Optimum-Assessment

SFTP Solution for Data Uploads to AWS S3 Buckets

Deepak S D

**Overview**

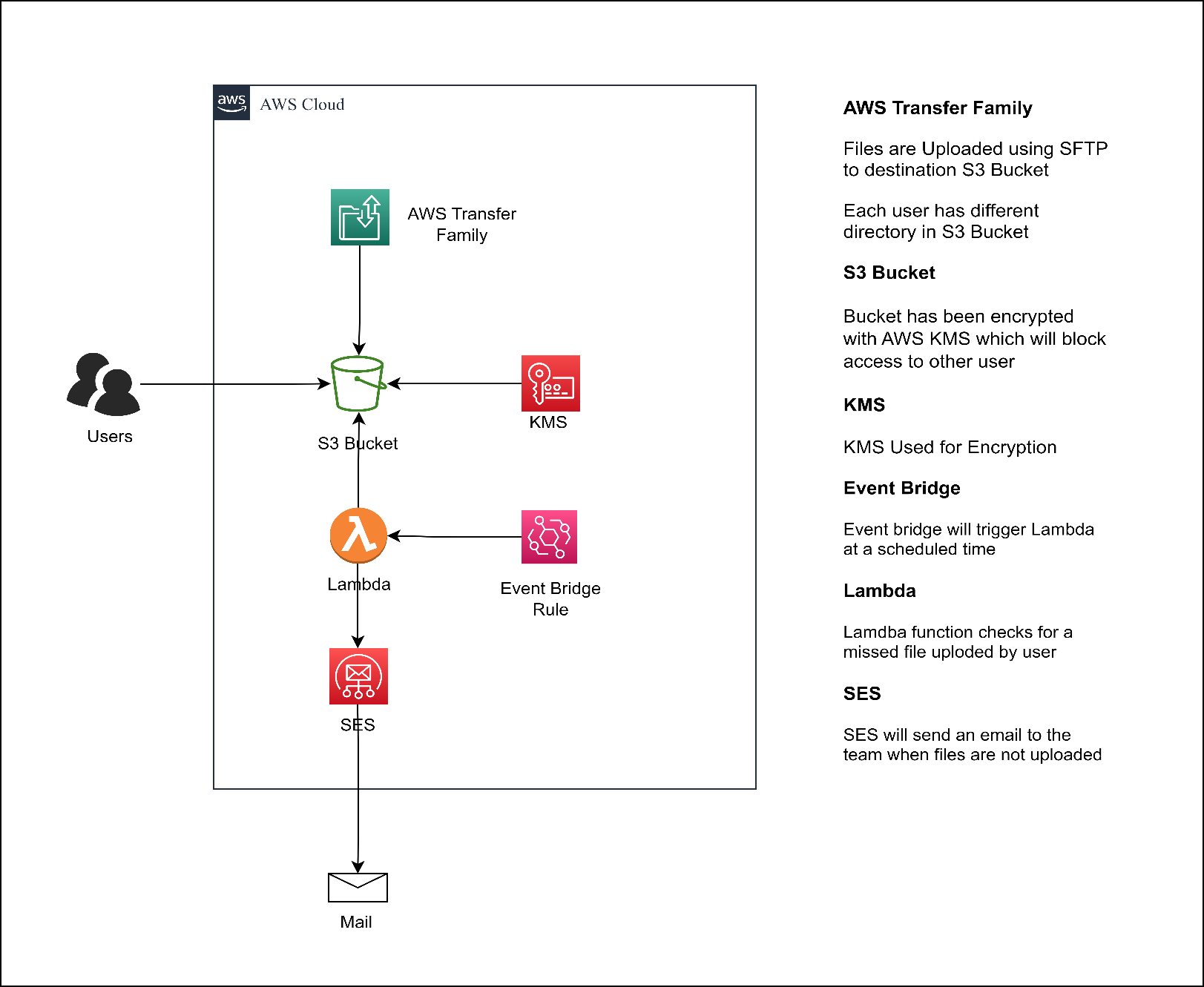
Problem:

As a global financial institution with operations, we are spanning multiple AWS regions. As a Cloud Engineer, you are responsible to help the organization by presenting, justifying, and implementing simple solutions for simple problems and manageable/reproducible solutions for complex challenges. We work with multiple marketing agencies and companies all around the world and now as part of data lake project, we want to create a solution to those agencies to upload data to our S3 buckets. Being not very tech savvy, they ask for SFTP connection for daily or weekly uploads.

