WHIZLABS

0 K

Home / AWS / Guided Lab / Create an S3 Bucket event to get SNS Email Notification on Object upload using Terraform

Create an S3 Bucket event to get SNS Email Notification on Object upload using Terraform

Level: **Intermediate**

Amazon S3 Amazon SNS Amazon Web Services Terraform

Required Points	10
Lab Duration	00:45:00
Average Start time	Less than a minute

Start Lab →

Need help?

How to use Hands on Lab

Troubleshooting Lab

FAQs

Submit Feedback | Share

Lab Overview

Cloud Architect, Cloud Developer, Cloud DevOps Engineer, Cloud Administrator

Storage, Serverless, Infrastructure

Lab Details

1. This lab guides you through the process of creating and subscribing to an SNS (Simple Notification Service) Topic. Additionally, you will create an S3 bucket and configure event notifications.You will use Terraform to create the required resources and then test their functionality using the AWS console.

2. Duration of the Lab: **45 minutes**
3. AWS Region: **US East (N. Virginia) us-east-1**

Introduction

What is SNS?

- SNS stands for Simple Notification Service.
- Provides a low-cost infrastructure for the mass delivery of messages, predominantly to mobile users.
- SNS acts as a single message bus that can message to a variety of devices and platforms.
- SNS uses the publish/subscribe model for push delivery of messages.
- SNS enables us to decouple microservices, distributed systems, and serverless applications using fully managed pub/sub.
- Publishers communicate asynchronously with subscribers by producing and sending a message to a topic, which is a logical access point and communication channel.
- Subscribers i.e., web servers, email addresses, SQS queues etc., consume or receive the message or notification over one of the supported protocols when they are subscribed to the topic.
- Recipients subscribe to one or more "topics" within SNS.
- Using SNS topics, the publisher systems can fan out messages to a large number of subscriber endpoints for parallel processing, including Amazon SQS queues, AWS Lambda functions, and HTTP/S webhooks.
- SNS is reliable in delivering messages with durability.
- SNS can help in automatically scale the workload.
- Using topic policies, you can keep messages private and secure.

What is S3?

- S3 stands for Simple Storage Service.
- It provides object storage through a web service interface.
- Each object is stored as a file with its metadata included and is given an ID number.
- Objects uploaded to S3 are stored in containers called “Buckets”, whose names are globally unique. They organize the Amazon S3 namespace at the highest level.
- Amazon S3 creates buckets in the region you specify.
- You can assign permissions to these buckets to provide or restrict data transactions.

- Applications use this ID number to access an object.
- Developers can access an object via a REST API.
- S3 supports upload of objects.
- It uses the same scalable storage infrastructure that Amazon.com uses to run its global e-commerce network.
- It's designed for storing online backup and archiving of data and applications on AWS.
- **Storage classes provided are:**

1. Standard
2. Standard_IA i.e., Standard Infrequent Access
3. Intelligent_Tiering
4. OneZone_IA
5. Glacier
6. Deep_Archive
7. RRS i.e., Reduced Redundancy Storage (Not recommended by AWS)

- Data access is provided through the S3 Console.
- Data stored can be either Public or Private based on user requirement.
- Data stored can be encrypted.
- We can define life-cycle policies which can help in automation of data transfer, retention and deletion.

