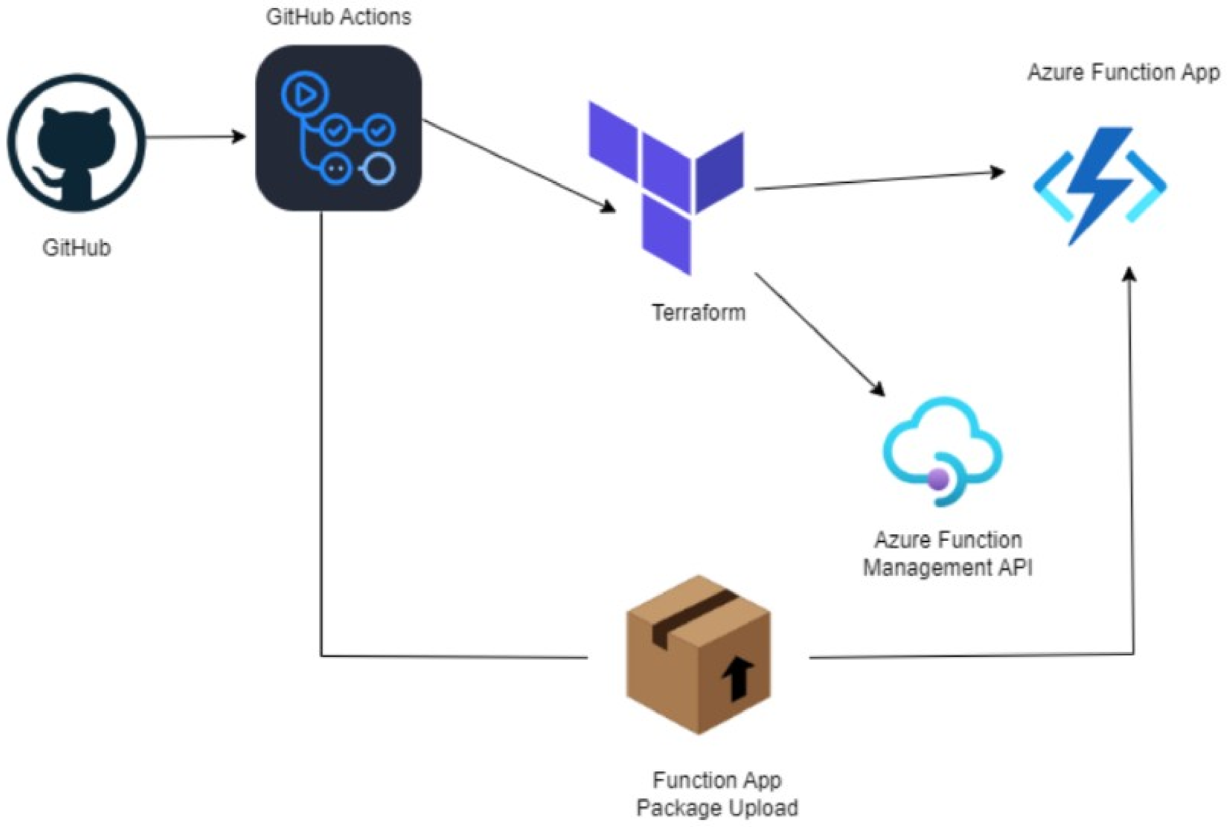
Case Study 2 – Azure API Management

GitHub Link : https://github.com/sriramsurendhran/surge-api-case-study-2.git
Author : Sriram SURENDHRAN

ArchitectureDesign :



Task Solution:
Azure API Management (APIM):
• APIM Instance:
An APIM instance named my-apim is created, acting as a gateway to manage APIs, including versioning, security, and scaling.

• Terraform Integration:
o Terraform is used to create and manage the APIM instance.
o A main.tf file in the Terraform directory contains the Resource Manager and APIM configurations.
o Using the Terraform client, resources are provisioned in Azure by executing the Terraform files.
• API Registration:
o An API named AzureApimAPI is registered within the APIM instance and exposed at the path /azureapim.
o Secure communication is ensured via HTTPS.
o The backend is configured to route requests to a Function App using its hostname.

• API Gateway:
o The APIM instance enforces security policies, transforms API requests, and routes them to backend services like Azure Functions.
o Application Insights is integrated to track usage, errors, and performance metrics for the Function App.
• Resource Group and Outputs:
o The created resource group, named apim-function-rg, is exposed as an output.
o Application Insights (azurerm_application_insights) is configured for telemetry and performance monitoring.

• Administration:
o Publisher details (email and name) are configured for APIM administration.

Function App:
• Migrates the logic to a serverless model to handle incoming API requests from APIM.
• Monitors API health and infrastructure.
• The connection string is passed to the Function App via an environment variable.

Terraform Automation:
• Automates the provisioning of resources such as APIM, Azure Functions, storage, monitoring, and security settings.
• Provides flexibility to create or delete resources by applying Terraform instructions.
• Careful execution is required during resource deletion.
Security:
• Authentication and Authorization:
o APIM enforces authentication using methods like OAuth 2.0 and API keys.
o Policies enforce JWT validation, including OpenID Connect integration with Azure Active Directory (AAD).
o API requests must include an Authorization header with a valid token (JWT).
o Unauthorized requests return a 401 Unauthorized response.
• Security Enhancements:
o A policy ensures the aud (audience) claim matches the API client ID.
o An additional security header (X-Frame-Options) prevents clickjacking.

Monitoring:
• Integrated Monitoring:
o Azure Monitor, Log Analytics, and Application Insights are configured to track usage and diagnose performance issues.

Azure APIM Configuration:
• Resource Group:
o A resource group named apim-function-rg is created in the India South location, serving as a container for Azure resources.

• Storage Account:
o A storage account named azueapimfunctionapp is created within the resource group for storing runtime data and package files.
o It uses the Standard performance tier and LRS (Locally Redundant Storage) replication.
• App Service Plan:
o An App Service Plan named function-app-plan is created with the Dynamic SKU (Y1 size) for Azure Functions.

o The reserved flag is enabled for specific configurations.

This setup ensures a secure, scalable, and well-monitored API management and Function App environment using Azure and Terraform.