Solution:

This document proposes a solution that utilizes AWS Transfer Family to establish an SFTP server, facilitating secure data transfer to S3 buckets for agencies. In addition, we will furnish detailed guidelines for implementing this solution, along with recommendations for adopting best practices to guarantee secure and streamlined file transfer.

# **Architecture Diagram**



**Prerequisites**

* **AWS Account:**

You will need an AWS account to create and manage the AWS resources required for the solution.

* **AWS CLI:**

The AWS Command Line Interface (CLI) is a tool that enables you to interact with AWS services from the command line. You will need to have AWS CLI installed on your local machine to run the Terraform code.

* **Terraform:**

Terraform is an IAC tool that allows you to create, manage, and version your infrastructure in a safe and repeatable way. You will need to have Terraform installed on your local machine to run the Terraform code.

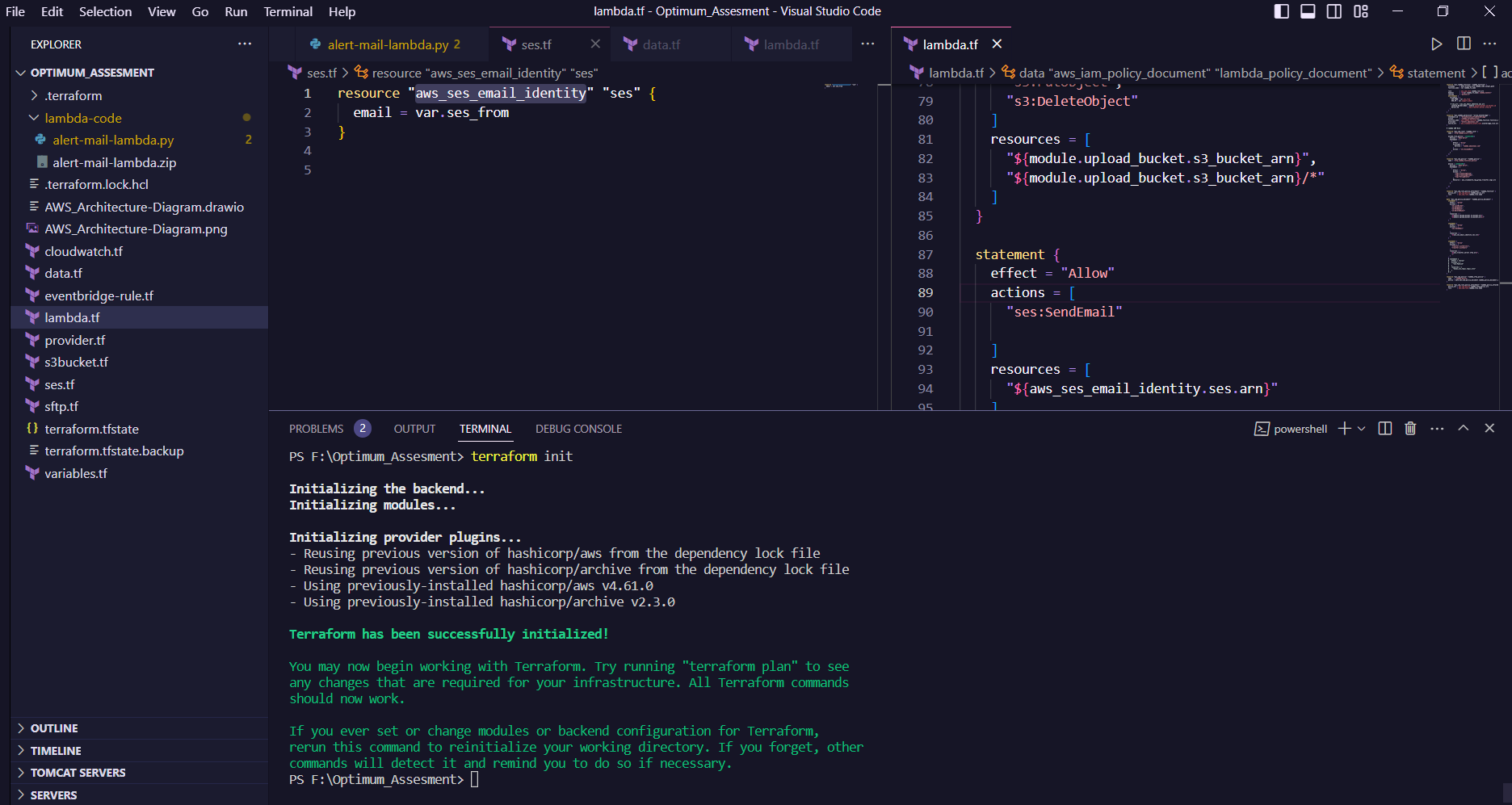
* **Access keys for an IAM user:**

