# Agentic AI Terraform Infrastructure Builder (PoC)

## 🎯 Objective

Build an Agentic AI system that:  
- Accepts natural language prompts  
- Converts them into Terraform code  
- Applies infrastructure (e.g., S3 buckets) on AWS  
- Works via a FastAPI backend and uses Claude 3.5 Sonnet from Amazon Bedrock

## 🏗️ Architecture Overview

Stack used:  
- LLM: anthropic.claude-3-5-sonnet-20240620-v1:0 via Amazon Bedrock  
- Cloud: AWS (with configured CLI profile)  
- IaC: Terraform  
- Backend: FastAPI  
- Auth: AWS STS for account validation  
- Local run: uvicorn for FastAPI server

## 🔄 Flow

1. User POSTs a natural language instruction + AWS CLI profile  
2. FastAPI sends it to Claude 3.5 Sonnet  
3. Claude responds with Terraform code  
4. Code is written to main.tf  
5. Terraform is initialized, validated, and applied via subprocess  
6. Success/failure output is returned

## 📥 Request Example

Endpoint: POST /generate-terraform/  
Request Body:  
{  
 "bucket\_name": "vaishaltestagentic",  
 "region": "us-east-1",  
 "profile": "cgi"  
}

## 🧠 Prompt Sent to Claude 3.5

You are an expert in AWS and Terraform.  
  
Based on the following inputs:  
- Bucket Name: vaishaltestagentic  
- Region: us-east-1  
  
Generate only Terraform code to create the S3 bucket with versioning enabled.  
Don't include explanation or extra text.

## 🧾 Claude Output Sample

provider "aws" {  
 region = "us-east-1"  
 profile = "cgi"  
}  
  
resource "aws\_s3\_bucket" "my\_bucket" {  
 bucket = "vaishaltestagentic"  
  
 versioning {  
 enabled = true  
 }  
  
 tags = {  
 Environment = "Agentic"  
 }  
}

## 🛠️ Backend Logic (main.py Summary)

1. FastAPI POST endpoint receives input JSON.  
2. Sends prompt to Claude 3.5 on Bedrock.  
3. Receives Terraform HCL code from the model.  
4. Writes HCL to main.tf  
5. Executes:  
 - terraform init  
 - terraform validate  
 - terraform apply -auto-approve  
6. Returns plan or apply output or error details.

## ✅ Output Example

HTTP 200  
{  
 "message": "Terraform applied",  
 "output": "...Terraform stdout with plan and apply logs..."  
}

## 📸 PoC Screenshots

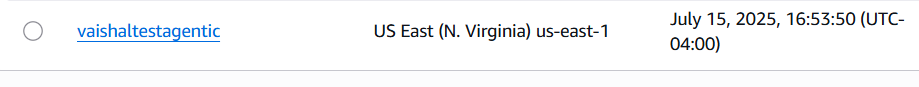
- Successful Swagger POST request  
- Output from Claude via JSON  
- Bucket 'my-vaishal-test-bucket' shown created in AWS Console

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.



## ⚠️ Edge Case Handled

- Wrong AWS account: STS assumed-role verified before execution.  
- Claude model changed: Explicit modelId ensured: anthropic.claude-3-5-sonnet-20240620-v1:0

## 🔒 Security Note

- Terraform executes using CLI credentials via provided AWS profile.  
- Claude is prompted in controlled, bounded context (no hallucination).  
- Bucket name is validated via terraform validate.

## 📁 Folder Structure

project/  
├── main.py # FastAPI backend  
├── main.tf # Generated Terraform file  
├── .aws/credentials # AWS profile setup  
├── requirements.txt # fastapi, boto3, openai, etc.

## 🪛 Future Enhancements

- Add support for other resource types (EC2, VPC, etc.)  
- Add prompt memory and refinement loop (Agentic feedback)  
- Use LangGraph for orchestration  
- Add user authentication to API