

Configuration Manual

MSc Research Project
Msc In Cloud Computing

Harsh Deore
Student ID: x23107219

School of Computing
National College of Ireland

Supervisor: Ahmed Makki

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Harsh Deore
Student ID:	x23107219
Programme:	Msc In Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Ahmed Makki
Submission Due Date:	12/12/2024
Project Title:	Configuration Manual
Word Count:	968
Page Count:	1

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:	12th December 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):	

Enhancing Cloud Flexibility: Optimizing Live Migration for Non-Web Applications Across Cloud Environments

Name: Harsh Deore
Student Id: x23107219

1. Introduction

This configuration manual provides a step-by-step guide for implementing the live migration of a stateful PostgreSQL database, using the Pagila sample database, between AWS and Azure cloud platforms. It details the setup of cloud environments, database configurations, migration tools, and security measures. By following this manual, users can achieve a seamless migration process with minimal downtime, data consistency, and enhanced performance, ensuring a practical framework for non-web application migrations across heterogeneous cloud environments.

2. System Requirements and Libraries

This section outlines the essential hardware, software, and tools required to implement the live migration process. It includes cloud platforms (AWS and Azure), PostgreSQL with the Pagila sample database, and tools such as CRIU for process checkpointing and WAL Shipping for data synchronization. Additionally, secure connectivity is ensured using SSL/TLS, AWS Key Management Service (KMS), and Azure Key Vault. These resources are critical for achieving a reliable and secure migration workflow.

3. Cloud Execution

3.1. EC2 and VM

Instance summary for i-02a4f2ef92fbb594d (harsh-ec2-ubuntu) [Info](#)

Updated less than a minute ago

Instance ID

[i-02a4f2ef92fbb594d](#)

IPv6 address

-

Hostname type

IP name: ip-172-31-47-230.eu-north-1.compute.internal

Answer private resource DNS name

IPv4 (A)

Auto-assigned IP address

[13.60.190.28](#) [Public IP]

IAM Role

-

IMDSv2

Optional
[EC2 recommends setting IMDSv2 to required](#) | [Learn more](#)

Operator

-

Public IPv4 address

[13.60.190.28](#) | [open address](#)

Instance state

Running

Private IP DNS name (IPv4 only)

[ip-172-31-47-230.eu-north-1.compute.internal](#)

Instance type

t3.micro

VPC ID

[vpc-067024280e6020f7d](#)

Subnet ID

[subnet-068728c323897b7d5](#)

Instance ARN

[arn:aws:ec2:eu-north-1:250738637992:instance/i-02a4f2ef92fbb594d](#)

Private IPv4 addresses

[172.31.47.230](#)

Public IPv4 DNS

[ec2-13-60-190-28.eu-north-1.compute.amazonaws.com](#) | [open address](#)

Elastic IP addresses

-

AWS Compute Optimizer finding

User: arn:aws:sts::250738637992:assumed-role/AWSReservedS
SO_MSCCLOUD_554ef120b0d7b74a/x23107219@student.ncir
Lie is not authorized to perform: compute-optimizer:GetEnroll
mentStatus on resource: * because no identity-based policy all
ows the compute-optimizer:GetEnrollmentStatus action
[Retry](#)

Auto Scaling Group name

-

Managed

false

Figure 1: Showcase of the EC2 instance on AWS summary

Essentials

Resource group (move) : [MyResourceGroup](#)

Status : Running

Location : West US 2 (Zone 1)

Subscription (move) : [Azure for Students](#)

Subscription ID : da8ab216-38ed-4010-9d67-295848df4ffb

Availability zone : 1 (Azure-selected zone)

Tags (edit) : [Add tags](#)

Copy to clipboard

Operating system : Linux (ubuntu 20.04)

Size : Standard D2s v3 (2 vcpus, 8 GiB memory)

Public IP address : [74.179.56.30](#)

Virtual network/subnet : [HarshVM-vnet/default](#)

DNS name : [Not configured](#)

Health state : -

Time created : 11/1/2024, 5:49 AM UTC

Properties

Monitoring

Capabilities (7)

Recommendations (2)

Tutorials

Virtual machine

Computer name : HarshVM

Operating system : Linux (ubuntu 20.04)

VM generation : V2

VM architecture : x64

Agent status : Ready

Agent version : 2.12.0.2

Hibernation : Disabled

Host group : -

Host : -

Proximity placement group : -

Colocation status : N/A

Networking

Public IP address : [74.179.56.30](#) (Network interface [harshvm143-fec23e7b](#))

Public IP address (IPv6) : -

Private IP address : 10.0.0.4

Private IP address (IPv6) : -

Virtual network/subnet : [HarshVM-vnet/default](#)

DNS name : [Configure](#)

