AWS Environment Limitations

Observed Limitations

During deployment, several AWS services could not be fully provisioned due to restrictive IAM policies enforced by the AWS Voclabs student environment. The following actions were explicitly denied by the sandbox account:

- IAM role creation (iam:CreateRole)
- IAM user creation (iam:CreateUser)
- RDS database instance creation (rds:CreateDBInstance)
- WAF Web ACL creation (wafv2:CreateWebACL)
- ACM certificate import (acm:ImportCertificate)
- S3 Object Lock configuration (s3:GetBucketObjectLockConfiguration)

1. IAM Roles and Policies

What Was Implemented in Code

- Trust policy allowing EC2 service to assume the role
- Attachment of AWS managed policies (e.g., CloudWatch logging, SSM access)
- Role designed following the principle of least privilege

How It Is Supposed to Work:

- Terraform creates the IAM role
- EC2 instances assume the role at runtime
- No long-term credentials are stored on instances
- Access to AWS services is controlled via attached policies

Restrictions Encountered:

- IAM roles could not be created
- IAM users could not be created
- Terraform failed at the IAM resource stage with 403 AccessDenied errors

2. RDS Database Instance

What Was Implemented in Code

- An RDS instance in private database subnets
- Storage encryption enabled
- Security group allowing access only from application security group
- No public accessibility

How It Is Supposed to Work

- RDS instance is created inside private subnets
- Database is unreachable from the internet
- Only application instances can connect via security group rules
- Encryption protects data at rest

Restriction Encountered
The sandbox account denied: *rds:CreateDBInstance*

3. AWS WAF Web ACL

What Was Implemented in Code:

- A WAFv2 Web ACL
- AWS Managed Rules (Common Rule Set)
- Association with the Application Load Balancer

How It Is Supposed to Work

- WAF inspects incoming HTTP/HTTPS requests
- Blocks common attacks such as SQL injection and XSS
- Provides CloudWatch metrics for visibility
- Acts as an application-layer perimeter defense

Restriction Encountered

- WAF resource definition was valid
- Creation failed at apply stage due to permission restrictions

4. ACM Certificate Import

What Was Implemented in Code

- TLS private key generation
- Self-signed certificate creation
- Import of certificate into AWS Certificate Manager (ACM)

How It Is Supposed to Work

- Certificate is imported into ACM
- Attached to HTTPS listener on ALB
- Enables encrypted HTTPS traffic

Restriction Encountered

- HTTPS listener could not be fully configured
- HTTP listener remained active

5. S3 Object Lock and Advanced Security Features

What Was Implemented in Code

- Query or configure Object Lock settings
- Enable secure storage and logging buckets
- Apply server-side encryption (SSE-S3)

How It Is Supposed to Work

- Object Lock prevents deletion or tampering of audit logs
- Enhances compliance and forensic integrity
- Encryption ensures data confidentiality

Restriction Encountered

- The sandbox explicitly denied: *s3:GetBucketObjectLockConfiguration*
- S3 buckets were created successfully
- Server-side encryption (SSE-S3) was enabled and visible in AWS Console

Mitigation and Validation

Despite these restrictions:

- All blocked resources were correctly defined in Terraform
- Terraform validation succeeded (terraform validate)
- Network infrastructure deployed successfully
- Security groups and routing enforced isolation
- Port scanning verified that:
    - Only intended ports were exposed
    - Databases and private subnets were unreachable externally

Evidences

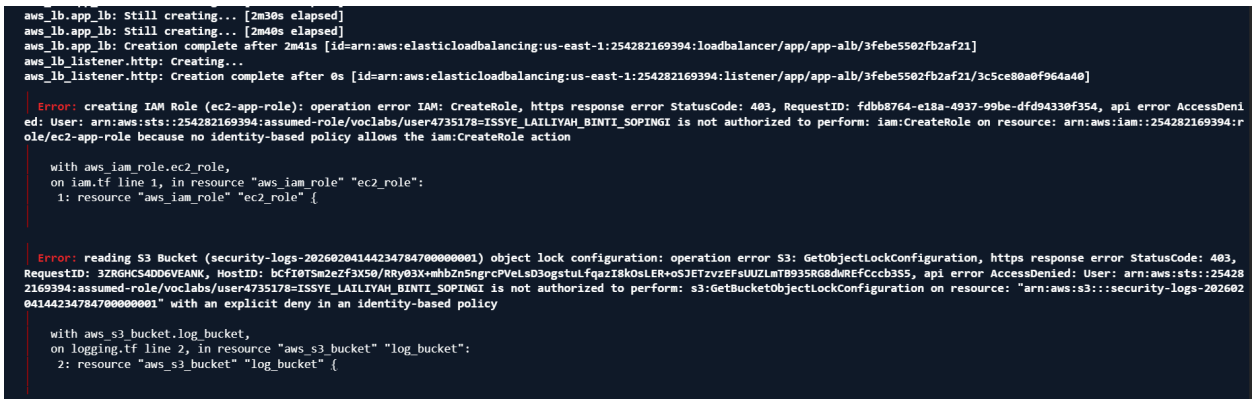

Figure 1: Terraform execution log showing successful Application Load Balancer (ALB) creation, followed by a permission denial when attempting to create an IAM role (iam:CreateRole). The error indicates that the AWS Academy student account does not allow IAM role creation due to restricted identity-based policies.

Figure 2: Terraform error output demonstrating access denial for S3 Object Lock configuration (s3:GetBucketObjectLockConfiguration) and Amazon RDS instance creation (rds:CreateDBInstance). These failures are caused by explicit policy restrictions in the AWS Academy student environment, despite the Terraform configuration being syntactically valid.

Figure 3: Terraform execution result showing access denial for importing an ACM certificate (acm:ImportCertificate) and creating an AWS WAFv2 Web ACL (wafv2:CreateWebACL). The errors confirm that advanced security services are restricted under the AWS Academy lab account and cannot be provisioned programmatically.