National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Achyut Gawade
Student ID: 22103228

School of Computing
National College of Ireland

Supervisor: Prof. Sean Heeney

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Achyut Gawade
Student ID:	22103228
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Prof. Sean Heeney
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	710
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:	
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):	

Configuration Manual

Achyut Gawade
22103228

1 AWS Cloud Account Setup

A working AWS cloud account is needed to perform below steps with access to IAM services. It is advisable to create a separate AWS IAM role which shall be used for interacting with the services later.

- Create new IAM role let's say `vmimport` in this case.
- Add inline policy as descrived in below text and attach to this IAM role.

```
Policy.Json
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:PutObject",
                "s3:GetBucketAcl"
            ],
            "Resource": [
                "arn:aws:s3:::final-bucket-ac",
                "arn:aws:s3:::final-bucket-ac/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ModifySnapshotAttribute",
                "ec2:CopySnapshot",
                "ec2:RegisterImage",
                "ec2:Describe*"
            ],
            "Resource": "*"
        }
```

- Fetch the AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY from the IAM module. Additionally the AWS_SESSION_TOKEN in case of assumed role. This will be needed to perform operations from python boto3 SDK with AWS services as shown in Figure 1.

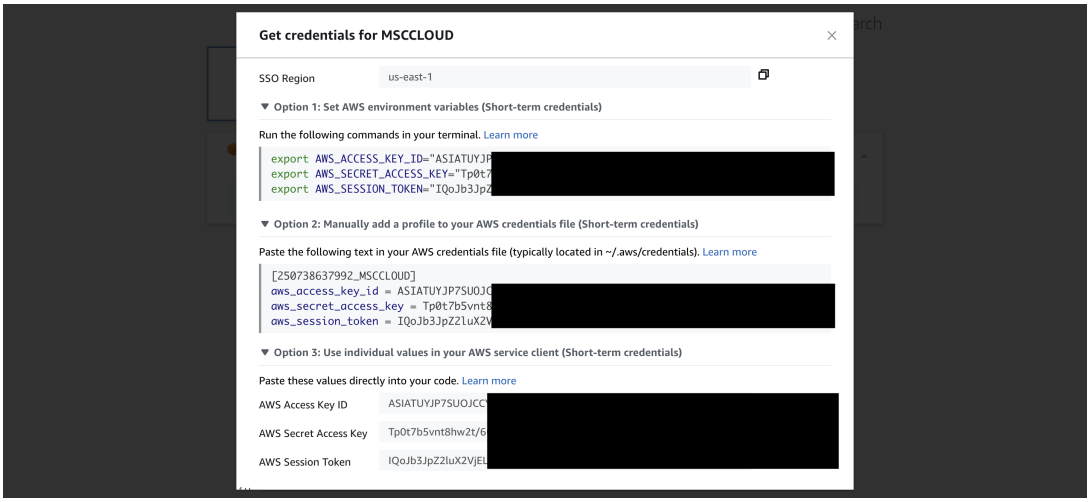

Figure 1: Azure Cloud Account Application Credential Details

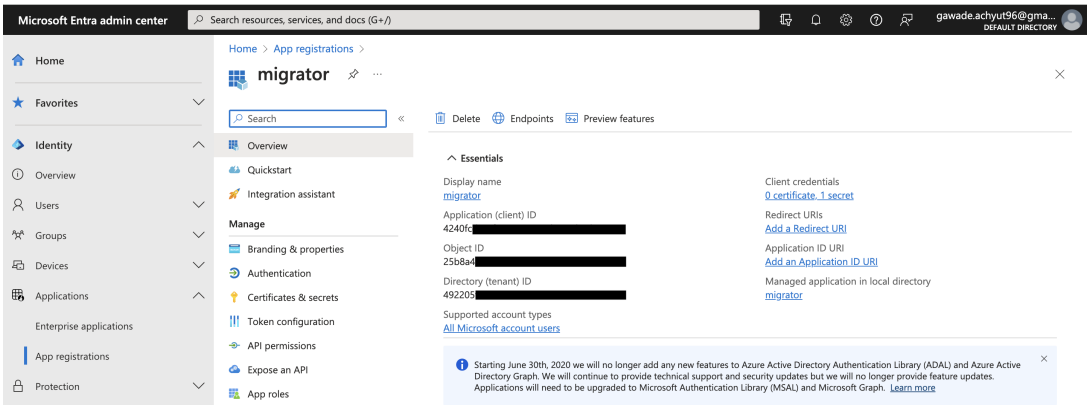

Figure 2: Azure Cloud Account Application Credential Details

- Spin-up a testbed EC2 instance for running the developed middleware with Ubuntu image.

