ASG – SNS-Cloudwatch with CloudFormation

An Auto Scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. An Auto Scaling group also enables you to use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies.

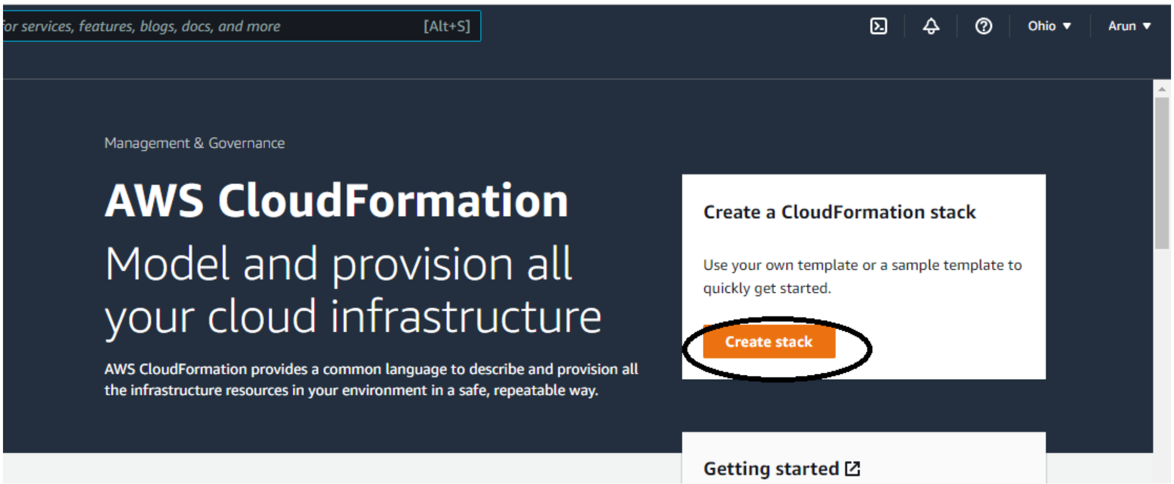
Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication.

The A2A pub/sub functionality provides topics for high-throughput, push-based, many-to-many messaging between distributed systems, microservices, and event-driven serverless applications. Using Amazon SNS topics, your publisher systems can fan out messages to a large number of subscriber systems, including Amazon SQS queues, AWS Lambda functions, HTTPS endpoints, and Amazon Kinesis Data Firehose, for parallel processing. The A2P functionality enables you to send messages to users at scale via SMS, mobile push, and email.

Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), IT managers, and product owners. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, and optimize resource utilization. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events. You get a unified view of operational health and gain complete visibility of your AWS resources, applications, and services running on AWS and on-premises. You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly.

Uploading Template file :

1. Create stack :



Management & Governance

AWS CloudFormation

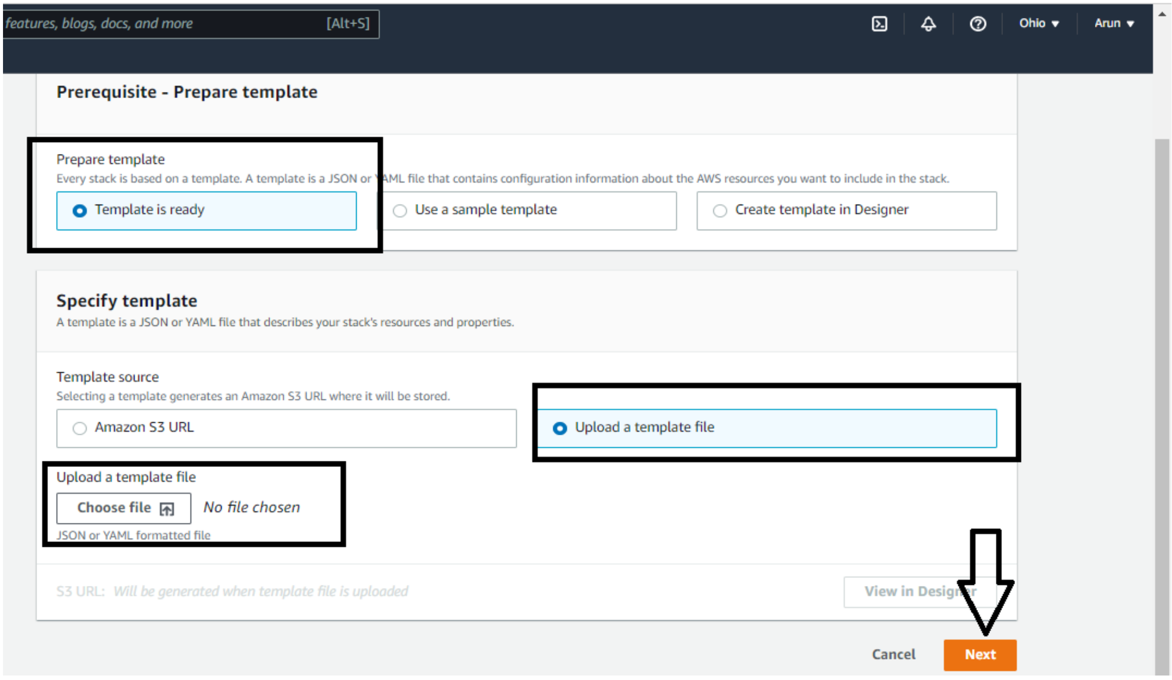
Model and provision all your cloud infrastructure

