa124ccb896","Availability Zone":"ap-southeast-1a"}

2.Using Cloudwatch with Slack to send Notification in Channel.

In This Case we will create SNS with Slack through Lambda Function.
I used the following Resource for that:
https://dev.to/alex_barashkov/how-to-send-aws-cloudwatch-alarms-to-slack-596e

Let’s Follow the steps:

1.Create Your Slack channel That will receive the Notifications.

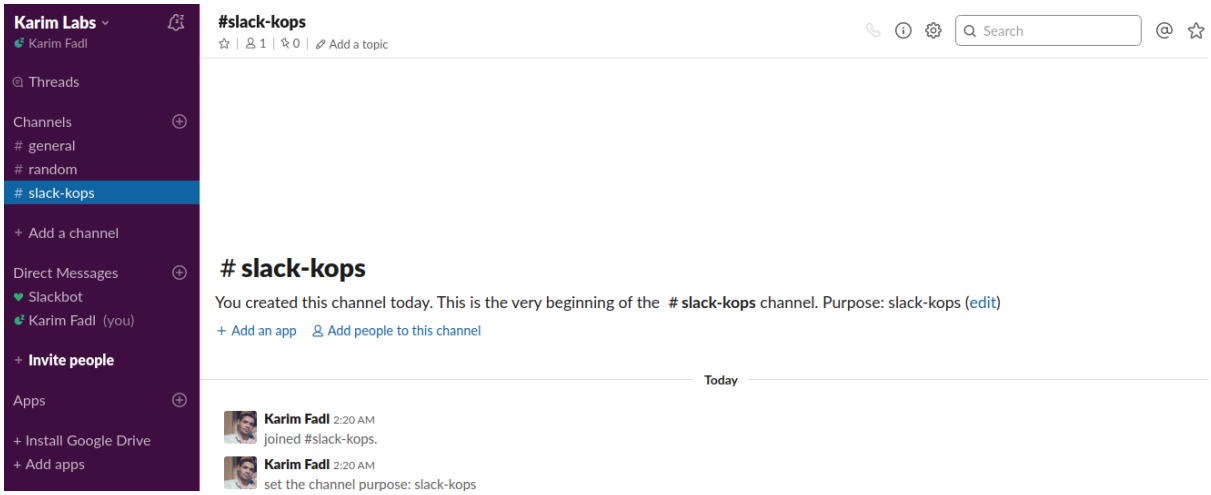
2.Create AWS Role To execute Lambda Function.

3.Create Lambda Functions to be Part from SNS Topic and Subscription.

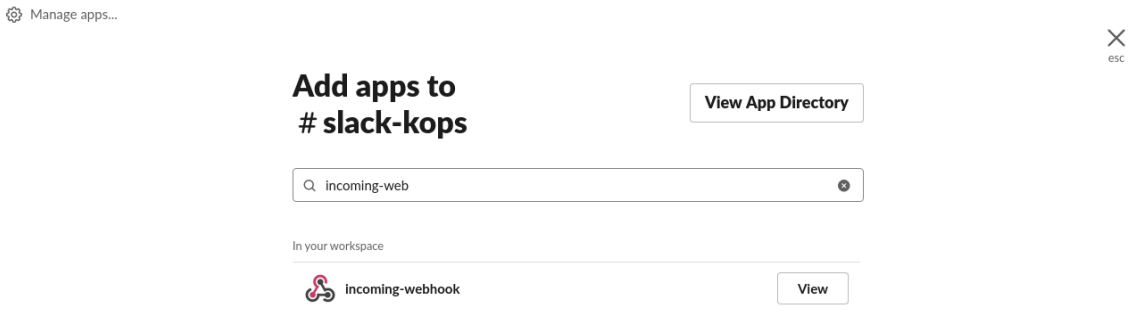
4.Test The Slack Notification if you terminated an instance.

1.Create Your Slack channel That will receive the Notifications.