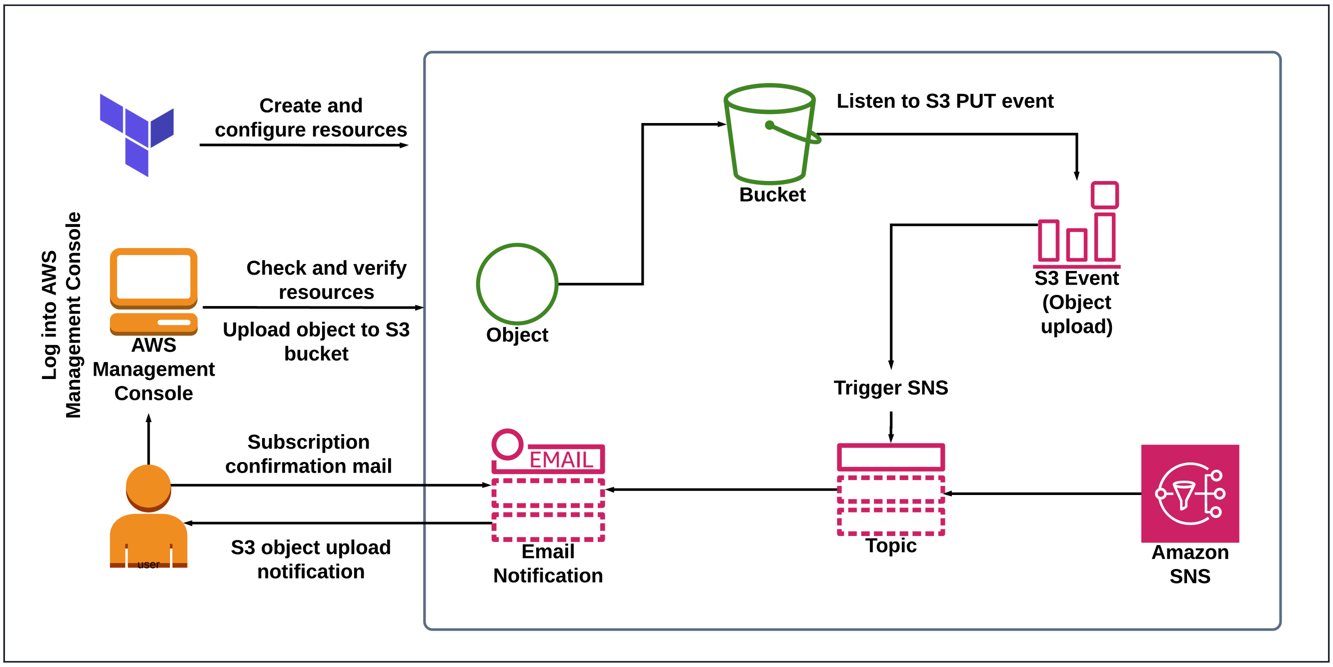
What is Terraform?

- It is an open-source IaaC (Infrastructure as a Code) software tool where you define and create resources using providers in the declarative configuration language example JSON.
- With Terraform, You can package and reuse the code in the form of modules.
- It supports a number of cloud infrastructure providers such as AWS, Azure, GCP, IBM Cloud, OCI, etc.
- Terraform has four major commands:
    - terraform init
    - terraform plan
    - terraform apply
    - terraform destroy

Prerequisites

- Install Terraform in your local machine using this official guide by Hashicorp.
  - To install Terraform using CLI, use this guide https://learn.hashicorp.com/tutorials/terraform/install-cli
  - To install Terraform by downloading, use this guide https://www.terraform.io/downloads.html
- Download and Install Visual Studio code editor using this guide https://code.visualstudio.com/download

Architecture Diagram


Log into AWS Management Console
Create and configure resources
Check and verify resources
Upload object to S3 bucket
AWS Management Console
Subscription confirmation mail
S3 object upload notification
user
Listen to S3 PUT event
Bucket
Object
S3 Event (Object upload)
Trigger SNS
EMAIL
Email Notification
Topic
Amazon SNS

Task Details

1. Sign in to AWS Management Console
2. Setup Visual Studio Code.
3. Create a Variables file.
4. Create SNS, S3 and its components in main.tf file
5. Create an Output file.
6. Confirm the installation of Terraform by checking the version.
7. Apply Terraform configurations.
8. Check the resources in AWS Console.
9. Upload an object into S3 bucket and test the SNS notification.
10. Validation of the lab.
11. Clean up AWS Resources.

Launching Lab Environment

1. To launch the lab environment, Click on the **Start Lab** button.
2. Please wait until the cloud environment is provisioned. It will take less than a minute to provision.
3. Once the Lab is started, you will be provided with **IAM user name**, **Password**, **Access Key**, and **Secret Access Key**.

Note : You can only start one lab at any given time

About Us Subscription Instructions and Guidelines FAQ’s Contact Us