DevOps Learning Journal - Day 3 (Terraform & AWS)

Key Activities

- Created IAM User with custom policy via Terraform.
- Created S3 Bucket with tags and bucket policy using modular Terraform setup.
- Setup EC2 with Apache using UserData.
- SSH into EC2 instance using `.pem` key after applying correct permissions.
- Explored difference between `user_data` and `remote-exec` provisioners.
- Accessed HTTP server from browser after successful EC2 + Apache setup.

Common Issues & Troubleshooting

Issue: `templatefile` path error

Error: Invalid value for "path" parameter

Fix: Used correct relative path: templatefile("${path.module}/install_httpd.sh", {})

Issue: Terraform prompting for input despite terraform.tfvars being present

Cause: Incorrect or missing variables.tf or wrong variable references.

Fix: Verified inputs.tf matches all variables used in main.tf and ensured correct var. references.

Issue: Bucket Policy AccessDenied (403)

Cause: S3 Block Public Access was ON by default.

Fix: Added block_public_acls = false, block_public_policy = false in aws_s3_bucket_public_access_block.

Question: Can I rename main.tf to s3_main.tf?

Answer: Yes, Terraform reads all .tf files in the root module, order doesn't matter.

Question: Will adding S3 module affect existing EC2?

Answer: No, Terraform compares the current state and applies only the delta.

Question: How to split modules cleanly?

Answer: Create new files like s3.tf, iam.tf, etc., in root. All .tf files are loaded.

Question: Do DevOps engineers memorize policies?

Answer: No. They understand the structure and use AWS documentation or policy generators.

Question: ec2:Describe* what does it mean?

Answer: Grants read-only access like describing instances, AMIs, volumes, etc.

Issue: Lost PEM file

Fix: Deleted EC2 and created a new one with a new key pair.

Issue: S3 Bucket Policy failed due to Resource: "*"

Clarification: * here means policy allows access to all matching resources of the specified action type.

SSH Issue

Fix: Applied chmod 400 <pem_file> to secure the key file and gained access.

Final Output

- Apache Web Page accessed via EC2 Public IP

- IAM User + S3 Bucket + EC2 deployed using modular Terraform

Next Steps (Day 4 Preview)

- Create AWS Billing Alert from Console.

- Begin Ansible Hands-on Labs.

- Increase Terraform + Ansible combo use cases.