

Implementation and Evaluation of a Zero Trust Architecture to Mitigate Multitenant Data Leakage In .Net Core Azure Applications Configuration Manual

MSc Research Project
Cloud Computing

Nidhi Anandan
Student ID: 23286873

School of Computing
National College of Ireland

Supervisor: Dr. Shivani Jaswal

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Nidhi Anandan
Student ID:	23286873
Programme:	Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Dr. Shivani Jaswal
Submission Due Date:	11/12/2025
Project Title:	Implementation and Evaluation of a Zero Trust Architecture to Mitigate Multitenant Data Leakage In .Net Core Azure Applications Configuration Manual
Word Count:	1000 +
Page Count:	15

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Nidhi Anandan
Date:	11th December 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI Acknowledgement Supplement

MSc Research Project

Implementation and Evaluation of a Zero Trust Architecture to Mitigate Multitenant Data Leakage In .Net Core Azure Applications

Your Number	Name/StudentCourse	Date
Nidhi Anandan/ 23286873	Msc. In Cloud Computing	11/12/2025

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click [here](#).

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used.**

[Insert Tool Name]	
[Insert Description of use]	
[Insert Sample prompt]	[Insert Sample response]

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

[Place evidence here]

Additional Evidence:

[Place evidence here]

Implementation and Evaluation of a Zero Trust Architecture to Mitigate Multitenant Data Leakage In .Net Core Azure Applications Configuration Manual

Nidhi Anandan
23286873

1 Prerequisites And Initial Setup

1.1 Visual Studio Installation

Install Visual Studio 2026 Community version from the following link.

<https://visualstudio.microsoft.com/downloads/>

Installation guide link - Microsoft Visual Studio Download (2025)

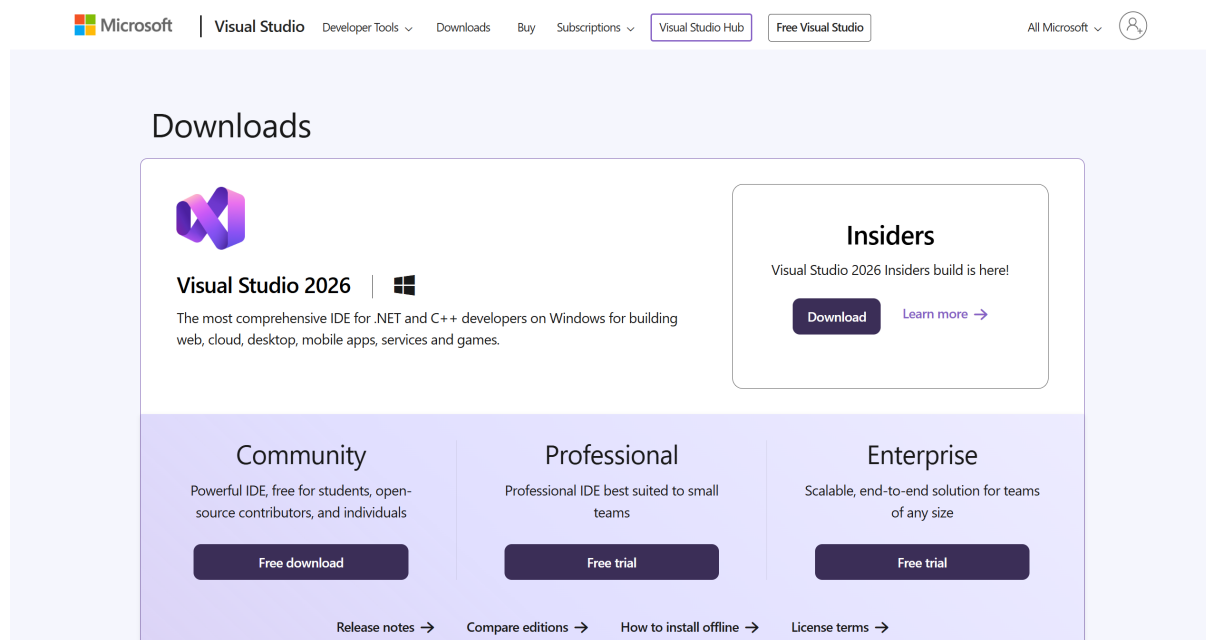


Figure 1: Visual Studio Installation.

1.2 Clone GitHub Repository to Visual Studio

Web Application and API Project Repository Link

https://studentncirl-my.sharepoint.com/:f:/g/personal/x23286873_student_ncirl_ie/IgBJ7vLwA7p1SZVgovkYFpTrAU8nDj4GcJbWuqXrIGshGDQ?e=homQj2

Github repository clone guide link - Github Repository Clone Guide (2025)

