National College of Ireland

Configuration Manual

MSc Research Project
Programme Name

Aishwarya Gatla
Student ID: x22172297

School of Computing
National College of Ireland

Supervisor: Shreyas Setlur Arun

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Aishwarya Gatla
Student ID:	x22172297
Programme:	Programme Name
Year:	2023-2024
Module:	MSc Research Project
Supervisor:	Shreyas Setlur Arun
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	8

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:	Aishwarya Gatla
Date:	13th 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):	

Configuration Manual

Aishwarya Gatla
x22172297

1 Introduction

As organisations increasingly use cloud technology to store and handle sensitive data, ensuring strong security measures becomes increasingly important. This need is addressed by the project "Enhancing Cloud Security Using a Hybrid Approach with AES and 3DES Methods," which implements a comprehensive security approach within the Amazon Web Services (AWS) environment. This setup handbook walks you through the process of installing and configuring the system to create a secure cloud architecture.

2 System Configuration

2.1 Software Specification

• Programming Language: Python
• Web Framework: Django
• Database: RDS Database PostgreSQL
• Email: Gmail account required for accessing the key

2.2 Hardware Requirement

• Processor: Intel ® Core ™ i5-CPU@ 1.60GHz
• RAM: 8.00 GB
• System Type: 64-bit operating system, x64-based processor

2.3 Encryption Algorithms

• AES (Advanced Encryption Standard)
• DES (Data Encryption Standard)
• 3DES (Triple DES)

2.4 Cloud Services

• AWS Elastic Beanstalk: Used for web deployment, ensuring a streamlined process with scalability and load balancing.
• EC2 Instance: Configured with essential security parameters and application requirements

3 Project Implementation

The Python programming language and the Django framework were used to create the complete application.The project's code was written in Visual Studio Code (VSCode). VSCode was chosen since it is a free application that is compatible with different systems and allows for programming in a variety of languages. The default text editor, however, is Visual Studio 2022's version 15.0.Ascertain that Python is installed on your system.

3.1 Environment Setup

The steps for setting the Django development environment in Visual Studio Code (VSC) are as follows:

3.1.1 Install Python

Python can be downloaded from the official website python.org and installed by following the installation instructions for your operating system. In my system, I have installed version 3.8.

3.1.2 Install VSC

Install Visual Studio Code, a powerful and free source-code editor accessible for a variety of operating systems. You can get it at code.visualstudio.com and install it by following the installation instructions.

3.1.3 Creating/Opening Django Project

Open Visual Studio Code and either utilise the File menu's "Open Folder" option or drag the freshly formed Django project folder into VSC as shown in the figure 1. This operation opens the project for editing in VSC.

3.1.4 Set Up Virtual Environment

You can use Virtual Environment to build a virtual environment within your project directory. Use the command as shown in Figure 2.

3.2 Database Server Setup

For the storing of application data, the project employs a Postgres database. With the help of this technology, we can connect our application to both local and cloud data storage. Navigate to RDS in the AWS Management Console, choose PostgreSQL as the engine, establish instance parameters, set access controls, and activate the RDS instance.The Postgres version used is 2.9.9.

3.3 Libraries Used

The requirements.txt file specifies the particular versions or constraints for the project's needed libraries and packages. The libraries that are used in this project are specified in the figure 5 and 6.

