National College of Ireland

Developing Interoperable Blockchain Protocols for Secure Data Migration In Multi Cloud Environments

Saeed Adetugboboh
Student ID: 23212365

Configuration Manual

School of Computing
National College of Ireland

Supervisor: REJWANUL HAQUE

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. SAEED ADETUGBOBOH

Student ID: ……X23212365……..……

Programme: CLOUD COMPUTING **Year:** ……1……..

Module: ……RESEARCH PROJECT……

Supervisor: ……REJWANUL HAQUE……..……

Submission Due Date: ……..……

Project Title: ……Developing Interoperable Blockchain Protocols for Secure Data Migration in Multi-Cloud Environments

Word Count: ……1710…… **Page Count**…9…

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: ……SAEED ADETUGBOBOH……

Date: ……3RD DECEMEBER 2024…

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

Saeed Adetugboboh
Student ID: 23212365

1 Introduction

This document provides a wide guide for configuring the blockchain-based multi-cloud data migration framework. The guide includes the prerequisites, step-by-step setup instructions, and troubleshooting techniques to ensure easy deployment. The framework takes advantage of blockchain for secure data migration, ensuring data integrity, compliance, and interoperability across multiple cloud platforms (AWS, Azure, GCP). Users are expected to have basic familiarity with blockchain, cloud environments, and system configurations.]

2 Setup Prerequisites

Component	Type/Version
Operating System	Ubuntu 20.04 / Windows 10+
Programming Language	Node.js v16+ / python 3.9+
Blockchain Network	Ethereum Sepolia Test
Cloud Tools	AWS CLI , Azure CLI , Google Cloud SDK
Storage	AWS S3 , Azure blob, GCP storage

Ensure that your machine meets the hardware requirements:

- **Minimum**: 4-core CPU, 8 GB RAM, 50 GB free disk space
- **Recommended**: 8-core CPU, 16 GB RAM, 100 GB SSD

3 AWS setup

You can create and configure an S3 bucket using either the AWS Management Console GUI. Follow the steps below for the preferred method:

3.1.1 Creating Bucket

1. Log in to the AWS Management Console.
2. Navigate to **S3** from the "Services" menu.
3. Click **Create bucket**.
4. In the "Create bucket" form:
    - Enter a unique bucket name.

- Select your desired AWS region.
- Configure the settings of enabling encryption and block public and versioning