Size

Size : Standard D2s v3

vCPUs : 2

RAM : 8 GiB

Figure 2: Showcase of the VM instance on Azure summary

3.2. Details about the AWS EC2 Instance

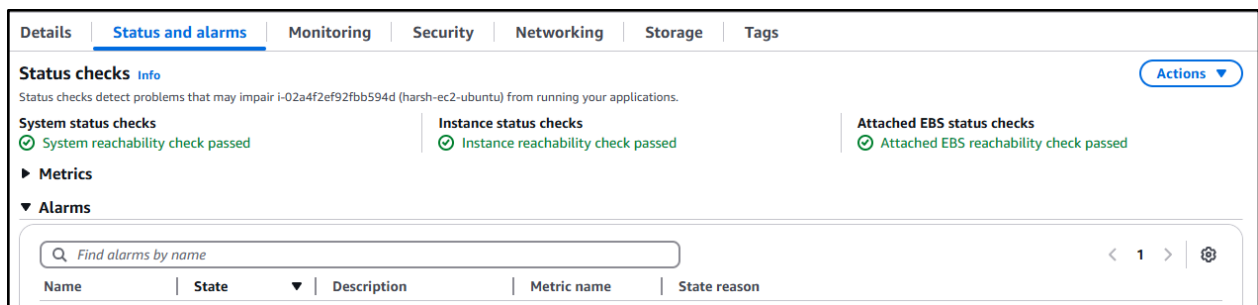


Figure 3: Status and reachability check for the EC2 instance

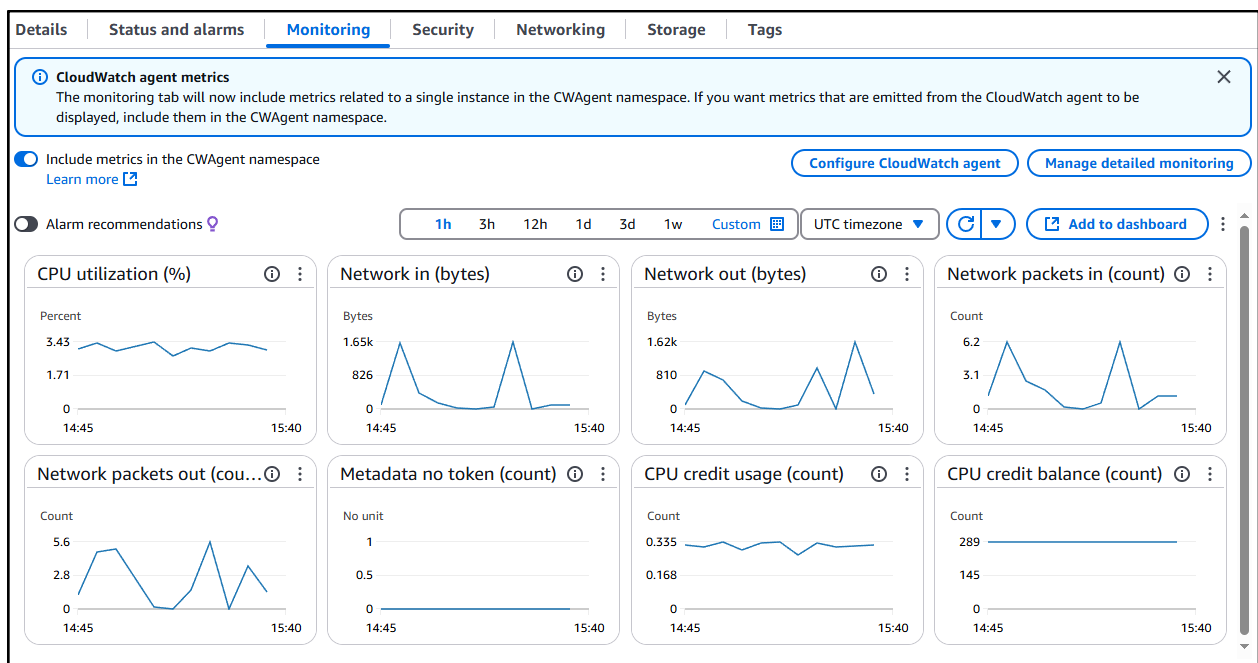


Figure 4: The monitor tab of the EC2 instance depicts the various CPU and Network utilization.

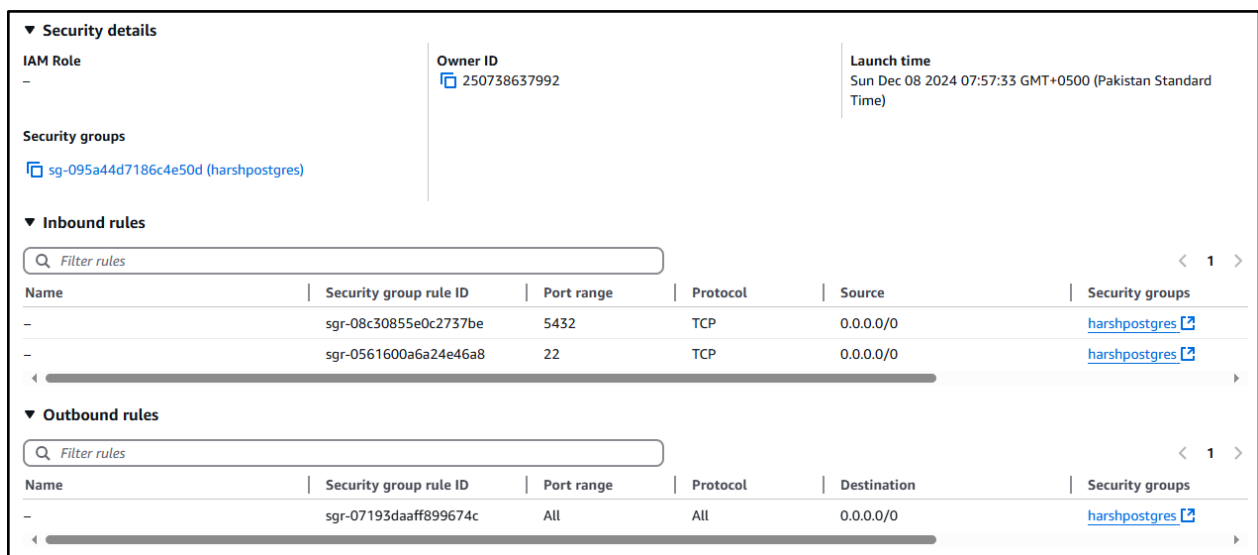


Figure 5: The inbound and the outbound rules for the EC2 instance.

