National College of Ireland

Data security on cloud using hybrid cryptography a PGP based encryption methodology

MSc Research Project
Cloud Computing

Ali Saif

Student ID: x22155996

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ali Saif
Student ID:	x22155996
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Sean Heeney
Submission Due Date:	14/12/2023
Project Title:	Data security on cloud using hybrid cryptography a PGP based encryption methodology
Word Count:	1020
Page Count:	10

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:	Ali Saif
Date:	14th December 2023

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):	

Data Security on cloud using hybrid cryptography a PGP based encryption methodology

Ali Saif
x22155996

1 Introduction

This manual explains the requirements and deployment procedures for data security using hybrid cryptography a PGP approach which is a data encryption and decryption technique that offers cryptographic privacy and data authentication.Tiwari; (2022)Praveen KumarTripathi, RatneshKumarShukla PGP is commonly used to secure data, but it can also be used to secure email transfer. PGP provides a high level of security by combining symmetric-key encryption and public-key cryptography.ofShen (2021)Yiming Shen .2021 PGP mechanism on Azure Cloud,

The following sections of report comprise the remaining portions of the entire document. Module 2 gives details of the system's configuration. Libraries required for Module 3, Database Tables in Module 4, Module 5: Using the PGP method Section 6: Cloud Implementation.Sousi et al. (2020)Dalia Yehya

2 System configuration requirements

2.1 ASP .Net Environment Setup

The application is written in the C# programming language. The code for the project is been written in Visual Studio Community 2022 on the.NET platform. Visual Studio has been downloaded and installed for free from the internet. Version 15.0 of Visual Studio 2022 is the preferred text editor.

2.2 Database Server Setup

The project stores application data in a Microsoft SQL database. With the help of this technology, I have linked my application to local and cloud data storage. Microsoft SQL Management Studio 18.11.1 is the version that is being used. .
The user needs to enter the authentication credentials in order to connect to the database. Online, open source versions of the SQL Management tool are available for free.

2.3 Pre-Requisite for the project implementation

.
-Install visual studio Community 2022.