5. Click **Create bucket** to finish the setup.
6. Note your bucket name and region for future use.

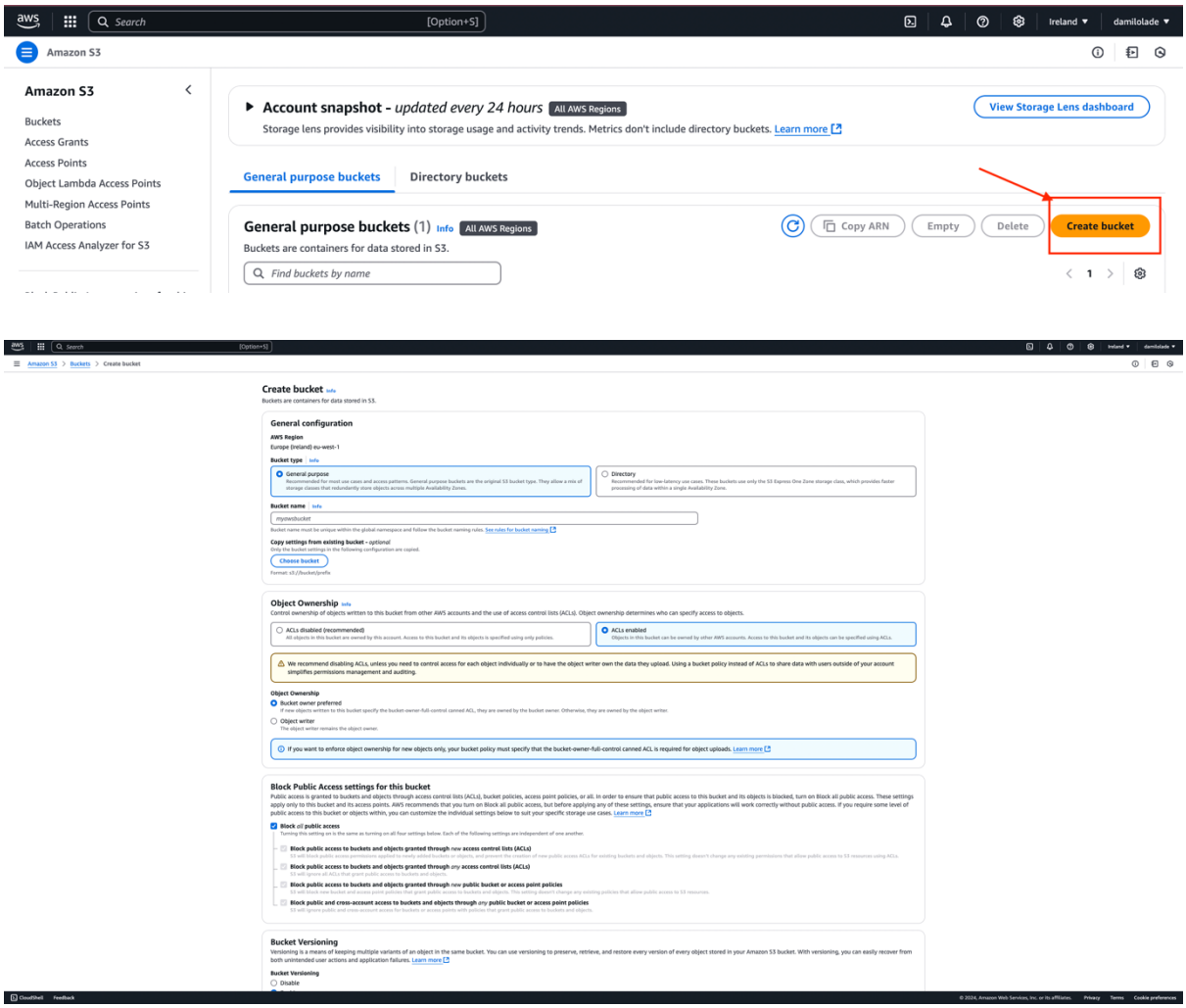


Figure 1. Creation of s3 bucket

3.1.2 Configuring API Keys

Generate AWS API keys via the **IAM Console** or AWS CLI:

GUI

I. Navigate to the **IAM Console**, create a new user, and attach the "AmazonS3FullAccess" policy.
II. Download the generated **Access Key ID** and **Secret Access Key**.
III. Store the keys as it will be needed later

CLI:

Use the following command to generate access keys:

```
aws iam create-access-key --user-name your-user-name
```

I. Add the keys to your .env file:

```
AWS_ACCESS_KEY_ID=your-access-key
AWS_SECRET_ACCESS_KEY=your-secret-key
```

Set Bucket Permissions

1. Ensure ACL is enabled

```
aws s3api put-bucket-acl --bucket your-bucket-name --acl public-read-write
```

4 Azure setup

1. Log in to the Azure Portal.
2. Navigate to **Storage Accounts** from the "All Services" menu.
3. Click **+ Create** to create a new storage account.

In the "Create a Storage Account" form:

- **Subscription**: Select your subscription.
- **Resource Group**: Choose an existing group or create a new one.
- **Storage Account Name**: Enter a globally unique name for your storage account.
- **Region**: Select your preferred region.
- **Performance**: Choose between Standard or Premium based on your requirements.
- **Replication**: Select your desired replication option (e.g., Locally Redundant Storage (LRS)).