2 Azure Cloud Account Setup

- Visit https://entra.microsoft.com/ and register an application
- Fetch client_id, client_secret, tenant_id, subscription_id from the registered application as shown in the Figure 2
- In Azure cloud account, go to the subscription service and add the registered application to grant access to the subscription resources from IAM module.

3 GCP Cloud Account Setup

- Visit https://console.cloud.google.com/ and create an application

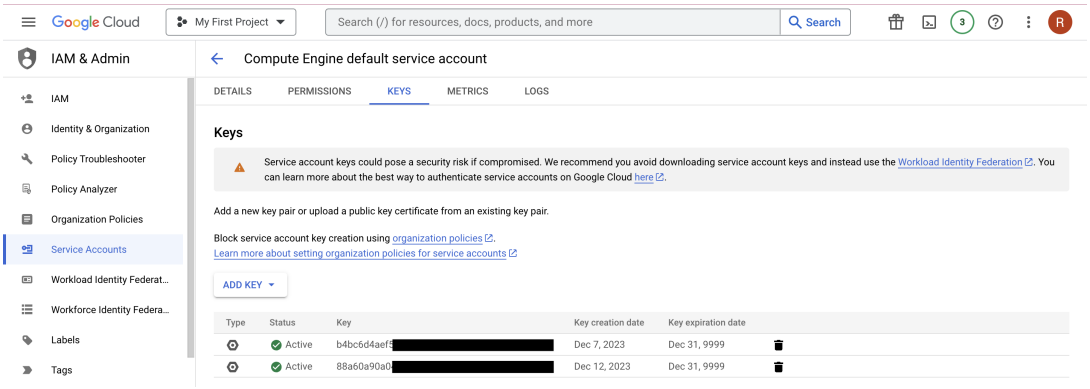

Figure 3: GCP Cloud Service Account Details

- Go to "IAM And Admin" section and generate a service account with access to Compute, Storage and Database services as shown in Figure 3. A Json file will be downloaded automatically which shall be used in next sections.

4 Resource Discovery Module Setup

- On the test server, update the server packages eg. sudo apt update.
- Install Terraform on testbed server as per the OS [1]
- In main.tf file add all the cloud account details fetched from Section 1, Section 2 and Section 3 such as access keys, token, client secrete and serviceAccount.json etc. in the fields shown in Figure 4
- In AzureDiscovery.py add the Azure account details from Section 2 as per Figure 5
- In AWSDiscovery.py add the AWS account details from Section 1 as per Figure 6
- In GCPDiscovery.py add the path to the serviceAccount.json file obtained in 3 as per Figure 6
- Below are the python librariries that needs to be configured which are mentioned in requirements.txt file as well.
  - azure.common.credentials
  - azure.mgmt.resource
  - boto3
  - google.cloud