AWS CloudFormation provides a common language to describe and provision all the infrastructure resources in your environment in a safe, repeatable way.

Create a CloudFormation stack

Use your own template or a sample template to quickly get started.

2. prepare template upload it to to stack



Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

● Template is ready ○ Use a sample template ○ Create template in Designer

Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

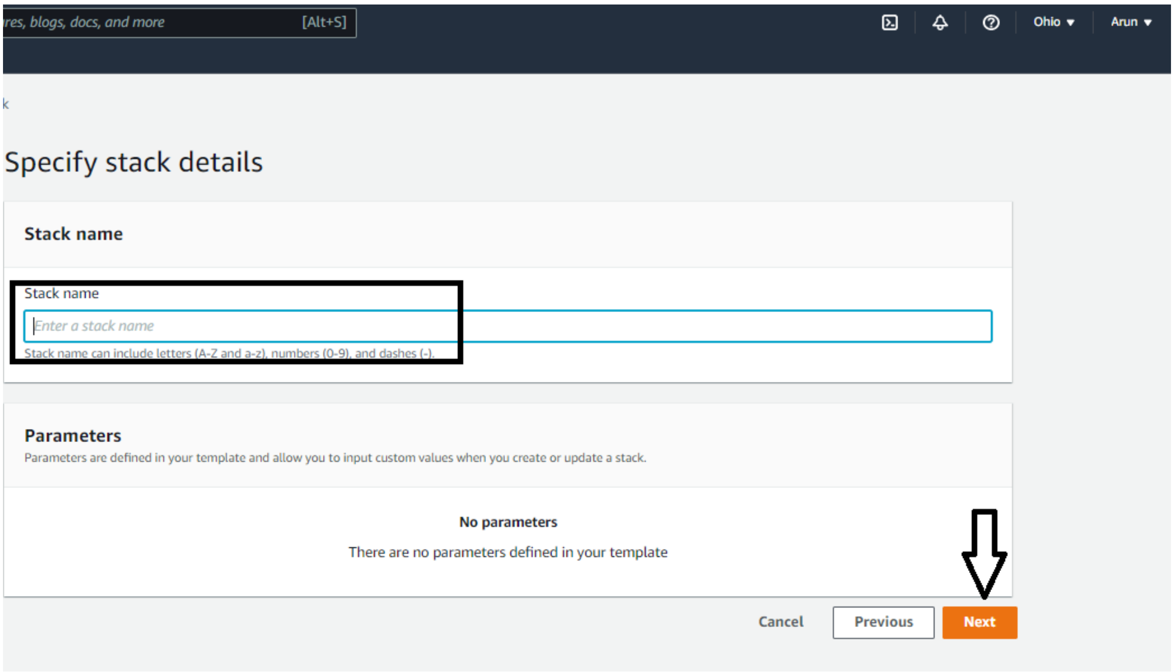
Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