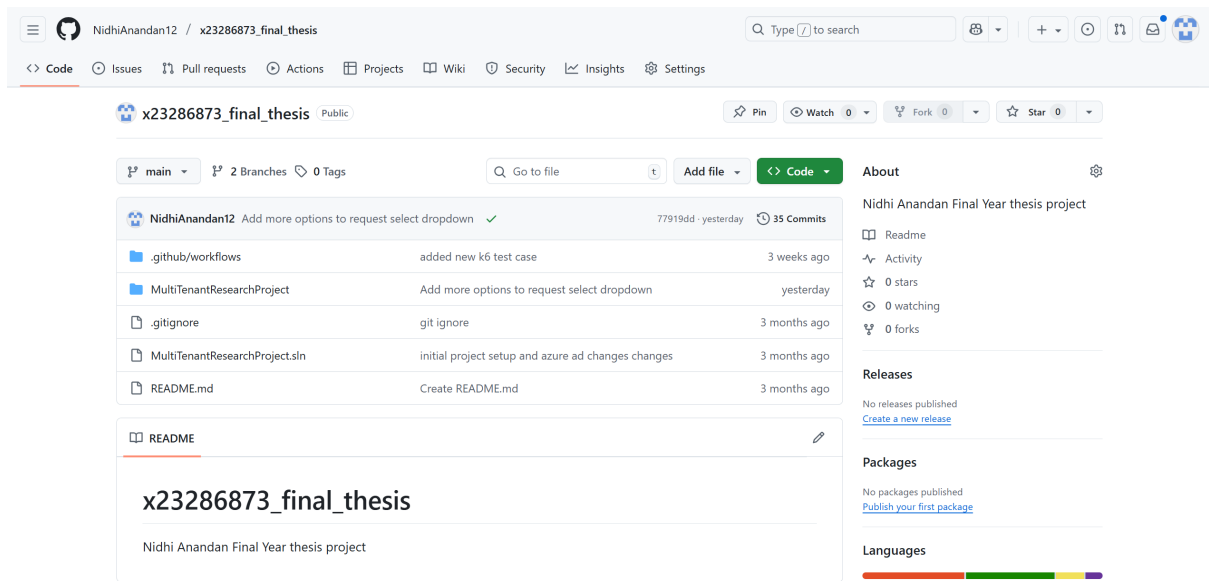


Figure 2: Github web application repository.

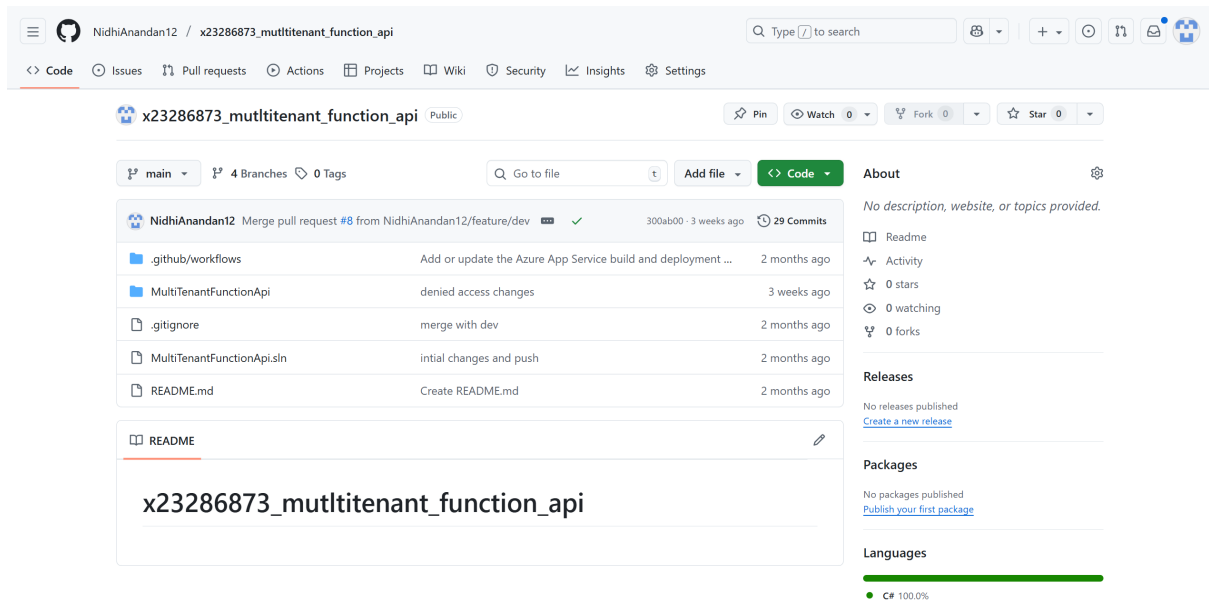


Figure 3: Github web API repository.

2 Azure Infrastructure Provisioning and Configuration

2.1 Azure Active Entra ID Setup

A new Entra ID needs to be setup. It is used to manage organizations where each entra represents an organization. But due to personal account restriction in creating multiple Entra ID's we will be using Default entra ID and create two new user groups to simulate multiple tenants and test users needs to be added to each user group. We will need to get the entra id details and group details which will be used while running the web application.

Azure Entra setup guide link - Azure Entra ID (2025)

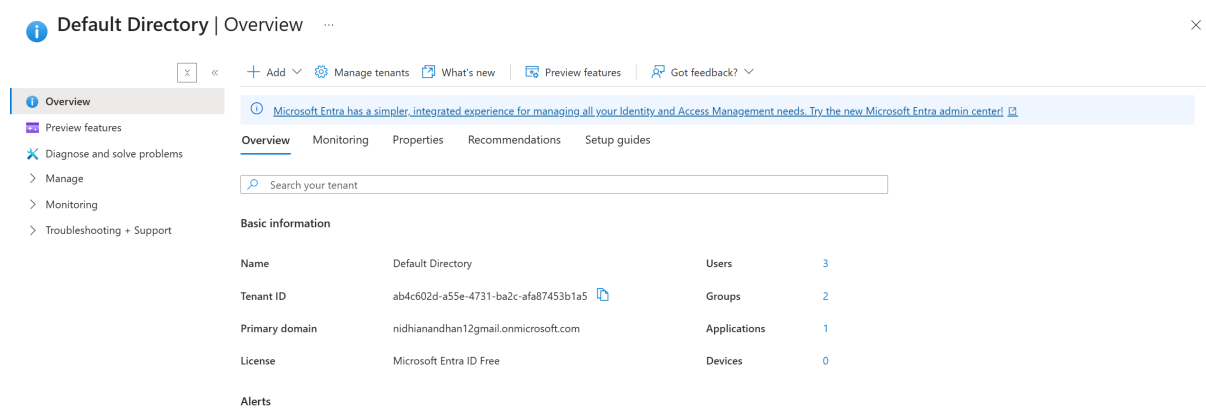


Figure 4: Azure Entra ID.