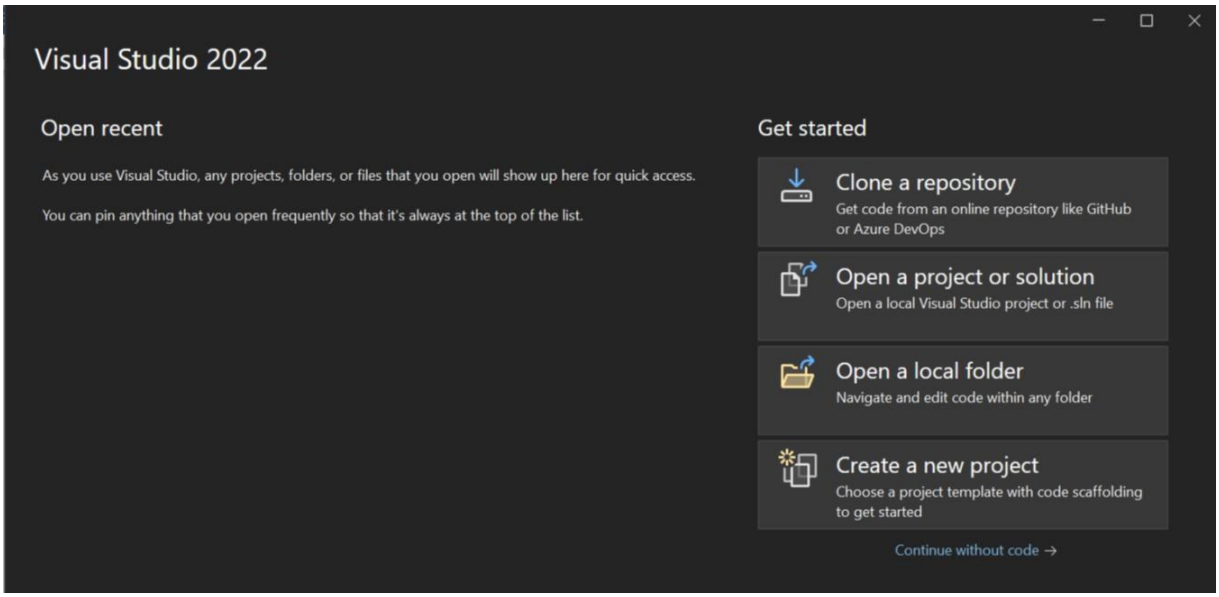


Figure 1: Workflow of the proposed system

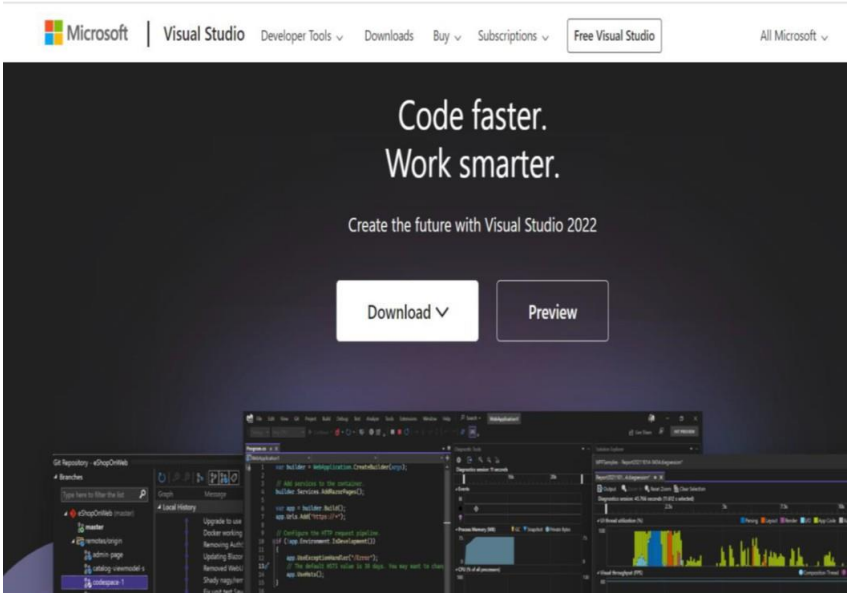


Figure 2: Download page of Microsoft Visual Studio

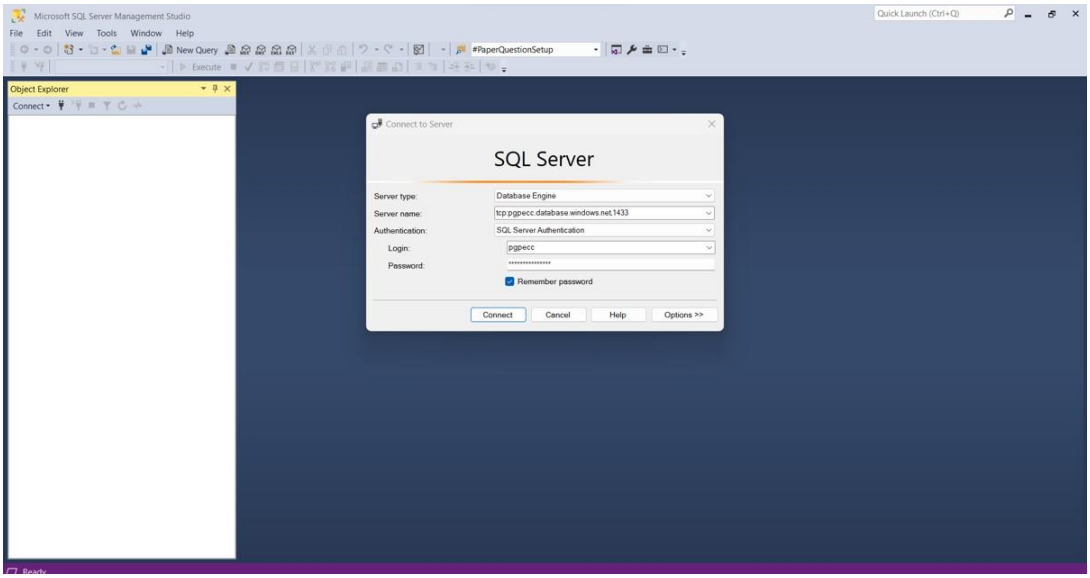


Figure 3: SQL Server Setup

-Microsoft SQL Server Management Studio.

-Microsoft Azure cloud account.

2.4 Hardware Requirements

.

- Processor: Intel i5 2.30 GHz .

-RAM: 16.0 GB .

-System congfig: x64- processor,64-bit operating system

2.5 Software specifications

.

- Windows 11.

- Language: C # .Net Framework.

- IDE used: Visual Studio 2022.

- Microsoft Sql Server 2019.

- SQL Server Management Studio.

.