4. Click **Review + Create** and then **Create** to deploy your storage account.
5. Once created, navigate to **Containers** in the storage account dashboard.
6. Click **+ Container** to create a new container:
   - **Name**: Enter a container name.
   - **Public Access Level**: Set as "Private".

4.1.1 Configuring API Keys

- Generate AWS API keys via the **IAM Console** or AWS CLI:

```
az storage account show-connection-string --name your-storage-account --
resource-group your-resource-group
```

5 Google cloud setup

1. Log in to the Google Cloud Console.
2. Navigate to Storage from the "Products and Services" menu.
3. Click + Create Bucket.
4. In the "Create a Bucket" form:
5. Bucket Name: Enter a globally unique name.
6. Location Type: Choose. “Region" and select a location.

7. Storage Class: Choose a class “Standard”
8. Access Control: Select "Uniform"
9. Click Create to finalize the bucket.

5.1.1 Configuring API Keys

- Navigate to **IAM & Admin → Service Accounts**.
- Click **+ Create Service Account**.
- Assign roles such as "Storage Admin."
- Download the JSON key file.

6 Blockchain Setup

6.1.1 6.1 Set Up Metamask:

- Install the Metamask browser extension.
- Create a wallet or import an existing one.
- Add the **Sepolia Testnet**:
    - Open Metamask → Click on the Network dropdown → Select "Add Network".
    - Use the following details:

```
Network Name: Sepolia Testnet
RPC URL: https://rpc.sepolia.org
Chain ID: 11155111
Currency Symbol: ETH
```

    - Click **Save**.

6.1.2 6.2 Obtain Sepolia Test ETH:

- Go to the Google Cloud Web3 Faucet.
- Enter your Metamask wallet address.
- Click **Receive 0.05 Sepolia ETH**.
- Verify the balance in your Metamask wallet under the Sepolia network.

