WHIZLABS

0

K

Home / AWS / Guided Lab / Access S3 bucket from EC2 Instance using Terraform

Access S3 bucket from EC2 Instance using Terraform

Level: **Intermediate**

Amazon EC2 Amazon S3 Identity And Access Management Amazon Web Services Terraform

Required Points	10
Lab Duration	01:30:00
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, Security, Compute, Infrastructure

Lab Details

1. This lab walks you through the steps to access the S3 bucket and its objects from EC2 Instance using IAM Role.

Privacy - Terms

2. Duration: **90 minutes**
3. AWS Region: **US East (N. Virginia) us-east-1.**

Introduction

What is EC2

- AWS defines it as Elastic Compute Cloud.
- It's a virtual environment where "you rent" to have your environment created, without purchasing.
- Amazon refers to these virtual machines as instances.
- Preconfigured templates can be used to launch instances. These templates are referred to as images. Amazon provides these images in the form of AMIs (Amazon Machine Images).
- Allows you to install custom applications and services.
- Scaling of infrastructure, i.e., up or down, is easy based on the demand you face.
- AWS provides multiple configurations of CPU, memory, storage, etc., through which you can pick the flavor that's required for your environment.

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?

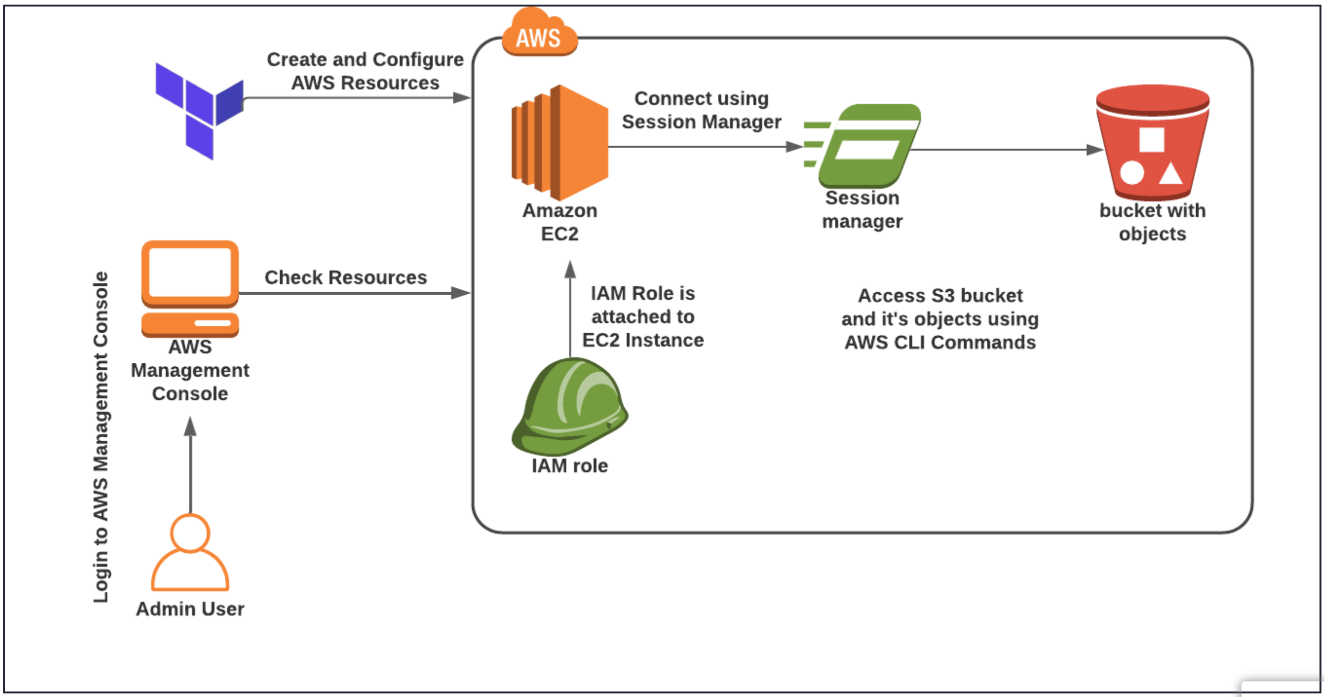
- It is an open-source IaaC (Infrastructure as a code) software tool where you define and create resources using providers in the declarative configuration language example JSON.
- With Terraform, You can package and reuse the code in form of modules.

- 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


AWS
Create and Configure AWS Resources
Connect using Session Manager
Amazon EC2
Session manager
bucket with objects
Check Resources
IAM Role is attached to EC2 Instance
Access S3 bucket and it's objects using AWS CLI Commands
AWS Management Console
IAM role
Login to AWS Management Console
Admin User

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 user name**, **Password**, **Access Key**, and **Secret Access Key**.

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