2.6 Running Web Application in Visual Studio

For running the web API,First open: PGP Encryption_ECC folder having following solution file :PGP Encryption_ECC.sln at VS Code 2022 , as in Fig 1

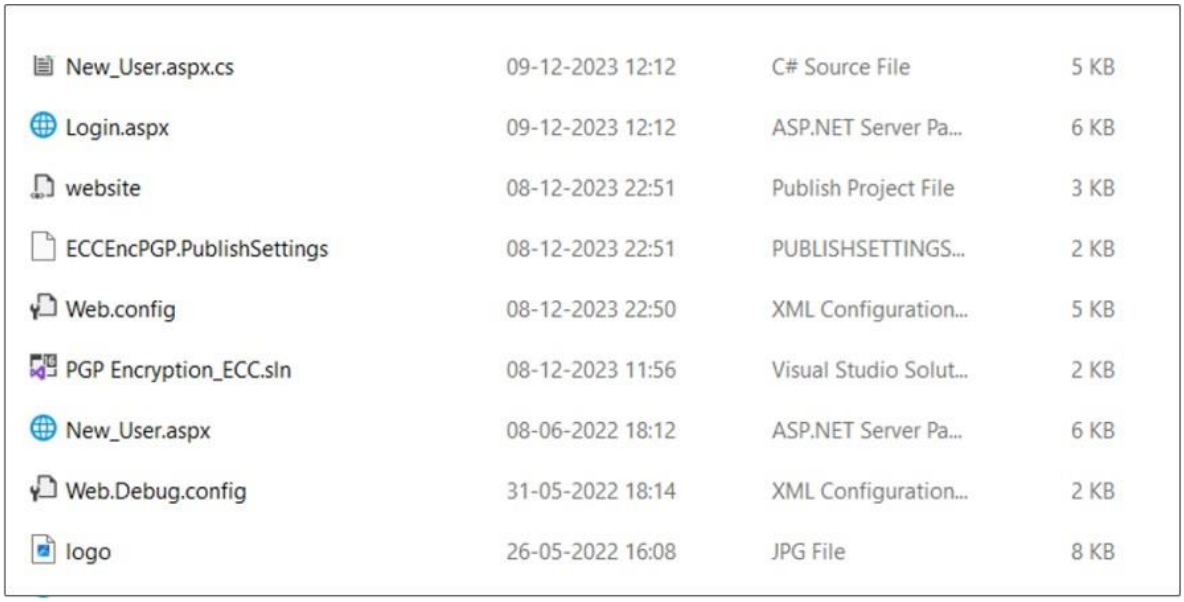


Figure 4: SQL Server Setup

2.7 Connecting SQL SERVER for DATABASE

.

3 Implementation of PGP Technique

Both symmetric and asymmetric methods are used to implement the PGP technique. As the data is encrypted with the help of symmetric method (AES), the secret key that is generated is encrypted using an asymmetric technique (ECC). It is essential to perform decryption of the secret key before decrypting the data. The encrypted secret key is sent to the user via e-mail.

.

4 Cloud Deployment

4.1 New Application Creation

The whole application is deployed on cloud. Projects on Azure can be created in practically any language and integrate cloud-based public services. This program makes use of the Windows Azure cloud service.