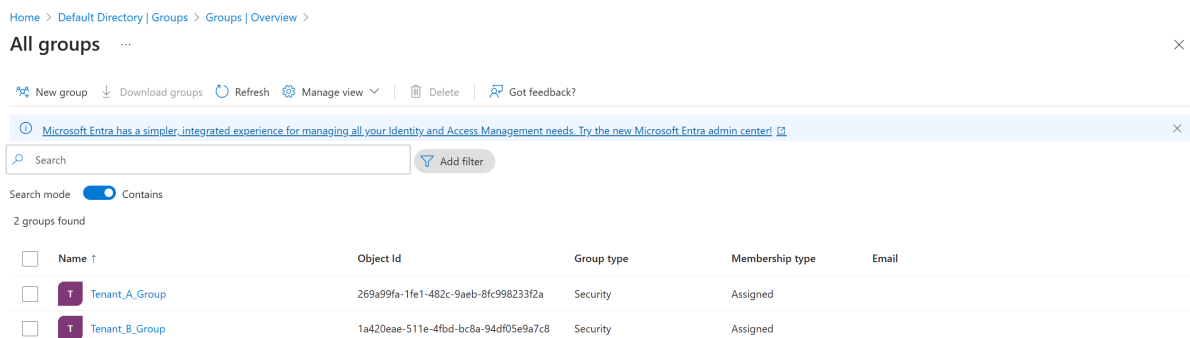


Figure 5: Azure Entra Groups.

2.2 Azure Cosmos Database Setup

A new Cosmos database needs to be created with any database name and other config details but to run this project a new cosmos account needs to be created as shown in fig.6 also database and container needs to be created. Database name - tenantdb, container - Task and add some sample data of partition keys with document types like Task Assignment and Work Assignment As shown in fig. 7. Also copy the connection

string, primary keys and secondary keys which will be used to access the database the keys are available in keys section as shown in fig.8

Azure Cosmos setup guide link - [Azure Cosmos Database \(2025\)](#)

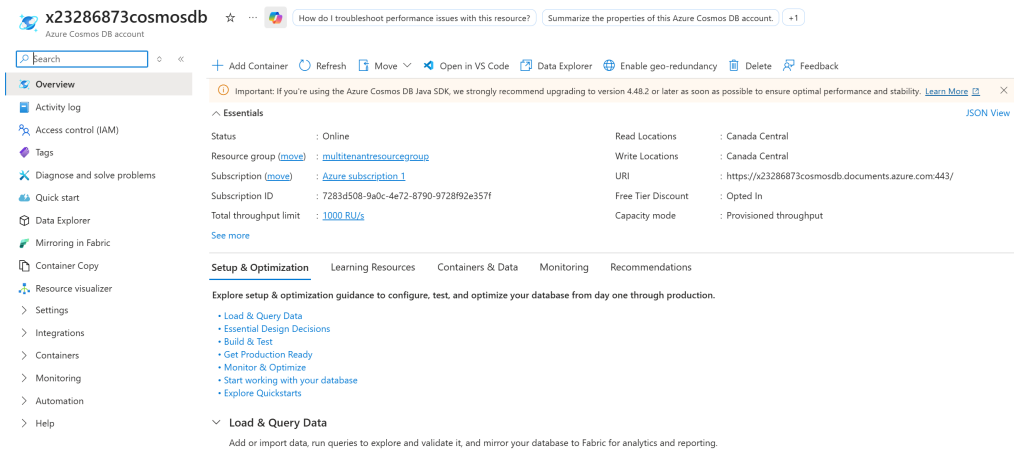


Figure 6: Azure Cosmos Database.

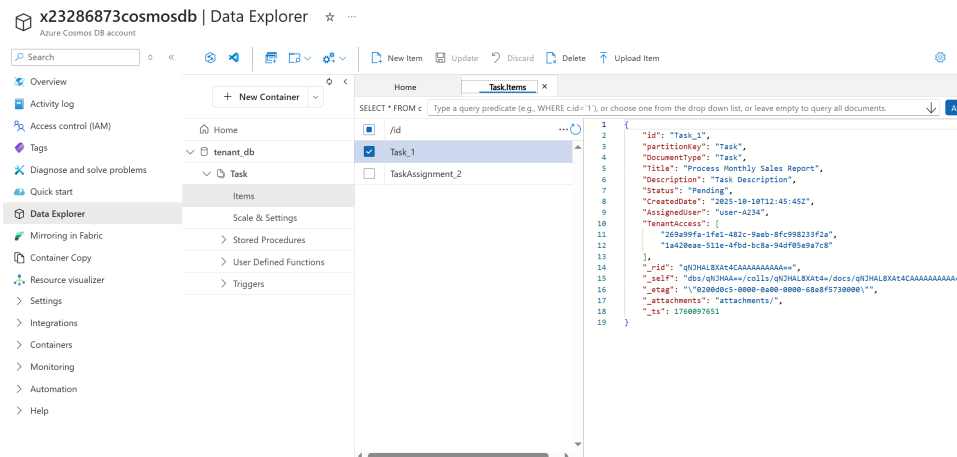


Figure 7: Azure Cosmos Database Sample Data.