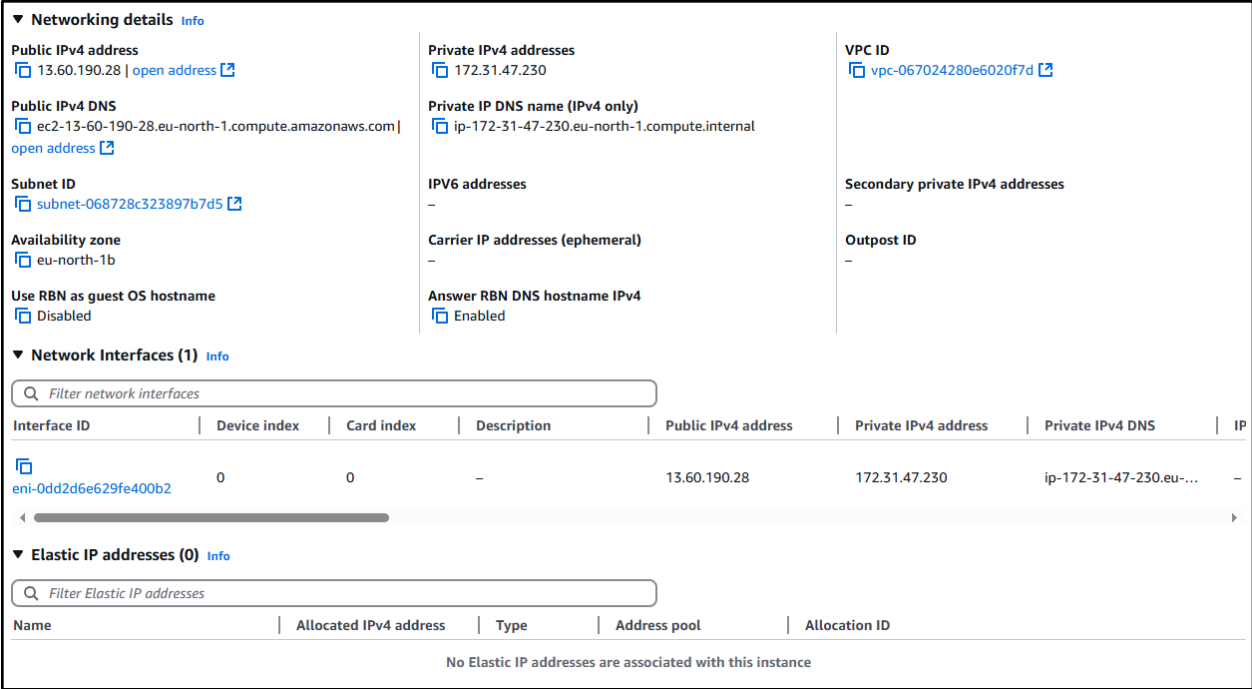


Figure 6: The network details showcase the public and the private IP of the EC2 instance.

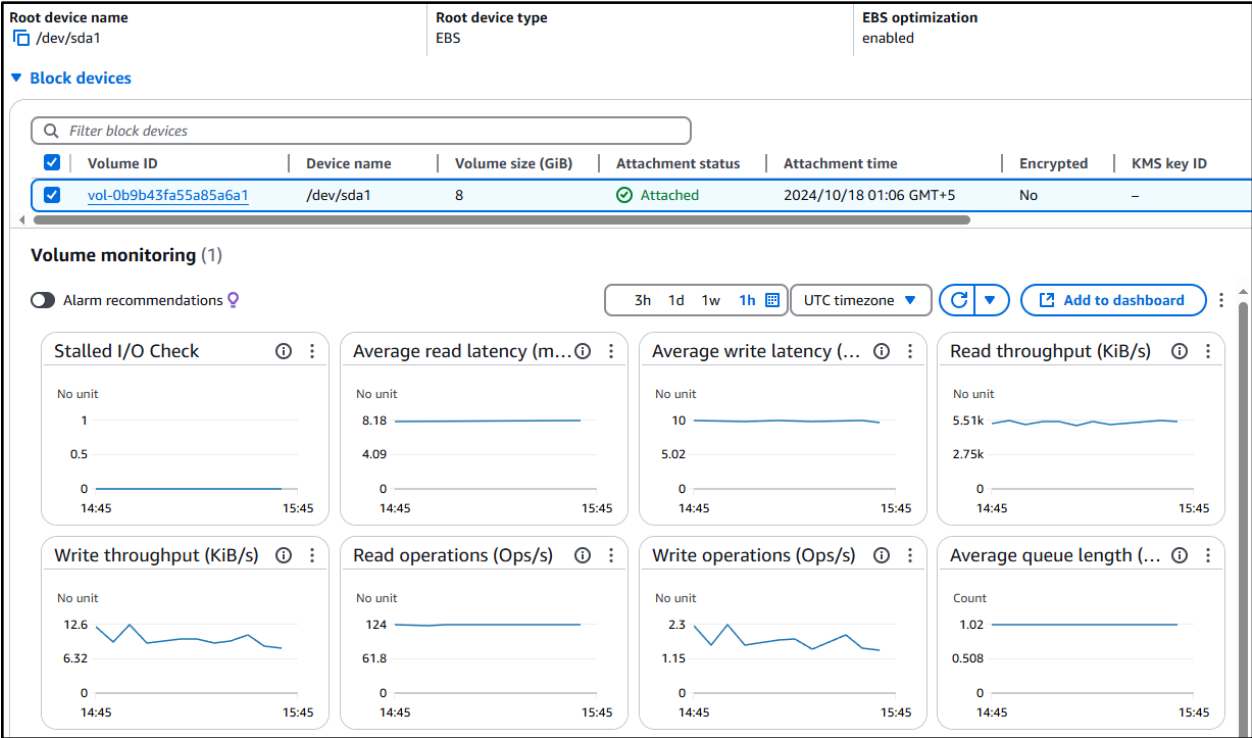


Figure 6: Representation of the EC2 instance’s root and volume privileges.