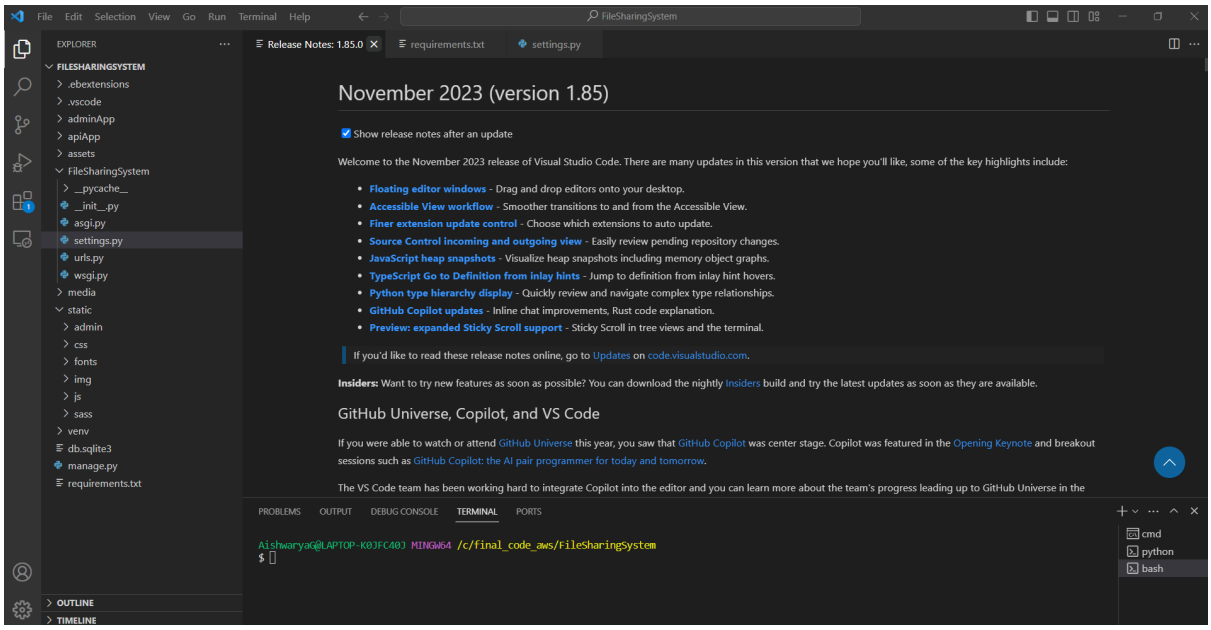

Figure 1: VSC

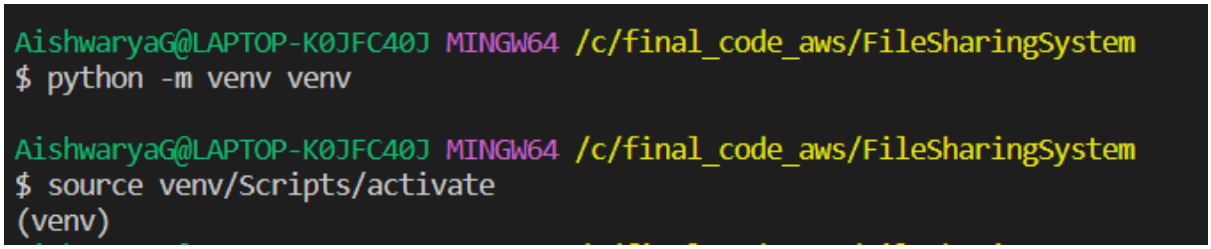

Figure 2: Command to setup environment and activate it

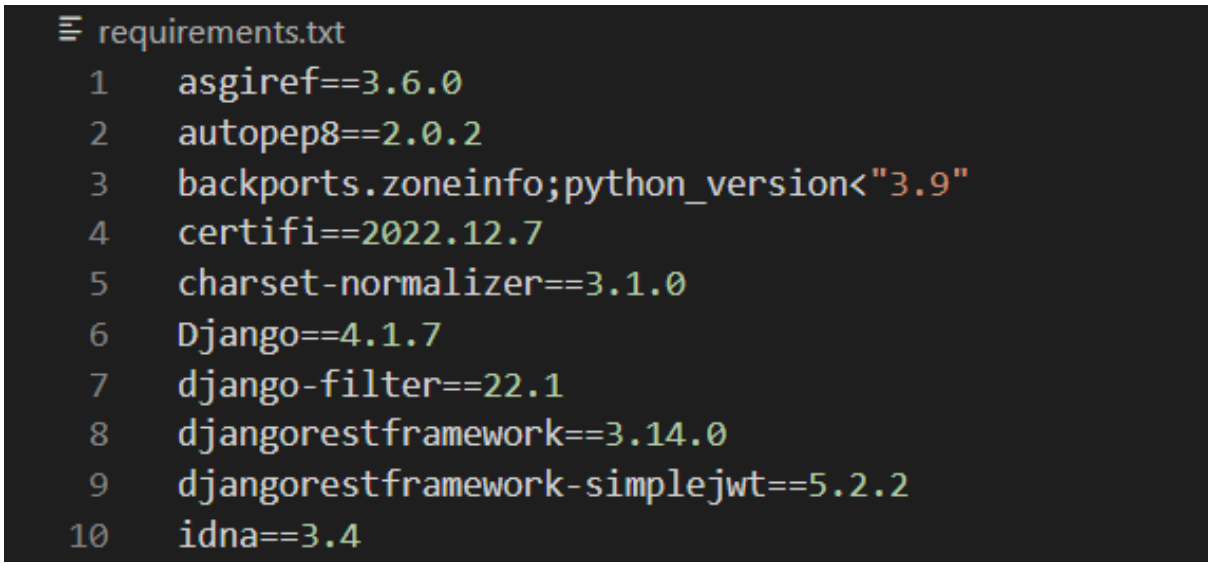

Figure 3: Imported Libraries

```
importlib-metadata==6.0.0
Markdown==3.4.1
Pillow==9.4.0
pycodestyle==2.10.0
PyJWT==2.6.0
pytz==2022.7.1
requests==2.28.2
sqlparse==0.4.3
tomli==2.0.1
urllib3==1.26.14
zipp==3.15.0
cryptography
pycryptodome
django-json==0.27
psycopg2
```

Figure 4: Imported Libraries

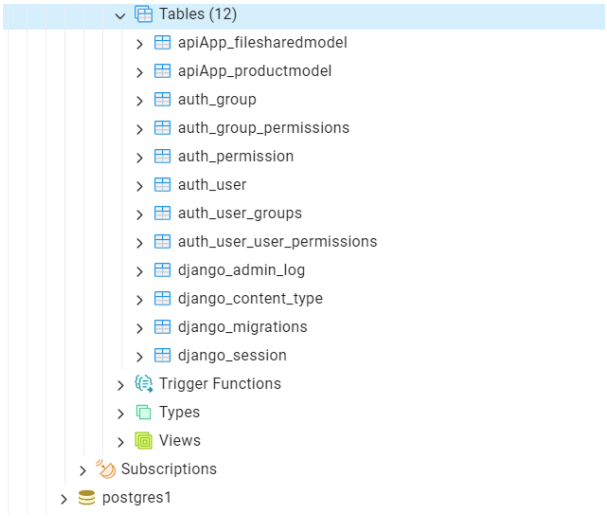

Figure 5: Database Tables

3.4 Database Tables Used

4 Deployment

The implementation incorporates powerful encryption algorithms, such as (AES+DES) and (AES+3DES), within the application's source to ensure data security.To begin, configure Amazon RDS and PostgreSQL to securely store encrypted data and user credentials. AWS services deliver important security parameters, Python/Django frameworks, and application requirements to EC2 instances. For web deployment, Elastic Beanstalk is employed, which speeds the process while ensuring scalability and load balancing.

4.0.1 Terraform should be installed in the AWS CloudShell

The AWS CloudShell is a browser-based shell that allows you to securely manage, examine, and interact with your AWS resources quickly and easily. Terraform may be installed via AWS CloudShell.Unknown (n.d.c)

1. Navigate to the AWS Management Console.
2. Click on the AWS CloudShell icon on the console navigation bar
3. With the AWS CloudShell type in:
   git clone https://github.com/tfutils/tfenv.git /.tfenv
4. Now, type the following command to create a symlink for tfenv/bin/* scripts onto the path /bin:
   ln -s /.tfenv/bin/ /bin/*
5. We may now install Terraform using the Terraform Version Manager. Enter the following command:
   tfenv install

The Figure 8 shows the terraform script used to for deploying the application on Elastic Beanstalk.