You will need to have an IAM user with administrative privileges and access keys to authenticate with the AWS CLI and create and manage AWS resources using Terraform.

# **Implementation Steps to Be Followed:**

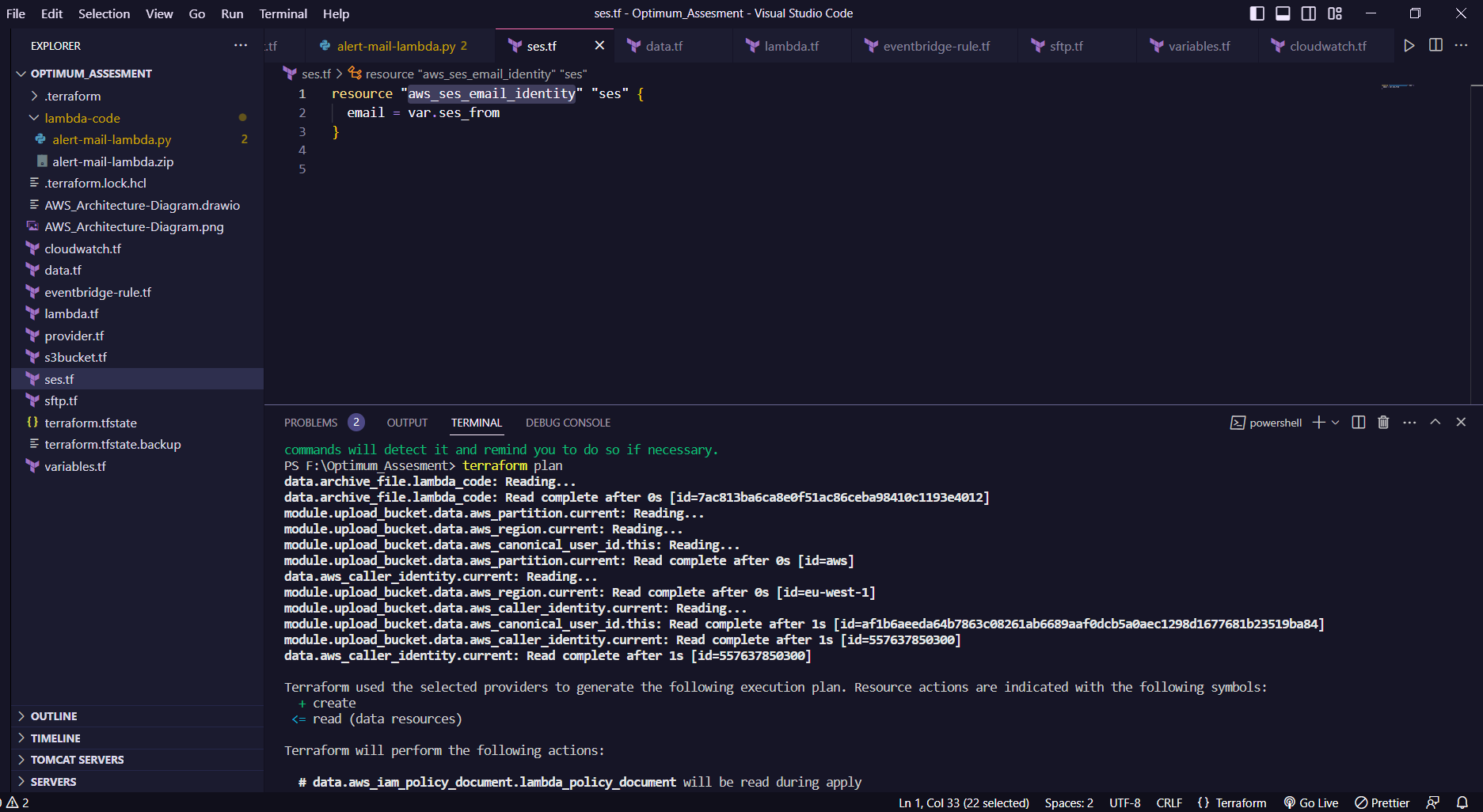
1. To obtain the code for the AWS solution provided for the stated problem, you can clone the repository onto your local machine.
2. In order to authenticate with your AWS account, terraform needs your AWS access and secret keys. You have two options to provide these credentials: you can set them as environment variables, or you can use an AWS CLI profile.
3. The 'aws configure' command can be used to establish a connection between AWS and Terraform. This command allows you to set up your AWS access and secret keys, which Terraform requires to interact with your AWS account.
4. To customize the Terraform script based on your requirements, you will need to modify the values in the 'variables.tf' files. For example, you may need to change the bucket name or create new users in the SFTP server.
5. Aws configure is the command to establish connection between aws and terraform.
6. **Terraform init**

