Automating Cloud Deployment with Serverless & Terraform

Casper Kristiansson (casperkr@kth.se)
Nicole Wijkman (nwijkman@kth.se)

Project Overview

Goal

Automate the deployment of the backend of a web application to AWS.

Technologies Used

Terraform: Infrastructure as code to provision AWS services like DynamoDB.

Serverless Framework: Simplifies deployment of Lambda functions and API Gateway for AWS.

Key Components

DynamoDB: For data storage.

API Gateway & Lambda: For backend logic such as API endpoints.

What is Terraform & Serverless?

Terraform: Manages infrastructure as code for consistent, automated deployments.

Serverless: Automates code execution without managing servers, focusing on function scalability.

Code Examples: Terraform & Serverless

Terraform (DynamoDB Table)

```hcl
resource "aws_dynamodb_table"
"demo_table" {
  name = "demo_table"
  billing_mode = "PAY_PER_REQUEST"
  hash_key = "id"

  attribute {
    name = "id"
    type = "S"
  }
}
```

Serverless (API Endpoint & Lambda)

```yaml
service: demo-app
provider:
  name: aws
  runtime: nodejs14.x

functions:
  apiHandler:
    handler: handler.api
    events:
      - http:
          path: /api
          method: get
```

Demo Plan

Step 1

Showcase the basic configurations of Serverless and Terraform

Step 2

Test the API endpoint with the initial configurations

Step 3

Destroy a database and remove a endpoint

Step 4

Test the API endpoint with the new configurations

Why Terraform?

What is Terraform?

1. Declarative Infrastructure-as-code tool.
2. Automates Cloud resource creation and managing.

Role in the Project

1. Provisioning DynamoDB table.

Why Serverless?

What is Serverless Framework?

1. Simplifies Serverless code and API Gateway deployment.
2. Scales without managing servers.

Role in the Project

1. Deploys and manages Lambda functions.
2. Connects API Gateway to Lambda functions.

Conclusion & Q&A

Key Takeaways

1. Automation using Terraform and Serverless simplifies Cloud infrastructure management.

1. Serverless is ideal for quickly deploying APIs, while Terraform is powerful for managing Cloud resources like DynamoDB.

Questions?