Terraform IAM Lab Debugging – TestUser Read-Only Access

■ Debugging Walkthrough

In this stage of the IAM Lab, we verified whether TestUser (a user added to the Developers group with the AmazonS3ReadOnlyAccess policy attached) correctly inherits read-only permissions on the S3 bucket created in Terraform.

Steps we ran:

- Listed S3 buckets and objects with TestUser → Success.
- Downloaded (GET) object `seed.txt` with TestUser → Success.
- Tried uploading (PUT) a new object as TestUser → Failed with AccessDenied (expected).
- Tried deleting (DELETE) an object as TestUser → Failed with AccessDenied (expected).

■■ Errors observed:

When TestUser attempted `s3:PutObject` and `s3:DeleteObject`, AWS returned AccessDenied. This is expected behavior because the TestUser only has AmazonS3ReadOnlyAccess, which allows read-only operations (listing and getting objects) but not writing or deleting.

■ How we confirmed it worked:

The AccessDenied errors are not problems — they prove the policy is being enforced. We confirmed the IAM policy inheritance by showing that TestUser can only perform read actions and is blocked from write/delete actions.

■ Lessons Learned

- Group policies in IAM are inherited by users automatically.
- AmazonS3ReadOnlyAccess allows listing and downloading objects but blocks uploads and deletes.
- AccessDenied messages can be a positive confirmation that least-privilege is working as intended.
- Testing with `aws s3` CLI and switching profiles (`--profile testuser` vs `--profile admin`) is the best way to validate IAM behavior.

■ Is the Lab Complete?

Yes. The lab's objective was to: - Create a group (Developers) - Attach policy AmazonS3ReadOnlyAccess - Add TestUser to the group - Verify TestUser inherits read-only access All

steps have been verified successfully. You can double-check by running: aws sts get-caller-identity --profile testuser aws s3 ls s3://$(terraform output -raw bucket_name) --profile testuser If TestUser can only list/download but not upload/delete, the lab is complete.