[1] https://developer.hashicorp.com/terraform/install

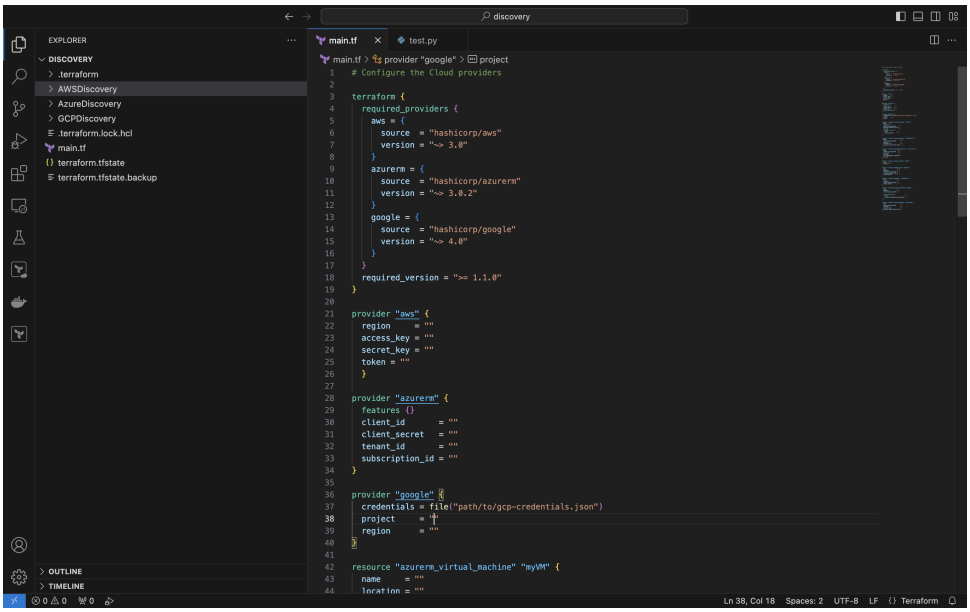

Figure 4: Terraform Cloud Provider Configurations

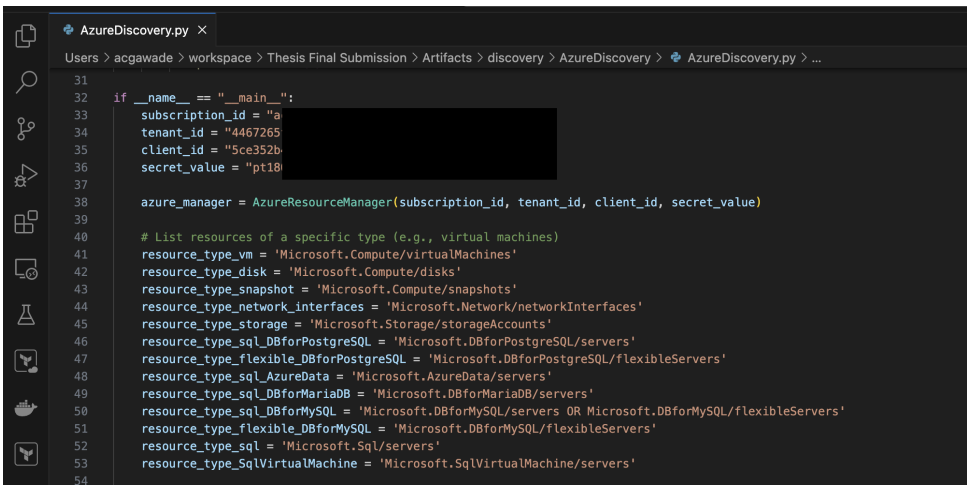

Figure 5: Azure Discovery Account Configuration

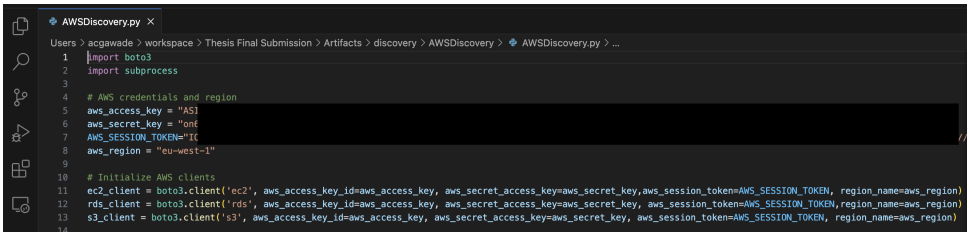

Figure 6: AWS Discovery Account Configuration

5 Virtual Machine Migration Module Setup

- Install Aria2[2] download utility on the test server.
- Install Qemu and Qemu-img [3] which is used for disk format conversions as shown in the Figure 8
- open .env file and set the required parameters such as account credentials, bucket details, IAM role created in Section 1 etc. as shown in Figure 7
- This module primarily contains 6 runnable python files which are azure_gcp_mgr.py, aws_gcp_mgr.py, gcp_aws_mgr.py, gcp_azure_mgr.py, azure_aws_mgr.py and aws_azure_mgr.py.
- Below are the python libraries that needs to be installed.
  - boto3
  - azure-identity
  - azure-storage-blob
  - azure-mgmt-compute
  - azure-mgmt-network
  - azure-mgmt-storage
  - azure-mgmt-resource
  - google-cloud
  - google-auth
  - google-cloud-build
  - google-cloud-storage
  - google-cloud-compute
  - python-dotenv

6 Database Migration Module Setup

- On the test server, update the server packages eg. sudo apt update
- Install MySQL on the server [4]
- Install postgresql-client or postgresql on the server [5]
- Start the MySQL service by running sudo systemctl start mysql
- Start the postgresql service.

