Configuration Manual

MSc Research Project

Cloud Computing

Vishwajeet Kumar

Student ID: 23319330

School of Computing

National College of Ireland

Supervisor: Jorge Mario Cortes Mendoza

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	……. ………………………vishwajeet kumar………………………………………………………………………………
Student ID:	…………23319330………………………………………………………………………………………………………..…….
Programme:	………………… **Msc** Cloud computing ……………………………………………
Year:	… 2025 …………………………..
Module:	…………………… MSCCLOUD Research Project ……………………………………………………………………………….………
Lecturer:	……………… Jorge Mario Cortes Mendoza ……………………………………………………………………………………….………
Submission Due Date:	……………………1st September…………………………………………………………………………………….………
Project Title:	……… A Novel Cloud-Native Autoscaling Framework Using MultiView-BiLSTM and Azure Time-Series Data
Word Count:	………2080……………………………… **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:	……………Vishwajeet kumar…………………………………………………………………………………… …
Date:	……………………11th aug 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):	

Configuration Manual

Vishwajeet kumar
Student ID: x23319330@student.ncirl.ie

1 Dataset Setup

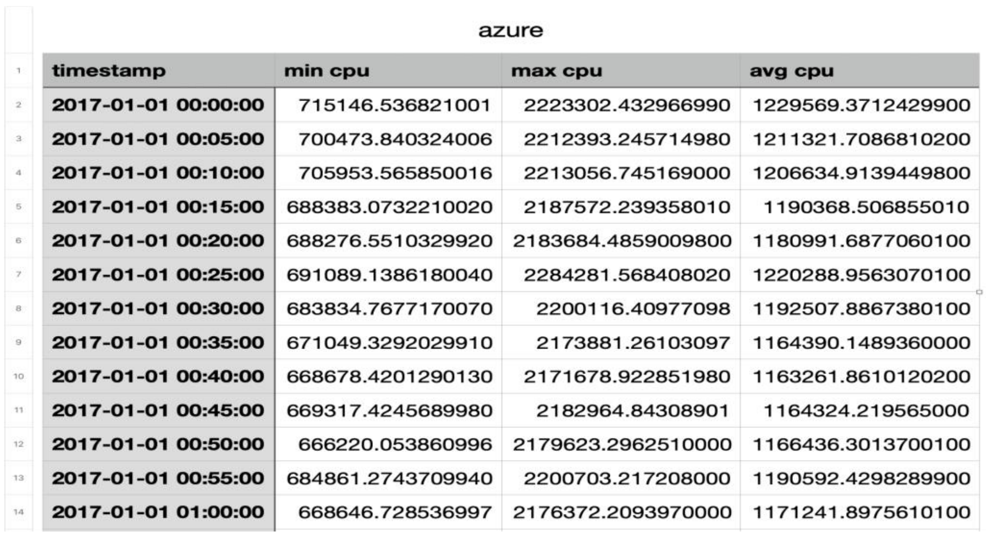
Dataset URL: https://github.com/Azure/AzurePublicDataset

About the Dataset:- The repository holds publicly available releases of Microsoft Azure traces to the advantage of the research and academic community (Eli, 2025), Nowadays there exist two classes of traces:

- **VM Traces:** one of the two representative traces, to Microsoft Azure in the year 2017 and 2019, two samples of the virtual machine (VM) workloads, and a third trace focused on the investigation of packing algorithms.
- **Azure Functions Traces:** a few representative traces of Azure Functions invocations, gathered in 2019 over a 2-week period, and of Azure Functions accessing blobs, between November and December of 2020.

This data is organized into the form of time-series data where measurements were taken at intervals of 5 minutes. Every single line will consist of three main numbers, namely, the minimum amount of efficient CPU consumption, maximum amount of efficient CPU consumption and average efficient CPU consumption over the interval of time captured.

