

ASG - SNS with CloudFormation

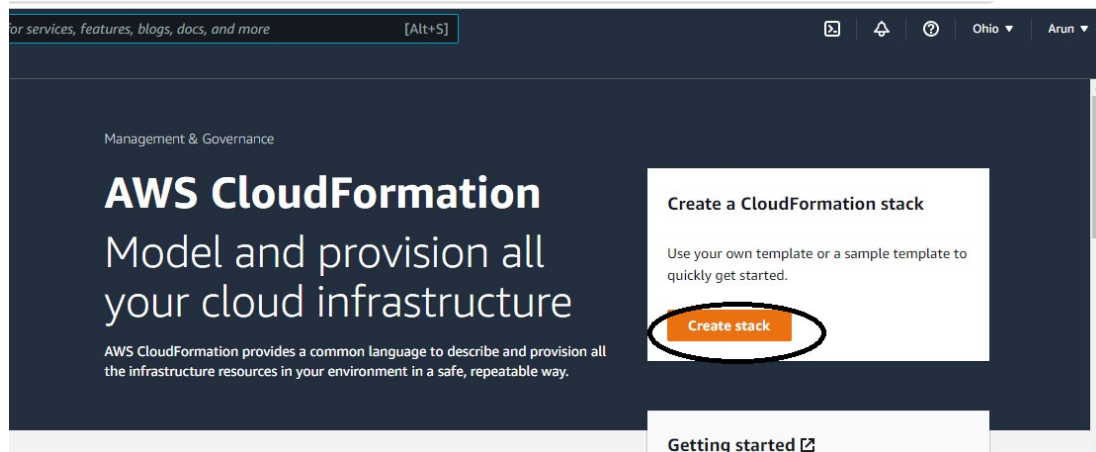
An Auto Scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. An Auto Scaling group also enables you to use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies.

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.

Uploading Template file :

1. Create stack :



2. prepare template upload it to to stack

The screenshot shows the 'Prerequisite - Prepare template' step in the AWS CloudFormation console. The 'Prepare template' section has three radio buttons: 'Template is ready' (selected), 'Use a sample template', and 'Create template in Designer'. The 'Specify template' section has two radio buttons: 'Amazon S3 URL' and 'Upload a template file' (selected). Below the 'Upload a template file' radio button is a 'Choose file' button with a file icon and the text 'No file chosen'. A large black arrow points down from the 'View in Designer' button to the 'Next' button at the bottom right.

features, blogs, docs, and more [Alt+S]

Ohio Arun

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

View in Designer

Cancel Next

3. Enter Stack name and Next

The screenshot shows the 'Specify stack details' step in the AWS CloudFormation console. The 'Stack name' section has a text input field with the placeholder 'Enter a stack name'. The 'Parameters' section shows 'No parameters' and 'There are no parameters defined in your template'. A large black arrow points down from the 'Next' button area to the 'Next' button at the bottom right.

res, blogs, docs, and more [Alt+S]

Ohio Arun

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 (-).

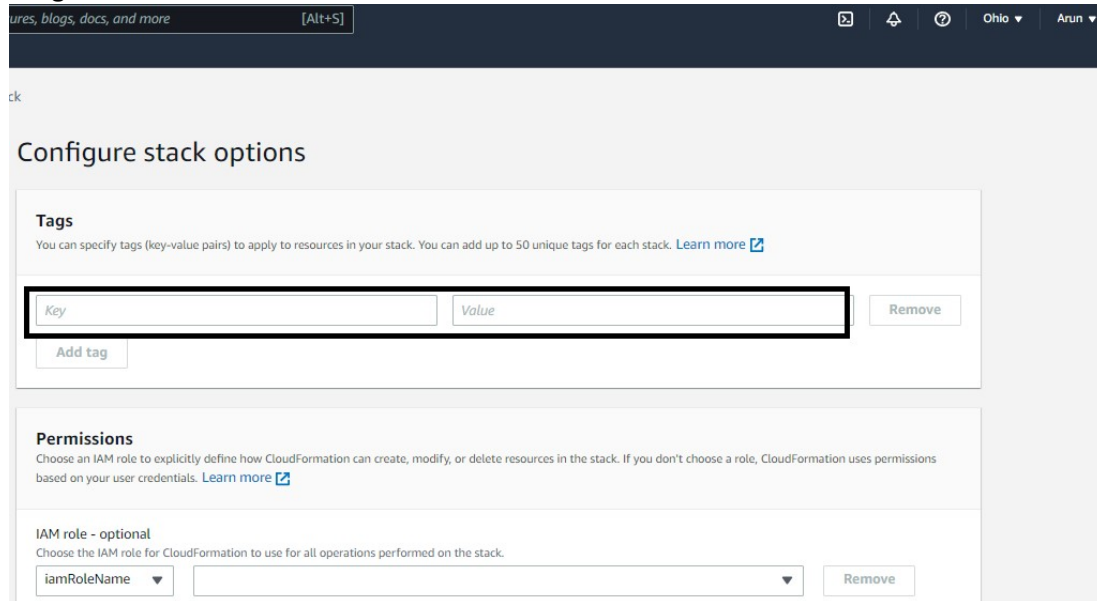
Parameters

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

Cancel Previous Next

4. Tags



features, blogs, docs, and more [Alt+S] Ohio Arun

ck

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](#)

Key	Value	Remove

Add tag

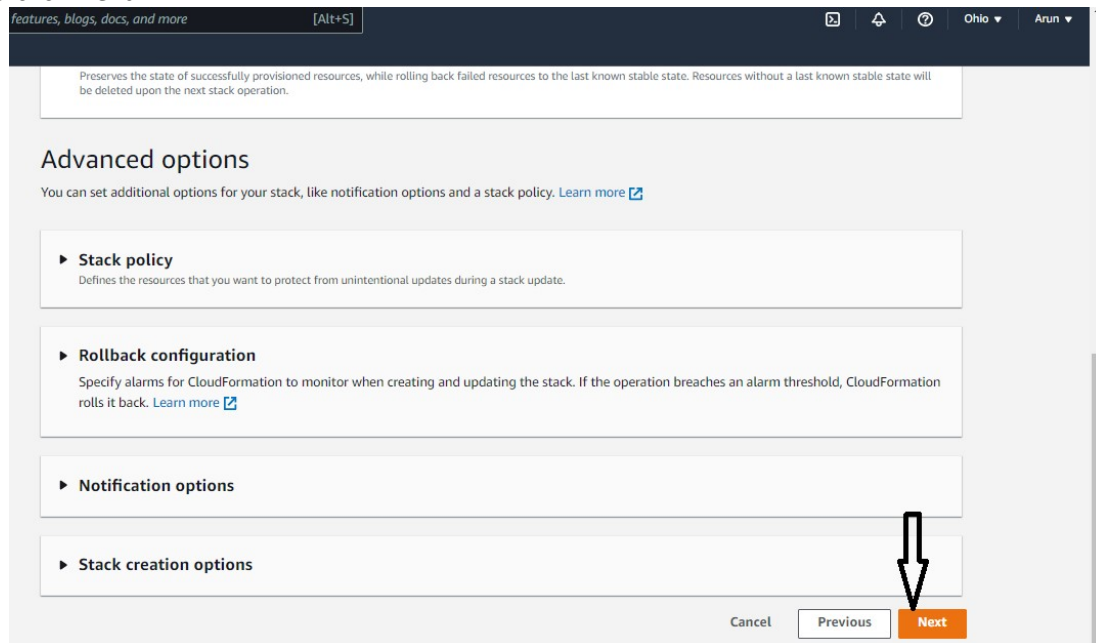
Permissions

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.

iamRoleName [v] [v] Remove

5.click Next



features, blogs, docs, and more [Alt+S] Ohio Arun

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**

Cancel Previous **Next**

5. Review and Acknowledge

more [Alt+S] Ohio Arun

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.

Cancel Previous Create change set **Create stack**

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

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

Here we are going to mention
for what actions we need
notifications

MySNSTopic:

Type: AWS::SNS::Topic

Properties:

Subscription:

- Endpoint: "arunhn.aws@gmail.com"

Protocol: "email"

In this we are going to
create simple notification
Service. Here email id is
used as endpoint.

TopicName: !Ref TopicName

Outputs:

MyTopicArn:

Description: Arn of Created SNS Topic

Value: !Ref MySNSTopic

more	[Alt+S]				Mumbai	ARUNA H N
Stack info	Events	Resources	Outputs	Parameters	Template	Change sets
Events (14)						
Search events						
Timestamp	Logical ID	Status	Status reason			
2022-06-17 13:14:51 UTC+0530	ASG-SNS	CREATE_COMPLETE				
2022-06-17 13:14:50 UTC+0530	MyAutoScalingGroup	CREATE_COMPLETE				
2022-06-17 13:13:58 UTC+0530	MyAutoScalingGroup	CREATE_IN_PROGRESS	Resource creation Initiated			
2022-06-17 13:13:57 UTC+0530	MyAutoScalingGroup	CREATE_IN_PROGRESS				
2022-06-17 13:13:56 UTC+0530	MySNSTopic	CREATE_COMPLETE				
2022-06-17 13:13:52 UTC+0530	MyAutoScalingLaunchConfiguration	CREATE_COMPLETE				

Events (14)

Timestamp	Logical ID	Status	Status reason
UTC+0530	MySNSTopic	LETE	
2022-06-17 13:13:52 UTC+0530	MyAutoScalingLaunchConfiguration	✔ CREATE_COMPLETE	-
2022-06-17 13:13:52 UTC+0530	MyAutoScalingLaunchConfiguration	ⓘ CREATE_IN_PROGRESS	Resource creation Initiated
2022-06-17 13:13:51 UTC+0530	MyAutoScalingLaunchConfiguration	ⓘ CREATE_IN_PROGRESS	-
2022-06-17 13:13:49 UTC+0530	MySecurityGroup	✔ CREATE_COMPLETE	-
2022-06-17 13:13:49 UTC+0530	MySecurityGroup	ⓘ CREATE_IN_PROGRESS	Resource creation Initiated
2022-06-17 13:13:45 UTC+0530	MySNSTopic	ⓘ CREATE_IN_PROGRESS	Resource creation Initiated
2022-06-17 13:13:44	MySNSTopic	ⓘ CREATE_IN_PROGRESS	-

Resources (4)

Logical ID	Physical ID	Type	Status	Status reason	Module
MyAutoScalingGroup	MyAutoScalingGroup-THPVWVBMFUON 🔗	AWS::AutoScaling::AutoScalingGroup	✔ CREATE_COMPLETE	-	-
MyAutoScalingLaunchConfiguration	ASG-SNS-MyAutoScalingLaunchConfiguration-wztv37twm26Y 🔗	AWS::AutoScaling::LaunchConfiguration	✔ CREATE_COMPLETE	-	-
MySNSTopic	arn:aws:sns:ap-south-1:450800326274:ASG-SNS 🔗	AWS::SNS::Topic	✔ CREATE_COMPLETE	-	-
MySecurityGroup	ASG-SNS-MySecurityGroup-51JZSQ5O8BTP 🔗	AWS::EC2::SecurityGroup	✔ CREATE_COMPLETE	-	-

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



Inbox x



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-THPVWVBMFUON

AutoScalingGroupARN: arn:aws:autoscaling:ap-south-1:450800326274:autoScalingGroup:1674606f-f636-4420-8997-ba5ceab7ef87:autoScalingGroupName/ASG-SNS-MyAutoScalingGroup-THPVWVBMFUON

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