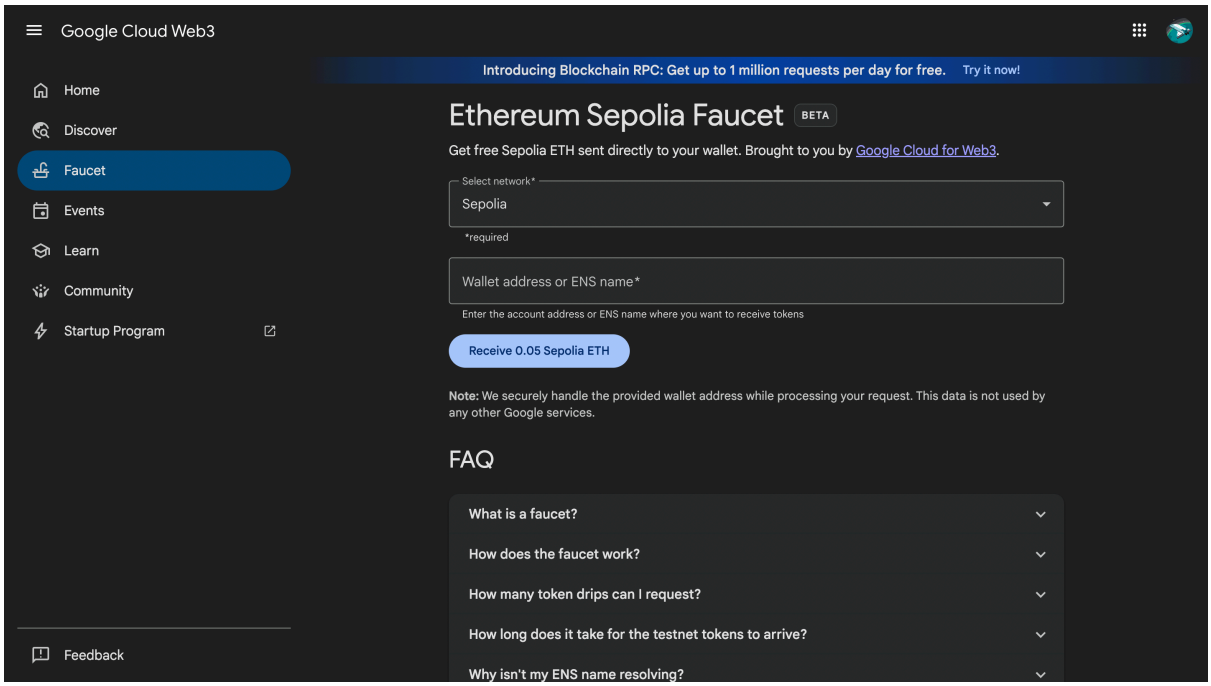


Figure 2. Faucet setup

6.1.3 Deploy Smart Contract Using Remix:

- Open Remix IDE.
- Create a new file in the contracts folder, e.g., MyContract.sol.
- Paste your Solidity code.
- Compile the contract using the Solidity Compiler in Remix.
- Connect your Metamask wallet to Remix:
    - In the Deploy & Run Transactions tab, select Injected Web3 as the environment.
    - Choose the account with Sepolia Testnet network in Metamask.
- Deploy the contract by clicking Deploy using the other default settings.

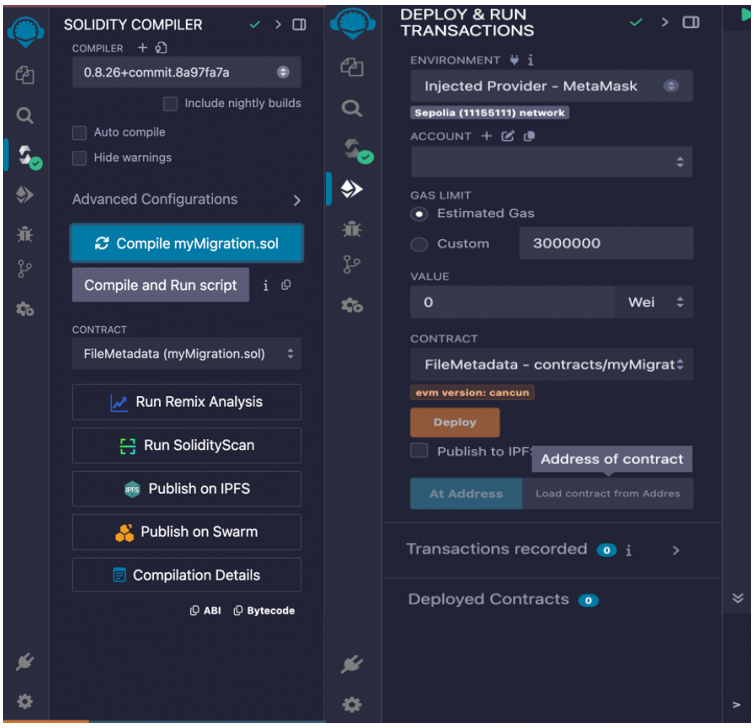
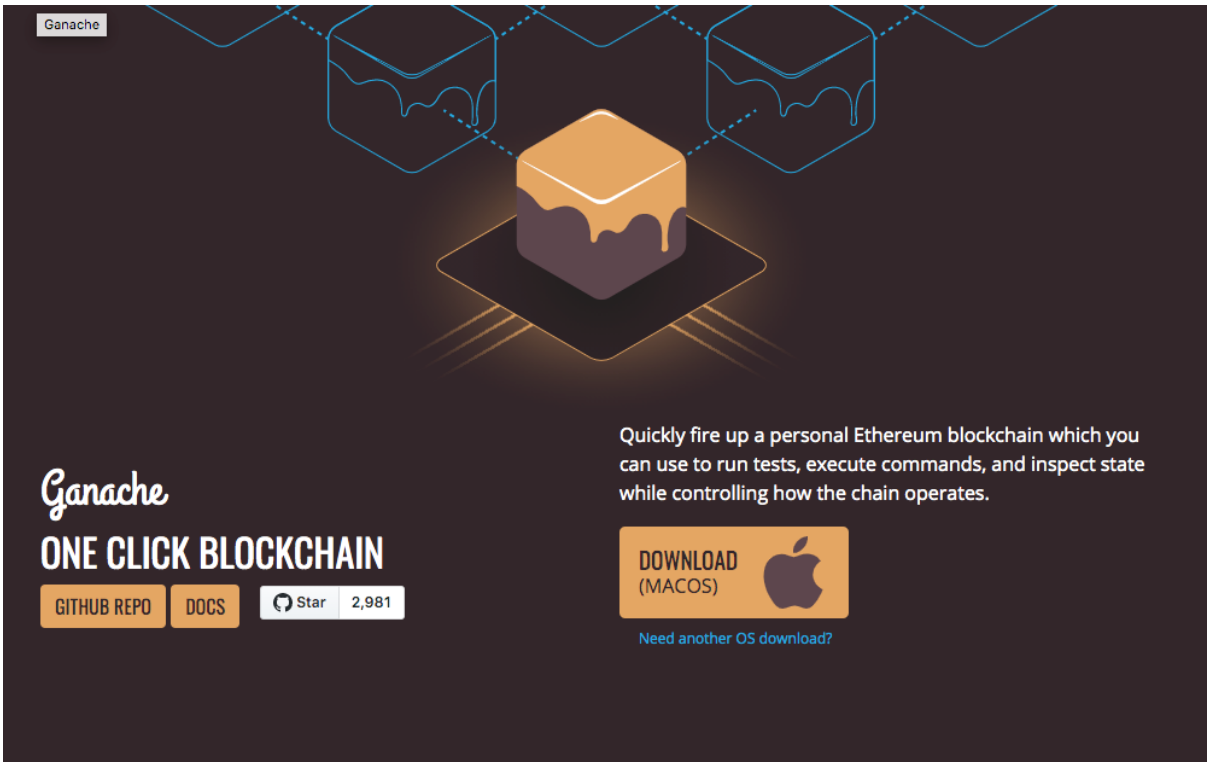


