WHIZLABS

Host a S3 Static Website using Terraform

Level: **Fundamental**

Amazon S3 Amazon Web Services Terraform

Required Points	10
Lab Duration	01:00:00
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

Storage, Serverless, Infrastructure

Lab Details

1. This lab walks you through the steps to create a Static website hosted in Amazon S3 Bucket using Terraform.

2. Duration: **60 minutes**
3. AWS Region: **US East (N. Virginia) us-east-1.**

Introduction

What is a Static Website?

- These are the most basic types of websites and are the easiest to create.
- A static web page is a web page that is delivered to the user's web browser exactly as stored.
- It holds fixed content, where each page is coded in HTML and displays the same information to every visitor.
- No web programming or database design is required when working with them.
- They are a safe bet when it comes to security since we do not have any interaction with databases or plugins.
- They are reliable, i.e., if an attack happens on the server, a redirection to the nearest safest node happens.
- Static websites are very fast because there is no true backend to fetch information from.
- Hosting the website is cheap due to the non-existence of any other components.
- Scaling the website is easy and can be done by just increasing the bandwidth.

What is Amazon 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.

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

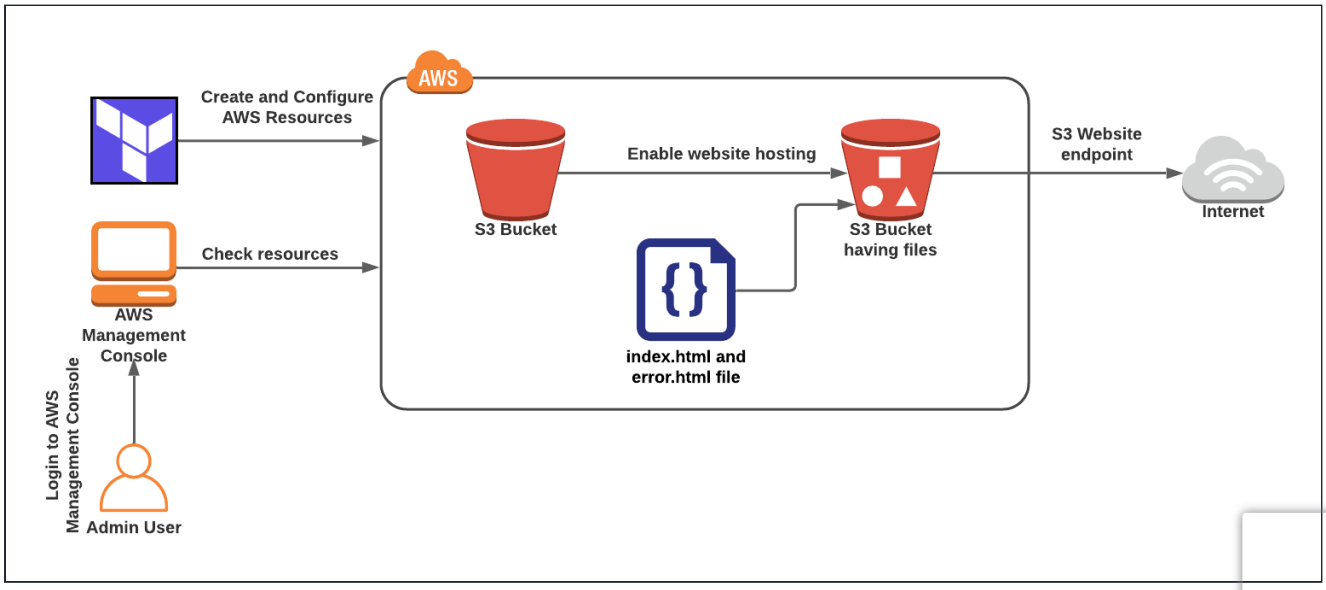
JSON.

- With Terraform, You can package and reuse the code in 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

Prerequisite

- 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



Task Details

1. Sign in to AWS Management Console
2. Setup Visual Studio Code
3. Create a variables file
4. Create an S3 Bucket and its components in main.tf file
5. Confirm the installation of Terraform by checking the version
6. Apply terraform configuration
7. Check the resources in AWS Console
8. Validation of the lab
9. Delete 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 username**, **Password**, **Access Key**, and **Secret Access Key**.

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