🔧 Tool Lambdas IAM Permissions

AWS Energy Data Insights Platform | Python-Based Analysis Tools

📋 Overview

Service Role: `RenewableToolsRole`

Function Names: `renewable-tools-*` (terrain, layout, simulation, report, windrose)

Purpose: Execute domain-specific analysis using Python scientific libraries (pandas, geopandas, py-wake, matplotlib)

Timeout: 300 seconds | **Memory:** 2048 MB

🛠️ Tool Lambda Functions

🗺️ Terrain Analysis

Fetches OSM data, analyzes topography, identifies constraints, generates interactive map visualization

📐 Layout Optimization

Optimizes turbine placement using wind data and terrain constraints, generates GeoJSON layout

💨 Wake Simulation

Simulates wake effects using py-wake, calculates energy production and losses

📊 Report Generation

Aggregates all analysis results into comprehensive PDF report with charts and tables

🌹 Wind Rose

Generates wind rose visualization from NREL wind data showing directional distribution

📦 S3 Permissions

`s3:PutObject` REQUIRED

Resource: `arn:aws:s3:::storage-bucket/renewable-projects/*`

Store generated visualizations (HTML maps, PNG charts, PDF reports) and analysis results (JSON data). This is the primary output mechanism for tool Lambdas.

s3:GetObject REQUIRED

Resource: arn:aws:s3:::storage-bucket/renewable-projects/*

Read artifacts from previous analysis steps. For example, layout optimization reads terrain analysis results, wake simulation reads layout data.

CloudWatch Logs Permissions

logs:CreateLogGroup REQUIRED

Resource: arn:aws:logs:*:*:log-group:/aws/lambda/renewable-tools-*:*

Standard Lambda logging permission. Each tool Lambda has its own log group.

logs:CreateLogStream REQUIRED

Resource: arn:aws:logs:*:*:log-group:/aws/lambda/renewable-tools-*:*

Create log streams for each tool invocation. Essential for debugging analysis failures.

logs:PutLogEvents REQUIRED

Resource: arn:aws:logs:*:*:log-group:/aws/lambda/renewable-tools-*:*

Write analysis logs including data processing steps, calculation results, and error details. Critical for troubleshooting scientific computations.

Complete IAM Policy JSON

```
{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "s3:PutObject", "s3:GetObject" ], "Resource": "arn:aws:s3:::storage-bucket/renewable-projects/*" }, { "Effect": "Allow", "Action": [ "logs:CreateLogGroup", "logs:CreateLogStream", "logs:PutLogEvents" ], "Resource": "arn:aws:logs:*:*:log-group:/aws/lambda/renewable-tools-*:*" } ] }
```

Minimal Permissions: Tool Lambdas have the most restrictive permissions in the system. They only need S3 access for artifact storage and standard CloudWatch logging. They don't access DynamoDB, invoke other Lambdas, or call external APIs (except public OSM/NREL endpoints).

Tool Lambda Workflow

1. **Invocation:** Orchestrator invokes tool Lambda with analysis parameters

2. **Data Fetching:** Tool fetches external data (OSM, NREL) or reads from S3
3. **Processing:** Executes scientific computations using Python libraries
4. **Visualization:** Generates charts, maps, or reports using matplotlib/folium
5. **Storage:** Stores artifacts in S3 under project directory
6. **Response:** Returns artifact metadata to orchestrator

📦 Python Dependencies

Tool Lambdas use Lambda Layers for Python dependencies:

- **Scientific:** numpy, pandas, scipy, geopandas
- **Visualization:** matplotlib, seaborn, folium, plotly
- **Wind Analysis:** py-wake (wake modeling)
- **Geospatial:** shapely, fiona, pyproj
- **Data Fetching:** requests, osmnx

🎯 Artifact Types Generated

- **Terrain Analysis:** Interactive HTML map with OSM features (151 features typical)
- **Layout Optimization:** GeoJSON with turbine coordinates and metadata
- **Wake Simulation:** JSON with energy production data and wake loss calculations
- **Report:** PDF with comprehensive analysis summary and visualizations
- **Wind Rose:** PNG/SVG wind rose diagram showing directional distribution