Figure 3. Remix setup configuration

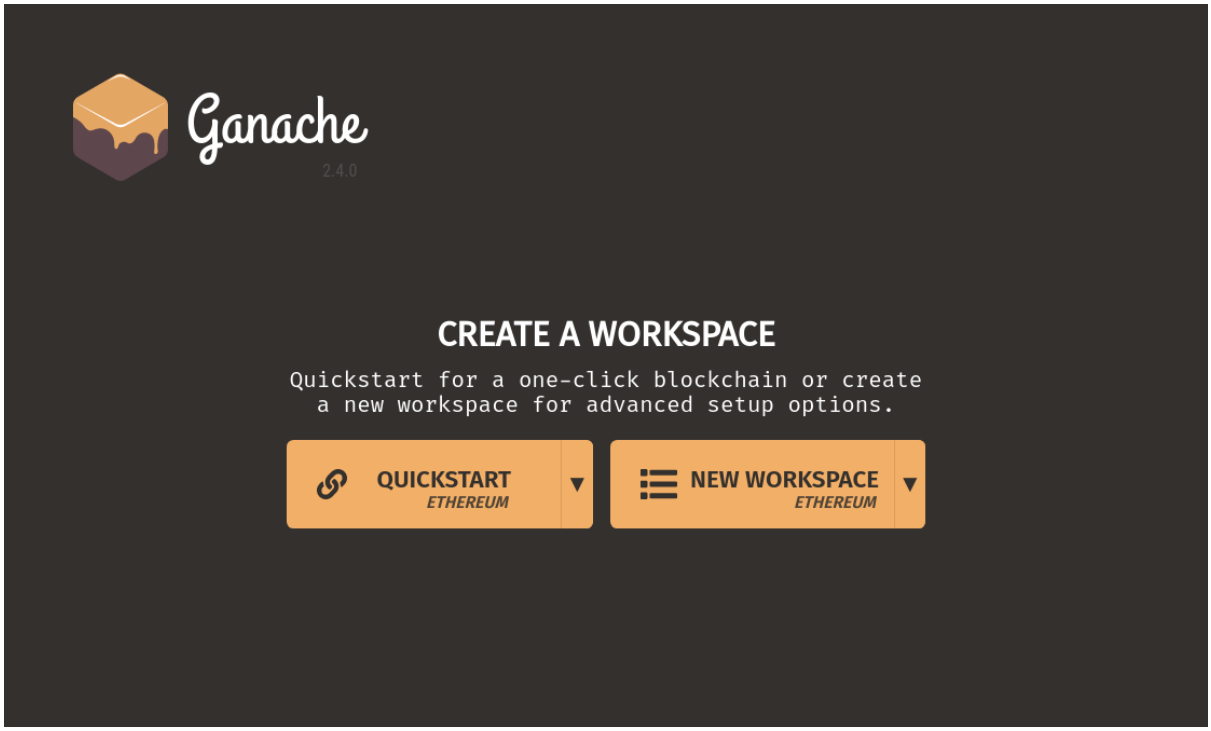
Here's how you can detail your **blockchain setup**, including using Sepolia for real-world testing and Ganache for local testing, along with all necessary configuration steps.