[2]https://aria2.github.io/
[3]https://www.qemu.org/download/
[4]https://www.mysql.com/downloads/
[5]https://www.postgresql.org/download/

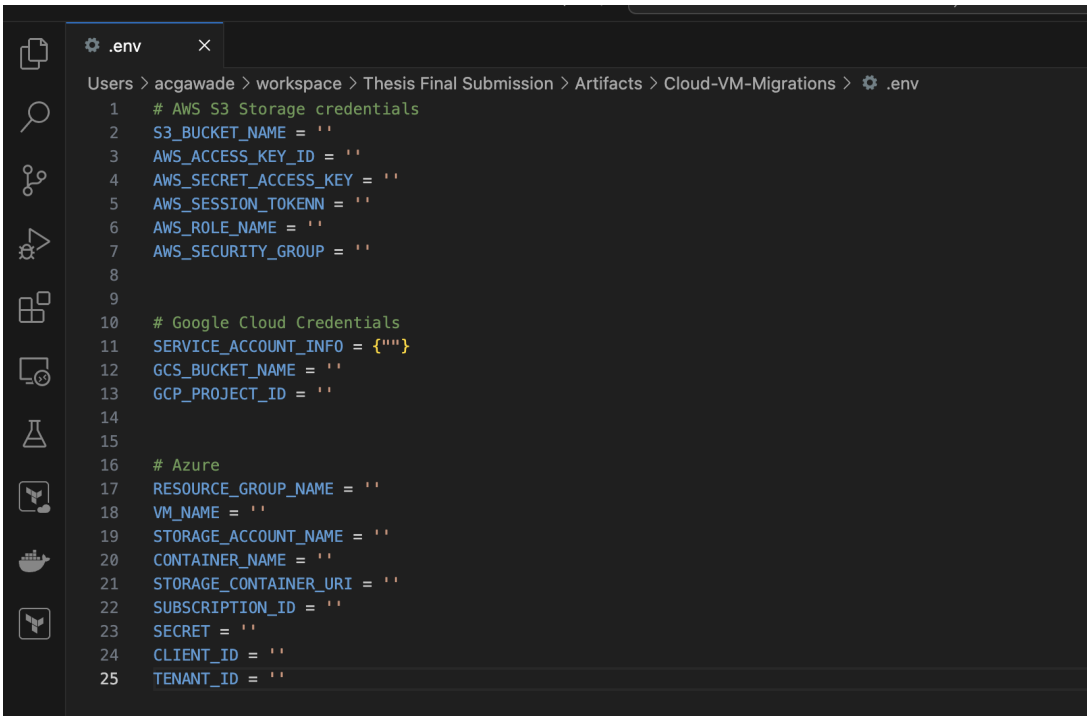

Figure 7: env file for VM Migration Configuration

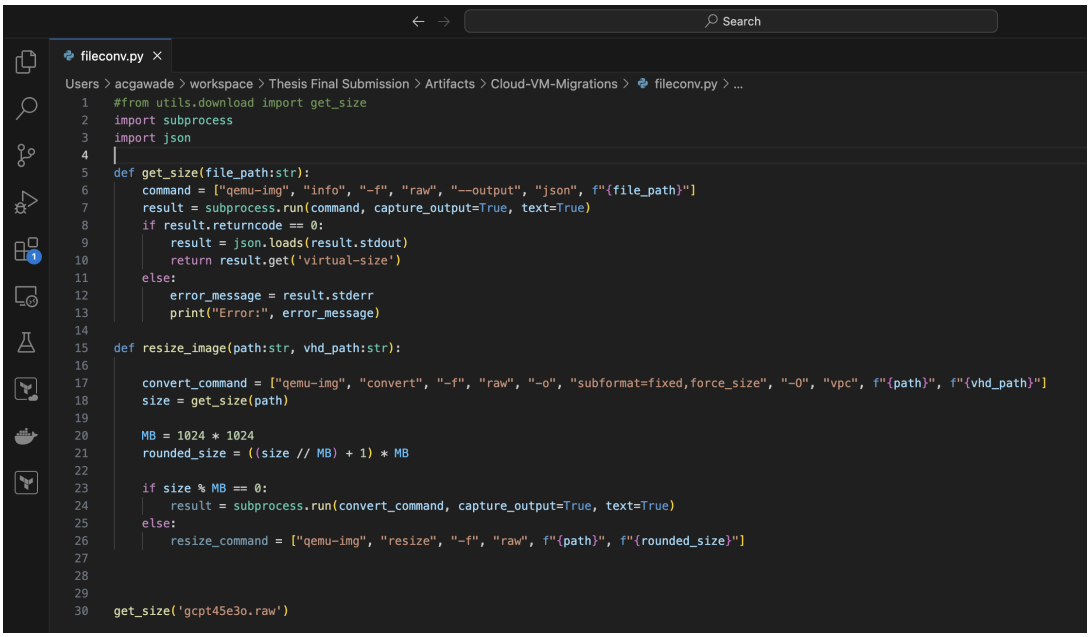

Figure 8: env file for VM Migration Configuration

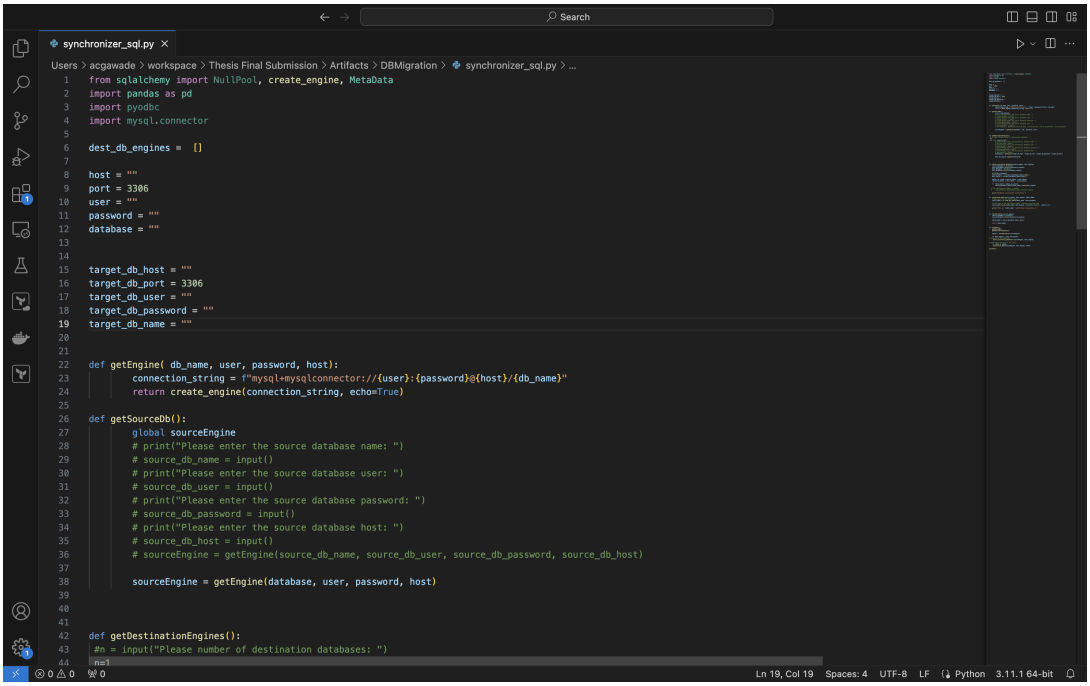

Figure 9: env file for VM Migration Configuration

- This module contains four executable python files which are for performing bulk migration, synchronisation, fetching the database size and verifying the hash of table data as shown in Figures 11, Figure 9 and Figure 12 and Figure 10 respectively.
- This module requires below python libraries
  - pandas
  - sqlalchemy
  - mysql.connector
  - pyodbc