○ Amazon S3 URL ● Upload a template file

Upload a template file
Choose file No file chosen
JSON or YAML formatted file

S3 URL: *Will be generated when template file is uploaded*

3. Enter Stack name and Next



Specify stack details

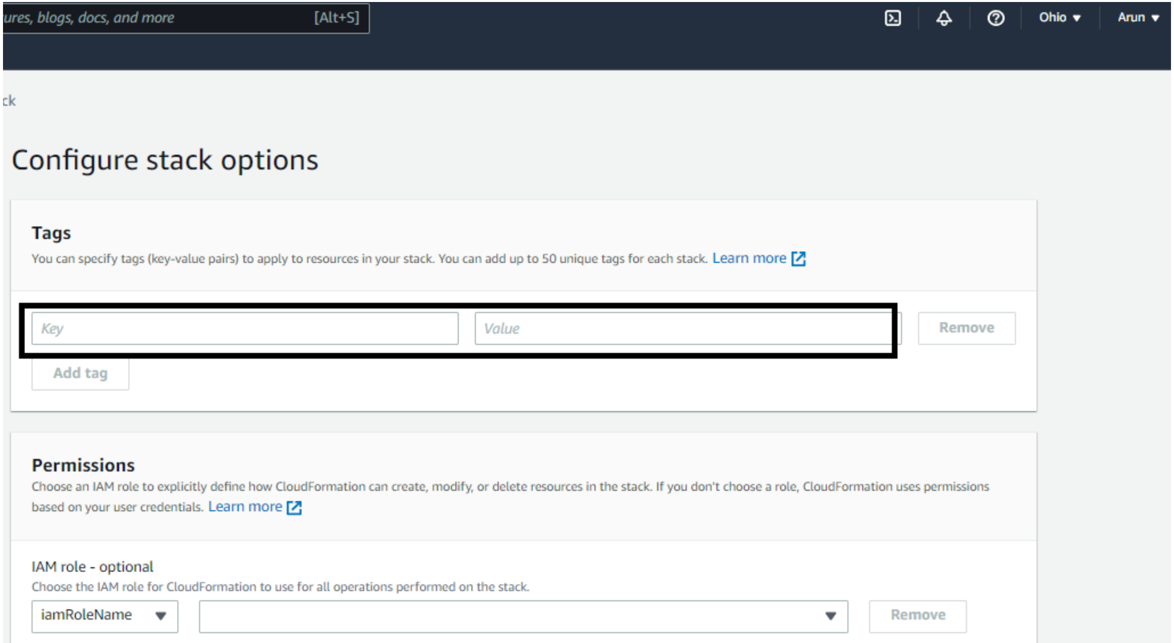
Stack name

Stack name
Enter a stack name
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

No parameters
There are no parameters defined in your template

4. Tags



Configure stack options

Tags

You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. Learn more

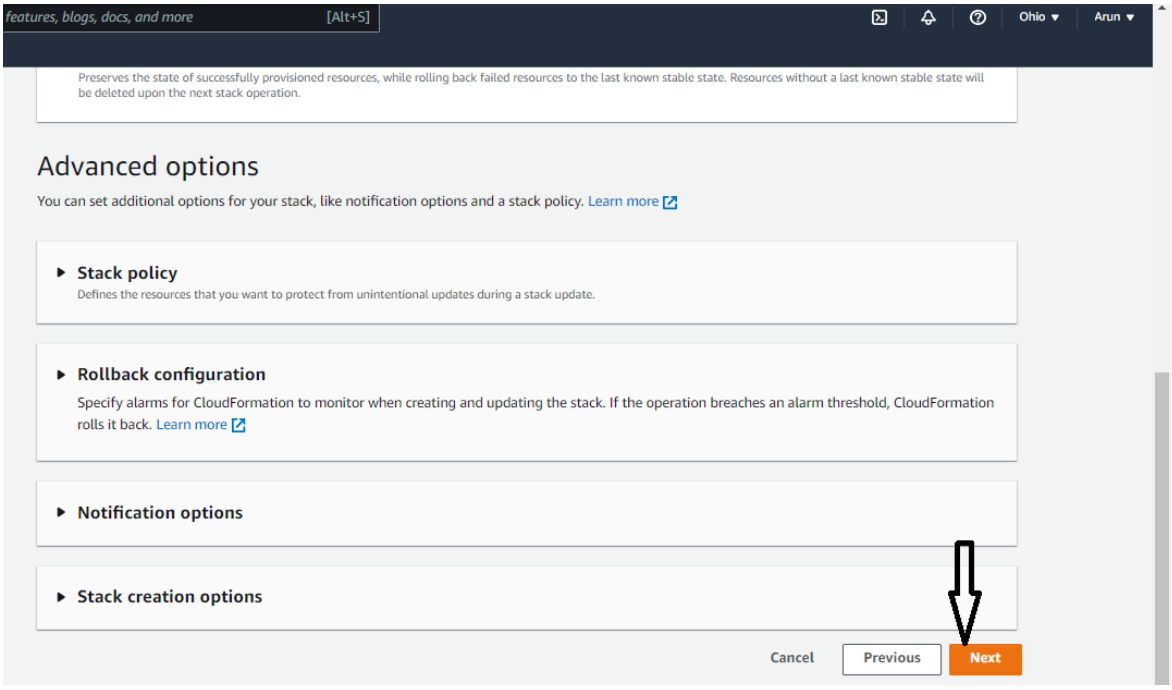
Permissions

Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. Learn more

IAM role - optional

Choose the IAM role for CloudFormation to use for all operations performed on the stack.

5.click Next



Preserves the state of successfully provisioned resources, while rolling back failed resources to the last known stable state. Resources without a last known stable state will be deleted upon the next stack operation.