An AWS S3 bucket will be used to store the dataset and be accessed programmatically through an AWS EC2 Cloud9 Jupyter Notebook, which can potentially be used to effectively retrieve and process the data to gain analytics.



azure

timestamp	min cpu	max cpu	avg cpu
2017-01-01 00:00:00	715146.536821001	2223302.432966990	1229569.3712429900
2017-01-01 00:05:00	700473.840324006	2212393.245714980	1211321.7086810200
2017-01-01 00:10:00	705953.565850016	2213056.745169000	1206634.9139449800
2017-01-01 00:15:00	688383.0732210020	2187572.239358010	1190368.506855010
2017-01-01 00:20:00	688276.5510329920	2183684.4859009800	1180991.6877060100
2017-01-01 00:25:00	691089.1386180040	2284281.568408020	1220288.9563070100
2017-01-01 00:30:00	683834.7677170070	2200116.40977098	1192507.8867380100
2017-01-01 00:35:00	671049.3292029910	2173881.26103097	1164390.1489360000
2017-01-01 00:40:00	668678.4201290130	2171678.922851980	1163261.8610120200
2017-01-01 00:45:00	669317.4245689980	2182964.84308901	1164324.219565000
2017-01-01 00:50:00	666220.053860996	2179623.2962510000	1166436.3013700100
2017-01-01 00:55:00	684861.2743709940	2200703.217208000	1190592.4298289900
2017-01-01 01:00:00	668646.728536997	2176372.2093970000	1171241.8975610100

2 Hardware Configuration

The development environment is being performed on an AWS EC2 t3.large instance, which has 8 GiB of RAM and 2 vCPUs and ensures a balanced compute and memory capacity that comes in handy during data processing and application development tasks in the cloud environment. The example (ID: i-0bef7989768470ec2) was started in us-east-1 region with the functioning in VPC vpc-00c062771410d1af5 and subnet subnet-0f06c86a064877d08. It was configured with an external IPv4 address (18.232.51.153) and a private IPv4 address (172.31.66.207) which had had an external access and an internal IPv4 address respectively. To be compatible with modern cloud infrastructure we used Amazon Linux 2023 as the operating environment and launched it in an Amazon Linux 2023 AMI and used a HVM virtualisation type and UEFI-preferred boot. It was set to an unlimited number of CPU credits, all with the purpose of creating a lasting performance of development workloads, and it was in combination with AWS cloud 9 IDE, which presents preconfigured, browser-based development environment with easy connection to AWS resources.

3 Software Configuration

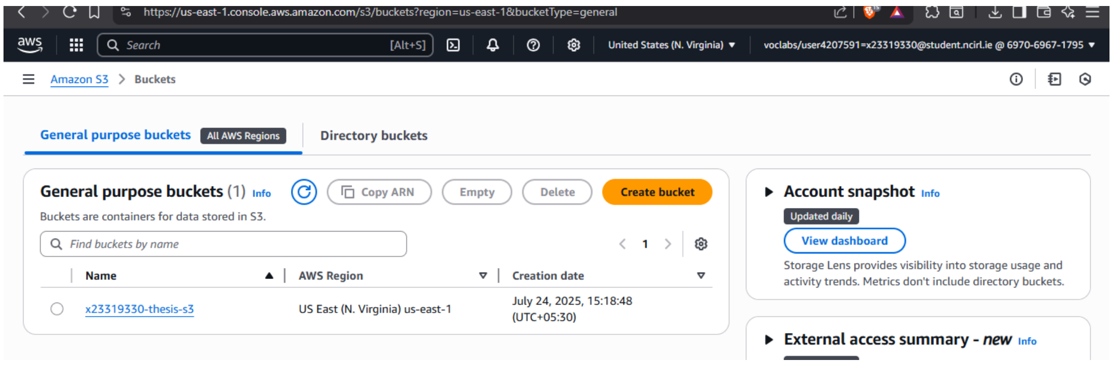
- **Python 3.10.12:** Was used as the main programming language, with high-level built-in data structures, support of dynamic typing and binding. Its object-oriented and interpreted characteristics were suitable to quick application development, scripting, and joining of numerous items, and offer clarity and scalability of the code in both little and vast projects (Python Documentation, 2025).
- **Visual Studio Code (VS Code):** As the local integrated development environment (IDE) this free and open-source text editor by Microsoft offered cross-platform support (Windows, Linux, macOS), integrated debugging, Git integration, and a vast marketplace of extensions to make the coding process more productive and simplify workflows (Visual Studio Code, 2025).
- **Kaggle:** Leveraged as a cloud-based machine learning and data science platform, with a hosted Jupyter Notebook interface (with free access to computational resources such as GPUs, and a large selection of publicly available datasets. It enabled experimentation, model building and teamwork without the necessity of local installation (kaggle, 2025).
- **Python Libraries:** The environment was set up with base libraries like Pandas, data manipulation, NumPy, for num processing, and Matplotlib and Seaborn, for static vis, and Plotly for static and interactive vis anal, Scikit for traditional machine learning, and TensorFlow with Kera for deep learn.
    - Pandas
    - Numpy
    - Matplotlib
    - Seaborn
    - Plotly
    - Scikit-learn
    - Tensorflow
    - Keras