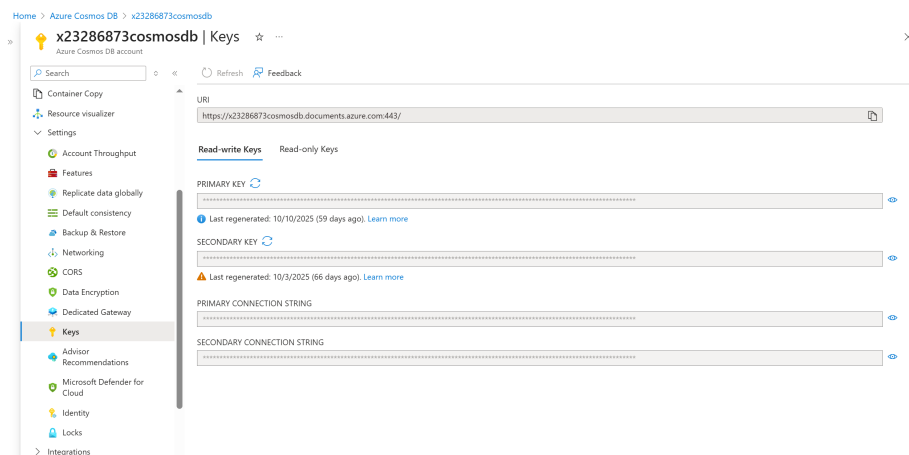


Figure 8: Azure Cosmos Database Key.

2.3 Azure Function API Setup

A new Azure function app which offers serverless API's needs to be created where the API's will be hosted as shown in fig.9 we can also push the latest changes to the function app if needed directly from visual studio.

Azure Function App setup guide link - Azure Function App (2025)

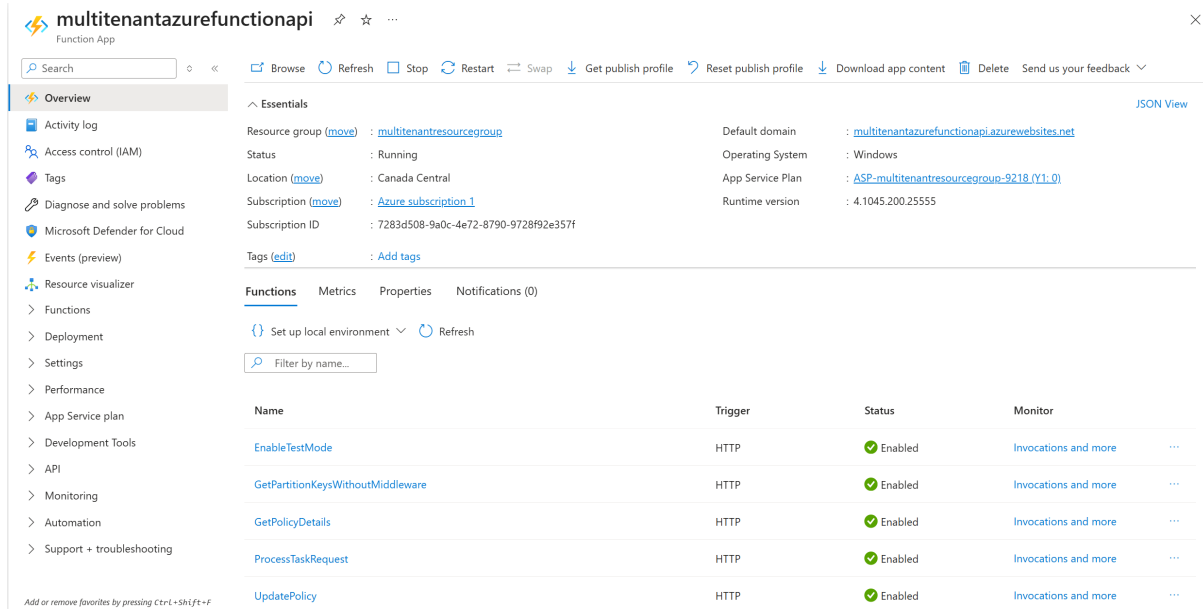


Figure 9: Azure Function API.

2.4 Azure App Service For Web Application

This is an optional step which needs to be done only if the web application needs to be hosted on the cloud. A new Azure app service needs to be created to host the web application with runtime version of .Net 8 as shown in fig.10. currently the application can be accessed in this url <https://ncimultitenantresearchproject.azurewebsites.net> which can be logged in with this email id - demouser@nidhianandhan12gmail.onmicrosoft.com and password - Ireland@2025

Azure App Service setup guide link - Azure App Service (2025)

2.5 Azure App Configuration Setup

Azure App configuration is used to store sensitive config values of .Net Core applications which can be accessed during runtime by the web application and Function App the following config values needs to be created as shown in the fig. 11

Azure App Configuration setup guide link - Azure App Configuration (2025)