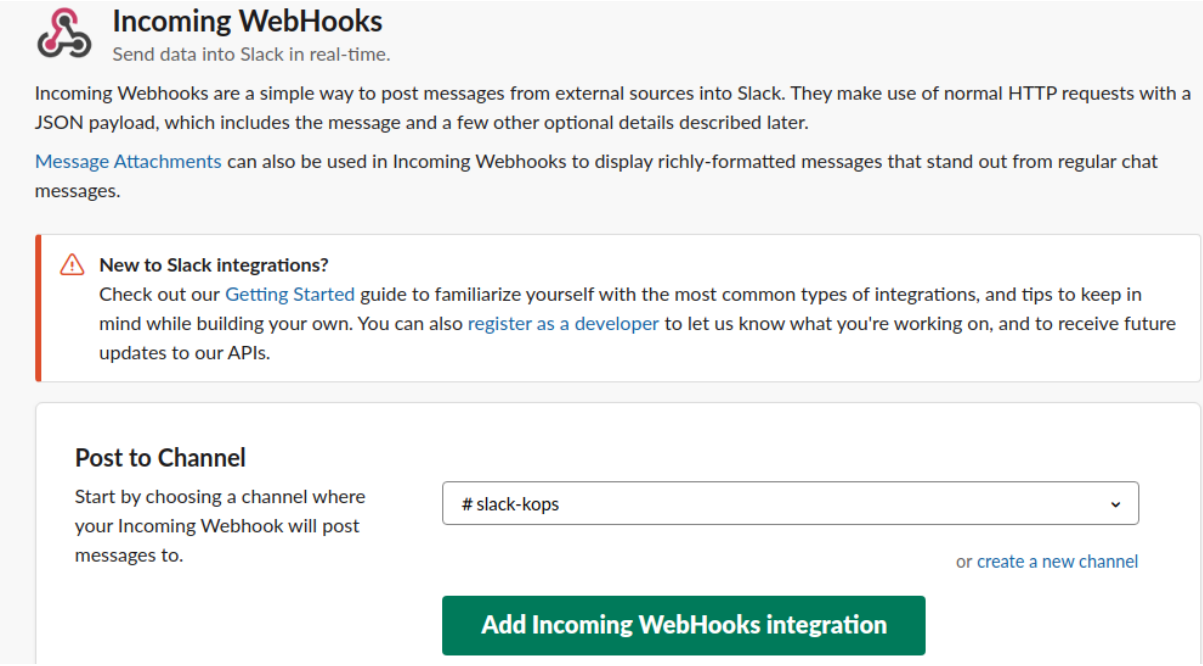
In My Slack workspace i created a new channel called “slack-kops”, to receive all kops notification on it.



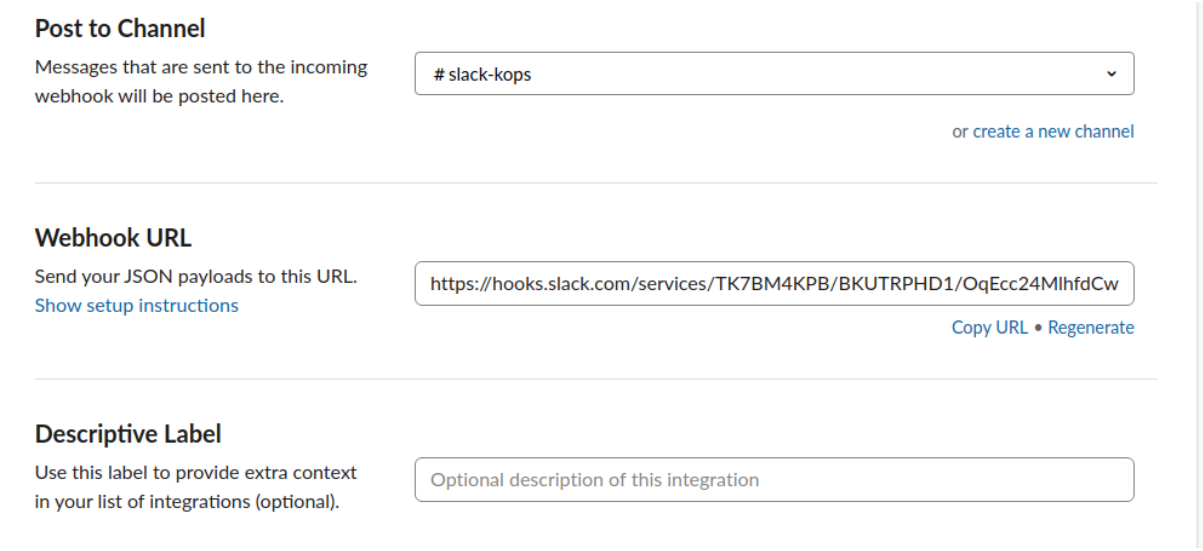
Next step i will click to “+add an app” to add “incoming webhook” app that will integrate with AWS SNS with tocken.