- **Docker and Amazon ECR:** Docker was used to containerise the Python notebooks, ensuring consistent environments across the development and deployment stages. These container images were stored in Amazon Elastic Container Registry (ECR), enabling scalable and efficient hosting within AWS infrastructure.

4 AWS Cloud Resources

● S3 Bucket

For this thesis, Amazon Simple Storage Service (Amazon S3) (aws, 2025) was adopted as a core cloud-based object storage platform to store and retrieve raw datasets in addition to output of exploratory data analysis (EDA). The s3 bucket was provisioned in N. Virginia (us-east-1) as S3 bucket x23319330-thesis-s3 where the low-latency access could be achieved by other AWS services e.g. EC2, Cloud9. The initial data, which was owned as a Microsoft Azure Public Dataset, was on this bucket in its raw avatar as it facilitated centralised and secure storage. In the course of analysis, intermediate files, such as transformed and cleaned datasets produced through pre-processing, were stored in the bucket, as well.

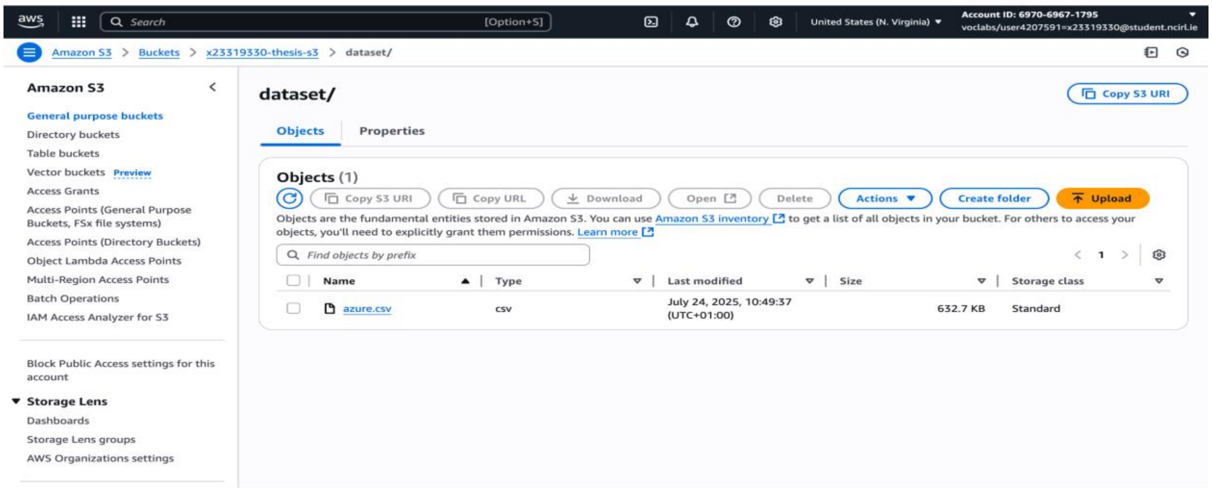

https://us-east-1.console.aws.amazon.com/s3/buckets?region=us-east-1&bucketType=general
aws
Search [Alt+S]
United States (N. Virginia)
voclabs/user4207591=x23319330@student.ncirl.ie @ 6970-6967-1795
Amazon S3 > Buckets
General purpose buckets
All AWS Regions
Directory buckets
General purpose buckets (1) Info
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Account snapshot Info
Updated daily
View dashboard
External access summary - new Info

Buckets are containers for data stored in S3.

Storage Lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets.

Name	AWS Region	Creation date
x23319330-thesis-s3	US East (N. Virginia) us-east-1	July 24, 2025, 15:18:48 (UTC+05:30)

Besides storing the dataset, the usage of the identical S3 bucket was utilised to store the results of EDA, such as statistical summaries, plots made, interactive figures, and visualisation pictures. Such outputs were retained in folders with an organised structure in the bucket to ensure ready accessibility and trace-ability. Making the visualisations and processed results persistent in S3 outside the compute as well as being able to easily download the results and add them to reports, presentations, and additional analysis was a requirement. Integration of the bucket with AWS SDK for Python (Boto3) created possibilities in file uploading, downloading, and management using a system literally from the Jupyter Notebook in the Cloud9 environment. This resulted in the raw data and all analysis artifacts created throughout the research to be stored in this manner in a scalable, durable and costs-efficient manner.


aws
Search [Option+S]
United States (N. Virginia)
Account ID: 6970-6967-1795
voclabs/user4207591=x23319330@student.ncirl.ie
Amazon S3 > Buckets > x23319330-thesis-s3 > dataset/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Vector buckets Preview
Access Grants
Access Points (General Purpose Buckets, FSx file systems)
Access Points (Directory Buckets)
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
dataset/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix

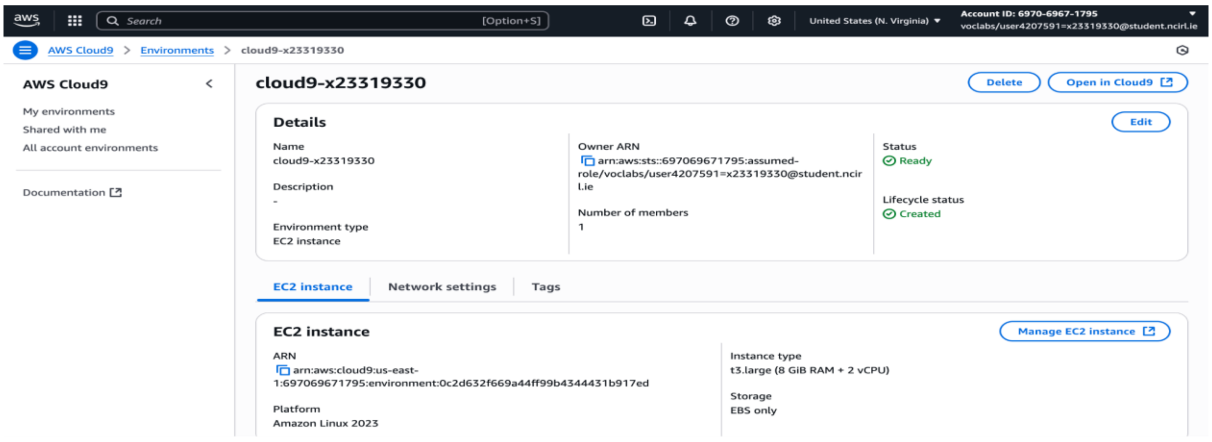
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
azure.csv	csv	July 24, 2025, 10:49:37 (UTC+01:00)	632.7 KB	Standard

● Cloud9 Environment

The chosen development environment implementation is the Cloud9 IDE, adopted under the name of cloud9-x23319330, installed on AWS EC2 t3. large with 8 Gb RAM and 2 vCPUs, Amazon Linux 2023. It offered a complete setup of browser based IDE seamlessly coupled with AWS services that can allow editing, executing, and debugging of the code without local environment(aws, 2025d). Environment was using only EBS-based storage, where project files and configuration have been persisted regardless of the underlying instance. Direct access to the AWS resources (S3, EC2, etc.)

made it possible to retrieve the datasets, analyze them and deploy them efficiently in a constant and scalable cloud-based development environment.



Account ID: 6970-6967-1795
voclabs/user4207591=x23319330@student.ncirl.ie

cloud9-x23319330

Details

Name
cloud9-x23319330

Description
-

Environment type
EC2 instance

Owner ARN
arn:aws:sts::697069671795:assumed-role/voclabs/user4207591=x23319330@student.ncirl.ie

Number of members
1

Status
Ready

Lifecycle status
Created

EC2 instance

ARN
arn:aws:cloud9:us-east-1:697069671795:environment:0c2d632f669a44ff99b4344431b917ed

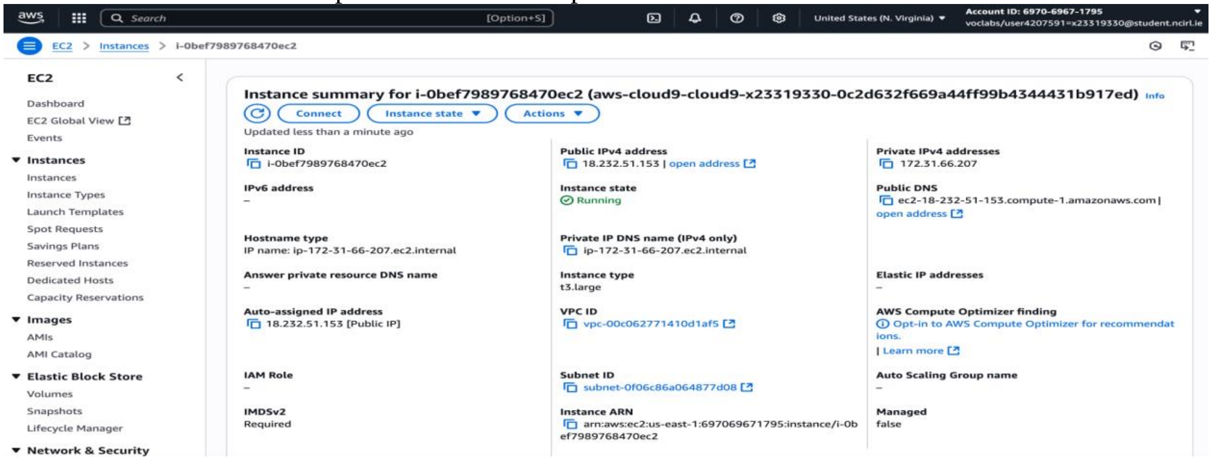
Platform
Amazon Linux 2023

Instance type
t3.large (8 GiB RAM + 2 vCPU)

Storage
EBS only

● EC2 Instance

The EC2 instance (aws, 2025a) attached at the Cloud9 environment (ID: i-0bef7989768470ec2) happened to be a t3.large instance with 8 GiB RAM and 2 vCPUs, and the Amazon Linux 2023 (ID: 2023) operating system. It was configured to ensure persistence of data with EBS as its only modality of storage and was implemented in the us-east-1 region of VPC vpc-00c062771410d1af5 and subnet subnet-0f06c86a064877d08, which had low latency access to AWS services like S3. Its external connection to the Internet went over IPv4 (18.232.51.153) and the internal network communication was using IPv4 (172.31.66.207). It supported HVM virtualisation and the preferred boot mode UEFI with infinite CPU credits to parse sustained compute workload.



Account ID: 6970-6967-1795
voclabs/user4207591=x23319330@student.ncirl.ie

Instance summary for i-0bef7989768470ec2 (aws-cloud9-cloud9-x23319330-0c2d632f669a44ff99b4344431b917ed)

Updated less than a minute ago

Instance ID
i-0bef7989768470ec2

IPv6 address
–

Hostname type
IP name: ip-172-31-66-207.ec2.internal

Answer private resource DNS name
–

Auto-assigned IP address
18.232.51.153 [Public IP]

IAM Role
–

IMDSv2
Required

Public IPv4 address
18.232.51.153 | open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-66-207.ec2.internal

Instance type
t3.large

VPC ID
vpc-00c062771410d1af5

Subnet ID
subnet-0f06c86a064877d08

Instance ARN
arn:aws:ec2:us-east-1:697069671795:instance/i-0bef7989768470ec2

Private IPv4 addresses
172.31.66.207

Public DNS
ec2-18-232-51-153.compute-1.amazonaws.com | open address

Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
| Learn more

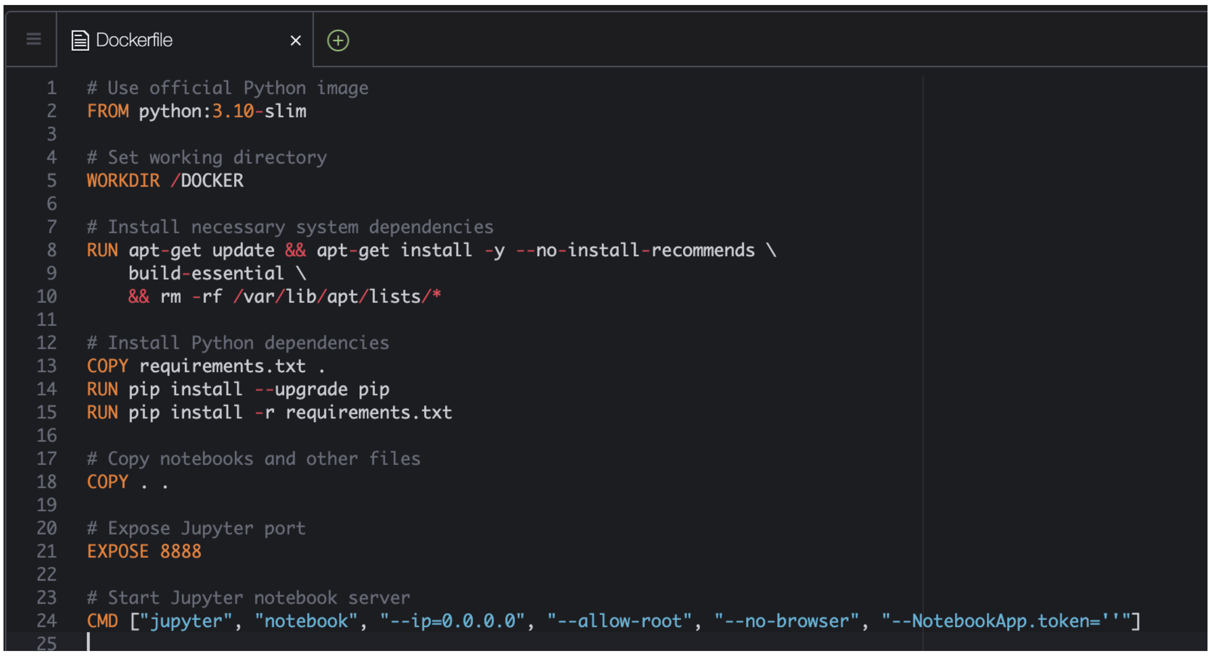
Auto Scaling Group name
–

Managed
false

The instance was characterized by the default auto-recovery that facilitated its high availability during the development process. As a background compute on which the Cloud9 IDE was to run this EC2 instance allowed us to run our code and process our data efficiently and could scale to meet our demand in a secure way.

● Docker, Elastic Container Registry (ECR) and Elastic Container Service (ECS)

For deployment, Docker (dockerdocs, 2025), Amazon Elastic Container Registry (ECR), (aws, 2025b). and Amazon Elastic Container Service (ECS) (aws, 2025c) had been designed to containerise, store and execute the Jupyter Notebook landscape in a managed way that allows scalability.



```
# Use official Python image
FROM python:3.10-slim

# Set working directory
WORKDIR /DOCKER

# Install necessary system dependencies
RUN apt-get update && apt-get install -y --no-install-recommends \
    build-essential \
    && rm -rf /var/lib/apt/lists/*

# Install Python dependencies
COPY requirements.txt .
RUN pip install --upgrade pip
RUN pip install -r requirements.txt

# Copy notebooks and other files
COPY . .

# Expose Jupyter port
EXPOSE 8888

# Start Jupyter notebook server
CMD ["jupyter", "notebook", "--ip=0.0.0.0", "--allow-root", "--no-browser", "--NotebookApp.token=''"]
```

This was done to create a Docker file on the Cloud9 environment so as to create a lightweight docker image which uses the official Python 3.10 slim image. So the workflow that was done with the assistance of Dockerfile was relating to defining a working directory (/DOCKER) and installing the necessary system package (such as build-essential) and then purported the Python dependencies which are written in the requirements.txt by means of pip. The Jupyter notebooks were included in the copying of all project files after which the notebook was accessible through port 8888 on the exposed container. Lastly, the Jupyter Notebook server was initiated (with configuration parameters: to enable remote connection (--ip=0.0.0.0) and to disable authentication tokens (--NotebookApp.token='')) to be able to connect to the ECS cluster with ease.

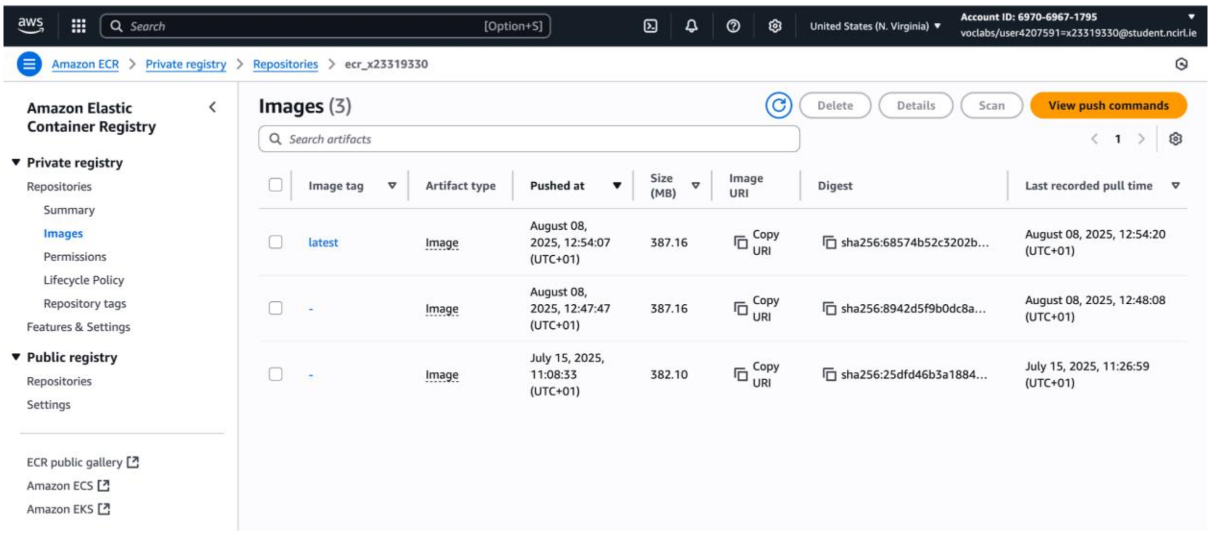


Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
latest	Image	August 08, 2025, 12:54:07 (UTC+01)	387.16	Copy URI	sha256:68574b52c3202b...	August 08, 2025, 12:54:20 (UTC+01)
-	Image	August 08, 2025, 12:47:47 (UTC+01)	387.16	Copy URI	sha256:8942d5f9b0dc8a...	August 08, 2025, 12:48:08 (UTC+01)
-	Image	July 15, 2025, 11:08:33 (UTC+01)	382.10	Copy URI	sha256:25dfd46b3a1884...	July 15, 2025, 11:26:59 (UTC+01)

After building the Docker image locally on Cloud9, it can be tagged and uploaded to an Amazon ECR repository as this gives a secure, scalable and highly available container image registry. With ECR repository, the versioning of container images and a simple retrieval of images was possible.

Due to machine dependency, deployment to Amazon ECS was utilised to create a new cluster with Docker configured and run the containerised Jupyter Notebook. ECS used the lifecycle, scaling, and networking of the task that ran the container image in ECR. Container was deployed to the ECS cluster where port 8888 was exposed to connect to the notebook, thus users could access the notebook in the stabilized and expandable cloud environment. With such a configuration, Jupyter notebooks could be deployed in a production-like AWS infrastructure in a reproducible, portable, and robust fashion.

- IAM Permissions

To enable secure and seamless access to the AWS resources used in this thesis, appropriate IAM policies were configured (aws, 2025e). The Cloud9 environment and related EC2 instance were given the permission to access Amazon S3 to store and retrieve dataset, including the s3:GetObject, s3:PutObject and s3:ListBucket permissions on a particular bucket (x23319330-thesis-s3). Also it required permissions to read Amazon ECR (ecr:GetDownloadUrlForLayer, ecr:BatchGetImage, ecr:BatchCheckLayerAvailability) to retrieve Docker images to be deployed. In the case of managing containers on Amazon ECS, the IAM role should have granted permission such as ecs:RunTask, ecs:DescribeTasks and ecs:UpdateService. The least-privilege access was guaranteed by these policies and limited access was granted to preserve security, but enable a smooth operation throughout the development, storage and deployment processes.

References

aws (2025a) What is Amazon EC2? - Amazon Elastic Compute Cloud. Available at: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html (Accessed: 10 August 2025).

aws (2025b) What is Amazon Elastic Container Registry? - Amazon ECR. Available at: https://docs.aws.amazon.com/AmazonECR/latest/userguide/what-is-ecr.html (Accessed: 10 August 2025).

aws (2025c) What is Amazon Elastic Container Service? - Amazon Elastic Container Service. Available at: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/Welcome.html (Accessed: 10 August 2025).

aws (2025) What is Amazon S3? - Amazon Simple Storage Service. Available at: https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html (Accessed: 10 August 2025).

aws (2025d) What is AWS Cloud9? - AWS Cloud9. Available at: https://docs.aws.amazon.com/cloud9/latest/user-guide/welcome.html (Accessed: 10 August 2025).

aws (2025e) What is IAM? - AWS Identity and Access Management. Available at: https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html (Accessed: 10 August 2025).

dockerdocs (200AD) Dockerfile reference, Docker Documentation. Available at: https://docs.docker.com/reference/dockerfile/ (Accessed: 10 August 2025).

Eli, C. (2025) 'Azure/AzurePublicDataset'. Microsoft Azure. Available at: https://github.com/Azure/AzurePublicDataset (Accessed: 10 August 2025).

kaggle (2025) Getting started on Kaggle | Data Science Resources. Available at: https://www.kaggle.com/docs (Accessed: 10 August 2025).

Python Documentation (2025) 3.10.18 Documentation. Available at: https://docs.python.org/3.10/ (Accessed: 10 August 2025).

Visual Studio Code (no date) Visual Studio Code - Code Editing. Redefined. Available at: https://code.visualstudio.com/ (Accessed: 10 August 2025).