3.3. Details about the Azure VM

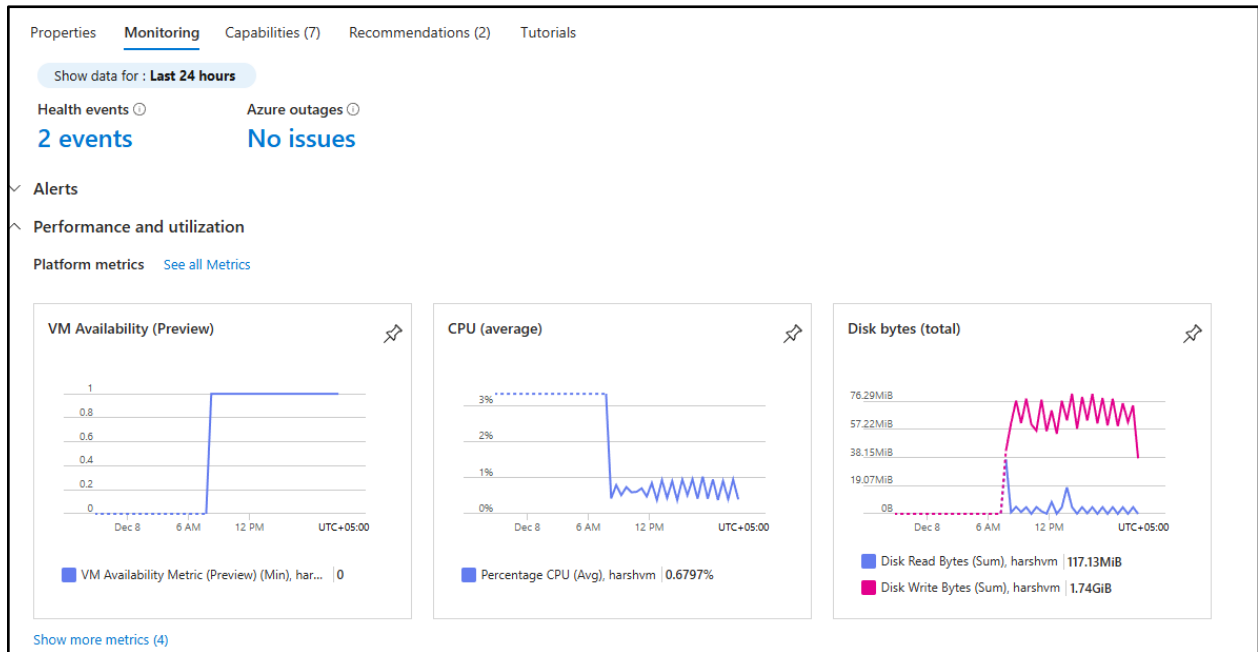


Figure 7: The monitoring tab of Azure VM shows the VM availability and the CPU usage etc.

Resources Recommendations (5)

Filter for any field... Type equals all Location equals all Add filter

Showing 1 to 8 of 8 records. Show hidden types

Name ↑↓	Type ↑↓	Location ↑↓
HarshKCluster	Kubernetes service	West US 2
HarshVM	Virtual machine	West US 2
HarshVM-ip-fec23e7b	Public IP address	West US 2
HarshVM-nsg	Network security group	West US 2
HarshVM-vnet	Virtual network	West US 2
harshvm143-fec23e7b	Network Interface	West US 2
HarshVM_disk1_8a5d2a9ec01c488687797c3af960eecb	Disk	West US 2
HarshVM_key	SSH key	West US 2

Figure 8: Resource Group shown consists of various important services of the VM available.

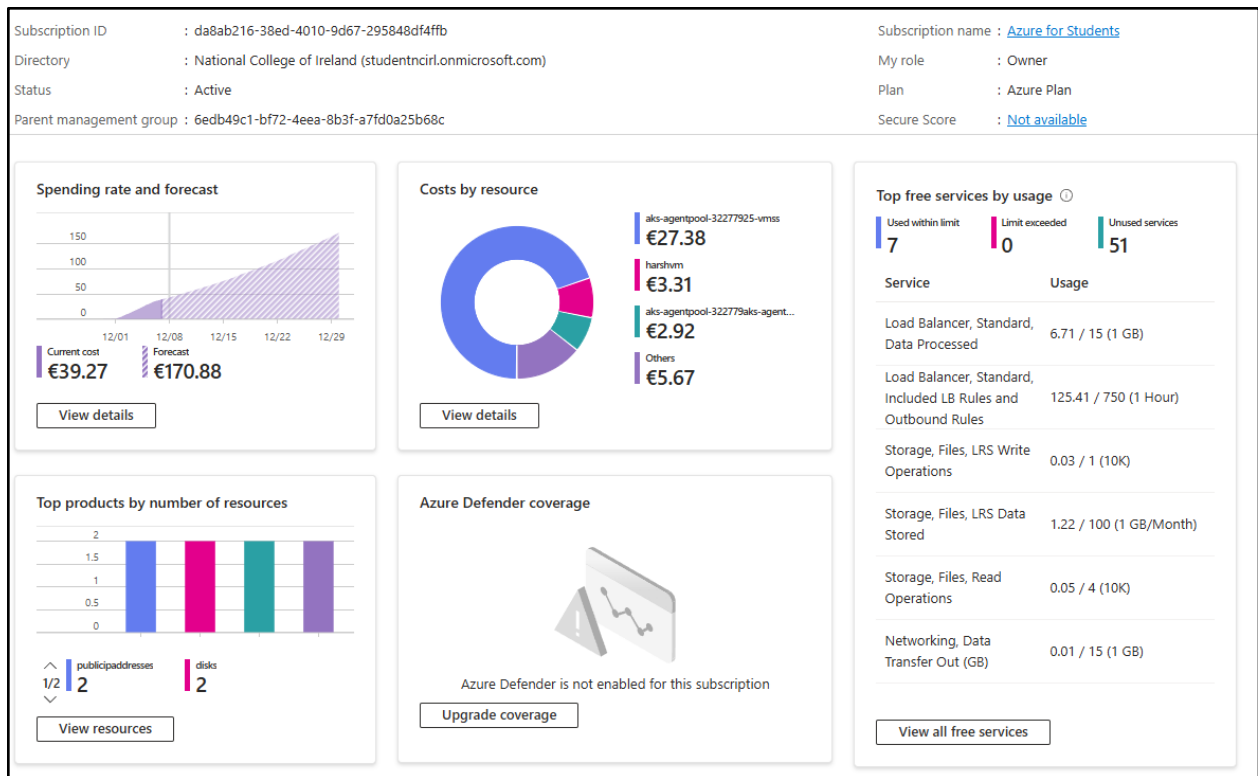


Figure 9: The subscription services of the VM are privileged by the NCI.

3.4. Code and Setup

```
harsh [ ~ ]$ ssh -i HarshVM_key.pem harshazure@74.179.56.30
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1074-azure x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro

System information as of Sun Dec  8 15:59:05 UTC 2024

System load:  0.23           Processes:            130
Usage of /:   16.0% of 28.89GB Users logged in:           0
Memory usage: 5%           IPv4 address for eth0: 10.0.0.4
Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.

  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

44 updates can be applied immediately.
16 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Dec  6 10:00:00 2024 from 4.224.8.127
harshazure@HarshVM:~$
harshazure@HarshVM:~$
```

Figure 10: The first step of the process was to SSH into the HarshVM on the Azure side.

```

harshazure@HarshVM:~$ ls -la -l
total 101508
drwxr-xr-x 12 harshazure harshazure      4096 Dec  6 10:26 .
drwxr-xr-x  3 root          root          4096 Nov  1 05:50 ..
drwxrwxr-x  5 harshazure harshazure      4096 Dec  3 06:46 .azure
-rw-----  1 harshazure harshazure    17069 Dec  6 11:19 .bash_history
-rw-r--r--  1 harshazure harshazure      220 Feb 25 2020 .bash_logout
-rw-r--r--  1 harshazure harshazure     3804 Dec  3 06:46 .bashrc
drwx-----  3 harshazure harshazure      4096 Dec  3 06:11 .cache
drwxrwxr-x  3 harshazure harshazure      4096 Dec  3 06:52 .kube
drwxrwxr-x  3 harshazure harshazure      4096 Dec  3 06:45 .local
drwxr-xr-x  9 harshazure harshazure      4096 Dec  3 05:56 .minikube
-rw-r--r--  1 harshazure harshazure       807 Feb 25 2020 .profile
drwx-----  2 harshazure harshazure      4096 Dec  5 17:07 .ssh
-rw-r--r--  1 harshazure harshazure         0 Nov  1 05:59 .sudo_as_admin_successful
-rw-rw-r--  1 harshazure harshazure       208 Dec  3 06:17 .wget-hsts
drwxrwxr-x  2 harshazure harshazure      4096 Nov  1 07:23 abc
drwxrwxr-x 19 harshazure harshazure      4096 Dec  3 18:18 cri-dockerd
drwxrwxr-x  3 harshazure harshazure      4096 Dec  3 06:11 go
drwxrwxr-x  2 harshazure harshazure      4096 Dec  6 10:30 incoming
-rw-rw-r--  1 harshazure harshazure 103820392 Dec  3 05:55 minikube-linux-amd64
-rw-----  1 harshazure harshazure       104 Dec  6 09:41 nohup.out
-rw-r--r--  1 harshazure harshazure       909 Dec  6 08:59 pagila_migration.log
-rw-rw-r--  1 harshazure harshazure     1217 Dec  3 06:55 postgres-deployment.yaml
-rw-rw-r--  1 harshazure harshazure       187 Dec  6 09:19 postgres-pvc.yaml
-rw-rw-r--  1 harshazure harshazure       237 Dec  6 09:17 postgres-secret.yaml
-rw-rw-r--  1 harshazure harshazure     2263 Dec  6 10:30 transfer.log
-rwxrwxr-x  1 harshazure harshazure     1254 Dec  6 09:30 transfer_logging.sh
-rwxrwxr-x  1 harshazure harshazure       497 Dec  6 09:31 watch_incoming.sh
harshazure@HarshVM:~$
harshazure@HarshVM:~$

```

Figure 11: The illustration depicts the file records of the Azure VM repository.

```

Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help
total 28
drwx----- 2 harshazure harshazure 4096 Dec 12 04:36 .
drwxr-xr-x 12 harshazure harshazure 4096 Dec 12 04:18 ..
-rw----- 1 harshazure harshazure 6101 Dec 12 04:20 authorized_keys
-r----- 1 harshazure harshazure 1831 Dec 12 03:39 azure_to_aws_key
-rw-rw-r-- 1 harshazure harshazure 405 Dec 12 04:36 azure_to_aws_key.pub
-rw-r--r-- 1 harshazure harshazure 666 Dec 12 03:32 known_hosts
harshazure@HarshVM:~/.ssh$ cd ..
harshazure@HarshVM:~$ cat ~/.ssh/azure_to_aws_key.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCFh6+1U9B6p8xheqRXVTplwevLowi0OpDaxtnsAqR0JX1Ks91BEDRnm0SioWMT2gOBx/smqoQiiVQekCcNuQx23jaGuuk3awbERudWakViLlYaKzhyvoiaRpjXpIRUtT
/aj8G1St0dkU2Svht7NABE3s7FJ5w6Xvf++zRPHMEE2p+B4pcjAybeVm5mnwGKUeSuNlqbgMANIMwUlfnaZIwXEntTrigtT0xvtbvbjUxnsFvYXMD0xtiC+csW2JLvo96W6VOpwU6v04+A7mjuWys4461vUeUTByzailUtCtB
TQg+9Nzpq7jg26145nKR9B5V19VEEXlmgBeAW8N9g+LmXX3d ubuntu@ip-172-31-47-230
harshazure@HarshVM:~$ scp -i ~/.ssh/azure_to_aws_key ~/testdb.dump ubuntu@13.60.13.56:~/
testdb.dump
harshazure@HarshVM:~$
100% 843KB 735.9KB/s 00:01

```

Figure 12: Sending Testdb.dump file to AWS from Microsoft Azure

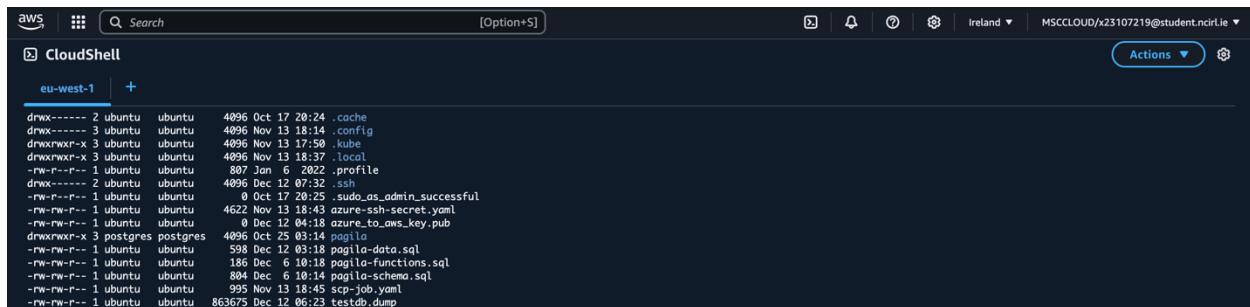


Figure 13: This Figure shows that AWS has received the Testdb.dump file.

```
[cloudshell-user@ip-10-134-54-168 ~]$ ssh -i ~/harsh-ec2-keypair.pem ubuntu@13.60.156.235
The authenticity of host '13.60.156.235 (13.60.156.235)' can't be established.
ED25519 key fingerprint is SHA256:bUuY4D0IE5tNCw/o3aELhnuSb1XoMHeZpYBysWNsGpQ.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: 51.20.4.4
  ~/.ssh/known_hosts:7: 16.171.5.198
  ~/.ssh/known_hosts:8: 16.170.247.153
  ~/.ssh/known_hosts:9: 13.61.100.56
  ~/.ssh/known_hosts:10: 13.60.29.184
  ~/.ssh/known_hosts:11: 16.170.245.206
  ~/.ssh/known_hosts:12: 13.60.231.193
  ~/.ssh/known_hosts:13: 13.60.190.28
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.60.156.235' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1019-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sun Dec  8 02:58:35 UTC 2024

System load:  1.38           Processes:           139
Usage of /:   64.3% of 7.57GB Users logged in:       0
Memory usage: 72%           IPv4 address for ens5: 172.31.47.230
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

24 updates can be applied immediately.
19 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Dec  8 02:58:36 2024 from 63.32.99.218
ubuntu@ip-172-31-47-230:~$
```

Figure 14: The figure shows the process of SSH into the ubuntu EC2 instance on the AWS side.

```
aws Search [Option+S] Ireland MSCLOUD/x23107219@student.ncirl.ie
CloudShell
eu-west-1 +
Expanded Security Maintenance for Applications is not enabled.
13 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
Last login: Thu Dec 12 07:14:18 2024 from 54.170.244.71
ubuntu@ip-172-31-47-230:~$ 2. ssh-keygen -t rsa -b 2048 -f ~/.ssh/aws_ec2_to_azure_vm_key
^Cubuntu@ip-172-31-47-230:~$ ssh-keygen -t rsa -b 2048 -f ~/.ssh/aws_ec2_to_azure_vm_key
Generating public/private rsa key pair.
/home/ubuntu/.ssh/aws_ec2_to_azure_vm_key already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/aws_ec2_to_azure_vm_key
Your public key has been saved in /home/ubuntu/.ssh/aws_ec2_to_azure_vm_key.pub
The key fingerprint is:
SHA256:nmMaj+Lj1BxRq0t+8ddNzqFpp9sD1ej8dfKHwWlrA ubuntu@ip-172-31-47-230
The key's randomart image is:
+---[RSA 2048]-----+
|
| ..
| +. . . . .
| + o . . . o o |
| +.. .S. E.o |
| .0 .+.. .oo |
| ..+0..== oo .|
| O+O.O==.+ +O .|
| .+.. O. O.....|
+---[SHA256]-----+
ubuntu@ip-172-31-47-230:~$
```

Figure 15: The figure shows RSA encryption with SHA 256

```
ubuntu@ip-172-31-47-230:~$ ls -a -l
total 72
drwxr-x--- 8 ubuntu  ubuntu  4096 Dec  6 10:18 .
drwxr-xr-x 3 root    root    4096 Oct 17 20:06 ..
-rw----- 1 ubuntu  ubuntu   7067 Dec  8 02:59 .bash_history
-rw-r--r-- 1 ubuntu  ubuntu    220 Jan  6 2022 .bash_logout
-rw-r--r-- 1 ubuntu  ubuntu   3808 Nov 13 18:12 .bashrc
drwx----- 2 ubuntu  ubuntu   4096 Oct 17 20:24 .cache
drwx----- 3 ubuntu  ubuntu   4096 Nov 13 18:14 .config
drwxrwxr-x 3 ubuntu  ubuntu   4096 Nov 13 17:50 .kube
drwxrwxr-x 3 ubuntu  ubuntu   4096 Nov 13 18:37 .local
-rw-r--r-- 1 ubuntu  ubuntu    807 Jan  6 2022 .profile
drwx----- 2 ubuntu  ubuntu   4096 Dec  6 10:30 .ssh
-rw-r--r-- 1 ubuntu  ubuntu     0 Oct 17 20:25 .sudo_as_admin_successful
-rw-rw-r-- 1 ubuntu  ubuntu   4622 Nov 13 18:43 azure-ssh-secret.yaml
drwxrwxr-x 3 postgres postgres 4096 Oct 25 03:14 pagila
-rw-rw-r-- 1 ubuntu  ubuntu    186 Dec  6 10:18 pagila-functions.sql
-rw-rw-r-- 1 ubuntu  ubuntu    804 Dec  6 10:14 pagila-schema.sql
-rw-rw-r-- 1 ubuntu  ubuntu    995 Nov 13 18:45 scp-job.yaml
ubuntu@ip-172-31-47-230:~$
ubuntu@ip-172-31-47-230:~$
```

Figure 16: The above illustration depicts the file records of the AWS EC2 repository.

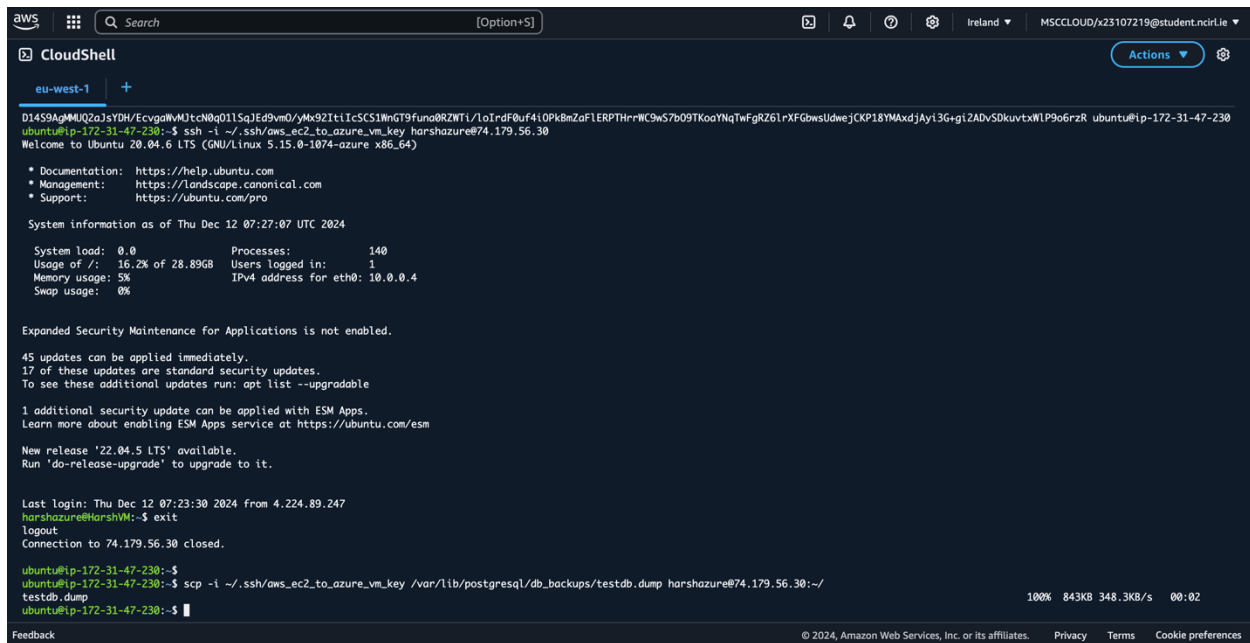


Figure 17: The Figure shows the Testdb.dump file is sent to Microsoft Azure from AWS

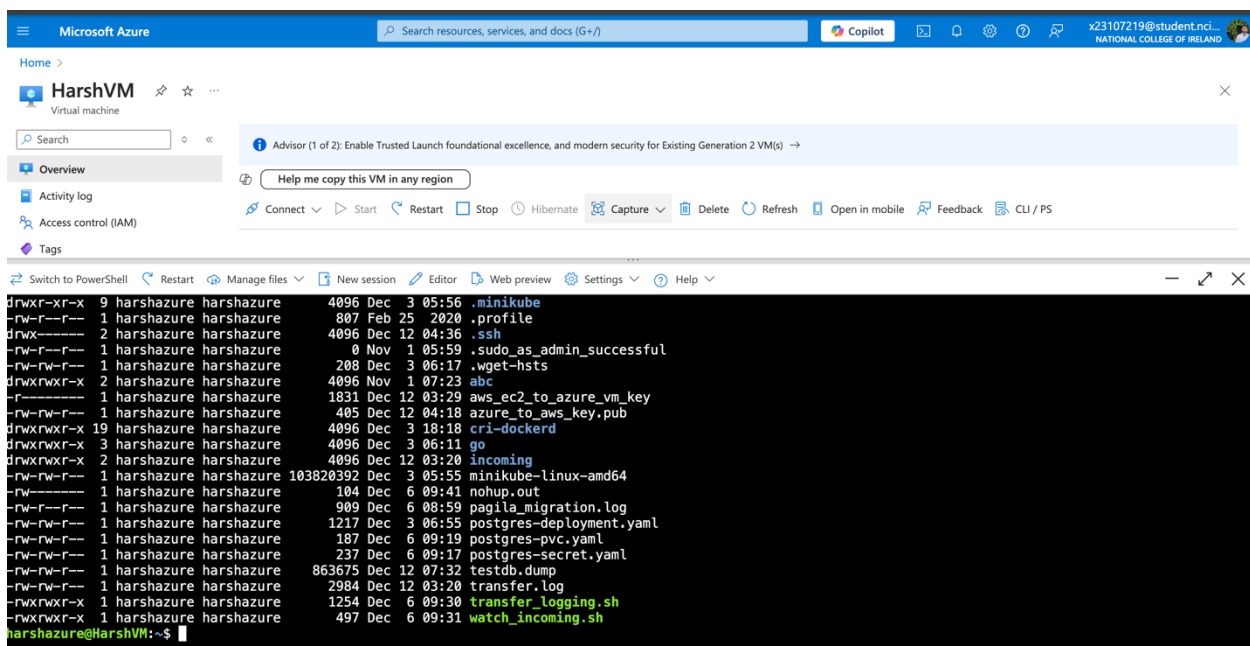


Figure 18: This Figure shows that Microsoft Azure has received the Testdb.dump file.

```

GNU nano 4.8
apiVersion: apps/v1
kind: Deployment
metadata:
  name: postgres
  namespace: my-namespace
spec:
  replicas: 1
  selector:
    matchLabels:
      app: postgres
  template:
    metadata:
      labels:
        app: postgres
    spec:
      containers:
        - name: postgres
          image: postgres:13
          ports:
            - containerPort: 5432
          env:
            - name: POSTGRES_USER
              valueFrom:
                secretKeyRef:
                  name: postgres-secret
                  key: postgres-user
            - name: POSTGRES_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: postgres-secret
                  key: postgres-password
            - name: POSTGRES_DB
              value: "pagila"

```

Figure 19: File contents of the postgres-deployment.yaml on the nano editor.



Figure 20: This figure shows Postgres setup for use

```
ubuntu@ip-172-31-47-230:~/pagila$ ls -a -l
total 33348
drwxrwxr-x 3 postgres postgres 4096 Oct 25 03:14 .
drwxr-x--- 8 ubuntu ubuntu 4096 Dec 6 10:18 ..
drwxrwxr-x 8 postgres postgres 4096 Oct 25 03:14 .git
-rw-rw-r-- 1 postgres postgres 1100 Oct 25 03:14 LICENSE.txt
-rw-rw-r-- 1 postgres postgres 8277 Oct 25 03:14 README.md
-rw-rw-r-- 1 postgres postgres 662 Oct 25 03:14 docker-compose.yml
-rw-rw-r-- 1 postgres postgres 7508866 Oct 25 03:14 pagila-data-apt-jsonb.sql
-rw-rw-r-- 1 postgres postgres 4903550 Oct 25 03:14 pagila-data-yum-jsonb.sql
-rw-rw-r-- 1 postgres postgres 3310905 Oct 25 03:14 pagila-data.sql
-rw-rw-r-- 1 postgres postgres 5334015 Oct 25 03:14 pagila-insert-data.sql
-rw-rw-r-- 1 postgres postgres 7628155 Oct 25 03:14 pagila-insert-data_apt-jsonb.sql
-rw-rw-r-- 1 postgres postgres 5024678 Oct 25 03:14 pagila-insert-data_yum-jsonb.sql
-rw-rw-r-- 1 postgres postgres 331269 Oct 25 03:14 pagila-schema-diagram.png
-rw-rw-r-- 1 postgres postgres 1812 Oct 25 03:14 pagila-schema-jsonb.sql
-rw-rw-r-- 1 postgres postgres 52422 Oct 25 03:14 pagila-schema.sql
ubuntu@ip-172-31-47-230:~/pagila$
```

Figure 21: File contents of the Pagila sample database and their privileges.