After you installed it you will start to Configure it by adding the app to our slack channel “slack-kops”



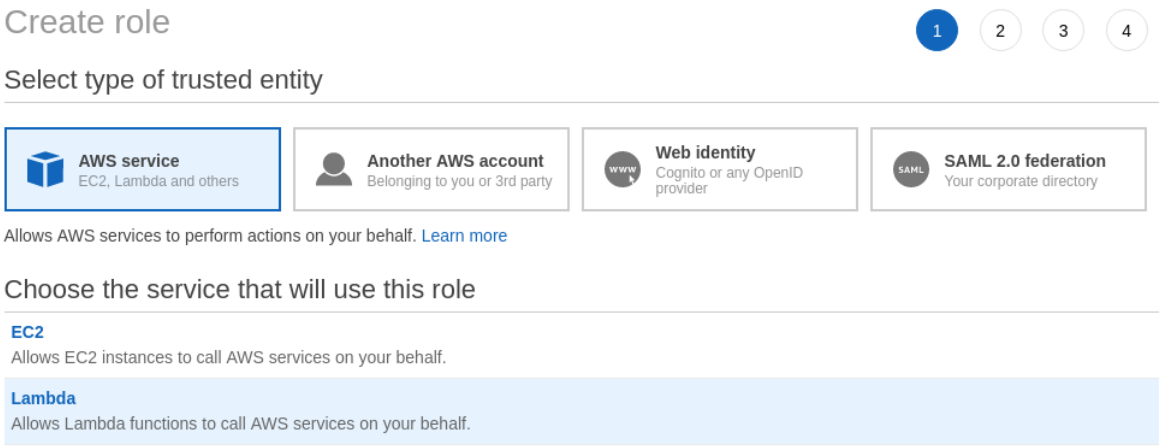
After that it’s automatically generate a tocken for your integration you must save it for future steps.
Also you can change some configuration like change slack channel logo.



2.Create AWS Role To execute Lambda Function.

