

# Step 1

The screenshot displays the AWS IoT console interface. At the top, the navigation bar includes the AWS logo, 'Services', 'Resource Groups', 'Cloud9', a notification bell, 'N. Virginia', and 'Support'. The left-hand navigation menu lists various IoT services: Monitor, Onboard, Manage, Greengrass, Secure, Defend, **Act** (with 'Rules' highlighted), Destinations, and Test. The main content area is titled 'Rules' and features a search bar labeled 'Search rules'. A 'Create' button is visible in the top right corner. Below the search bar, a single rule card is shown with the name 'thresholdtemperatures' and the status 'ENABLED'. The 'Act' menu item and the 'Create' button are highlighted with red boxes.

## Step 2

### Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

Name

thresholdtemperature

Description

Notify Users when Temperature rises

### Rule query statement

Indicate the source of the messages you want to process with this rule.

Using SQL version

2016-03-23

Rule query statement

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see [AWS IoT SQL Reference](#).

```
1 SELECT Temperature FROM 'ESP32/DHT11' where Temperature > 27;
```

### Set one or more actions

Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (\*.required)

Add action

## Step 2

### Select an action

Select an action.



Insert a message into a DynamoDB table  
DYNAMODB



Split message into multiple columns of a DynamoDB table (DynamoDBv2)  
DYNAMODBv2



Send a message to a Lambda function  
LAMBDA



Send a message as an SNS push notification  
SNS



Send a message to an SQS queue  
SQS



Send a message to an Amazon Kinesis Stream  
AMAZON KINESIS



Republish a message to an AWS IoT topic  
AWS IOT REPUBLISH



Store a message in an Amazon S3 bucket  
S3

### Step 3



Send a message as an SNS push notification

\*SNS target

No topic selected

Create

Select

Message format

RAW

Choose or create a role to grant AWS IoT access to perform this action.

No role selected

Select

## Create

Name

SendTempNotification

Cancel

Create

## Step 4



Send a message as an SNS push notification

\*SNS target

SendTempNotification

Create

Clear

Select

Message format

RAW

Choose or create a role to grant AWS IoT access to perform this action.

No role selected

Create Role

Select

## Create a new role

A new IAM role will be created in your account. An inline policy will be attached to the role providing scoped-down permissions allowing AWS IoT to access resources on your behalf.

Name

PublishNotificatio|

Cancel

Create role

## Step 5

# Configure action



Send a message as an SNS push notification  
SNS

### \*SNS target

SendTempNotification

Create

Clear

Select

### Message format

RAW

Choose or create a role to grant AWS IoT access to perform this action.

PublishNotification

Policy Attached ✓

Create Role

Select

Cancel

Add action

## Step 6



Send a message as an SNS push notification  
SendTempNotification

Remove

Edit ▶

Add action

### Error action

Optionally set an action that will be executed when something goes wrong with processing your rule.

Add action

### Tags

Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair. [Learn more](#) about tagging your AWS resources.