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

provider "aws" {
  region = "ap-south-1"
}

resource "aws_elastic_beanstalk_application" "my_app" {
  name = "MyElasticBeanstalkApp1"
}

resource "aws_elastic_beanstalk_environment" "my_environment" {
  name        = "MyElasticBeanstalkApp1"
  application = aws_elastic_beanstalk_application.my_app.name
  solution_stack_name = "64bit Amazon Linux 2 v3.5.9 running Python 3.8"

  setting {
    namespace = "aws:autoscaling:launchconfiguration"
    name      = "InstanceType"
    value     = "t2.micro"
  }

  setting {
    namespace = "aws:elasticbeanstalk:environment"
    name      = "EnvironmentType"
    value     = "SingleInstance"
  }

  setting {
    namespace = "aws:elasticbeanstalk:application:environment"
    name      = "PYTHONPATH"
    value     = "/opt/python/current/app:/opt/python/run/venv/lib/python3.8/site-packages"
  }
   setting {
```

Figure 6: Terraform Script

4.0.2 CLI for AWS Elastic Beanstalk

The EB Command Line Interface (CLI) is a tool for managing and deploying AWS Elastic Beanstalk applications and environments.Unknown (n.d.b) The command to install is :
$pip install awsebcli

To begin using Elastic Beanstalk, you must first create an AWS account. Register or log in to the AWS account.

4.0.3 Configure EB – Initialize Your App

With the AWS Elastic Beanstalk CLI operational, the first step is to construct a Beanstalk environment on which to host the application. This should be run from the project directory.Unknown (n.d.a)
$ eb init

4.0.4 Configure EB – Create an Environment

Command to create environment
$ eb create

4.0.5 Database Setup

Navigate to AWS Management Console, and search for RDS. Click on create database as shown in figure 7. Then, proceed as follows:

1. Select the "Configuration" option.
2. Scroll all the way to the bottom of the page, and then click the link "create a new RDS database" under the "Data Tier" section.
3. Change the "DB Engine" setting on the RDS setup page to "postgres".
4. Create a "Master Username" and a "Master Password" for yourself.
5. Save your changes.

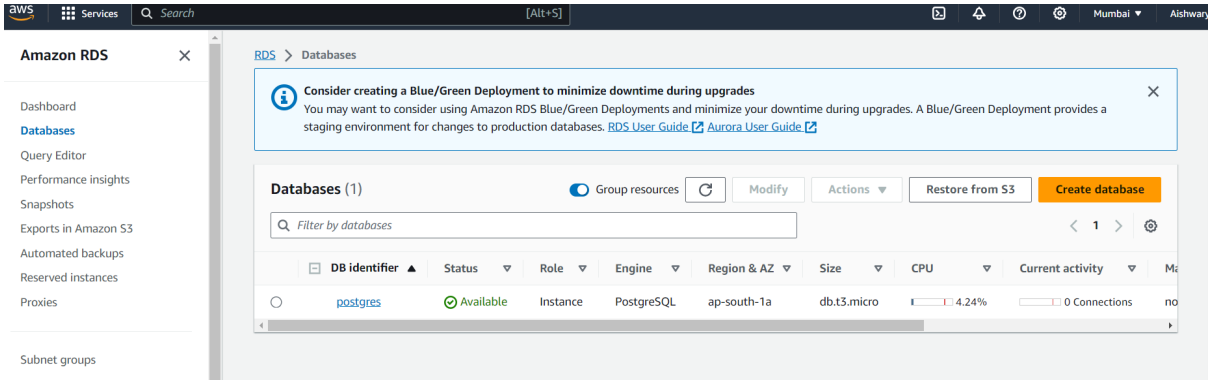

Figure 7: RDS: Create Database

The database configuration specifics can be found in the figure 8 and 9.

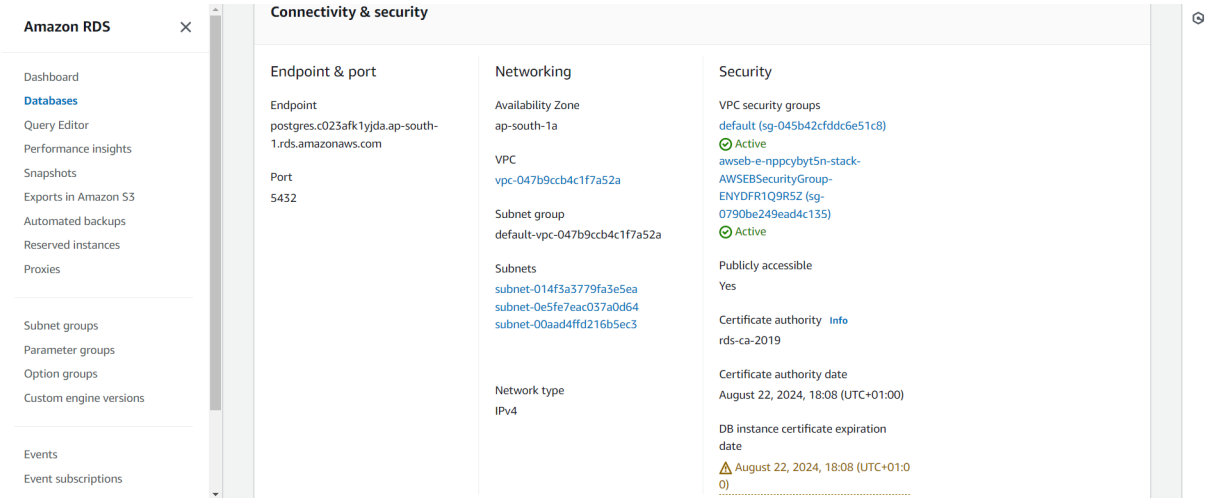

Figure 8: Postgres Configuration in AWS

Security group	Type	Rule
awseb-e-nppcybyt5n-stack-AWSEBSecurityGroup-ENYDFR1Q9R5Z (sg-0790be249ead4c135)	CIDR/IP - Inbound	0.0.0.0/0
awseb-e-nppcybyt5n-stack-AWSEBSecurityGroup-ENYDFR1Q9R5Z (sg-0790be249ead4c135)	CIDR/IP - Outbound	0.0.0.0/0
default (sg-045b42cfddc6e51c8)	EC2 Security Group - Inbound	sg-045b42cfddc6e51c8
default (sg-045b42cfddc6e51c8)	CIDR/IP - Outbound	0.0.0.0/0

Replication (1)

Filter by Replication

1

DB identifier	Role	Region & AZ	Replication source	Replication state	Lag
postgres	Instance	ap-south-1a	-	-	-

Figure 9: Security Roles

The RDS will be created for you by Beanstalk. Now we need to link our Django programme to the RDS. Beanstalk will assist us in this situation by providing a number of environment variables on the EC2 instances that specify how to connect to the Postgres server. To use those environment variables, we simply need to alter our settings.py file.

References

Unknown (n.d.a). Deploying a django app and postgresql to aws elastic beanstalk.
URL: *https://realpython.com/deploying-a-django-app-and-postgresql-to-aws-elastic-beanstalk/*

Unknown (n.d.b). Install terraform in the aws cloudshell.
URL: *https://www.techielass.com/install-terraform-in-the-aws-cloudshell/*

Unknown (n.d.c). Terraform on aws cloudshell.
URL: *https://github.com/t2yijaeho/Terraform-on-AWS-CloudShell*