To ensure that the required providers and modules are available for use, you must first run the **command 'terraform init'** in the terminal to initialize the working directory. This command downloads the necessary components and prepares the environment for use with the Terraform script.



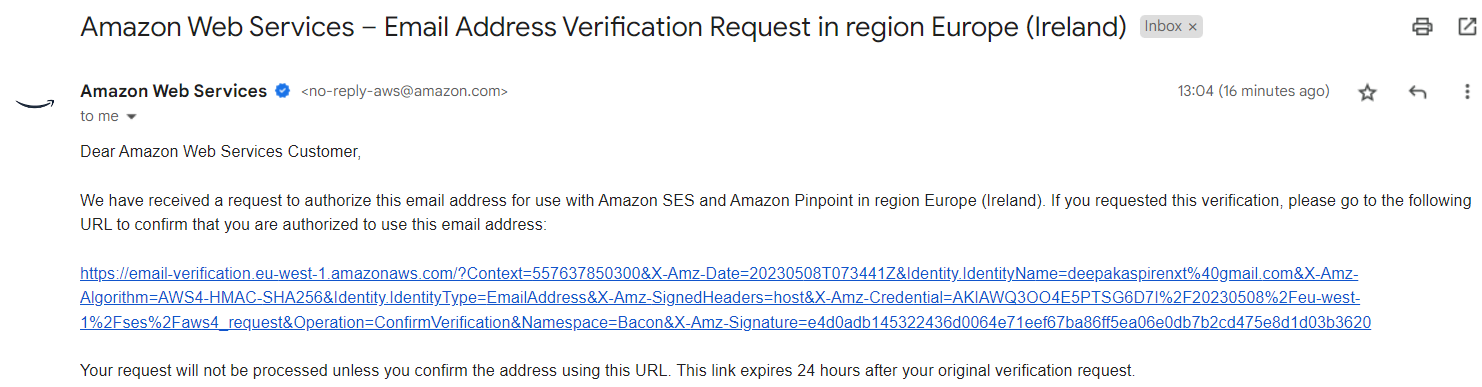
1. Terraform Plan

To preview the modifications that will be applied to your infrastructure, use the **'terraform plan'** command. This will generate a report detailing the changes Terraform will make, including any additions, modifications, or deletions to your resources.