7 Storage Migration

- Add AWS, Azure, GCP account details in StorageMigrator.py as shown in the Figure 13
- Add bucket names for all the cloud buckets along with mentioning the migration scenario and setting DELETE_AFTER_TRANSFER flag
- This module also contains a HashVerifier.py file which can be used to verify the accuracy of files migrated from source to destination cloud bucket as shown in 14.
- Below are the python librariries that needs to be configured which are also provided in requirements.txt
  - azure.common.credentials

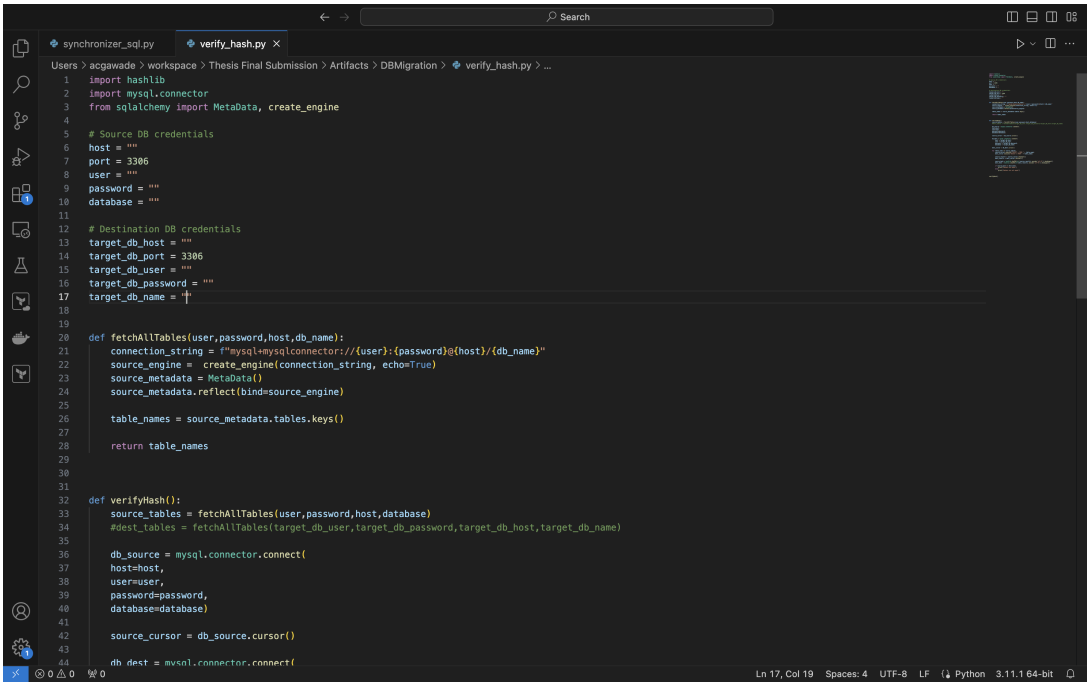

Figure 10: env file for VM Migration Configuration

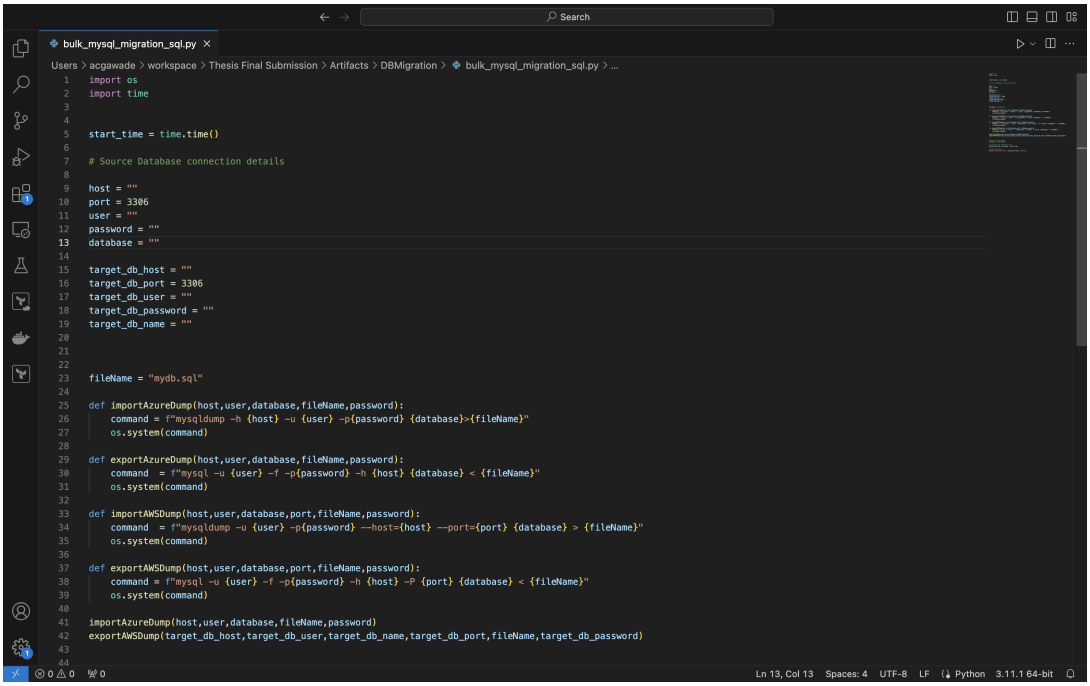

Figure 11: env file for VM Migration Configuration

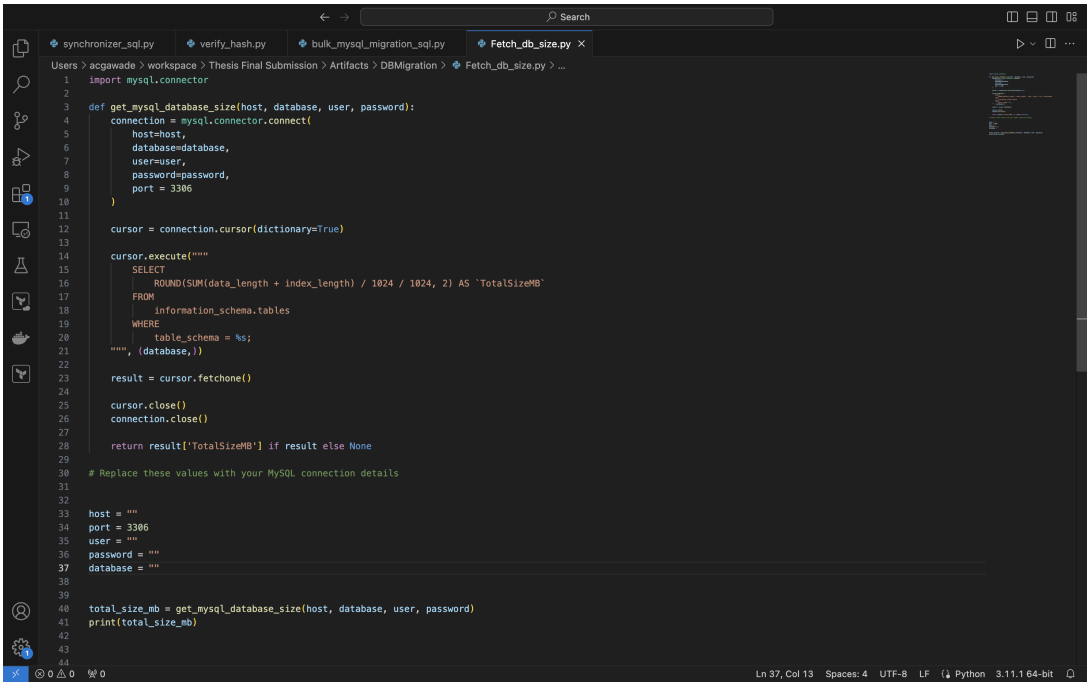

Figure 12: env file for VM Migration Configuration

- azure.mgmt.resource
- boto3
- google.cloud

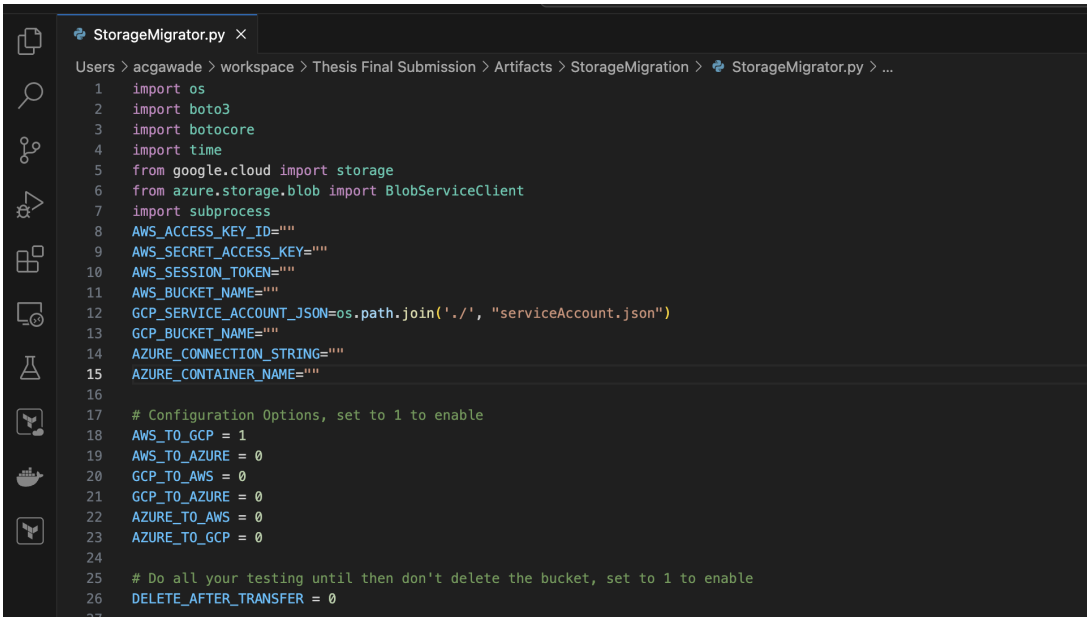

Figure 13: Storage Migration Accounts And Bucket Configurations

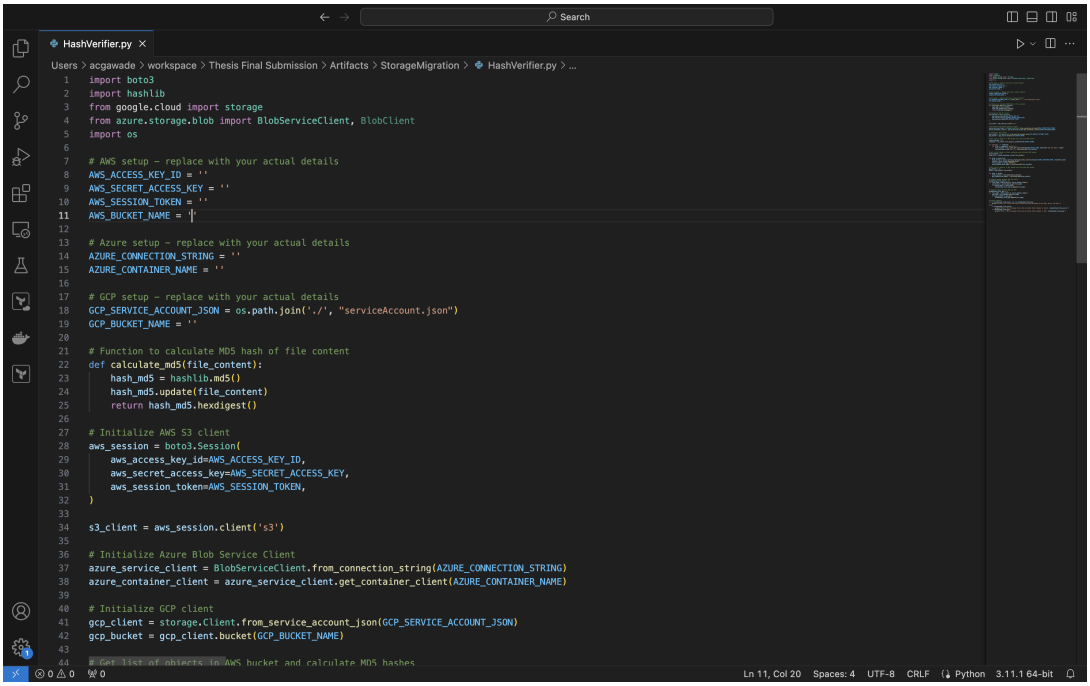

Figure 14: Hash Verifier Programm for Storage Migration