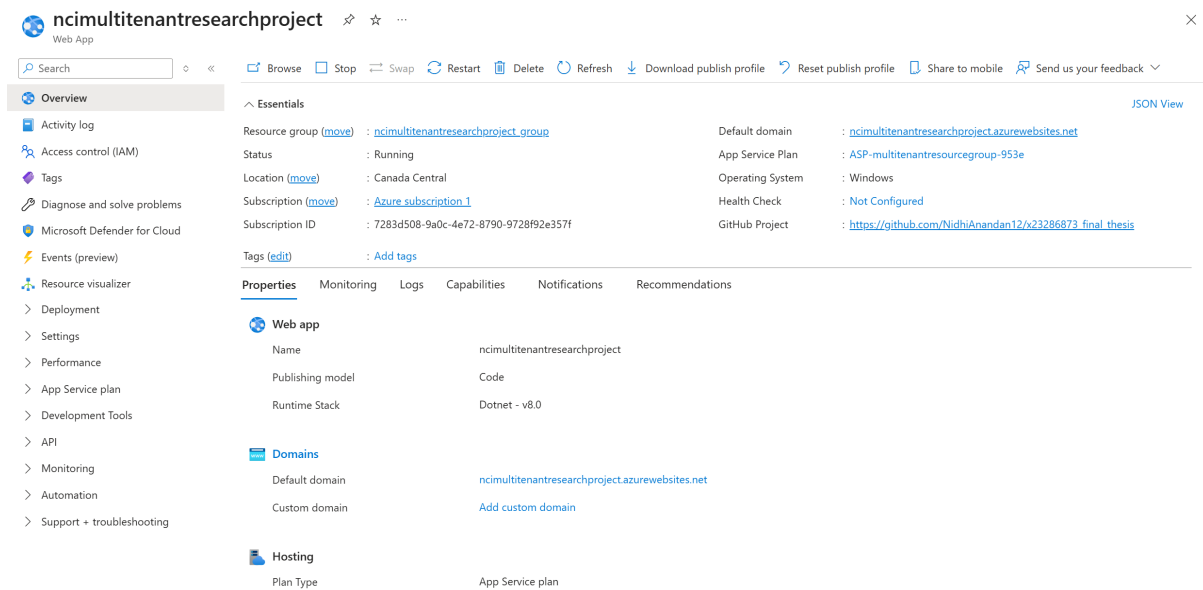


Figure 10: Azure Web Application App Service.

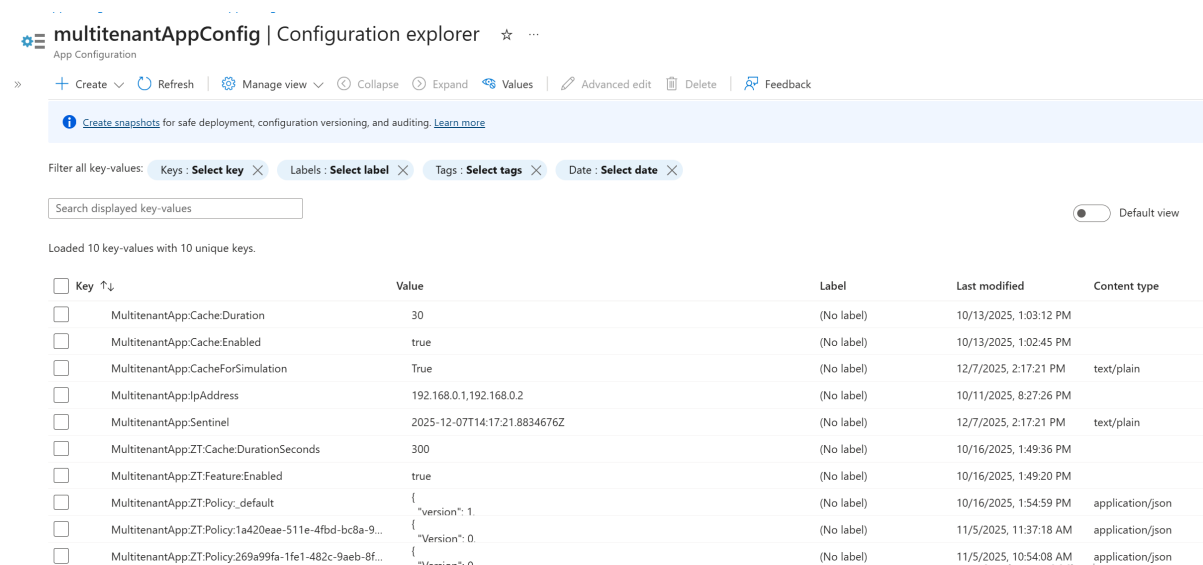


Figure 11: Azure App Configuration.

The following json should be added for in app configuration for each tenant level policy as application/json content type.

```
{
  "version": 1,
  "tenantId": "269a99fa-1fe1-482c-9aeb-8fc998233f2a",
  "displayName": "Tenant B",
  "allowedPartitions": [
    "TaskAssignment"
  ],
  "rules": {
    "ipAddress": {
      "enabled": true,
      "weight": 10,
      "mode": "AllowList",
      "allowedCidrs": [
        "192.168.0.5"
      ]
    },
    "geoLocation": {
      "enabled": true,
      "weight": 10,
      "allowedCountries": [
        "DE",
        "IE"
      ]
    },
    "accessTime": {
      "enabled": true,
      "weight": 10,
      "timezone": "UTC",
      "allowedWindows": [
        {
          "days": [
            "Mon",
            "Tue",
            "WED"
          ],
          "start": "08:00",
          "end": "18:00"
        }
      ]
    },
    "readAccess": {
      "enabled": false,
      "weight": 80
    },
    "writeAccess": {
      "enabled": false,
```

```

        "weight": 90
    },
    "denyAccess": {
        "enabled": false,
        "weight": 70
    }
},
"thresholds": {
    "low": 70,
    "medium": 80,
    "high": 90
},
"actions": {
    "onLow": "deny",
    "onMedium": "allow-read",
    "onHigh": "allow-write"
}
}