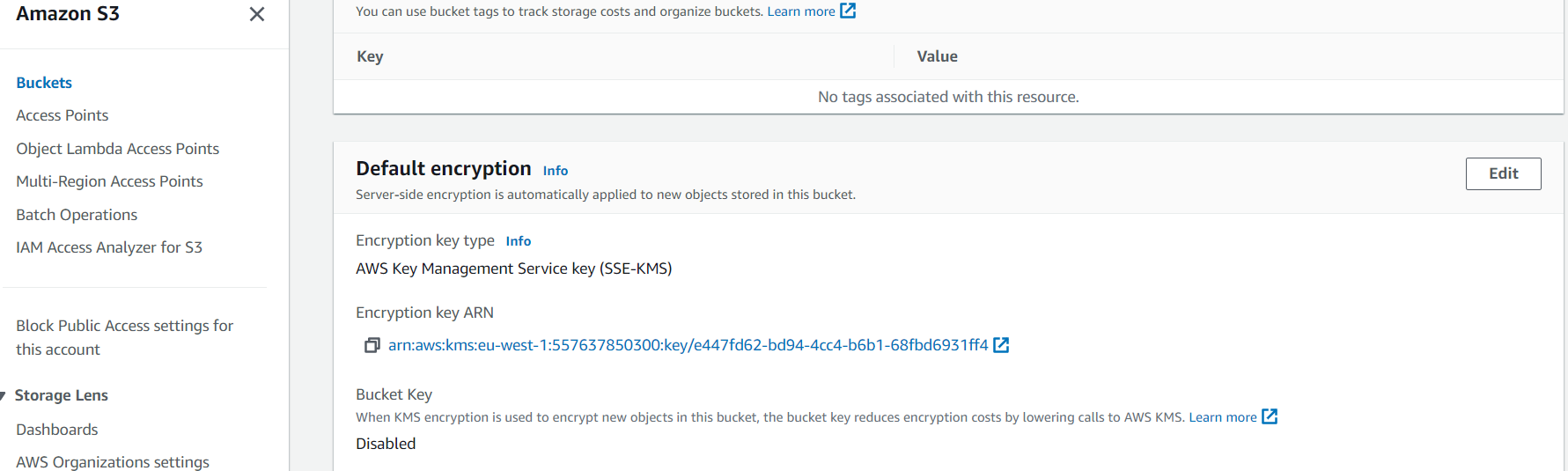


1. Terraform Apply

To create the resources in your AWS account and apply the changes you've made, use the **'terraform apply'** command. This command will execute the changes outlined in the Terraform script and create the corresponding infrastructure in your AWS account.

1. Upon completion of the 'terraform apply' command, you will receive an email containing a subscription link. Please follow the link and accept the subscription from your side to complete the process.

**Security**



* The S3 bucket is secured with AWS KMS encryption, ensuring that only authorized agencies can access it. This measure restricts access to the bucket from other services, thereby enhancing its security.
* The IAM has been configured to provide minimal permissions to each agency, limiting their access only to their respective directories. This measure ensures that agencies can only upload files to their own directories and prevents them from uploading files to other directories.

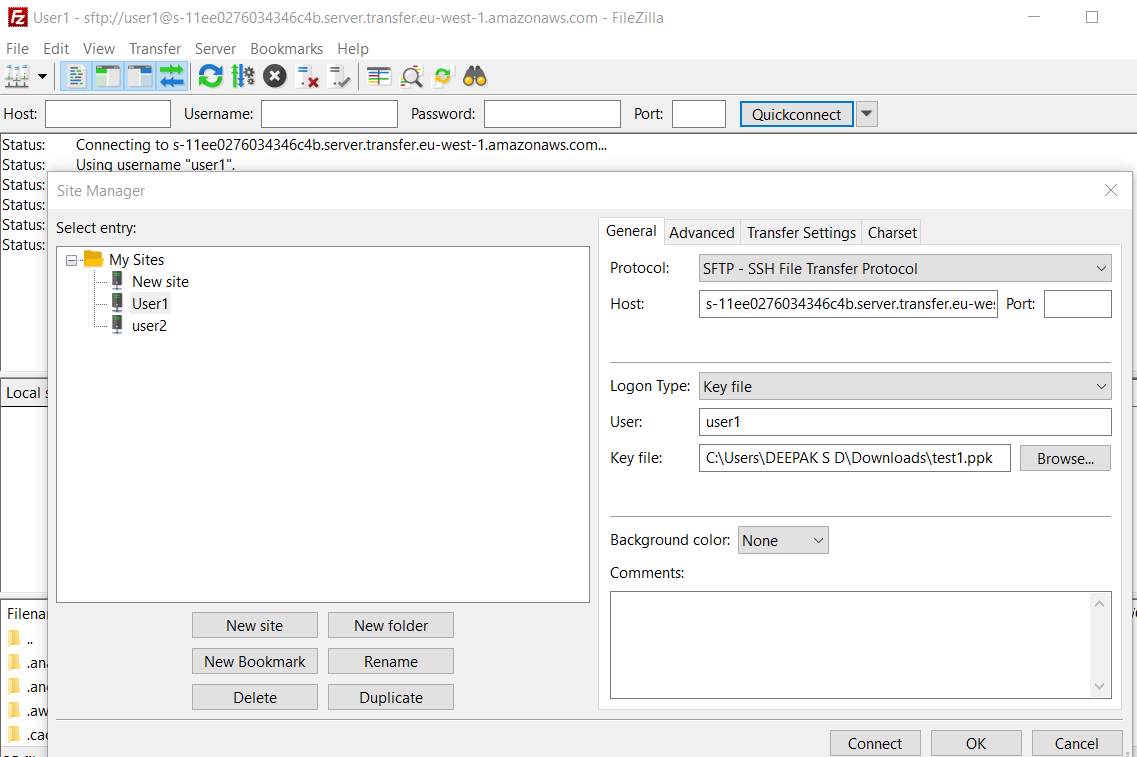
**Alert and Monitoring:**

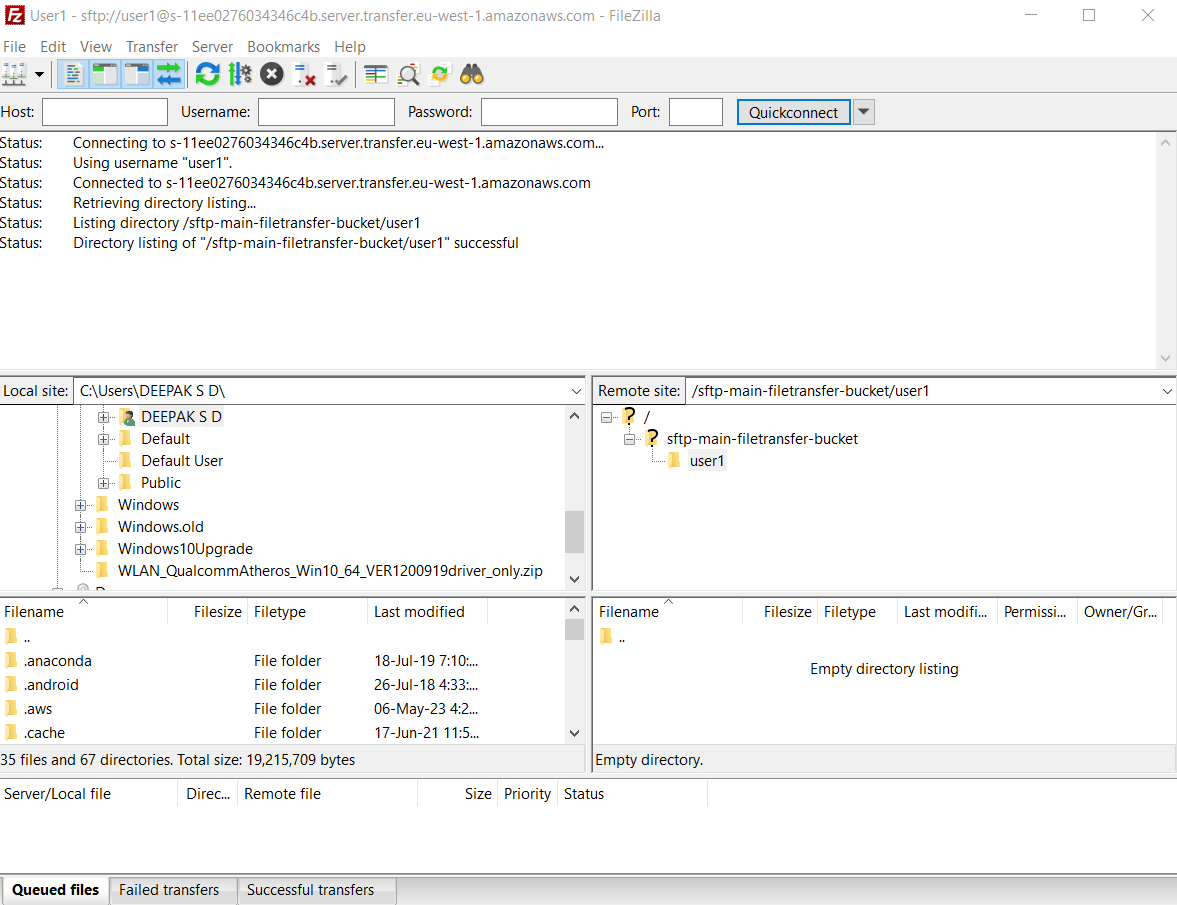
1. AWS Transfer Family has been configured with CloudWatch logs to enable monitoring of its activities. This measure enhances visibility into the transfer of files and provides additional insights for analysis.
2. An Event Bridge rule has been set up to trigger a Lambda function at 7 pm Ireland time every day. The function investigates the S3 bucket for any data files uploaded for the current day. If an agency fails to upload any files for the day, the Lambda function sends an alert to the SRE team via AWS SES. This measure helps ensure that all agencies are meeting their daily data upload requirements and allows the SRE team to take corrective action as needed.

**Final Step Include:**

To access the organization's system, the client logs in via an FTP tool using the credentials provided to them. This allows them to securely transfer data and files to the organization's designated SFTP server.

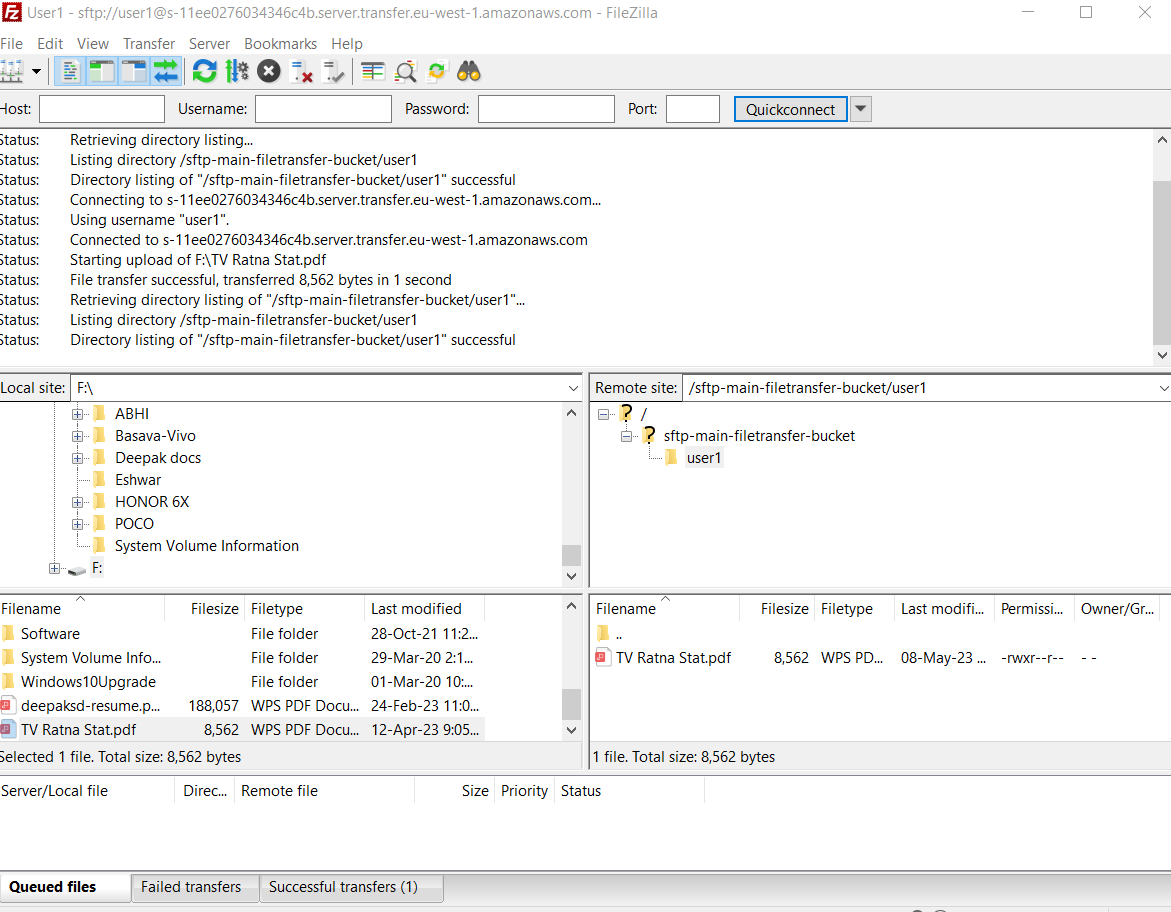
1. Before logging in, add a different SSH public key file to each user.
2. Copy the SFTP endpoint.
3. Go to File > Site Manager and select the SFTP protocol.
4. Paste the copied endpoint into the appropriate field.
5. Select the login key type and upload the SSH public key for the user.
6. Enter the username and connect to the server.

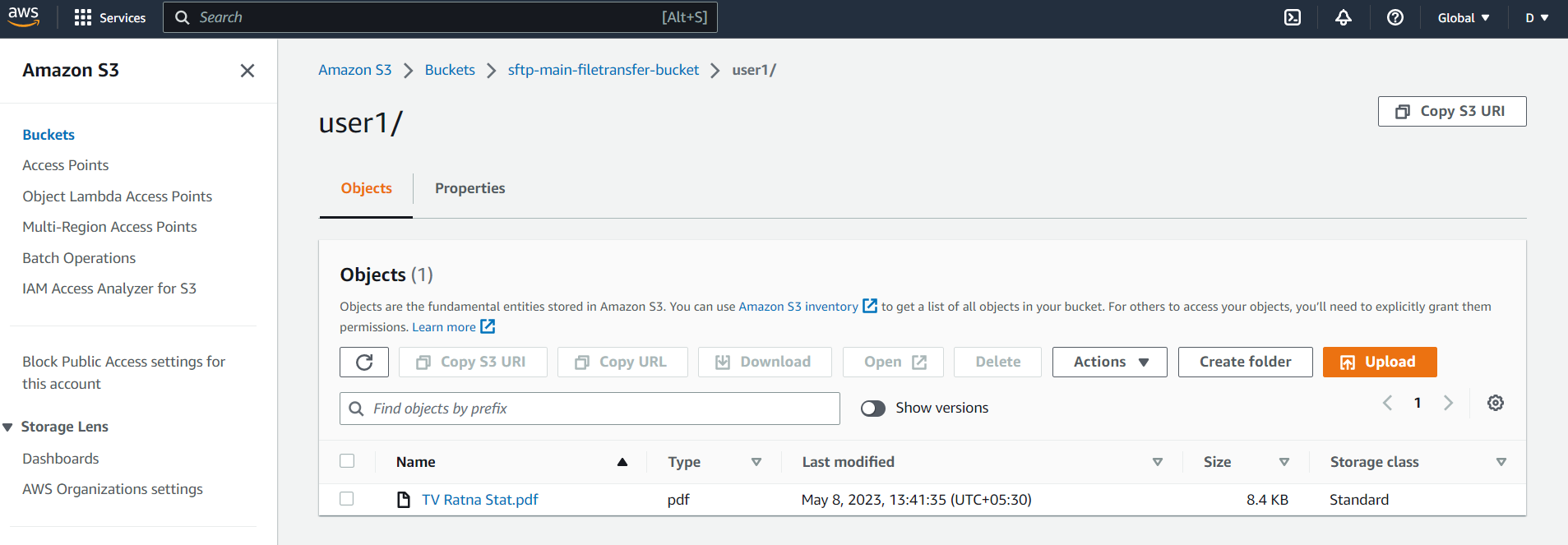




Upon successful connection, each agency will be automatically directed to their designated directory. From there, they can begin uploading their files securely to their respective directories using the provided credentials.

Below image shows the uploaded file.





THANK YOU