Containerised Node.js App with Terraform-Driven Infrastructure

This repository demonstrates a complete workflow for running a Node.js application inside Docker and provisioning infrastructure using Terraform.

Features:

- ❖ Containerised **Node.js app**
- ❖ Infrastructure provisioning via **Terraform**
- ❖ Automated workflows using **Bash scripts**
- ❖ Easy local development with **Docker**

Repository structure (tree):

```
aws-ecs-terraform-demo/
├── README.md
├── docker/
│   ├── Dockerfile
│   ├── .dockerignore
│   └── app.js
├── scripts/
│   └── analysis.sh
├── LinuxTasks.md
├── terraform/
│   ├── versions.tf
│   ├── providers.tf
│   ├── variables.tf
│   ├── outputs.tf
│   ├── main.tf
│   ├── backend.tf.example
│   └── modules/
│       ├── network/
│       │   ├── main.tf
│       │   ├── variables.tf
│       │   └── outputs.tf
│       ├── security/
│       │   ├── main.tf
│       │   ├── variables.tf
│       │   └── outputs.tf
│       ├── ecr/
│       │   ├── main.tf
│       │   ├── variables.tf
│       │   └── outputs.tf
│       ├── iam/
│       │   ├── main.tf
│       │   ├── variables.tf
│       │   └── outputs.tf
│       ├── ecs/
│       │   ├── main.tf
│       │   ├── variables.tf
│       │   └── outputs.tf
│       └── alb/
│         ├── main.tf
│         ├── variables.tf
│         └── outputs.tf
│
```

Prerequisites:

- □ Node.js (if running locally without Docker)
- □ Docker
- □ Terraform
- □ Bash shell (Linux/Mac/WSL)

Running the Node.js App with Docker:

Run Locally:

- cd app
- npm install
- node index.js

Docker Setup:

- Build the Docker image

  ```
  docker build -t nodejs-app .
  ```

- Run the container

  ```
  docker run -d -p 3000:3000 nodejs-app
  ```

- Open in browser

  http://localhost:3000

Provisioning Infrastructure with Terraform:

➔ Navigate to the Terraform directory:

```
cd tf-code
```

➔ Initialize Terraform:

```
terraform init
```

➔ Review the execution plan:

```
terraform plan
```

➔ Apply infrastructure:

```
terraform apply
```

Bash Automation:

```
./scripts/analysis.sh          # Default threshold 80%

./scripts/analysis.sh 90       # Custom threshold (90%)
```

Example output:

```
=== Disk Usage Report (threshold 80%) ===
Filesystem              Type    Size     Use%   Mount
/dev/sda1               ext4    50G      82     /     <-- HIGH
```

Future Improvements:

- □ Add CI/CD pipeline (GitHub Actions / GitLab CI)
- □ Integrate monitoring/logging (Prometheus & Grafana)

Author

Shrutika Khaparde
DevOps & Cloud Engineer