6.1.4 Local Testing with Ganache

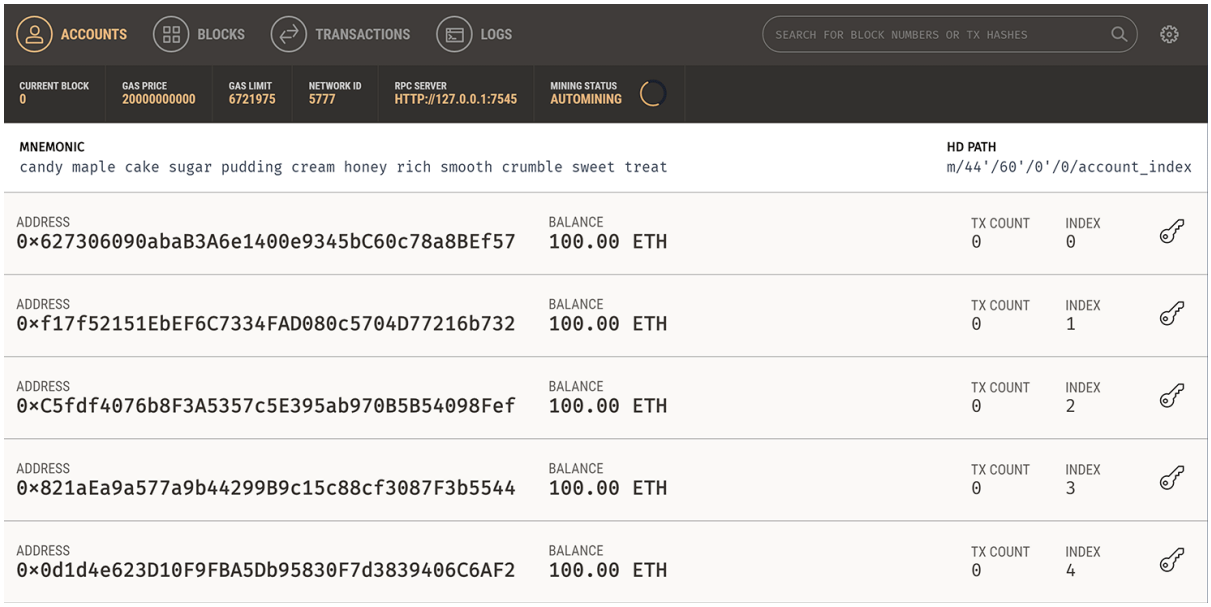
1. **Install Ganache**: Navigate to https://www.trufflesuite.com/ganache to download Ganache



2. Click the download button for your respective OS
    - Launch Ganache and create a new workspace.



3. Click the quick start button to start the local webserver with 10 accounts with 100 credits each
    - Note the RPC URL, network ID, and pre-funded accounts displayed in the interface.



