

# DevOps Kitchen Workshop

## S3 Bucket - Lifecycle

### *#day\_three - File transition to storage classes*

ForgTech company wanna test your ability to type down a clean code by Deploying the structure of resources. This will help you to build a good reputation.

ForgTech is required to build an S3 Bucket and configure file transition to multiple storage classes for archiving and low-cost purposes.

The S3 requires to have directories as follows:

1. /log
2. /outgoing
3. /incoming

and file transition must match the following rules:

- Transition all files under /log to infrequent access (i.e. Standard-IA) 30 consecutive days after creation time.
- Transition all files under /log to Archive access (i.e. Glacier) 90 consecutive days after creation time.
- Transition all files under /log to Deep Archive access (i.e. Glacier Deep Archive) 180 consecutive days after creation time.
- Remove all files under /log 365 consecutive days after creation time.
- Transition all files under /outgoing with tag "notDeepArchive" to infrequent access (i.e. Standard-IA) 30 consecutive days after creation time.
- Transition all files under /outgoing to Archive access (i.e. Glacier) with the tag "notDeepArchive" 90 consecutive days after creation time.
- Transition all files under /incoming with size between 1MB to 1G to infrequent access (i.e. Standard-IA) 30 consecutive days after creation time.
- Transition all files under /incoming with size between 1MB to 1G to Archive access (i.e. Glacier) 90 consecutive days after creation time.

Use IaC Terraform to build all resources and consider the below requirements specifications. and ensuring that you can destroy the S3 (*i.e. using terraform destroy command*) even if the bucket is not empty.

1. Resources must be created at the us-east-1 region.
2. Resources must have tags as below:
  - a. Key: "Environment", Value: "terraformChamps"
  - b. Key: "Owner", Value: "<Your\_first\_name>"
3. Preferd to use variables

### Code Sample:

- [Day Two sample/Blog](#)

- [Day Two sample/video](#)
- [Day Two sample/Code](#)
- [Day Two sample/Storage Classes](#)

FrogTech company has the challenge of automating a manual process, They are using S3 as a public storage for external/internal parties authenticating using IAM credentials.

There are two main S3 buckets 1. `frogtech-us-external1` and 2. `frogtech-us-internal1`, The Challenge lies in the manual actions that operators do daily in order to move files from one S3 to another.

You're requested to automate this process using Lambda function, utilizing the native SDK of AWS (*i.e. Python boto3 SDK,*) FrogTech engineers have no idea about lambda. Therefore you as an expert, should provide them with a document containing the **basics** of lambda, besides explaining:

1. Function event.
2. Function context.
3. Function environment variables.
4. layers.
5. Differences between Synchronous and Asynchronous.

as well as provide a diagram explaining the entire process of the created resources; Use IaC Terraform to build all resources and consider the below requirements specifications.

1. Resources must be created at the us-east-1 region.
2. Resources must have tags as below:
  - a. Key: "Environment", Value: "terraformChamps"
  - b. Key: "Owner", Value: "<Your\_first\_name>"
3. Preferd to use variables.

## References:

1. [📖 What is AWS Lambda? - AWS Lambda](#)
2. [💡 \[AWS\] Lambda - S3 Trigger + Terraform Project 10](#)
3. [💡 \[AWS\] Lambda Concepts Essentials](#)