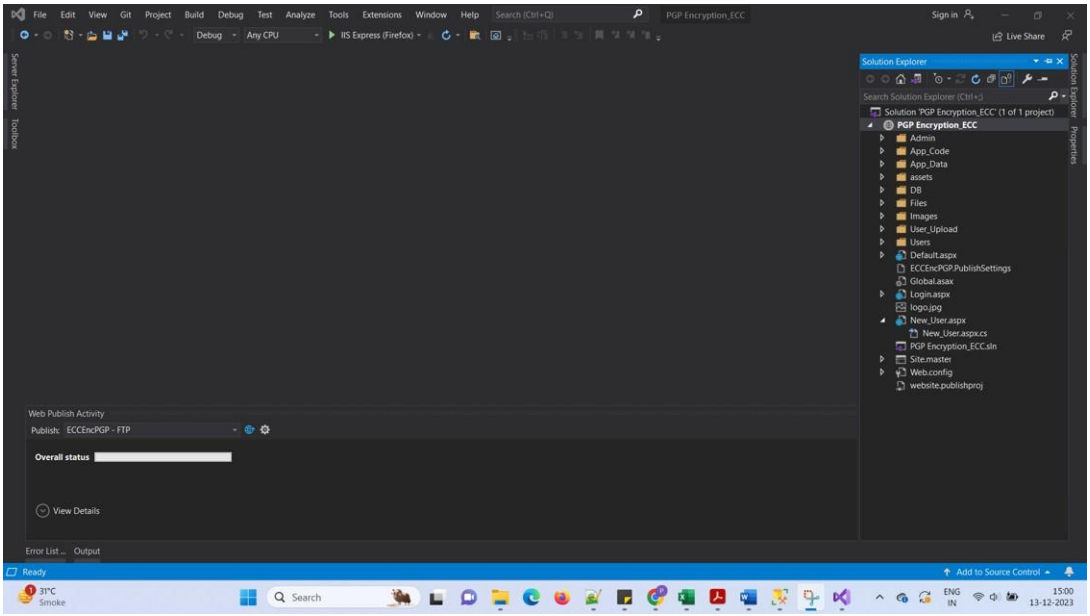


Figure 5: Project loaded in Visual Studio

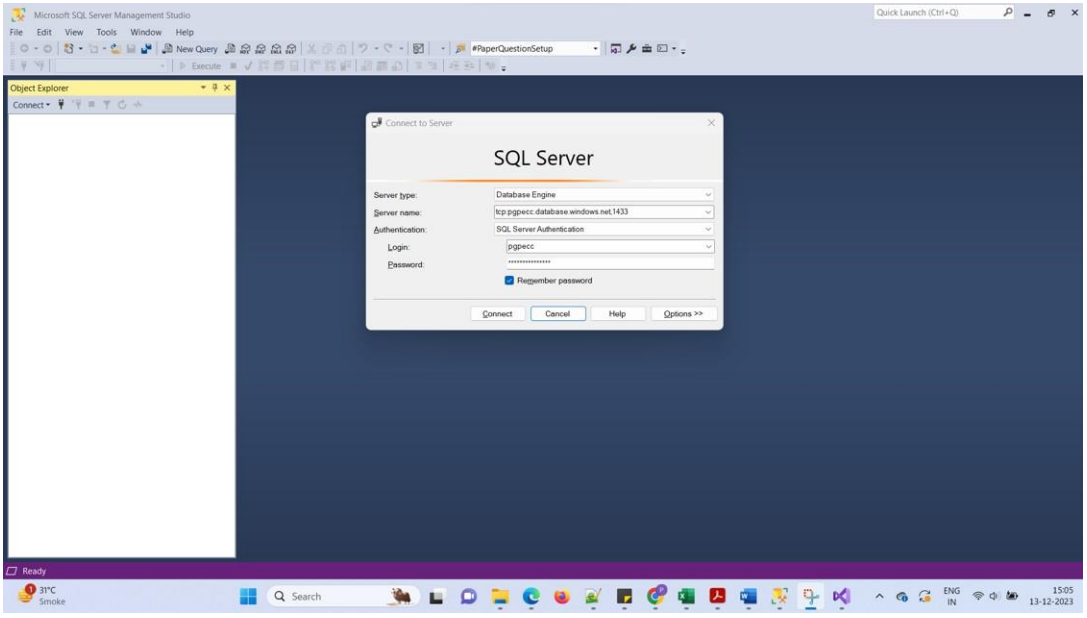
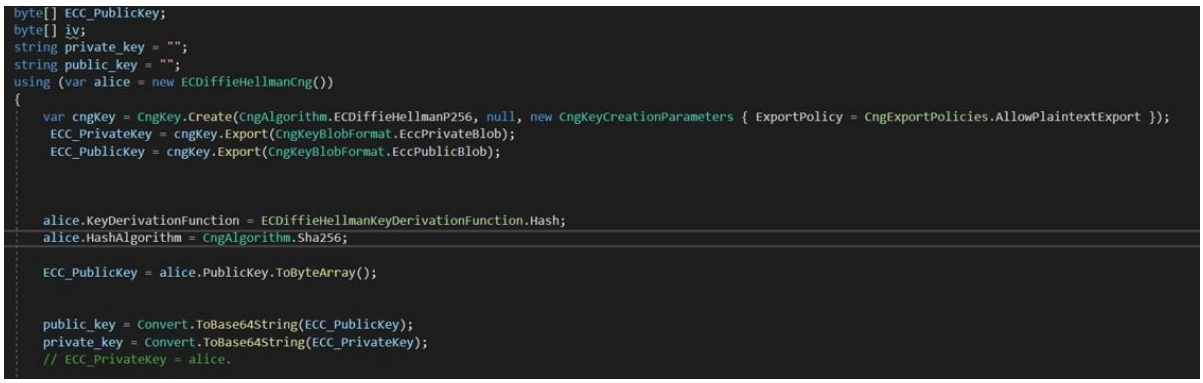


Figure 6: SQL Server Setup

```
string EncryptionKey = key;
using (Aes encryptor = Aes.Create())
{
    Rfc2898DeriveBytes pdb = new Rfc2898DeriveBytes(EncryptionKey, new byte[] { 0x49, 0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76 });
    encryptor.Key = pdb.GetBytes(32);
    encryptor.IV = pdb.GetBytes(16);
    using (FileStream fsOutput = new FileStream(outputFilePath, FileMode.Create))
    {
        using (CryptoStream cs = new CryptoStream(fsOutput, encryptor.CreateEncryptor(), CryptoStreamMode.Write))
        {
            using (FileStream fsInput = new FileStream(inputFilePath, FileMode.Open))
            {
                int data;
                while ((data = fsInput.ReadByte()) != -1)
                {
                    cs.WriteByte((byte)data);
                }
            }
        }
    }
    FileInfo fi = new FileInfo(inputFilePath);

    long size = fi.Length;
```

Figure 7: Code for AES Key Generation



```
byte[] ECC_PublicKey;
byte[] iv;
string private_key = "";
string public_key = "";
using (var alice = new ECDiffieHellmanCng())
{
    var cngKey = CngKey.Create(CngAlgorithm.ECDiffieHellmanP256, null, new CngKeyCreationParameters { ExportPolicy = CngExportPolicies.AllowPlaintextExport });
     ECC_PrivateKey = cngKey.Export(CngKeyBlobFormat.EccPrivateBlob);
     ECC_PublicKey = cngKey.Export(CngKeyBlobFormat.EccPublicBlob);



    alice.KeyDerivationFunction = ECDiffieHellmanKeyDerivationFunction.Hash;
    alice.HashAlgorithm = CngAlgorithm.Sha256;

    ECC_PublicKey = alice.PublicKey.ToByteArray();


    public_key = Convert.ToBase64String(ECC_PublicKey);
    private_key = Convert.ToBase64String(ECC_PrivateKey);
    // ECC_PrivateKey = alice.
```

Figure 8: Code for ECC Key Generation

```
string EncryptionKey = key;
using (Aes encryptor = Aes.Create())
{
    Rfc2898DeriveBytes pdb = new Rfc2898DeriveBytes(EncryptionKey, new byte[] { 0x49, 0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76 });
    encryptor.Key = pdb.GetBytes(32);
    encryptor.IV = pdb.GetBytes(16);
    using (FileStream fsOutput = new FileStream(outputFilePath, FileMode.Create))
    {
        using (CryptoStream cs = new CryptoStream(fsOutput, encryptor.CreateEncryptor(), CryptoStreamMode.Write))
        {
            using (FileStream fsInput = new FileStream(inputFilePath, FileMode.Open))
            {
                int data;
                while ((data = fsInput.ReadByte()) != -1)
                {
                    cs.WriteByte((byte)data);
                }
            }
        }
    }
    FileInfo fi = new FileInfo(inputFilePath);

    long size = fi.Length;
```

Figure 9: Code for AES Key Generation

Figure 10: Configuration File for Database Connection

4.2 Creation of Azure App Service

Middleware API is hosted by Azure App service. This is where web service deployment happens. How to host an online service:

· Open the Azure dashboard.

· Select Azure App Service and establish a new service.

· Go with the ASP.NET 4.8 runtime stack selection.

· Once App Service has been created, download ”Get Publish profile.”

· To publish to the web server (Azure), utilize the publish profile in your code structure.

· After copying the publish profile to the Visual Studio workspace folder. When you do a right-click on the project, choose ”Publish.” The Azure account will receive publication of the App service layer. The account and app service data are contained in the publish settings.

.

· At the time it is published, copy the domain URL.

.

4.3 Create Cloud Database

4.3.1 Creation of Azure SQL Database

Go to SQL database and create an instance to create a database. Once a database instance has been built. The connection string is information in the code base that allows you to connect to a database. The screenshot depicts the option to generate Database in Cloud. The database connection details are different in cloud platform

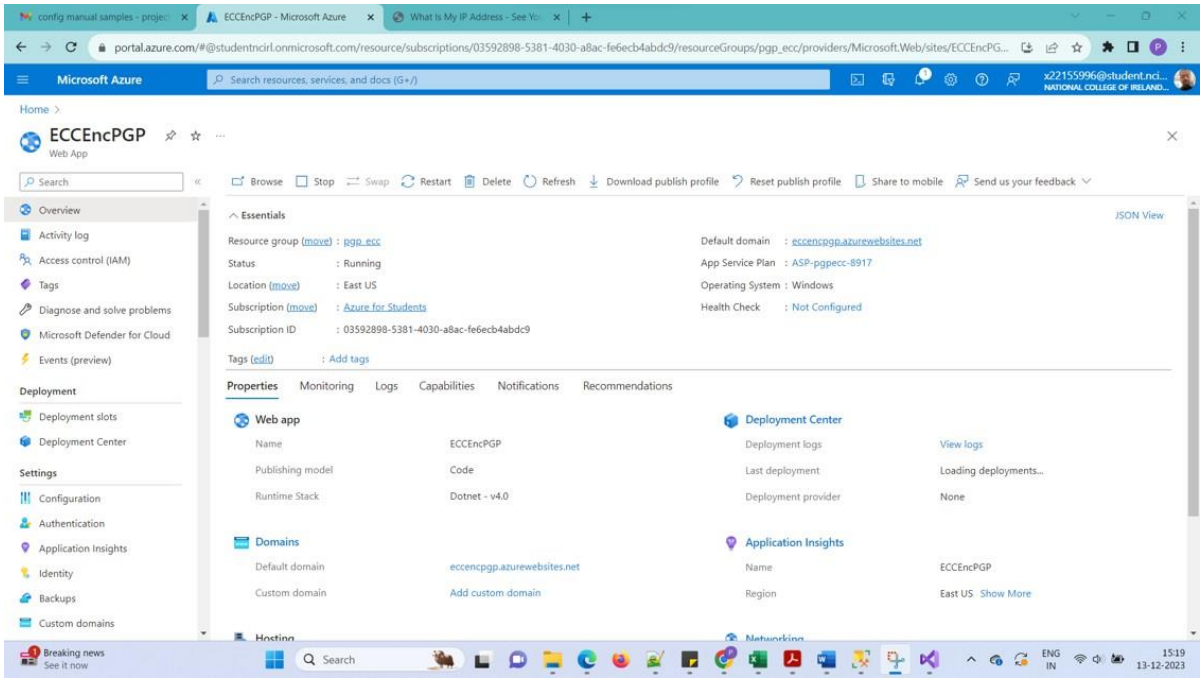


Figure 11: Application Deployment in Cloud

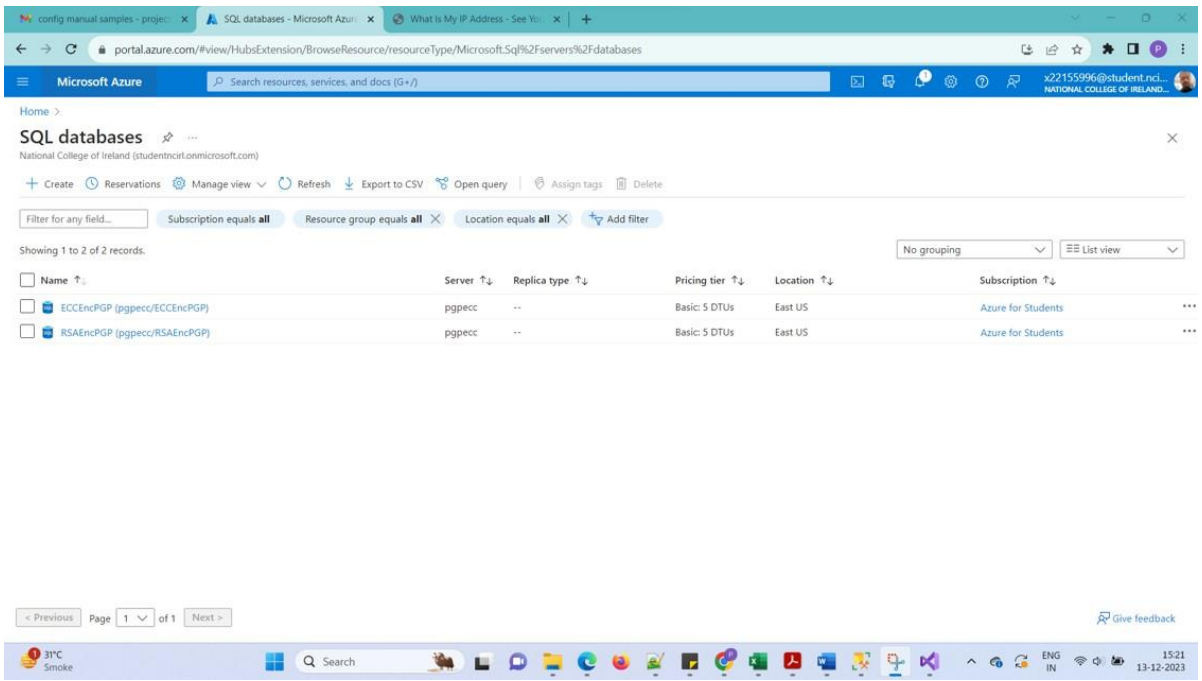


Figure 12: Create Database

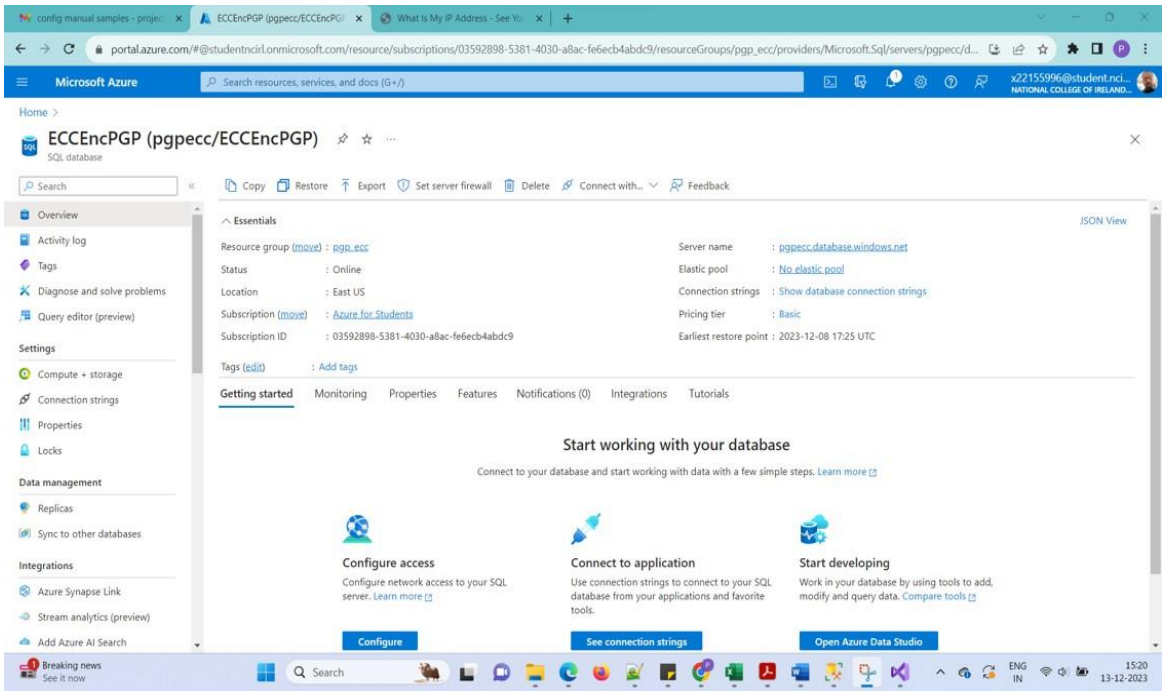


Figure 13: Database created for Application in Cloud

4.4 Publishing the Updated details in Azure Cloud Environment

There may be an upgrade to the current code base, the same ought to be released on the cloud. In a cloud environment, local changes can be updated. An option to publish modifications from a local workstation to a cloud environment is shown in Fig. 14 below.

4.4.1 Get Publish Profile details from Azure

.

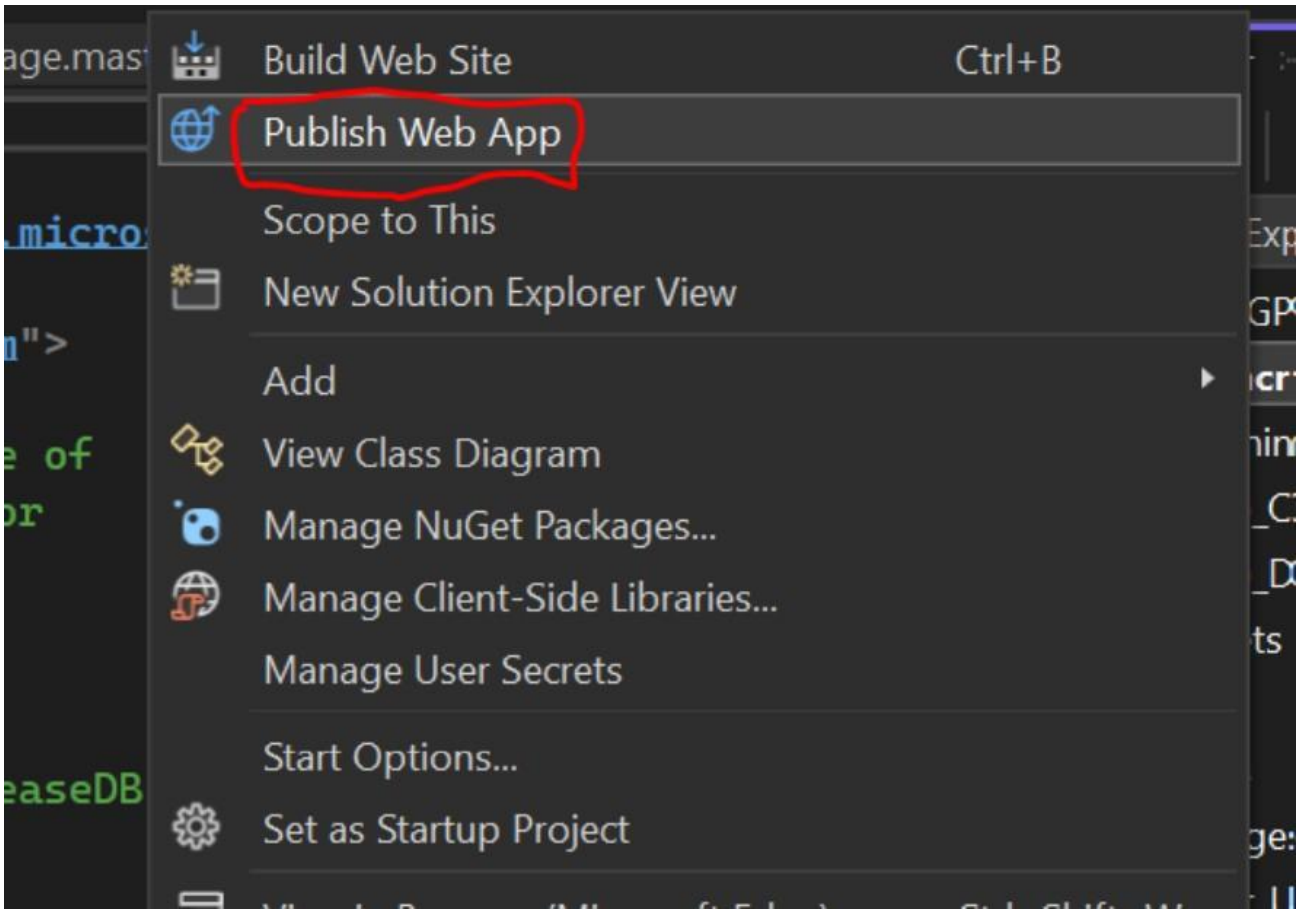


Figure 14: Publish Code Changes in Cloud

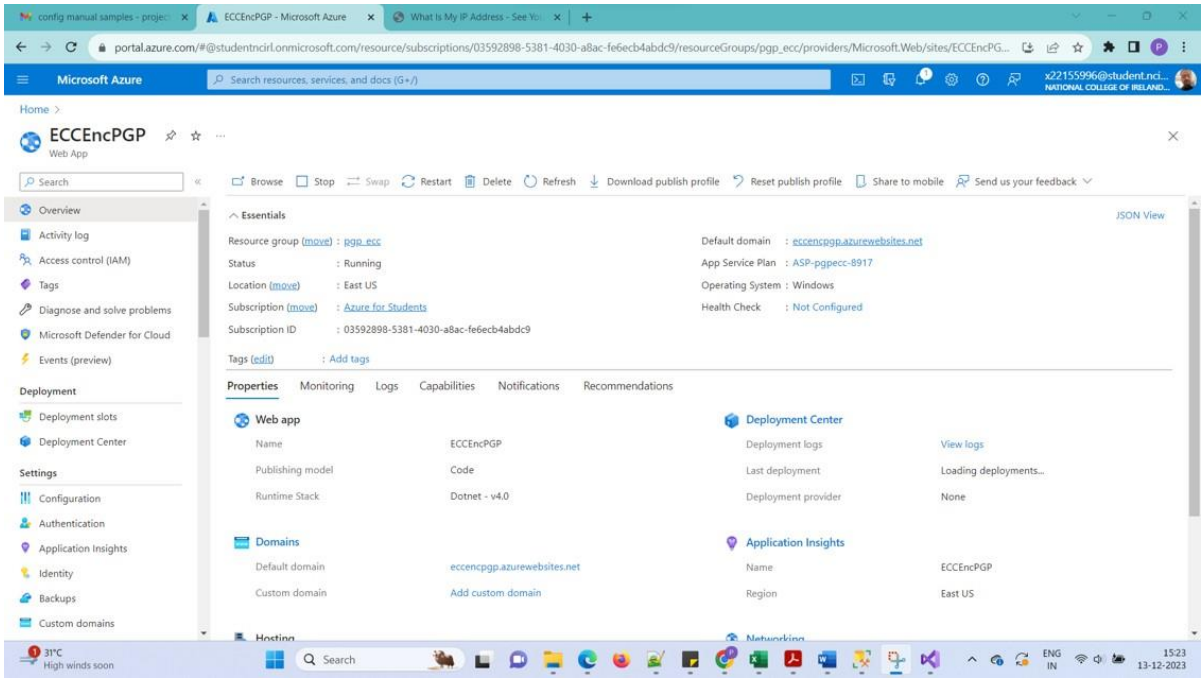


Figure 15: Get Publish Profile details from Azure

References

Shen, Y. (2021). End-to-end encrypted messaging based on pgp with forward secrecy, *Journal of Physics: Conference Series* **27**(5): 1–20.
URL: *https://iopscience.iop.org/article/10.1088/1742-6596/1873/1/012031/pdf*

Sousi, A.-L., Yehya, D. and Joudi, M. (2020). Es encryption: Study evaluation, p. 20.
URL: *https://www.researchgate.net/publication/346446212*$_A$*ES*$_E$*ncryption*$_S$*tudy*$_E$*valuation*

Tiwari;, P. K. T. R. K. S. N. K. (2022). Enhancing security of pgp with steganography, *11th International Conference on System Modeling Advancement in Research Trends (SMART)* **7**(1): 12–21.
URL: *https://ieeexplore.ieee.org/document/10046709*