```

3 Repository Application Setting Configuration

3.1 Admin Web Application App Settings

Replace the app settings values in the repository with the newly created Azure Infrastructure based on entra id, function API url config values but for demo purposes this are the existing values in the repository.

```

{
  "AzureAd": {
    "Instance": "https://login.microsoftonline.com/",
    "Domain": "nidhianandhan12gmail.onmicrosoft.com",
    "TenantId": "ab4c602d-a55e-4731-ba2c-afa87453b1a5",
    "ClientId": "777813d1-6c67-4e74-b1ee-0ce920e9ba92",
    "CallbackPath": "/signin-oidc",
    "ClientSecret": "AJq8Q~QIXizUJOnJd2TpEsesXzdaKKkhUPA20bKc"
  },
  "Graph": {
    "BaseUrl": "https://graph.microsoft.com/v1.0",
    "Scopes": "User.Read GroupMember.Read.All"
  },
  "FunctionAppUrl": {
    "GetAllPartitionKeysUrl": "",
    "GetAllPartitionKeysWithMiddlewareUrl": "",
    "EnableTestModeUrl": ""
  },
  "AllowedHosts": "*"
}

```

3.2 Azure Function API App Settings

Replace the app settings values in the repository with the newly created Azure Infrastructure based on cosmos database (primary key, connection string) config values but for demo purposes we can leave the existing values as it is in the program.cs file as shown in 12.

```
builder.Services.AddSingleton<ICosmosDataAdapter>(options =>
{
    // Retrieve configuration values from the "AzureCosmoDBSettings" section
    //string url = builder.Configuration.GetSection("CosmosDb").GetValue<string>("URL");
    //string primaryKey = builder.Configuration.GetSection("CosmosDb").GetValue<string>("PrimaryKey");
    //string dbName = builder.Configuration.GetSection("CosmosDb").GetValue<string>("DatabaseId");
    //string containerName = builder.Configuration.GetSection("CosmosDb").GetValue<string>("ContainerId");

    string url = "https://x23286873cosmosdb.documents.azure.com:443/";
    string primaryKey = "Rku3bLEzPuMikOfTA7RgW0L9sWA0V8BNZKocFY6f2NJV32cIcm1pNKEsm1uMz9irak2JGIpeJg2DACDb7tTxyQ==";
    string dbName = "tenant_db";
    string containerName = "Task";

    // Initialize the CosmosClient
    var cosmosClient = new CosmosClient(url, primaryKey);

    // Return a new instance of TodoService, injecting the required dependencies
    return new CosmosDataAdapter(cosmosClient, dbName, containerName);
});
```

Figure 12: Azure Function App Settings.

3.3 Azure Function Execution using Swagger

You can execute the Azure Function directly from the Admin Dashboard under Azure Functions Tab, as shown in fig. 13. we can pass the tenant id and valid details corresponding to the tenant and get the API response. Some important details for each tenant ID is Tenant A - 269a99fa-1fe1-482c-9aeb-8fc998233f2a Tenant B - 1a420eae-511e-4fbd-bc8a-94df05e9a7c8

3.4 NuGet Package of Zero Trust Middleware

This middleware is also packaged as a reusable Nuget Component. Nuget is a package manager for .Net. which means that other SaaS developers can easily reuse this middleware in their .Net Core applications with minimal code changes. This aligns with the modular design principles suggested and discussed by researchers (Viswanathan et al.; 2024), as shown in fig. 14 this package can be installed using this command. `dotnet add package Azure-Function-API-ZeroTrust-Middleware --version 1.0.1`

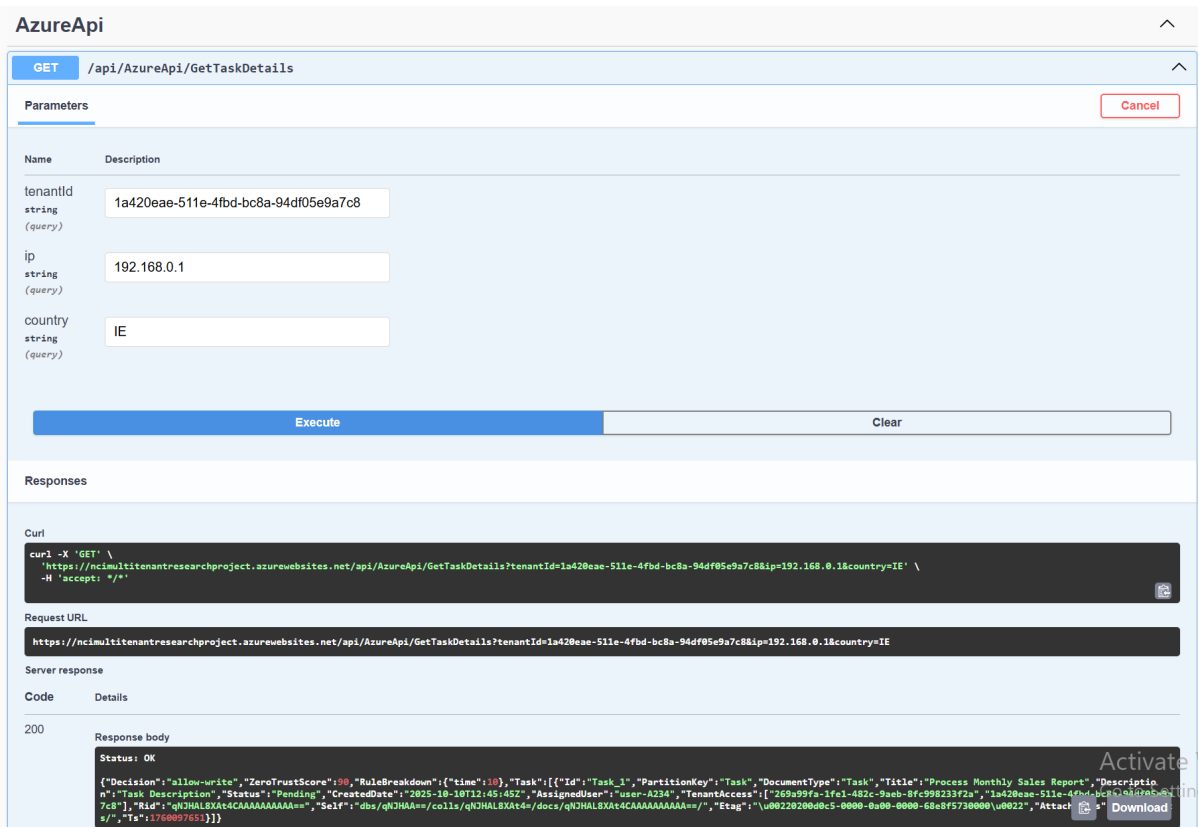


Figure 13: Azure Function App Settings.

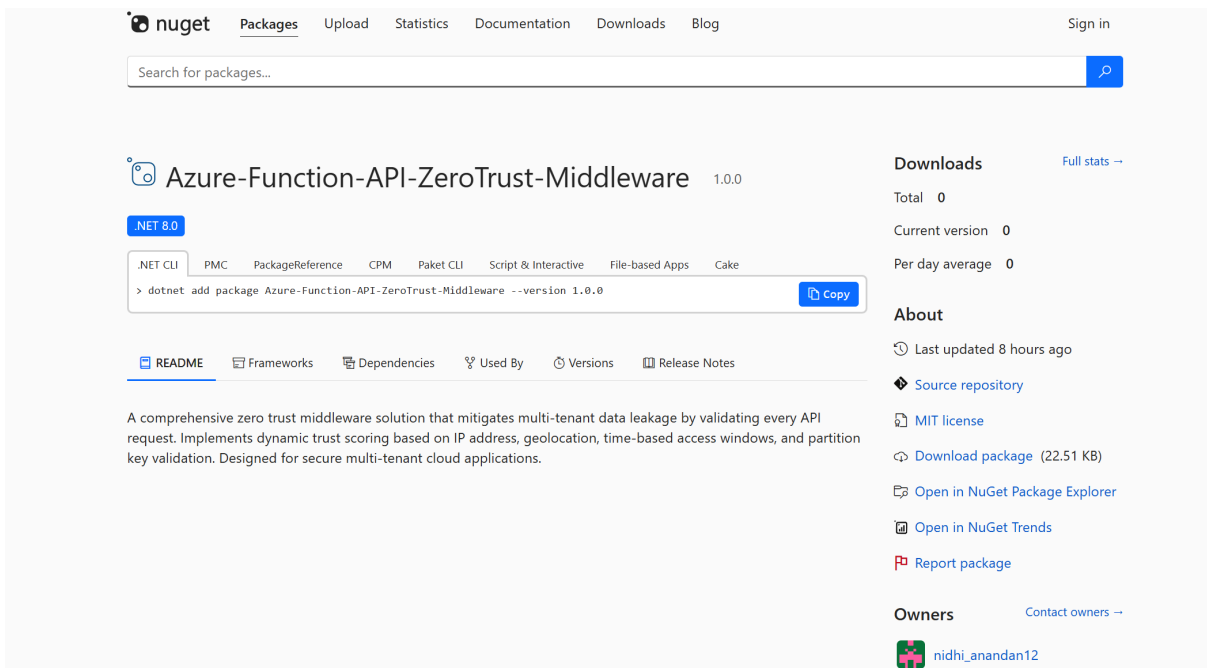


Figure 14: Nuget Package.

4 Performance Testing and Validation

4.1 K6 Load Test

Grafana k6 cloud scripts are used to perform performance and load testing of the middleware k6 can be installed using this command - `choco install k6`. Performance metrics can be viewed in Grafana dashboard. The entire Grafana Dashboard design can be exported directly as json format it is in this share point url - <https://nidhianandan12.grafana.net/d/nigqcgt/zero-trust-dashboard?orgId=1&from=now-6h&to=now&timezone=browser> and can be logged in using User Id - `nidhianandhan12@gmail.com` and password - `Ireland@2025`

This following k6 script is used to compare the performance of the API with and without the middleware.

```
import http from 'k6/http';
import { check, sleep } from 'k6';
import { Trend, Counter, Gauge } from 'k6/metrics';

// -----
// CUSTOM METRICS
// -----
export let latency = new Trend('latency_ms');
export let trustScore = new Trend('trust_score');
export let allowCount = new Counter('allow_count');
export let denyCount = new Counter('deny_count');
export let decisionType = new Counter('decision_type');
export let ipRuleHits = new Counter('rule_breakdown_ip');
export let geoRuleHits = new Counter('rule_breakdown_geo');
export let timeRuleHits = new Counter('rule_breakdown_time');
export let positiveRequests = new Counter('positive_requests');
export let negativeRequests = new Counter('negative_requests');

// NEW: total test duration per API
export let totalDuration = new Gauge('total_duration_ms');

// -----
// CONFIG
// -----
export let options = {
  vus: 10,
  iterations: 500,
};

// -----
// API URLs
// -----
const middlewareUrl =
  "https://multitenantazurefunctionapi.azurewebsites.net/api/ProcessTaskRequest;

const noMiddlewareUrl =
  "https://multitenantazurefunctionapi.azurewebsites.net/api/
```

```

        GetPartitionKeysWithoutMiddleware";

// -----
// TOTAL DURATION TRACKING
// -----
let startTimes = {
    middleware: 0,
    no_middleware: 0
};

let endTimes = {
    middleware: 0,
    no_middleware: 0
};

// -----
// Simulated Request Headers
// -----
const positiveHeaders = {
    'Content-Type': 'application/json',
    'x-tenant-id': '_default',
    'x-client-ip': '192.168.0.1',
    'x-country-code': 'IE',
};

const negativeHeaders = {
    'Content-Type': 'application/json',
    'x-tenant-id': '_default',
    'x-client-ip': '10.0.0.1',
    'x-country-code': 'US',
};

// -----
// TEST LOGIC
// -----
export default function () {

    const isPositive = Math.random() < 0.5;
    const headers = isPositive ? positiveHeaders : negativeHeaders;

    isPositive ? positiveRequests.add(1) : negativeRequests.add(1);

    // -----
    // 1) Middleware Request
    // -----

    if (startTimes.middleware === 0) {
        startTimes.middleware = new Date().getTime();
    }

    const resWith = http.get(middlewareUrl, { headers });

```

```

latency.add(resWith.timings.duration, { type: 'middleware' });
endTimes.middleware = new Date().getTime();

if (resWith.status === 200 || resWith.status === 403) {
  try {
    const body = JSON.parse(resWith.body);
    const decision = body.Decision;
    const score = body.ZeroTrustScore;
    const breakdown = body.RuleBreakdown || {};

    trustScore.add(score, { decision: decision });

    decision === "deny" ? denyCount.add(1) : allowCount.add(1);
    decisionType.add(1, { decision: decision });

    if (breakdown.ip) ipRuleHits.add(breakdown.ip);
    if (breakdown.geo) geoRuleHits.add(breakdown.geo);
    if (breakdown.time) timeRuleHits.add(breakdown.time);

  } catch (e) {
    console.log("JSON parse failed:", resWith.body);
  }
}

check(resWith, {
  'middleware status OK/403': r => r.status === 200 || r.status === 403
});

// -----
// 2) NO Middleware Request
// -----

if (startTimes.no_middleware === 0) {
  startTimes.no_middleware = new Date().getTime();
}

const resWithout = http.get(noMiddlewareUrl, { headers });

latency.add(resWithout.timings.duration, { type: 'no_middleware' });
endTimes.no_middleware = new Date().getTime();

check(resWithout, {
  'no_middleware status 200': r => r.status === 200
});

sleep(0.1);
}

```

5 Important Links And Details

- Admin dashboard: <https://ncimultitenantresearchproject.azurewebsites.net/>
User ID: demouser@nidhianandhan12gmail.onmicrosoft.com Password: Ireland@2025
- Grafana dashboard: <https://nidhianandan12.grafana.net/d/nigqcgzt/zero-trust-dashboard?orgId=1&from=now-6h&to=now&timezone=browser>
User ID: nidhianandhan12@gmail.com Password: Ireland@2025
- YouTube demo video: https://youtu.be/1_01tPtPUW0
- Code repository / SharePoint: https://studentncirl-my.sharepoint.com/personal/x23286873_student_ncirl_ie/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fx23286873%5Fstudent%5Fncirl%5Fie%2FDocuments%2FFinal%5FRepository&ga=1
- NuGet Package : <https://www.nuget.org/packages/Azure-Function-API-ZeroTrust-Middleware>
- Tenant Details
Tenant A ID - 269a99fa-1fe1-482c-9aeb-8fc998233f2a
Tenant B ID - 1a420eae-511e-4fbd-bc8a-94df05e9a7c8

References

Azure App Configuration (2025). Azure app configuration setup. Accessed: 2025-12-08.
URL: *<https://learn.microsoft.com/en-us/azure/azure-app-configuration/quickstart-azure-app-configuration-create?tabs=azure-portal>*

Azure App Service (2025). Azure app service setup. Accessed: 2025-12-08.
URL: *<https://learn.microsoft.com/en-us/azure/app-service/quickstart-dotnetcore?tabs=net80&pivots=development-environment-vs>*

Azure Cosmos Database (2025). Azure cosmos database setup. Accessed: 2025-12-08.
URL: *<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/how-to-dotnet-get-started>*

Azure Entra ID (2025). Azure entra id user groups setup. Accessed: 2025-12-08.
URL: *<https://learn.microsoft.com/en-us/training/modules/manage-users-and-groups-in-aad>*

Azure Function App (2025). Azure function app setup. Accessed: 2025-12-08.
URL: *<https://learn.microsoft.com/en-us/azure/azure-functions/functions-create-function-app-portal?tabs=core-tools&pivots=flex-consumption-plan>*

Github Repository Clone Guide (2025). Clone a git repository in visual studio. Accessed: 2025-12-08.

URL: *<https://learn.microsoft.com/en-us/visualstudio/version-control/git-clone-repository?view=visualstudio>*

Microsoft Visual Studio Download (2025). Visual studio 2026 download. Accessed: 2025-12-08.

URL: *<https://visualstudio.microsoft.com/downloads/>*

Viswanathan, J., Kumar, S. U. et al. (2024). Zero trust security for web applications in microservice-based environments, pp. 488–494.