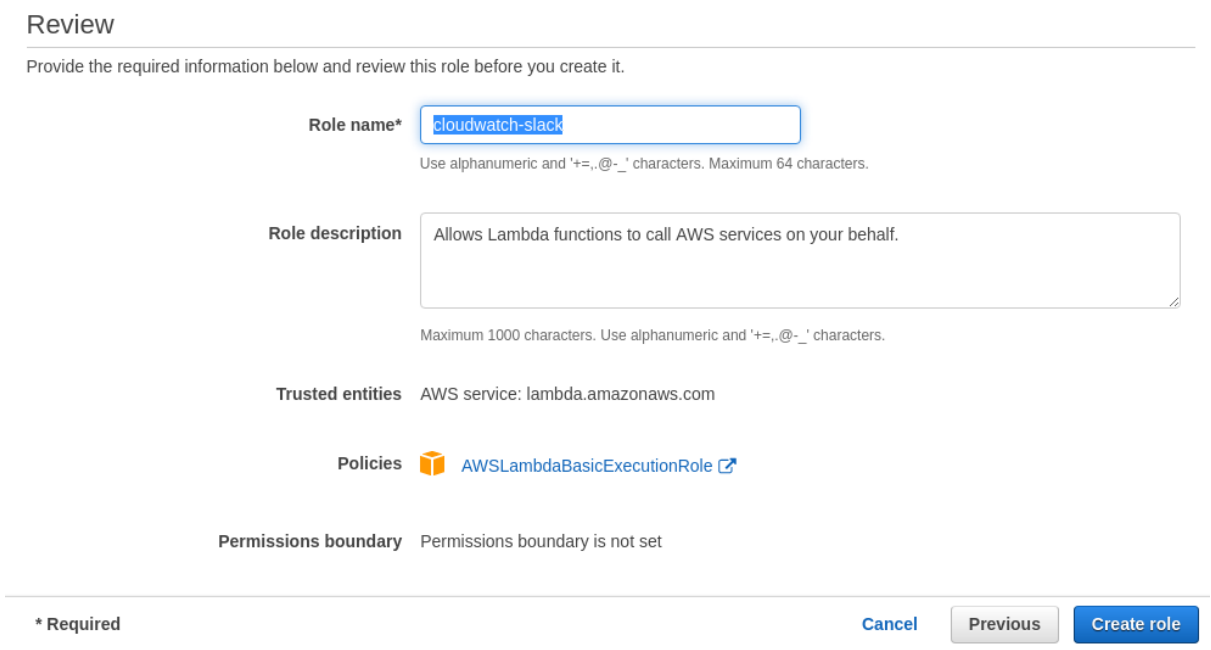
In this Step we create AWS custom role for Lambda function that will running to create SNS.

Through AWS console >> IAM >> Role >> chose AWS service >> Lambda.

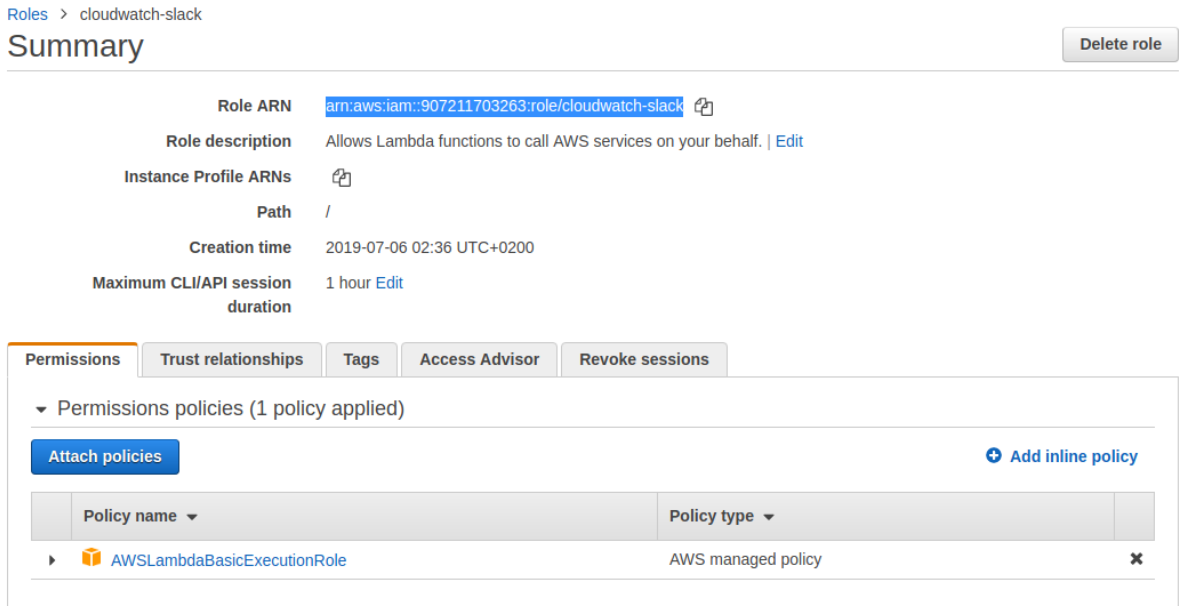


Chose “AWSLambdaBasicExecutionRole” Policy to attach to this Role.

Add Role Name for example “cloudwatch-slack”



In Summary section Save the ARN of this Role we will use it in the next steps.



3.Create Lambda Functions to be Part from SNS Topic and Subscription.

To deploy the AWS Lambda function, you need to clone the Repo:
https://github.com/assertible/lambda-cloudwatch-slack

and have Node.js installed on your local machine.

Especially for that guide, we contributed significantly in the repository to get a better configuration process.

```
git clone git@github.com:assertible/lambda-cloudwatch-slack.git
cd lambda-cloudwatch-slack
cp .env.example .env
```

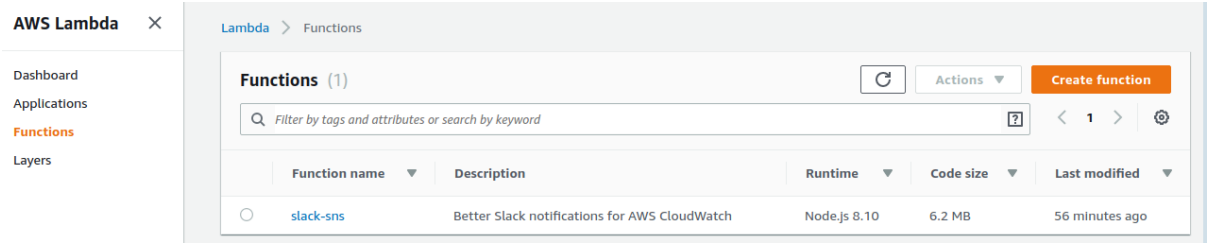
Open your .env file and fill in the environment variables.

Change The following Env :
UNENCRYPTED_HOOK_URL >> Slack URL WebHook
AWS_FUNCTION_NAME >> Choose any Name
AWS_REGION >> for example : ap-southeast-1
AWS_ROLE >> "arn:aws:iam::907211703263:role/cloudwatch-slack"
AWS_ACCESS_KEY_ID
AWS_SECRET_ACCESS_KEY

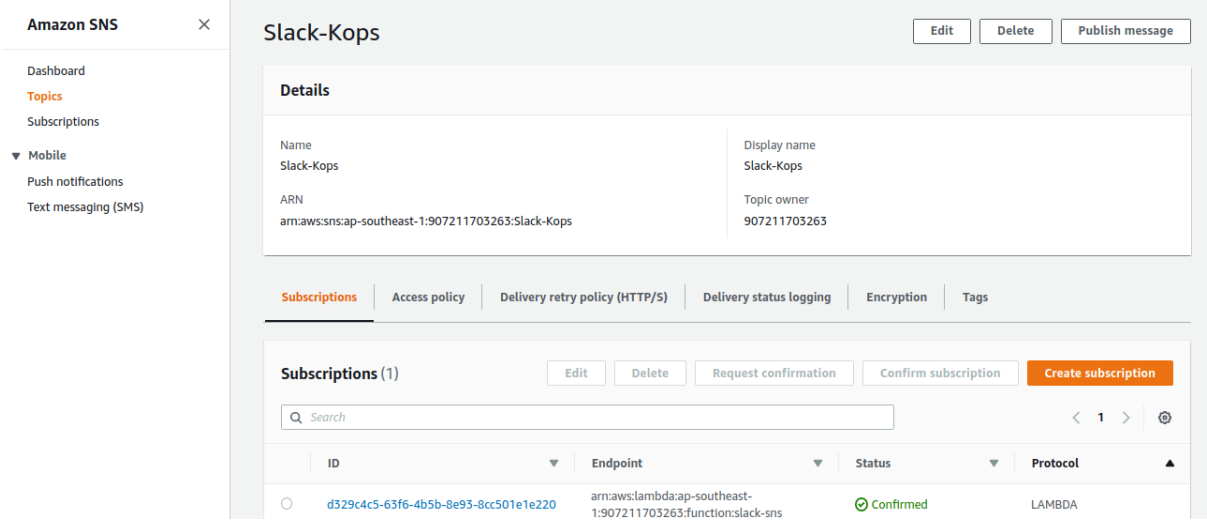
Now You can Run the script with two Steps:

```
npm install
npm run deploy
```

Now check the function that created, Open your AWS console in AWS service Lambda Function and see your function.

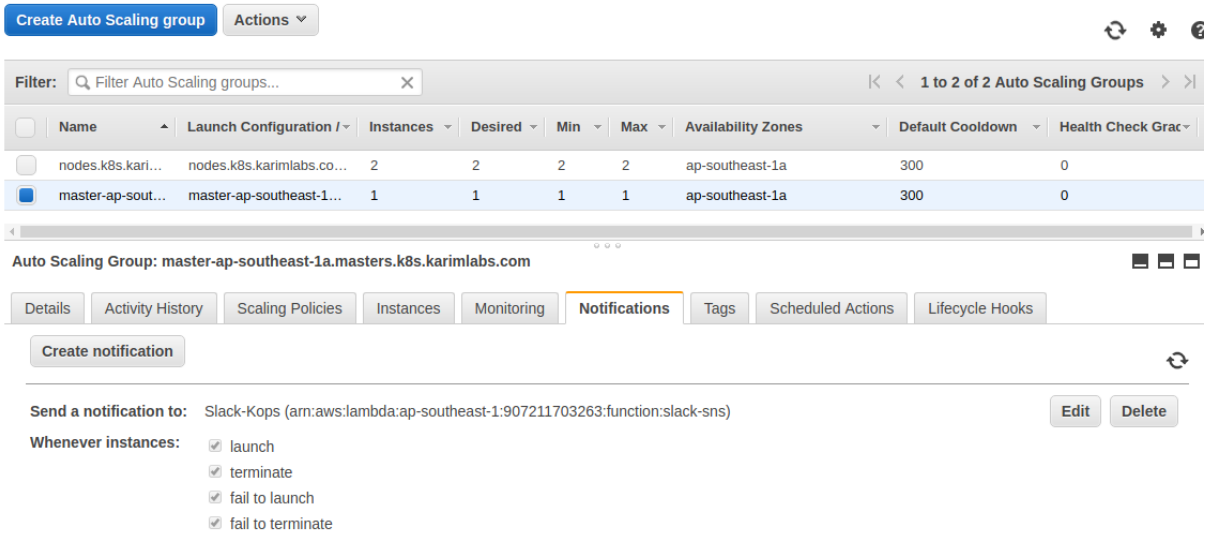


Now we can create a new SNS topic called for example “Slack-Kops” with “Lambda” type subscription.

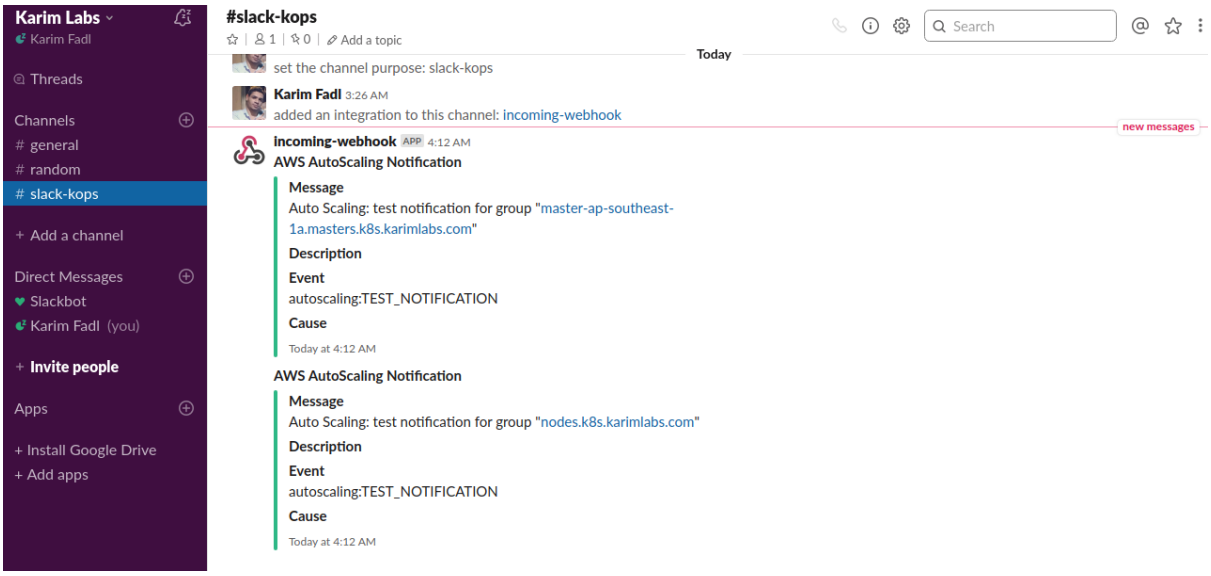


4.Test The Slack Notification if you terminated an instance.

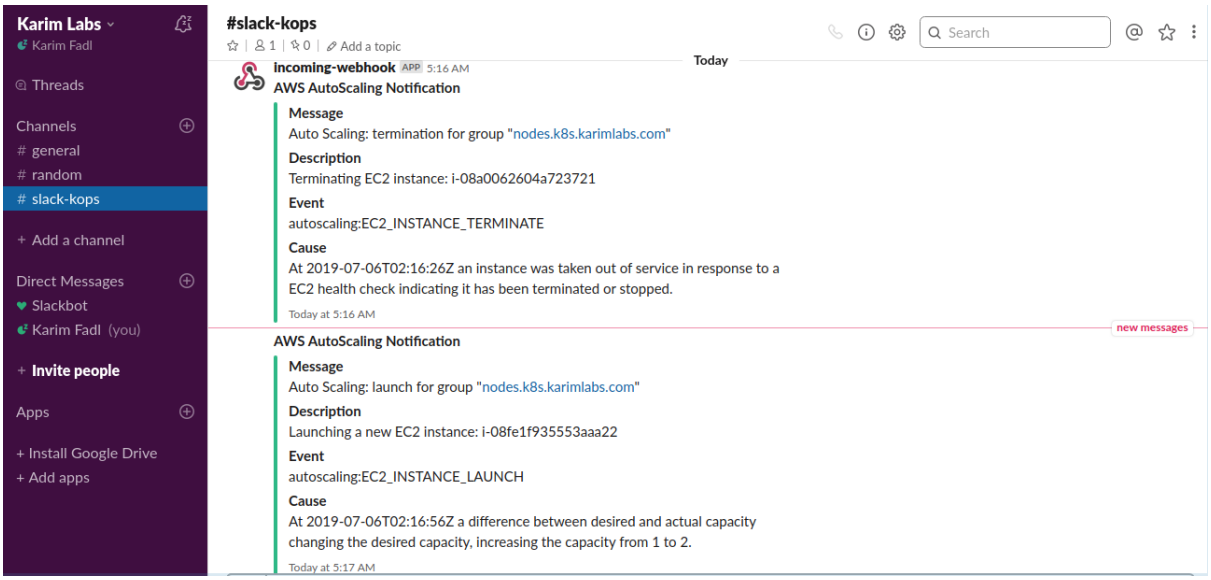
Like we did in SNS of Email with Autoscaling, we open Autoscaling section in AWS Console and create a Notification “For Master and Nodes Instances Groups” but in this time to “Slack-Kops” Topic and see what will happen.



You can check your Slack Channel, you will find a Notification for that.



Let’s Terminate one of nodes Instances and see if slack will receive that.



We have it right Now Enjoy :)