Ganache is now set up you should see a screen similar to the above, note the HTTP server as this will become important

4. **Configure Ganache in Metamask**:
    - Add the Ganache network to Metamask:
        - Network Name: Ganache Local
        - RPC URL: http://127.0.0.1:7545
        - Chain ID: 1337
        - Currency Symbol: ETH
5. **Deploy Smart Contract Using Remix**:
    - In Remix, select **Web3 Provider** as the environment.
    - Enter http://127.0.0.1:7545 (Ganache RPC URL) as the provider.
    - Compile and deploy your smart contract as outlined above

7 Usage Instructions

1. **Uploading Files**:
    - Navigate to the frontend application at http://localhost:3000.
    - Click on the **Upload File** section.
    - Select a file to upload by clicking the "Browse" button.
    - Choose the target cloud provider (e.g., AWS, Azure, or GCP) from the dropdown.
    - Click **Upload**.
    - Once the file is successfully uploaded, the system will display:
        - A unique **File ID**.
        - The **File Hash** (generated via cryptographic hashing).
        - The cloud provider where the file is stored.
2. **Migrating Files Between Cloud Providers**:
    - Go to the **Migrate File** section.
    - Select the file you want to migrate (use the File ID from the upload step).
    - Specify the source cloud provider (e.g., AWS).
    - Specify the destination cloud provider (e.g., Azure).
    - Click **Migrate**.
    - Once migration completes:
        - The system will display the status.
        - A new **File Hash** will be generated and recorded on the blockchain for verification.
3. **Verifying File Integrity via Blockchain**:
    - Navigate to the **Verify File Integrity** section.
    - Enter the File Hash or File ID into the input field.
    - Click **Verify**.
    - The system will compare the hash stored on the blockchain with the one provided:
        - If the hashes match, it will display a success message.
        - Otherwise, an error message will appear indicating data corruption or mismatch.

8 Troubleshooting

8.1.1 Blockchain Issues

1. **Smart Contract Not Deploying**:

- **Cause**: Incorrect RPC URL or insufficient test ETH in the wallet.
- **Solution**:
  - Verify the BLOCKCHAIN_PROVIDER in your .env file is correct (e.g., https://rpc.sepolia.org for Sepolia).
  - Ensure the wallet used for deployment has sufficient Sepolia ETH (use a faucet).
  - Check the contract compilation in Remix for syntax errors.

2. **Wallet Connectivity Problems**:
   - **Cause**: Metamask not connected to the correct blockchain network.
   - **Solution**:
     - Open Metamask and switch to the Sepolia testnet or your local Ganache network.
     - Reconnect Metamask to your frontend by refreshing the browser and re-logging in.

8.1.2 Cloud Platform Issues

1. **Access Denied Errors for Buckets/Storage Accounts**:
   - **Cause**: Incorrect API keys or insufficient permissions.
   - **Solution**:
     - Verify the credentials in your .env file: Ensure equivalent credentials are provided for Azure and GCP.
     - Check IAM roles and permissions for the respective cloud provider:
       - AWS: Ensure the user has "S3FullAccess".
       - Azure: Verify the storage account's connection string.
       - GCP: Ensure the service account has the "Storage Admin" role.
2. **Misconfigured API Keys or Tokens**:
   - **Cause**: Missing or invalid credentials in. env.
   - **Solution**:
     - Regenerate API keys or service account credentials from your cloud platform’s console.
     - Update the .env file with the correct values and restart the backend server.

8.1.3 Application Errors

1. **API Server Not Starting**:
   - **Cause**: Missing dependencies or incorrect environment variables.
   - **Solution**:
     - Run the following commands:

```
cd appDirectory

npm install
```

     - Verify the .env file for completeness.
     - Start the server: