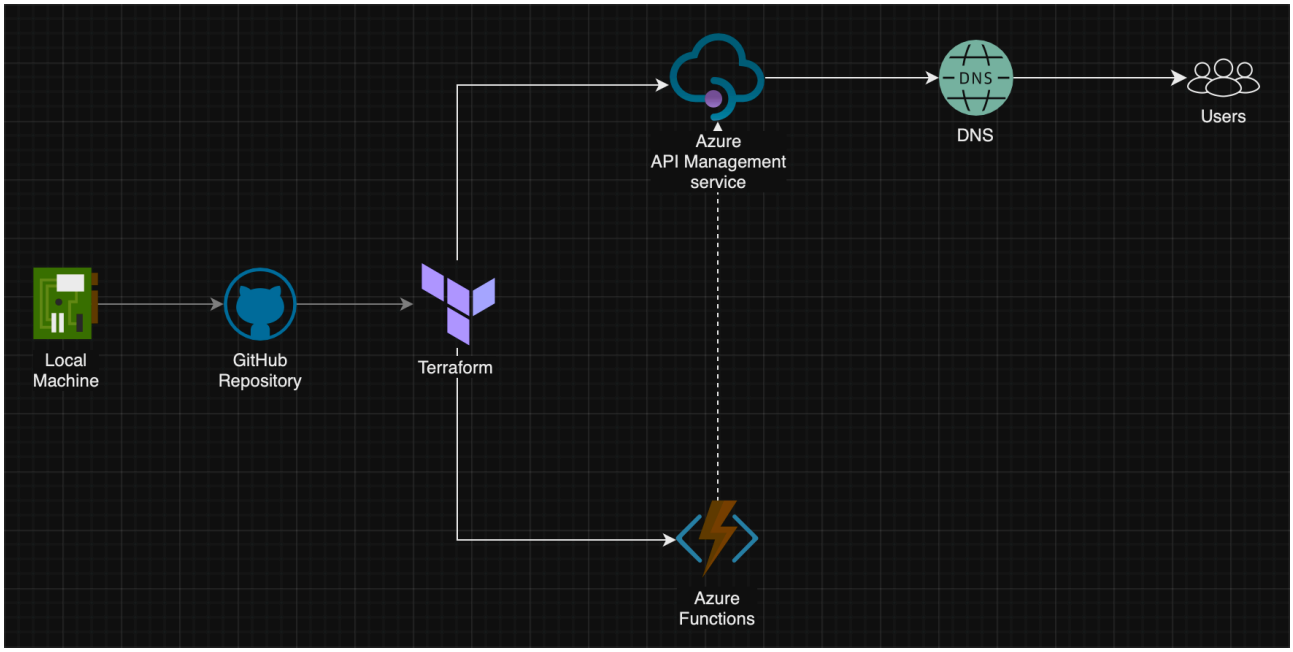
Assignment result

1. A detailed design document outlining the solution architecture.

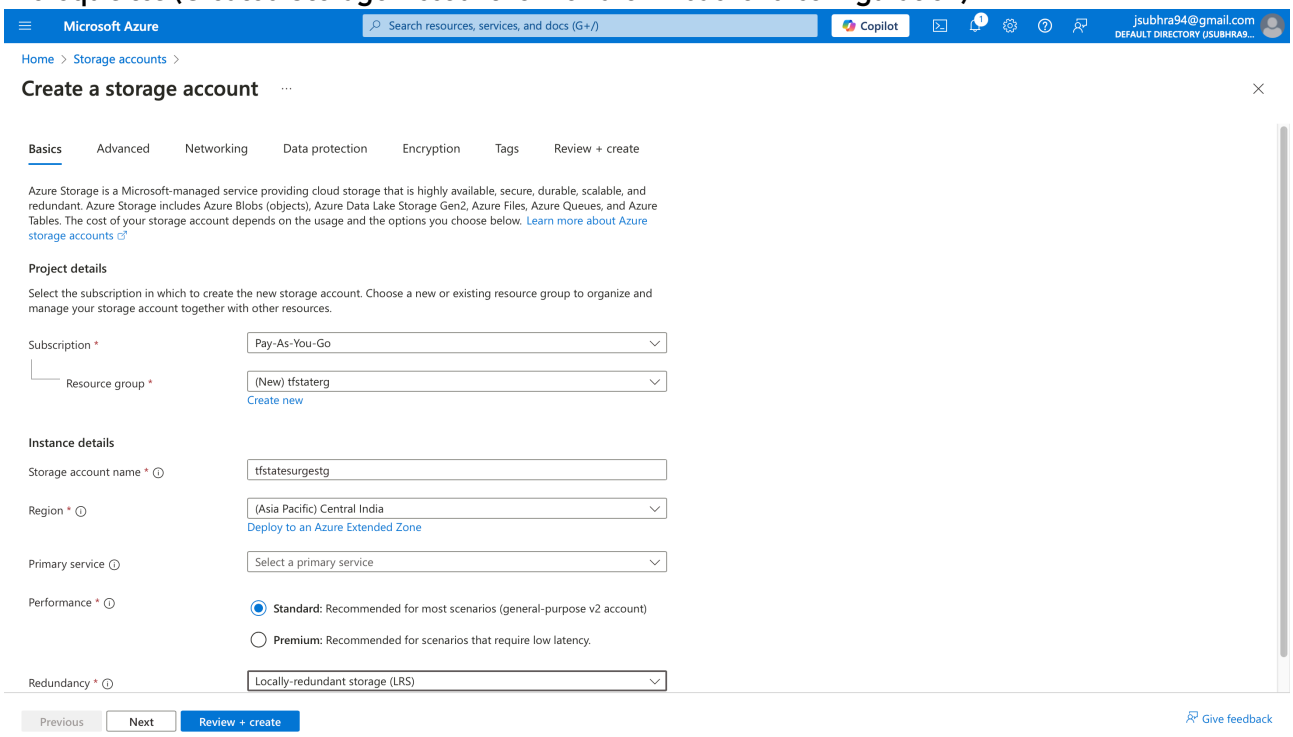
Architecture:



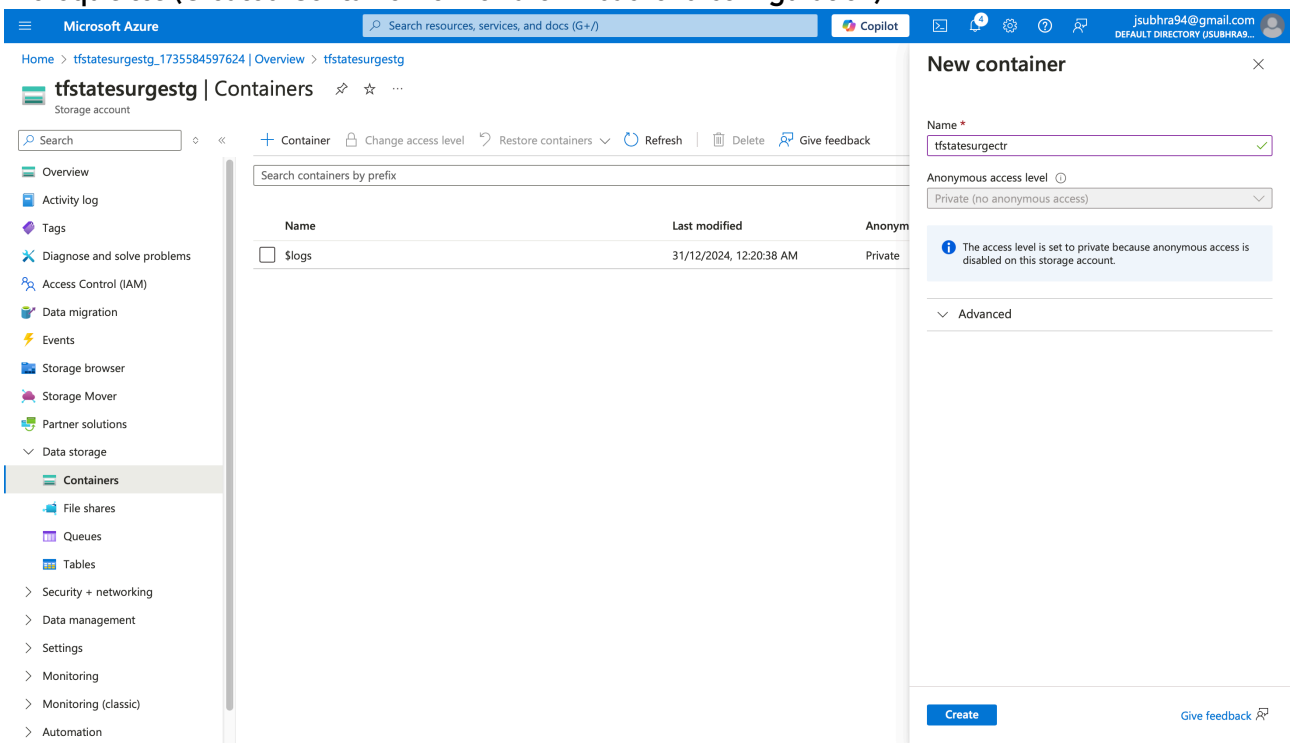
2. A Terraform configuration file that automates the deployment of the APIM and Function App.

Link: api-management-assignment

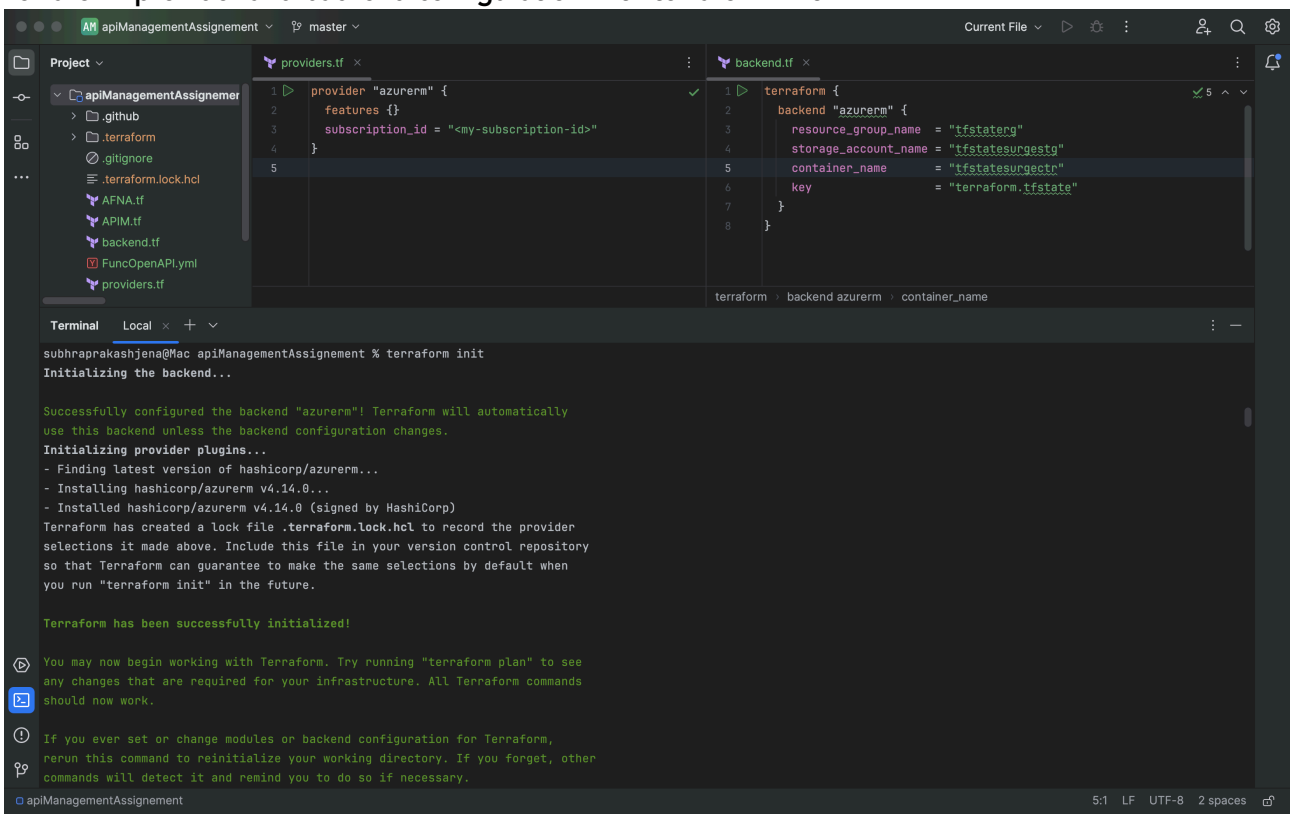
Prerequisites (Created Storage Account for Terraform backend configuration):



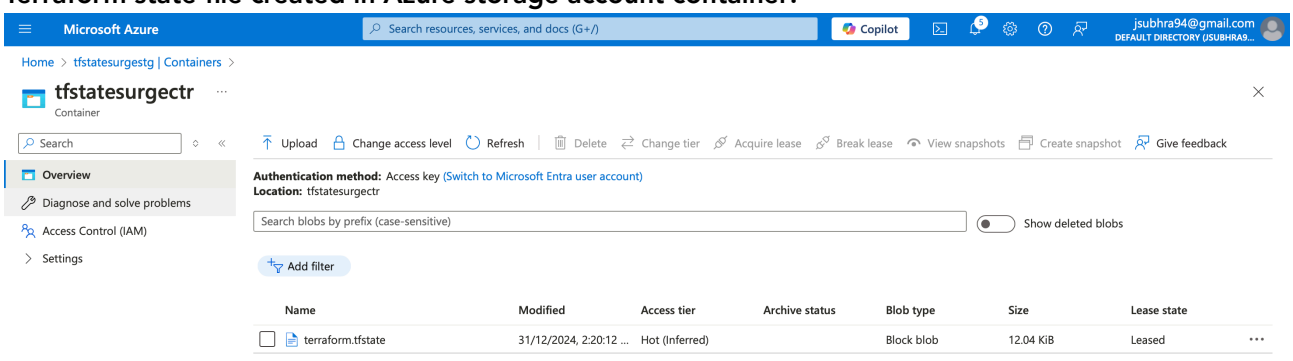
Prerequisites (Created Container for Terraform backend configuration):



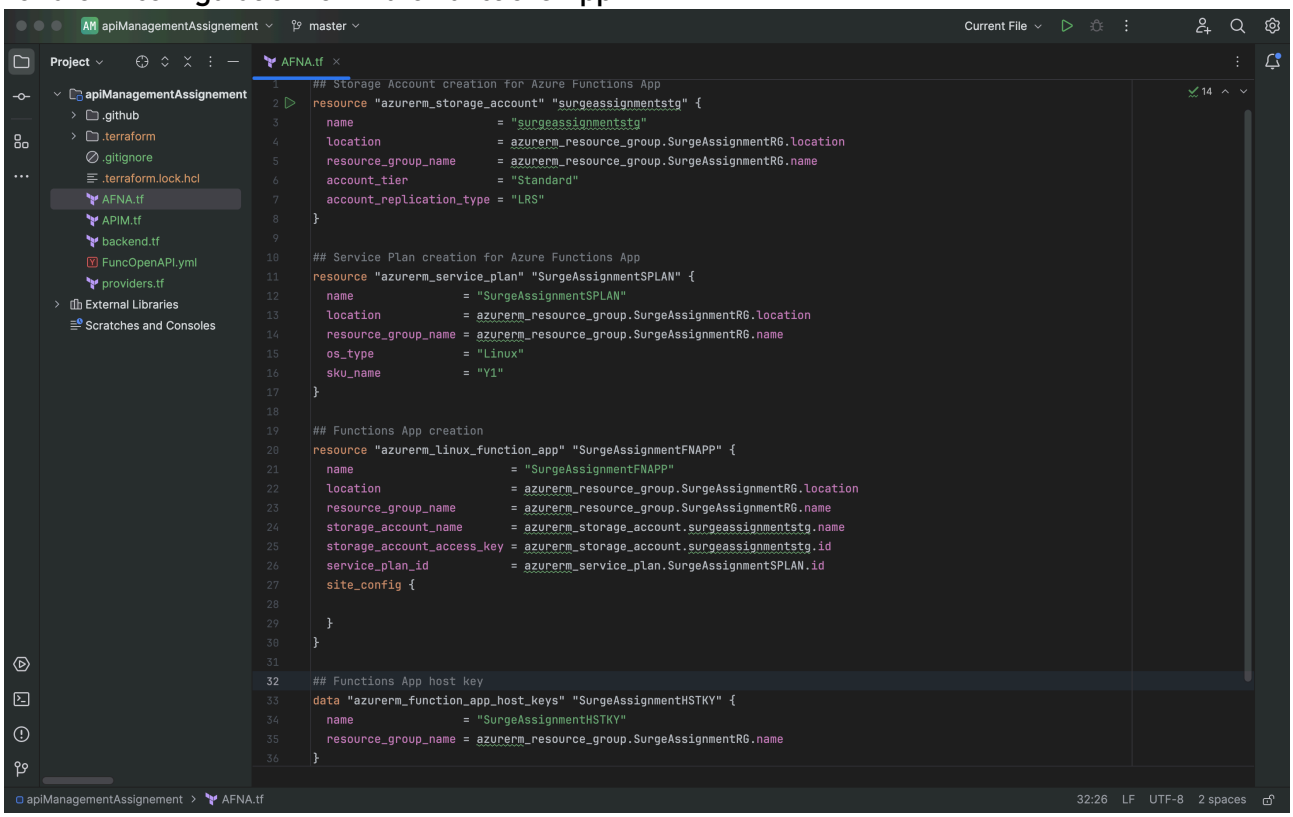
Terraform provider and backend configuration with terraform init:



Terraform state file created in Azure storage account container:

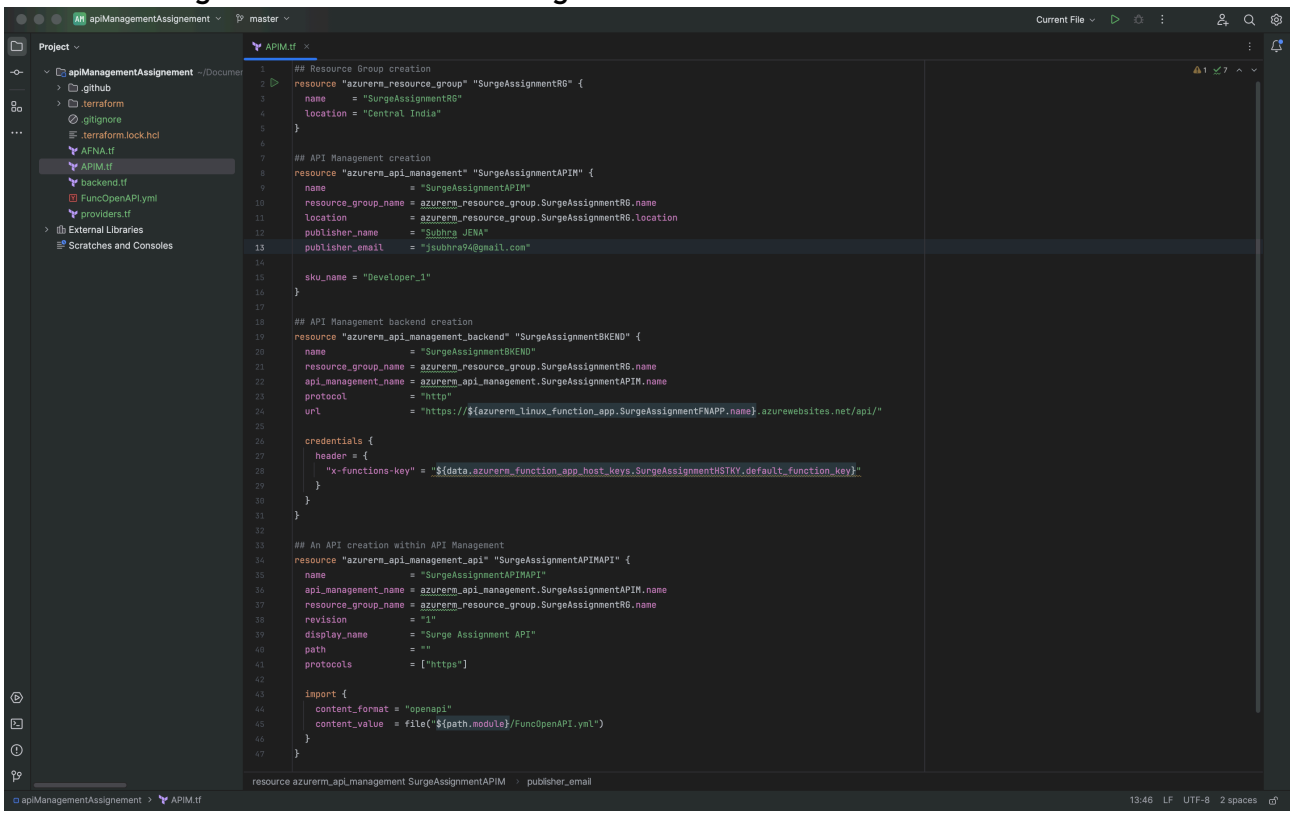


Terraform configuration for Azure Functions App:



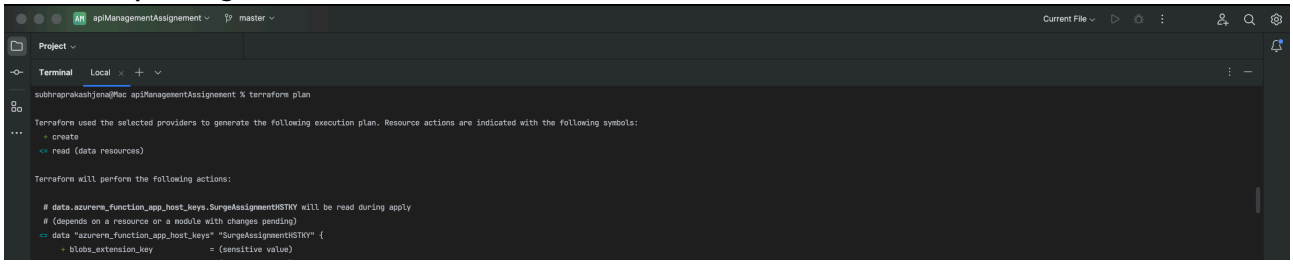
```
## Storage Account creation for Azure Functions App
resource "azurerm_storage_account" "surgeassignmentstg" {
  name                     = "surgeassignmentstg"
  location                 = azurerm_resource_group.SurgeAssignmentRG.location
  resource_group_name      = azurerm_resource_group.SurgeAssignmentRG.name
  account_tier             = "Standard"
  account_replication_type = "LRS"
}

## Service Plan creation for Azure Functions App
resource "azurerm_service_plan" "SurgeAssignmentSPLAN" {
  name                = "SurgeAssignmentSPLAN"
  location            = azurerm_resource_group.SurgeAssignmentRG.location
  resource_group_name = azurerm_resource_group.SurgeAssignmentRG.name
  os_type             = "Linux"
  sku_name            = "Y1"
}

## Functions App creation
resource "azurerm_linux_function_app" "SurgeAssignmentFNAPP" {
  name                       = "SurgeAssignmentFNAPP"
  location                   = azurerm_resource_group.SurgeAssignmentRG.location
  resource_group_name        = azurerm_resource_group.SurgeAssignmentRG.name
  storage_account_name       = azurerm_storage_account.surgeassignmentstg.name
  storage_account_access_key = azurerm_storage_account.surgeassignmentstg.id
  service_plan_id            = azurerm_service_plan.SurgeAssignmentSPLAN.id
  site_config {

  }
}

## Functions App host key
data "azurerm_function_app_host_keys" "SurgeAssignmentHSTKY" {
  name                = "SurgeAssignmentHSTKY"
  resource_group_name = azurerm_resource_group.SurgeAssignmentRG.name
}
```

Terraform configuration for Azure API Management Service:

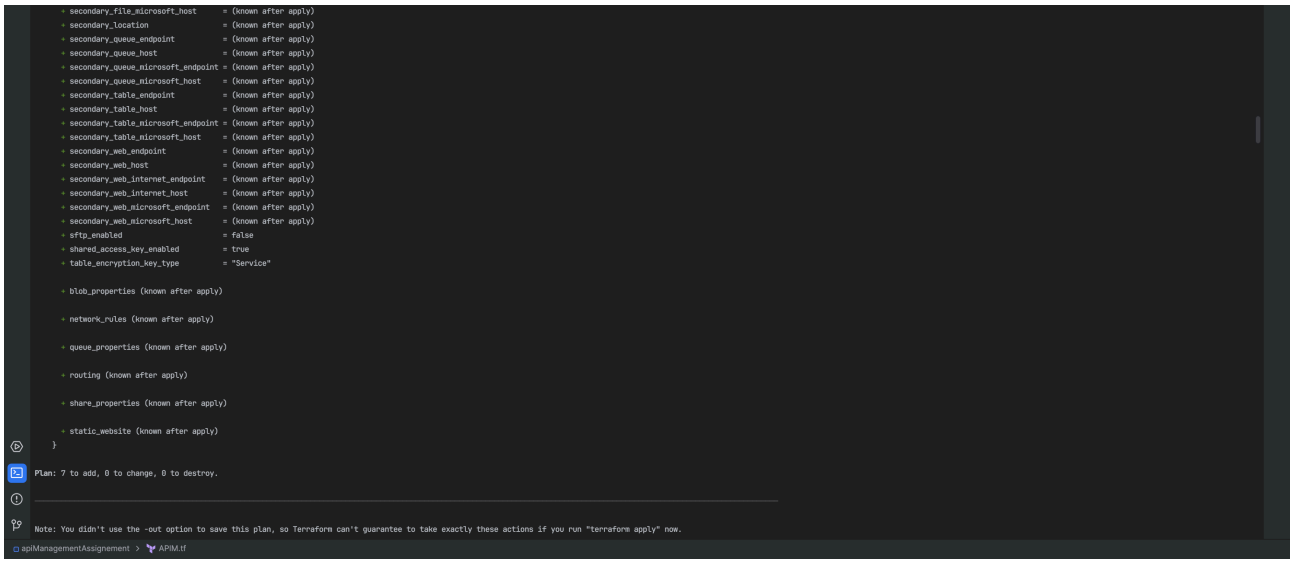


```
## Resource Group creation
resource "azurerm_resource_group" "SurgeAssignmentRG" {
  name     = "SurgeAssignmentRG"
  location = "Central India"
}

## API Management creation
resource "azurerm_api_management" "SurgeAssignmentAPIM" {
  name                = "SurgeAssignmentAPIM"
  resource_group_name = azurerm_resource_group.SurgeAssignmentRG.name
  location            = azurerm_resource_group.SurgeAssignmentRG.location
  publisher_name      = "Subhra JENA"
  publisher_email     = "jsubhra94@gmail.com"

  sku_name = "Developer_1"
}

## API Management backend creation
resource "azurerm_api_management_backend" "SurgeAssignmentBKEND" {
  name                = "SurgeAssignmentBKEND"
  resource_group_name = azurerm_resource_group.SurgeAssignmentRG.name
  api_management_name = azurerm_api_management.SurgeAssignmentAPIM.name
  protocol            = "http"
  url                 = "https://${azurerm_linux_function_app.SurgeAssignmentFNAPP.name}.azurewebsites.net/api/"

  credentials {
    header = {
      "x-functions-key" = "${data.azurerm_function_app_host_keys.SurgeAssignmentHSTKY.default_function_key}"
    }
  }
}

## An API creation within API Management
resource "azurerm_api_management_api" "SurgeAssignmentAPIMAPI" {
  name                = "SurgeAssignmentAPIMAPI"
  api_management_name = azurerm_api_management.SurgeAssignmentAPIM.name
  resource_group_name = azurerm_resource_group.SurgeAssignmentRG.name
  revision            = "1"
  display_name        = "Surge Assignment API"
  path                = ""
  protocols           = ["https"]

  import {
    content_format = "openapi"
    content_value  = file("${path.module}/FuncOpenAPI.yml")
  }
}
```

Terraform plan logs:

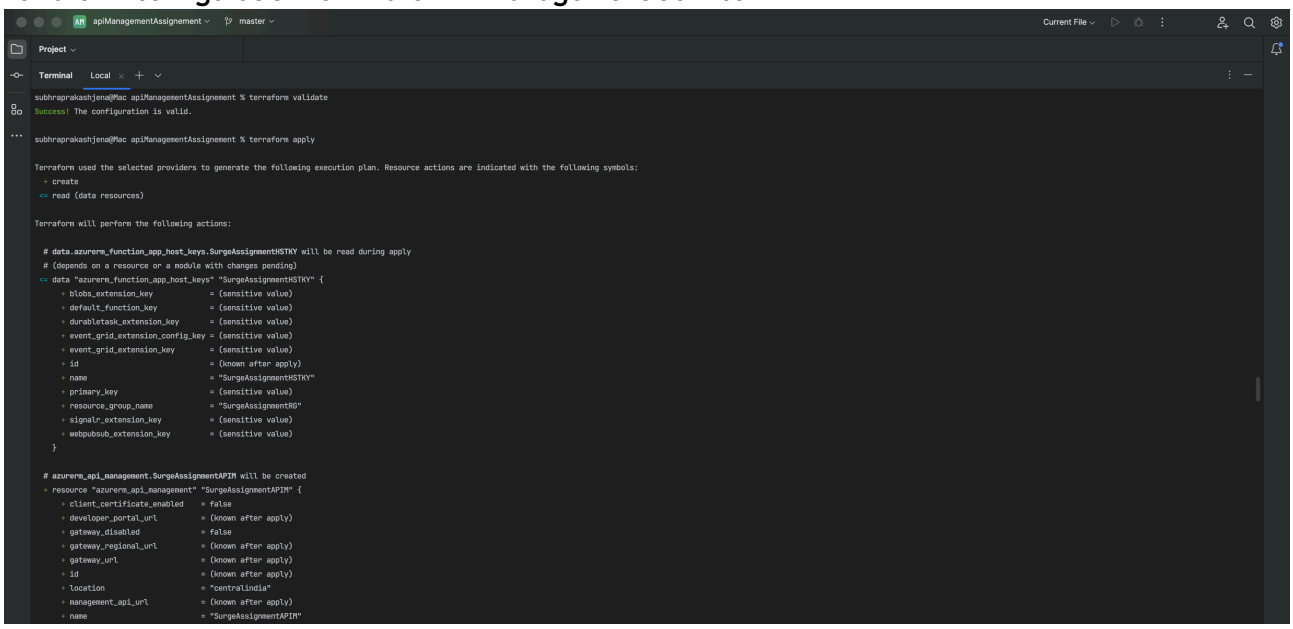


```
subhraprakashjena@Mac apiManagementAssignement % terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create
 <= read (data resources)

Terraform will perform the following actions:

  # data.azurerm_function_app_host_keys.SurgeAssignmentHSTKY will be read during apply
  # (depends on a resource or a module with changes pending)
 <= data "azurerm_function_app_host_keys" "SurgeAssignmentHSTKY" {
      + blobs_extension_key             = (sensitive value)
```



```
      + secondary_file_microsoft_host      = (known after apply)
      + secondary_location                 = (known after apply)
      + secondary_queue_endpoint           = (known after apply)
      + secondary_queue_host               = (known after apply)
      + secondary_queue_microsoft_endpoint = (known after apply)
      + secondary_queue_microsoft_host     = (known after apply)
      + secondary_table_endpoint           = (known after apply)
      + secondary_table_host               = (known after apply)
      + secondary_table_microsoft_endpoint = (known after apply)
      + secondary_table_microsoft_host     = (known after apply)
      + secondary_web_endpoint             = (known after apply)
      + secondary_web_host                 = (known after apply)
      + secondary_web_internet_endpoint    = (known after apply)
      + secondary_web_internet_host        = (known after apply)
      + secondary_web_microsoft_endpoint   = (known after apply)
      + secondary_web_microsoft_host       = (known after apply)
      + sftp_enabled                       = false
      + shared_access_key_enabled          = true
      + table_encryption_key_type          = "Service"

      + blob_properties (known after apply)

      + network_rules (known after apply)

      + queue_properties (known after apply)

      + routing (known after apply)

      + share_properties (known after apply)

      + static_website (known after apply)
    }

Plan: 7 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
```

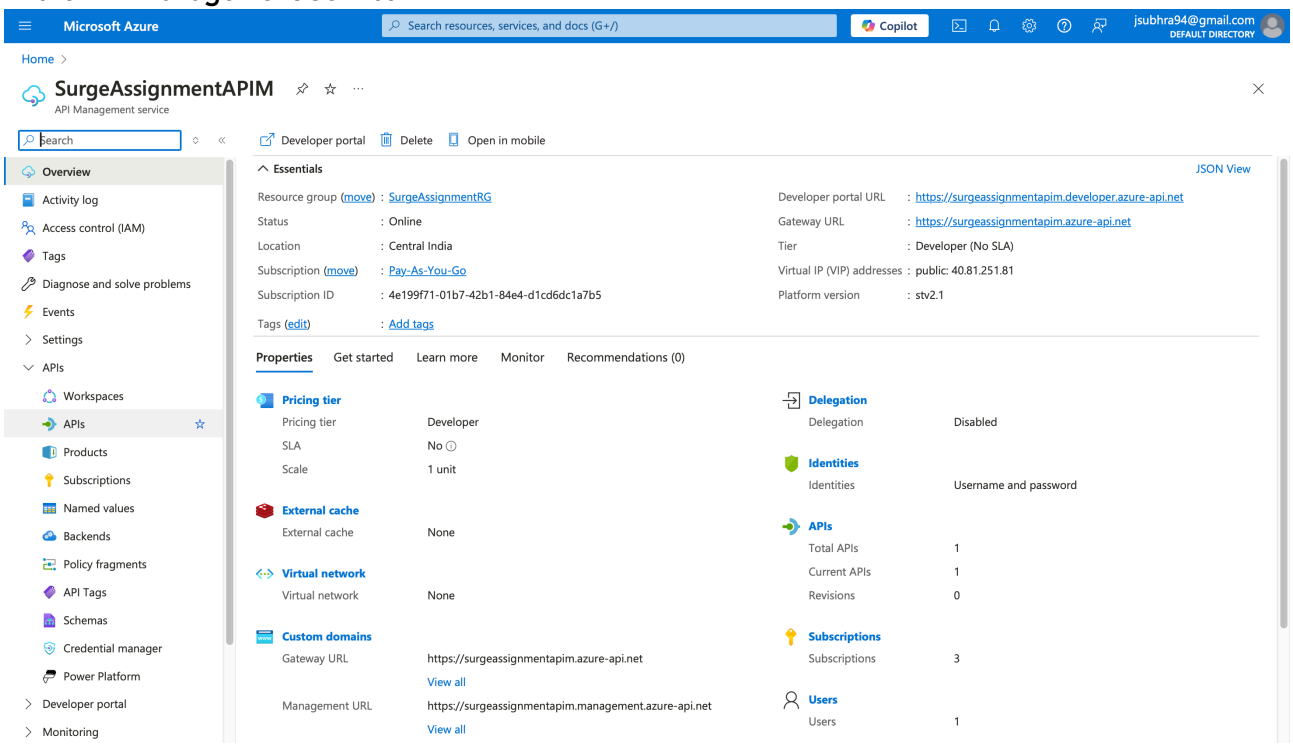
Terraform configuration for Azure API Management Service:



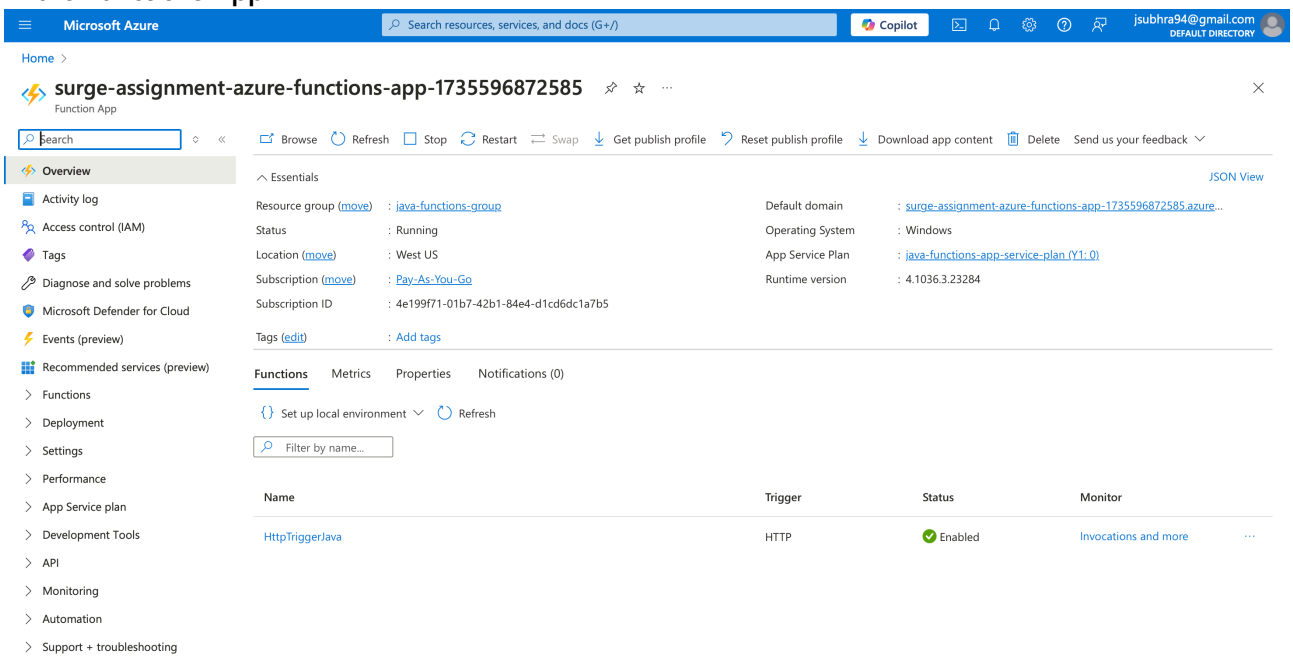
```
subhraprakashjena@Mac apiManagementAssignement % terraform validate
Success! The configuration is valid.

subhraprakashjena@Mac apiManagementAssignement % terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create
 <= read (data resources)

Terraform will perform the following actions:

  # data.azurerm_function_app_host_keys.SurgeAssignmentHSTKY will be read during apply
  # (depends on a resource or a module with changes pending)
 <= data "azurerm_function_app_host_keys" "SurgeAssignmentHSTKY" {
      + blobs_extension_key             = (sensitive value)
      + default_function_key            = (sensitive value)
      + durabletask_extension_key       = (sensitive value)
      + event_grid_extension_config_key = (sensitive value)
      + event_grid_extension_key        = (sensitive value)
      + id                              = (known after apply)
      + name                            = "SurgeAssignmentHSTKY"
      + primary_key                     = (sensitive value)
      + resource_group_name             = "SurgeAssignmentRG"
      + signalr_extension_key           = (sensitive value)
      + webpubsub_extension_key         = (sensitive value)
    }

  # azurerm_api_management.SurgeAssignmentAPIM will be created
  + resource "azurerm_api_management" "SurgeAssignmentAPIM" {
      + client_certificate_enabled    = false
      + developer_portal_url          = (known after apply)
      + gateway_disabled              = false
      + gateway_regional_url          = (known after apply)
      + gateway_url                   = (known after apply)
      + id                            = (known after apply)
      + location                      = "centralindia"
      + management_api_url            = (known after apply)
      + name                          = "SurgeAssignmentAPIM"
```

```
      + queue_properties (known after apply)

      + routing (known after apply)

      + share_properties (known after apply)

      + static_website (known after apply)
    }

Plan: 7 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

azurerm_resource_group.SurgeAssignmentRG: Creating...
azurerm_resource_group.SurgeAssignmentRG: Still creating... [10s elapsed]
```

Azure API Management Service:

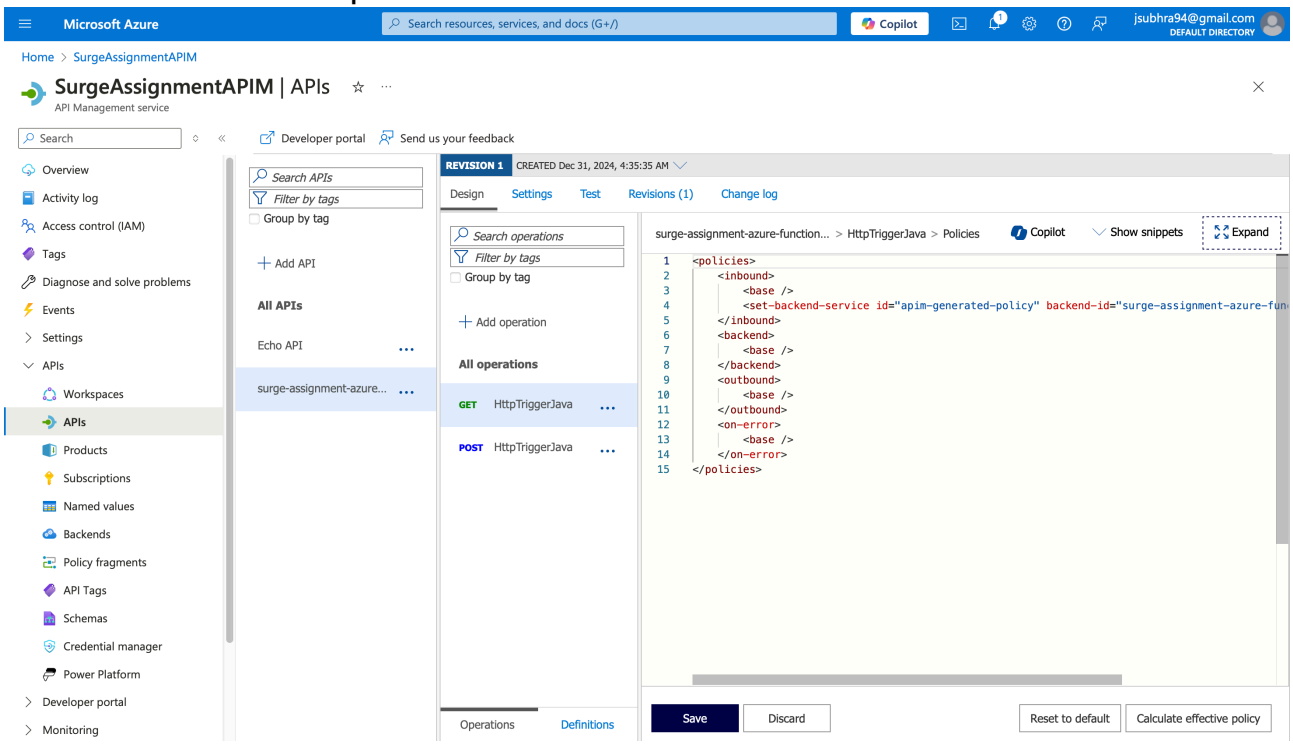


Azure Functions App:

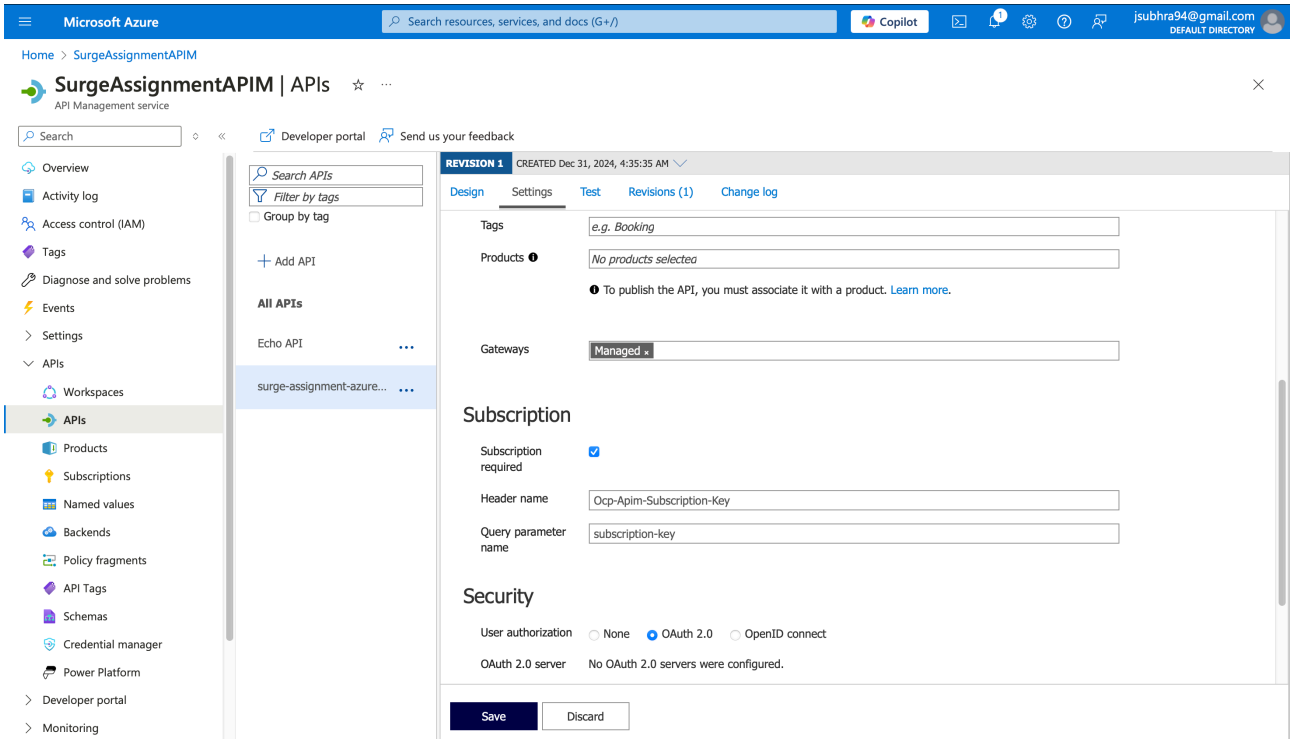


3. An APIM configuration file that defines the API policies and security settings.

API inbound and outbound policies:

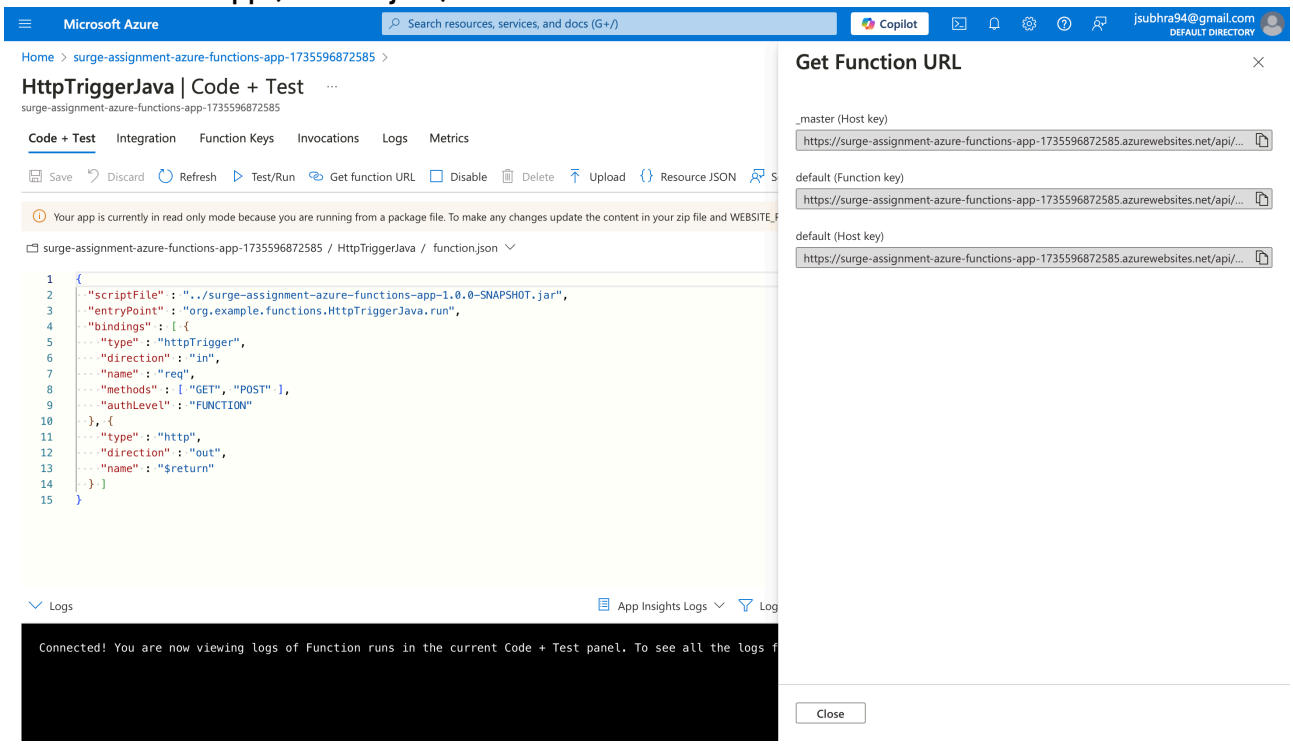


API enabled with OAuth2 authorisation to make it secure:

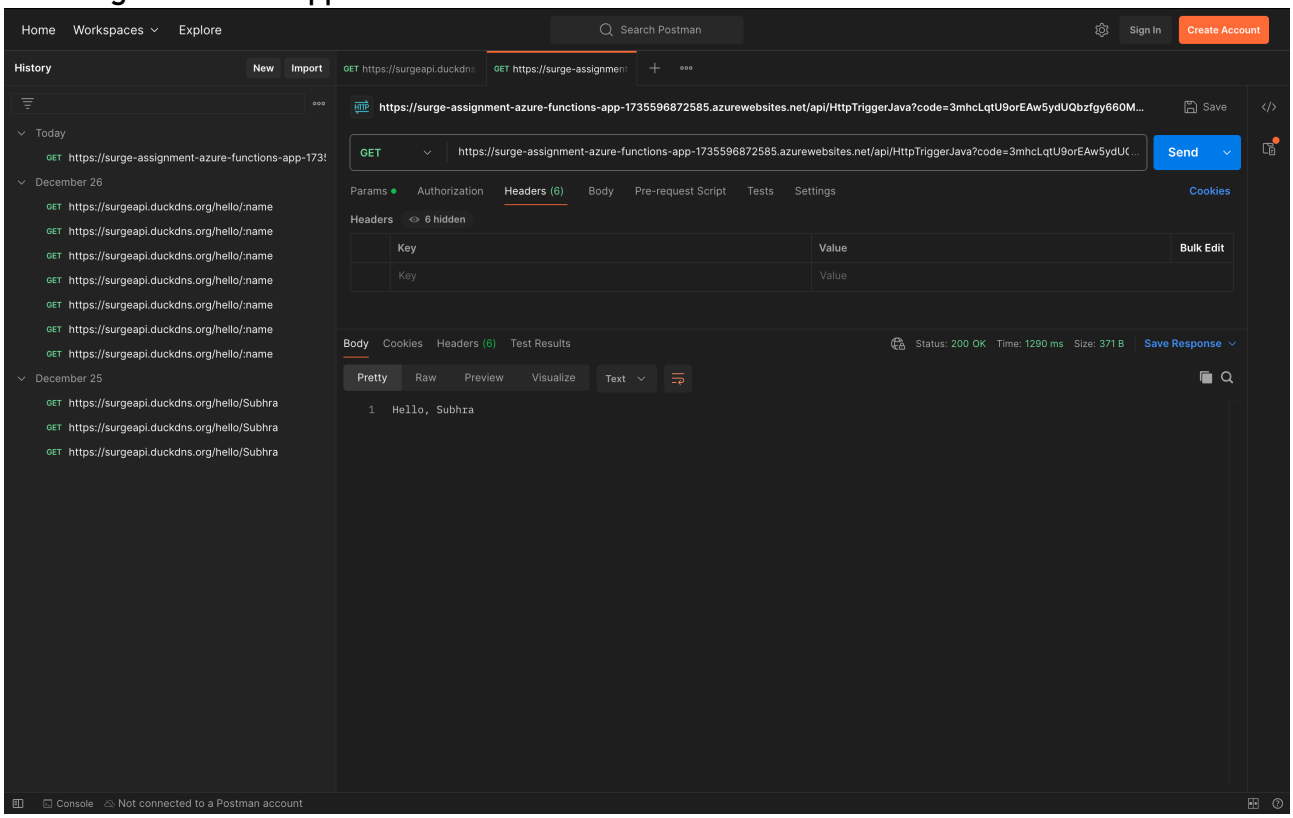


4. A Function App configuration file that defines the API handlers and security settings.

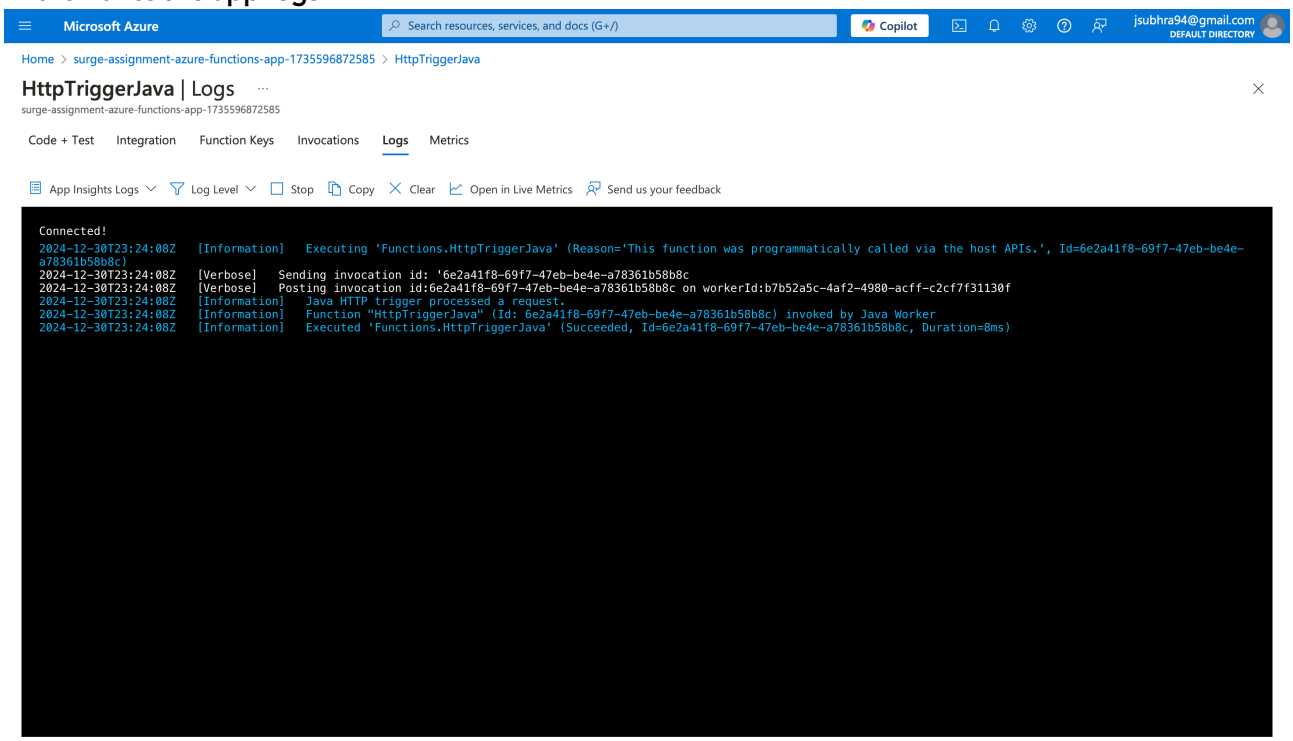
Azure Functions App (function.json):



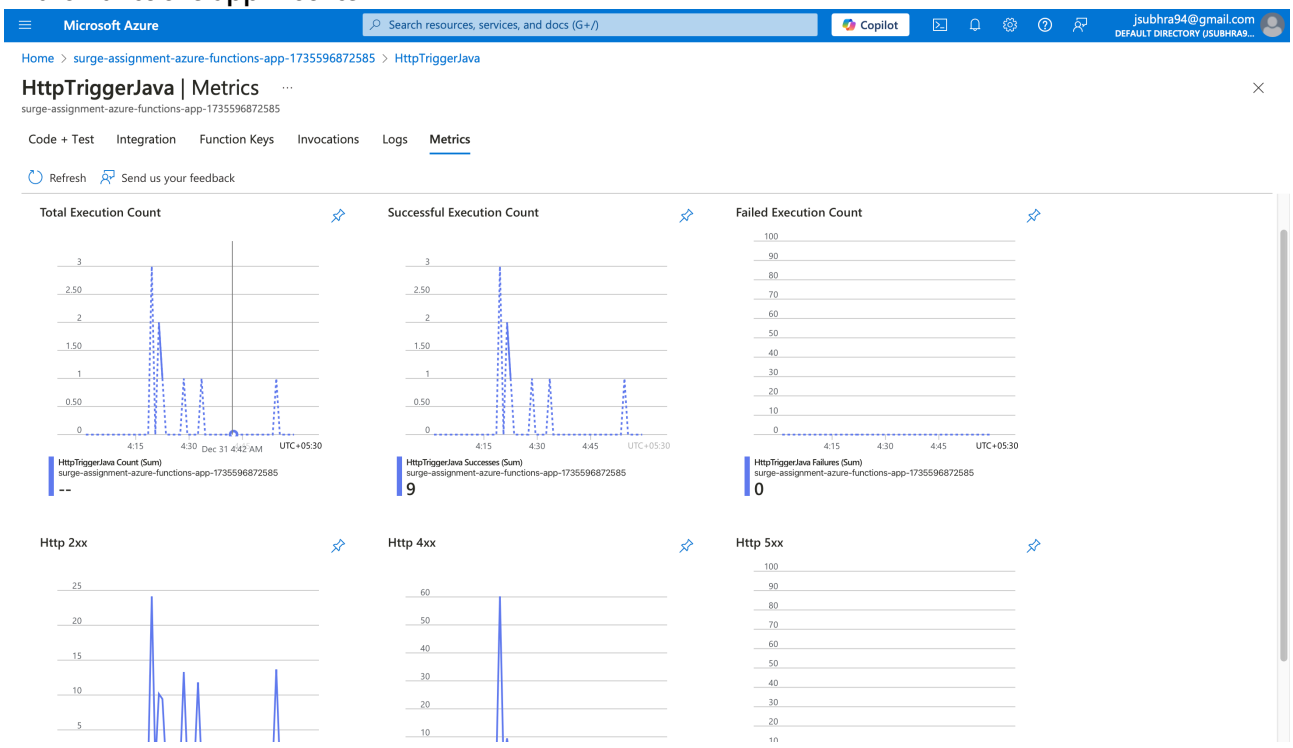
Accessing the function app:



Azure Functions app logs:



Azure Functions app metrics:



5. A report that summarises the solution and its benefits.

- Simplified API lifecycle management.
- Reduced operational overhead through serverless architecture.
- Enhanced security and compliance with advanced policies.
- Automated, consistent infrastructure deployment.
- Insightful analytics to improve performance and user experience.