#### Tag name

#### Value

Clear

Add another

Cancel

Create rule

# Step 7

aws Services ^ Resource Groups v Cloud9 \* N. Virginia v

History

Simple Notification Service

IoT Core

S3

Billing

Console Home

CodeCommit

Group A-Z

Simple Notification Service  
SNS managed message topics for Pub/Sub

Integration

EC2 AWS IQ ↗ Amazon SageMaker Step Functions

Lightsail ↗ Support Amazon CodeGuru Amazon EventBridge

ECR Managed Services Amazon Comprehend Amazon MQ

ECS Amazon Forecast Simple Notification Service

EKS Amazon Fraud Detector Simple Queue Service

Lambda Blockchain Amazon Kendra SWF

Batch Amazon Managed Blockchain Amazon Lex

Elastic Beanstalk Amazon Machine Learning

Serverless Application Repository Satellite AWS Cost Management

AWS Outposts Ground Station Amazon Personalize AWS Cost Explorer

EC2 Image Builder Quantum Technologies Amazon Polly AWS Budgets

Storage AWS Braket ↗ Amazon Rekognition AWS Marketplace Subscriptions

S3 Amazon Transcribe

EFS Amazon Translate

FSx AWS DeepLens

S3 Glacier AWS DeepRacer

Storage Gateway Amazon Augmented AI

AWS Backup Amazon Connect

Database Amazon Pinpoint

RDS Simple Email Service

DynamoDB Amazon Chime ↗

WorkMail

Analytics

Athena

EMR

CloudSearch

Elasticsearch Service

close

lambda

SQS

include s, SQS TP(S)



# Step 8

Amazon SNS ×

- Dashboard
- Topics**
- Subscriptions
- ▼ Mobile
  - Push notifications
  - Text messaging (SMS)

Amazon SNS > Topics

Topics (2) Edit Delete Publish message Create topic

	Name	ARN
<input type="radio"/>	SendTempNotification	arn:aws:sns:us-east-1:949814168619:SendTempNotification
<input type="radio"/>	thresholdtemperatures	arn:aws:sns:us-east-1:949814168619:thresholdtemperatures

## SendTempNotification

Edit Delete Publish message

**Details**

Name SendTempNotification	Display name SendTempNotification
ARN arn:aws:sns:us-east-1:949814168619:SendTempNotification	Topic owner 949814168619

**Subscriptions** | Access policy | Delivery retry policy (HTTP/S) | Delivery status logging | Encryption | Tags

Subscriptions (0) Edit Delete Request confirmation Confirm subscription Create subscription

ID	Endpoint	Status	Protocol
No subscriptions found You don't have any subscriptions to this topic. <span>Create subscription</span>			

## Step 9

# Create subscription

### Details

Topic ARN

arn:aws:sns:us-east-1:949814168619:Send ✕

Protocol


The type of endpoint to subscribe

Email ▼

Endpoint

An email address that can receive notifications from Amazon SNS.

ask.gourav7@gmail.com

 After your subscription is created, you must confirm it. [Info](#)

#### ► Subscription filter policy - *optional*

This policy filters the messages that a subscriber receives. [Info](#)

#### ► Redrive policy (dead-letter queue) - *optional*


Send undeliverable messages to a dead-letter queue. [Info](#)

Cancel

Create subscription

## Step 10

Check SNS confirmation Email in your mailbox and Confirm Subscription. [Sometime it may come to Gmail Promotion tab or SPAM folder]

AWS Notification - Subscription Confirmation  Inbox x



**SendTempNotification** <no-reply@sns.amazonaws.com>

to ask.gourav7 ▾

You have chosen to subscribe to the topic:

**arn:aws:sns:us-east-1:949814168619:SendTempNotification**

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):

[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)

## Step 11

If you are done, make sure you delete your Rule Positively without failure.

The screenshot shows the AWS IoT Rule Editor interface for a rule named "thresholdtemperature". The rule is currently "ENABLED". A dropdown menu is open under the "Actions" header, with the "Delete" option highlighted in red. The interface includes a sidebar with "Overview" and "Tags" sections. The main content area is divided into sections: "Description" (Notify Users when Temperature rises), "Rule query statement" (containing a SQL query: `SELECT Temperature FROM 'ESP32/DHT11' where Temperature > 27`), and "Actions" (containing one action: "Send a message as an SNS push notification").

**RULE**  
**thresholdtemperature**  
ENABLED

**Actions** ▾

- Disable
- Enable
- Delete**

**Overview**

Tags

**Description**

Notify Users when Temperature rises

**Rule query statement** Edit


The source of the messages you want to process with this rule.

```
SELECT Temperature FROM 'ESP32/DHT11' where Temperature > 27
```

Using SQL version 2016-03-23

**Actions**

Actions are what happens when a rule is triggered. [Learn more](#)

 Send a message as an SNS push notification Remove Edit ▶  
SendTempNotification

[Add action](#)

**Error action**