Advanced options

You can set additional options for your stack, like notification options and a stack policy. Learn more

Stack policy

Defines the resources that you want to protect from unintentional updates during a stack update.

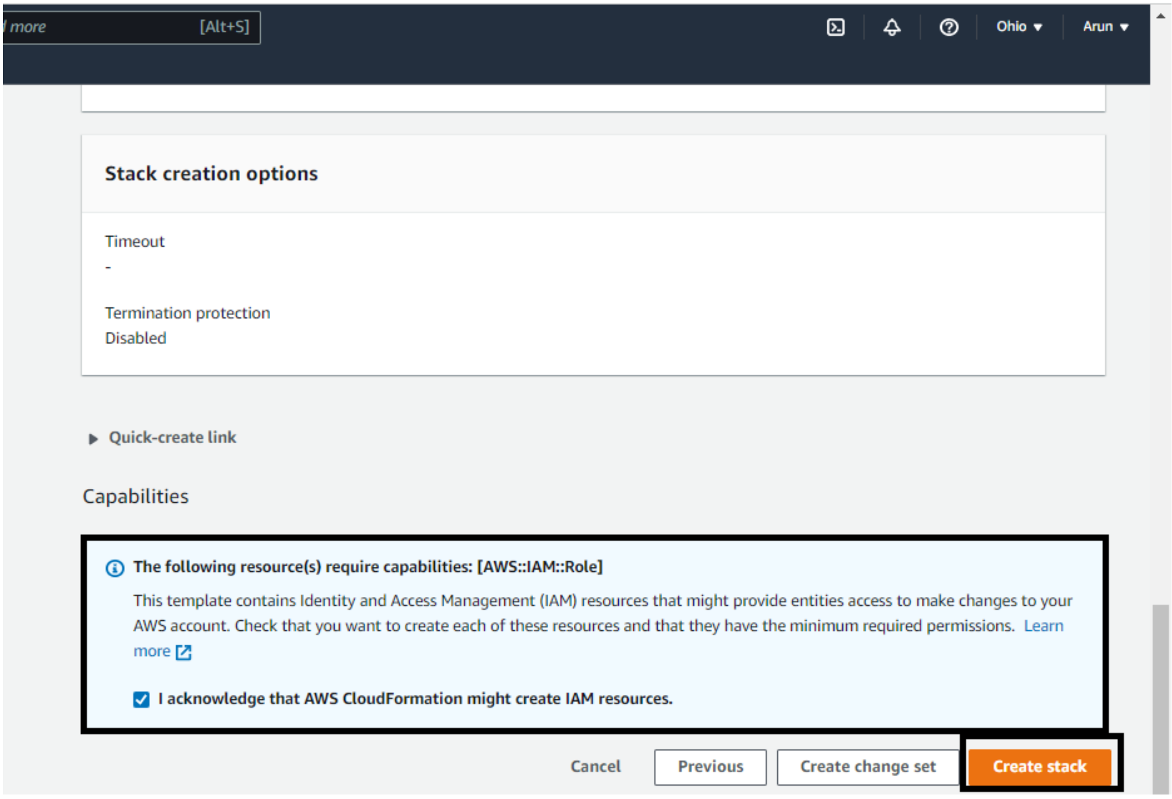
Rollback configuration

Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. Learn more

Notification options

Stack creation options

5. Review and Acknowledge



Stack creation options

Timeout
-

Termination protection
Disabled

▶ Quick-create link

Capabilities

The following resource(s) require capabilities: [AWS::IAM::Role]

This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. Learn more

☑ I acknowledge that AWS CloudFormation might create IAM resources.

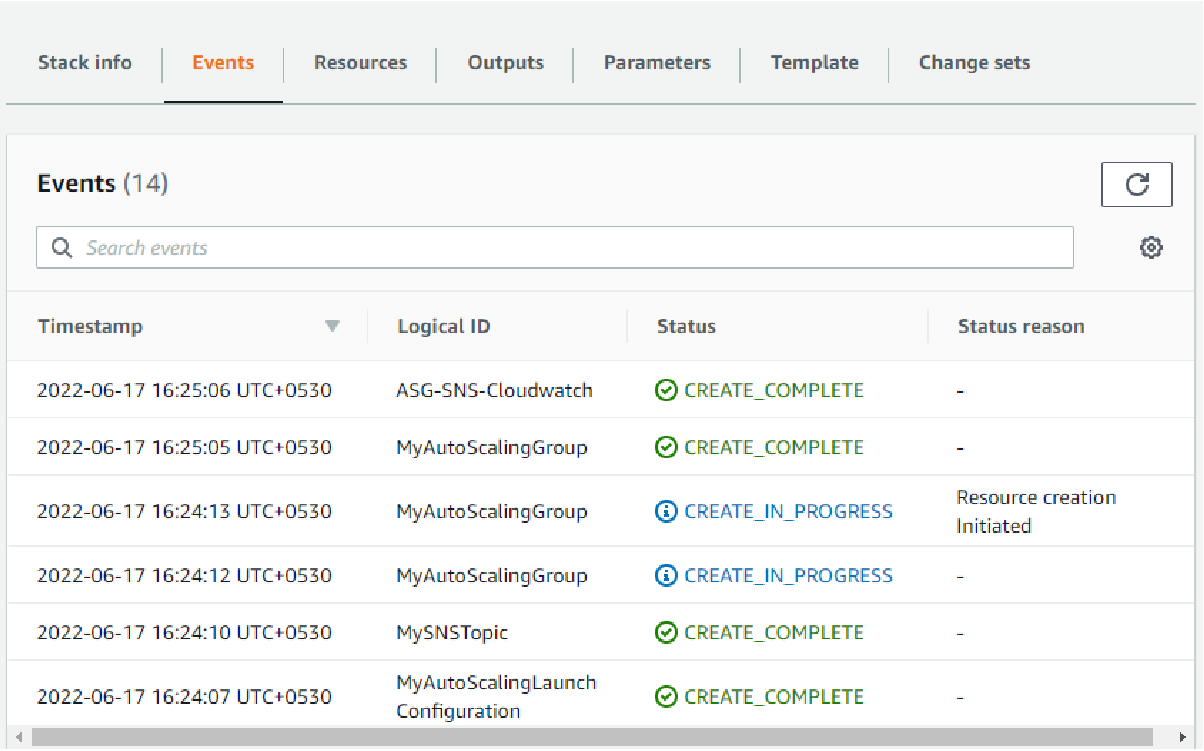
Yaml Code :

```
Parameters:
  KeypairName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instance
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  MyImageId:
    Type: String
    Default: ami-08df646e18b182346
    Description : ami-08df646e18b182346 is for Mumbai region
  TopicName:
    Type: String
    Description:  Topic Name
    Default: my-topic
Resources:
  MyAutoScalingLaunchConfiguration:
    Type: AWS::AutoScaling::LaunchConfiguration
    Properties:
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          yum update -y
          yum install -y httpd
          systemctl start httpd
          systemctl enable httpd
          echo "Hello World" > /var/www/html/index.html
      KeyName: !Ref KeypairName
      ImageId: !Ref MyImageId
      SecurityGroups:
        - !Ref MySecurityGroup
      InstanceType: t2.micro
  MySecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Allowing SSH from everywhere
      SecurityGroupIngress:
        - IpProtocol: tcp
          ToPort: '22'
          FromPort: '22'
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          ToPort: '80'
          FromPort: '80'
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          ToPort: '443'
          FromPort: '443'
          CidrIp: 0.0.0.0/0
  MyAutoScalingGroup:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      AvailabilityZones: !GetAZs
      MinSize: '2'
      MaxSize: '4'
      LaunchConfigurationName: !Ref MyAutoScalingLaunchConfiguration
      MetricsCollection:
      - Granularity: 1Minute
      NotificationConfigurations:
      - TopicARN: !Ref MySNSTopic
        NotificationTypes:
        - autoscaling:EC2_INSTANCE_LAUNCH
        - autoscaling:EC2_INSTANCE_LAUNCH_ERROR
        - autoscaling:EC2_INSTANCE_TERMINATE
        - autoscaling:EC2_INSTANCE_TERMINATE_ERROR
        - autoscaling:TEST_NOTIFICATION
  MySNSTopic:
   Type: AWS::SNS::Topic
   Properties:
     Subscription:
```

In this MetricsCollection we are enabling Cloudwatch

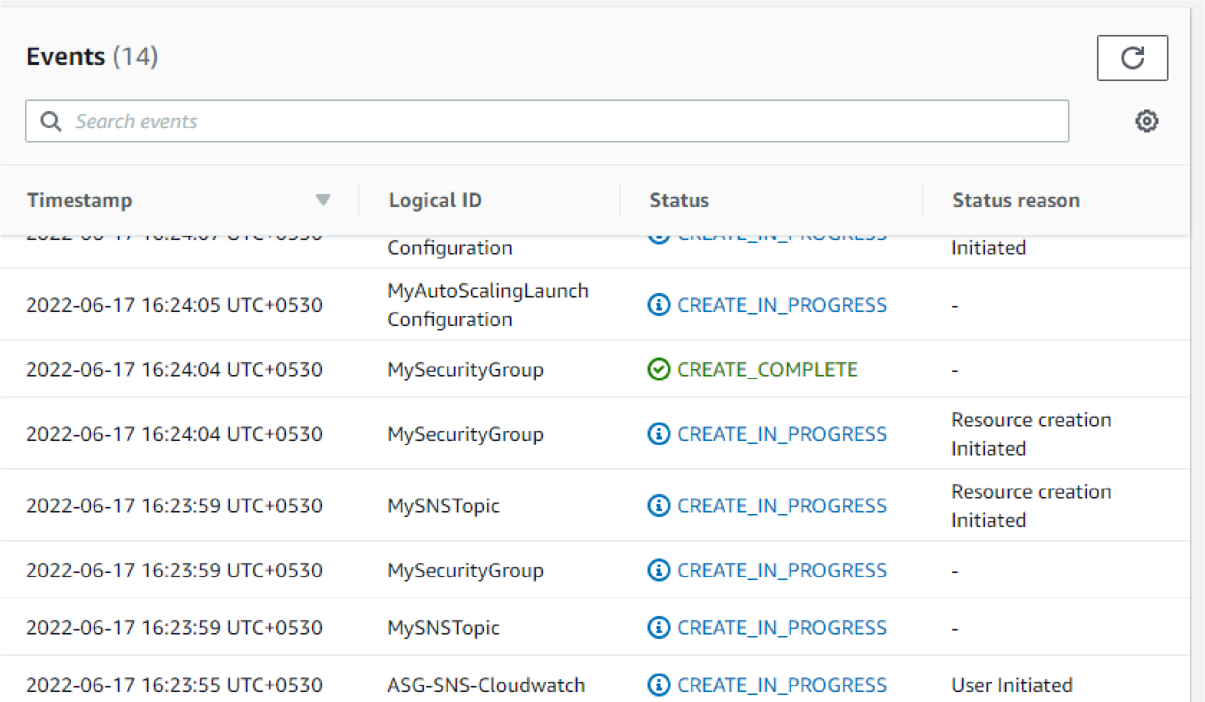
```
      - Endpoint: "arunhn.aws@gmail.com"
        Protocol: "email"
    TopicName: !Ref TopicName

Outputs:
  MyTopicArn:
    Description: Arn of Created SNS Topic
    Value: !Ref MySNSTopic
```



Events (14)

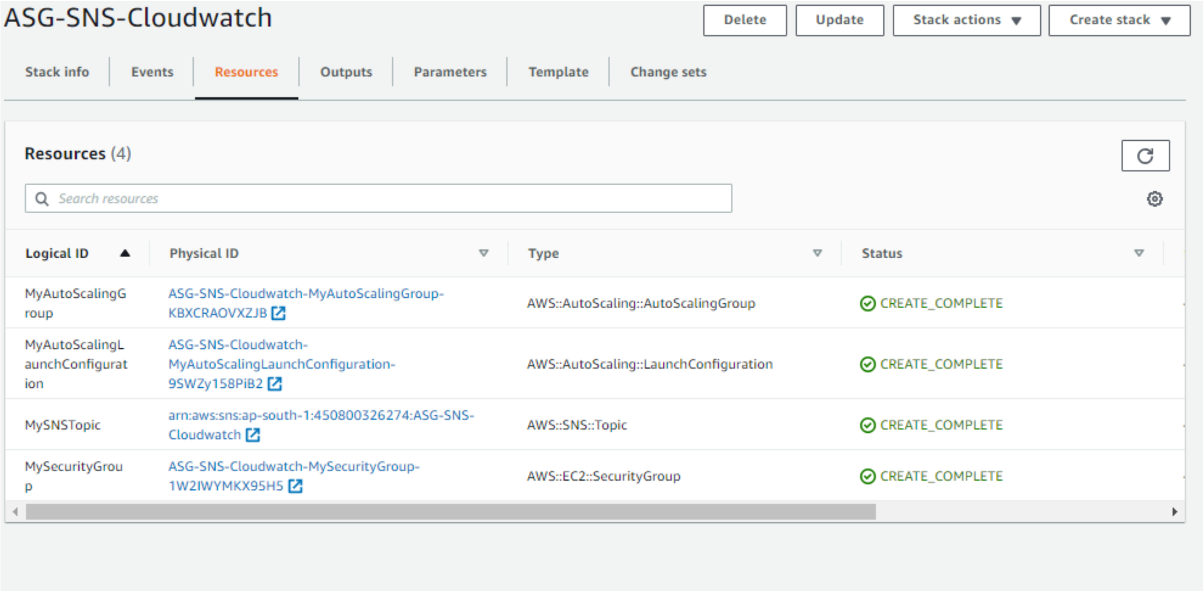
Timestamp	Logical ID	Status	Status reason
2022-06-17 16:25:06 UTC+0530	ASG-SNS-Cloudwatch	✓ CREATE_COMPLETE	-
2022-06-17 16:25:05 UTC+0530	MyAutoScalingGroup	✓ CREATE_COMPLETE	-
2022-06-17 16:24:13 UTC+0530	MyAutoScalingGroup	ⓘ CREATE_IN_PROGRESS	Resource creation Initiated
2022-06-17 16:24:12 UTC+0530	MyAutoScalingGroup	ⓘ CREATE_IN_PROGRESS	-
2022-06-17 16:24:10 UTC+0530	MySNSTopic	✓ CREATE_COMPLETE	-
2022-06-17 16:24:07 UTC+0530	MyAutoScalingLaunch Configuration	✓ CREATE_COMPLETE	-



Events (14)

Search events

Timestamp	Logical ID	Status	Status reason
	Configuration		Initiated
2022-06-17 16:24:05 UTC+0530	MyAutoScalingLaunchConfiguration	ⓘ CREATE_IN_PROGRESS	-
2022-06-17 16:24:04 UTC+0530	MySecurityGroup	✓ CREATE_COMPLETE	-
2022-06-17 16:24:04 UTC+0530	MySecurityGroup	ⓘ CREATE_IN_PROGRESS	Resource creation Initiated
2022-06-17 16:23:59 UTC+0530	MySNSTopic	ⓘ CREATE_IN_PROGRESS	Resource creation Initiated
2022-06-17 16:23:59 UTC+0530	MySecurityGroup	ⓘ CREATE_IN_PROGRESS	-
2022-06-17 16:23:59 UTC+0530	MySNSTopic	ⓘ CREATE_IN_PROGRESS	-
2022-06-17 16:23:55 UTC+0530	ASG-SNS-Cloudwatch	ⓘ CREATE_IN_PROGRESS	User Initiated

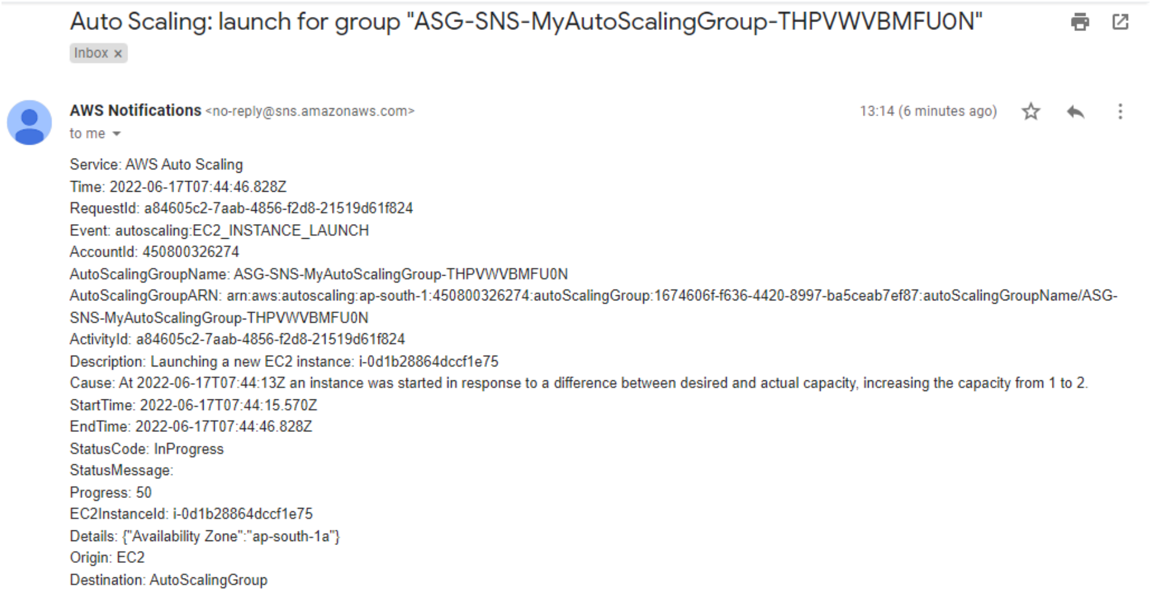


ASG-SNS-Cloudwatch

Resources (4)

Search resources

Logical ID ▲	Physical ID	Type	Status
MyAutoScalingGroup	ASG-SNS-Cloudwatch-MyAutoScalingGroup-KBXCRAOVXZJB	AWS::AutoScaling::AutoScalingGroup	✓ CREATE_COMPLETE
MyAutoScalingLaunchConfiguration	ASG-SNS-Cloudwatch-MyAutoScalingLaunchConfiguration-9SWZy158PiB2	AWS::AutoScaling::LaunchConfiguration	✓ CREATE_COMPLETE
MySNSTopic	arn:aws:sns:ap-south-1:450800326274:ASG-SNS-Cloudwatch	AWS::SNS::Topic	✓ CREATE_COMPLETE
MySecurityGroup	ASG-SNS-Cloudwatch-MySecurityGroup-1W2IWYMKX95H5	AWS::EC2::SecurityGroup	✓ CREATE_COMPLETE



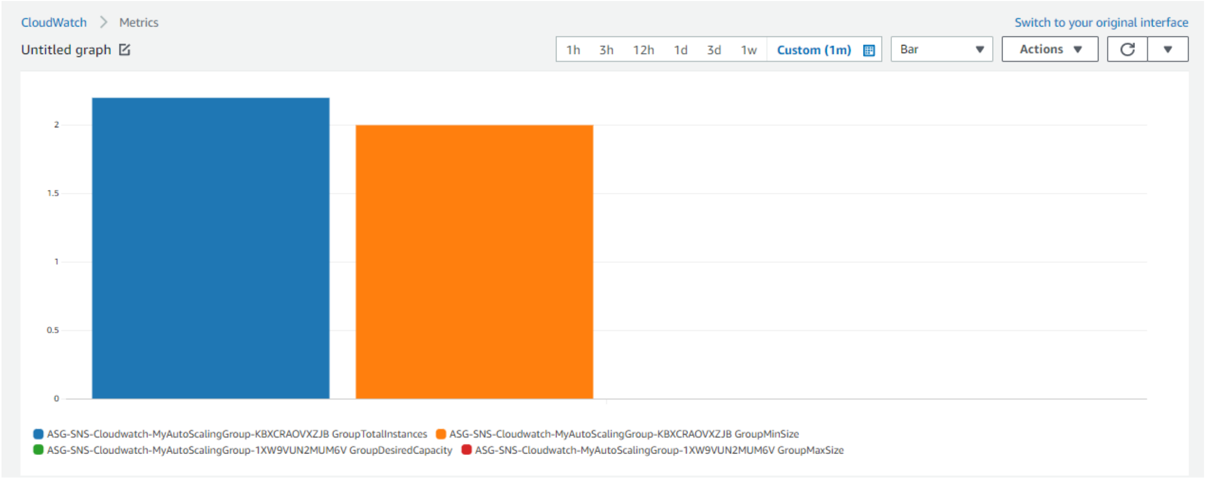
Auto Scaling: launch for group "ASG-SNS-MyAutoScalingGroup-THPVWVBMFU0N"

AWS Notifications <no-reply@sns.amazonaws.com>

to me

13:14 (6 minutes ago)

Service: AWS Auto Scaling
Time: 2022-06-17T07:44:46.828Z
RequestId: a84605c2-7aab-4856-f2d8-21519d61f824
Event: autoscaling:EC2_INSTANCE_LAUNCH
AccountId: 450800326274
AutoScalingGroupName: ASG-SNS-MyAutoScalingGroup-THPVWVBMFU0N
AutoScalingGroupARN: arn:aws:autoscaling:ap-south-1:450800326274:autoScalingGroup:1674606f-f636-4420-8997-ba5ceab7ef87:autoScalingGroupName/ASG-SNS-MyAutoScalingGroup-THPVWVBMFU0N
ActivityId: a84605c2-7aab-4856-f2d8-21519d61f824
Description: Launching a new EC2 instance: i-0d1b28864dccf1e75
Cause: At 2022-06-17T07:44:13Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.
StartTime: 2022-06-17T07:44:15.570Z
EndTime: 2022-06-17T07:44:46.828Z
StatusCode: InProgress
StatusMessage:
Progress: 50
EC2InstanceId: i-0d1b28864dccf1e75
Details: {"Availability Zone":"ap-south-1a"}
Origin: EC2
Destination: AutoScalingGroup



CloudWatch > Metrics
Switch to your original interface
Untitled graph
1h
3h
12h
1d
3d
1w
Custom (1m)
Bar
Actions
2
1.5
1
0.5
0
ASG-SNS-Cloudwatch-MyAutoScalingGroup-KBXCRAOVXZJB GroupTotalInstances
ASG-SNS-Cloudwatch-MyAutoScalingGroup-KBXCRAOVXZJB GroupMinSize
ASG-SNS-Cloudwatch-MyAutoScalingGroup-1XW9VUN2MUM6V GroupDesiredCapacity
ASG-SNS-Cloudwatch-MyAutoScalingGroup-1XW9VUN2MUM6V GroupMaxSize