🔐 Lambda Authorizer IAM Permissions

AWS Energy Data Insights Platform | Custom JWT Authorizer

📋 Overview

Service Role: `CustomAuthorizerRole`

Function Name: `custom-authorizer`

Purpose: Validates JWT tokens from Cognito User Pool for API Gateway HTTP API requests

Timeout: 30 seconds | **Memory:** 256 MB

🔑 Required IAM Permissions

`cognito-idp:DescribeUserPool` REQUIRED

Resource: `arn:aws:cognito-idp:us-east-1:*:userpool/us-east-1_*`

Retrieve user pool configuration for JWT validation. Required to get JWKS (JSON Web Key Set) endpoint for token signature verification.

`cognito-idp:DescribeUserPoolClient` REQUIRED

Resource: `arn:aws:cognito-idp:us-east-1:*:userpool/us-east-1_*`

Get client configuration for validating token audience (aud claim). Ensures tokens are issued for the correct application.

`logs:CreateLogGroup` REQUIRED

Resource: `arn:aws:logs:us-east-1:*:log-group:/aws/lambda/authorizer:*`

Create CloudWatch log group for Lambda function logs. Standard Lambda permission for logging.

`logs:CreateLogStream` REQUIRED

Resource: `arn:aws:logs:us-east-1:*:log-group:/aws/lambda/authorizer:*`

Create log streams within the log group. Required for writing logs to CloudWatch.

`logs:PutLogEvents` REQUIRED

Resource: `arn:aws:logs:us-east-1:*:log-group:/aws/lambda/authorizer:*`

Write log events to CloudWatch. Essential for debugging and monitoring authorization failures.

📄 Complete IAM Policy JSON

```
{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "cognito-idp:DescribeUserPool", "cognito-idp:DescribeUserPoolClient" ], "Resource": "arn:aws:cognito-idp:us-east-1:*:userpool/us-east-1_*" }, { "Effect": "Allow", "Action": [ "logs:CreateLogGroup", "logs:CreateLogStream", "logs:PutLogEvents" ], "Resource": "arn:aws:logs:us-east-1:*:log-group:/aws/lambda/authorizer:*" } ] }
```

💡 **Security Best Practice:** This role follows the principle of least privilege. It only has permissions to read Cognito configuration (not modify) and write logs. No permissions to access user data or other AWS resources.

🔄 Authorization Flow

1. API Gateway receives request with JWT token in Authorization header
2. Invokes Lambda Authorizer with token
3. Authorizer calls `DescribeUserPool` to get JWKS endpoint
4. Downloads public keys from JWKS endpoint
5. Verifies JWT signature using public key
6. Validates token expiration, issuer, and audience
7. Returns IAM policy allowing or denying access
8. API Gateway caches authorization decision for 5 minutes