3.5. Kubernetes Service Details

Essentials

Resource group

Power state

Cluster operation status

Subscription

Location

Subscription ID

Tags

MyResourceGroup

Running

Succeeded

Azure for Students

West US 2

da8ab216-38ed-4010-9d67-295848df4ffb

Add tags

Kubernetes version

API server address

Network configuration

Node pools

Container registries

1.29.10

harshkcluster-dns-aulmz0ut.hcp.westus2.azmk8s.io

Azure CNI Overlay

1 node pool

Attach a registry

Get started

Properties

Monitoring

Capabilities (5)

Recommendations (3)

Tutorials

Kubernetes services

Encryption type

Virtual node pools

Encryption at-rest with a platform-managed key

Not enabled

Node pools

Node pools

Kubernetes versions

Node sizes

1 node pool

1.29.9

Standard_DS2_v2

Configuration

Kubernetes version

Auto Upgrade Type

Automatic upgrade scheduler

Node security channel type

Security channel scheduler

Authentication and Authorization

Local accounts

1.29.10

Patch

Every week on Sunday (recommended)

Node Image

Every week on Sunday (recommended)

Local accounts with Kubernetes RBAC

Enabled

Networking

API server address

Network configuration

Pod CIDR

Service CIDR

DNS service IP

Cilium dataplane

Network Policy

Load balancer

Private cluster

Authorized IP ranges

Application Gateway ingress controller

harshkcluster-dns-aulmz0ut.hcp.westus2.azmk8s.io

Azure CNI Overlay

10.244.0.0/16

10.0.0.0/16

10.0.0.10

Not enabled

None

Standard

Not enabled

Not enabled

Not enabled

Integrations

Container insights

Workspace resource ID

Service Mesh - Istio

Not enabled

-

Not enabled

Figure 22: The details of the Kubernetes service setup for the Azure file reception and its essentials.

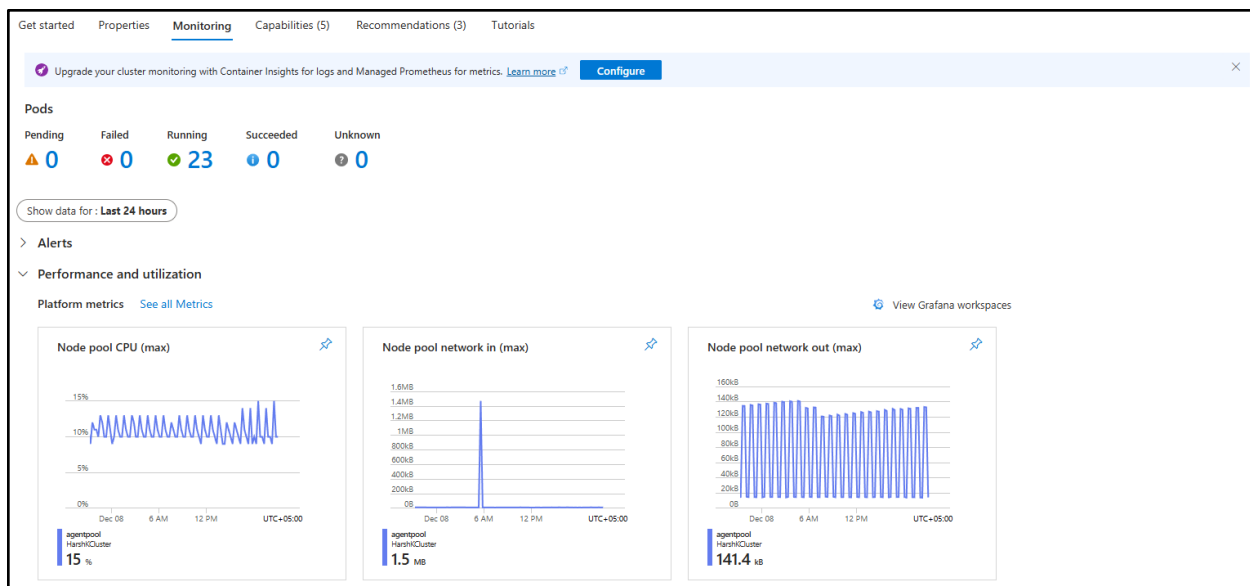


Figure 23: The monitoring tab of the Kubernetes service tracks the status of the Pods and their usage.

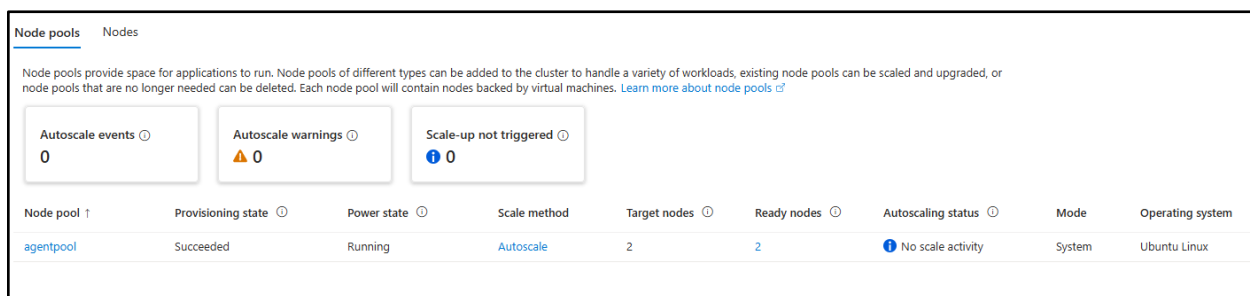


Figure 24: The details of the Node Pool activated and used for the file migration services.

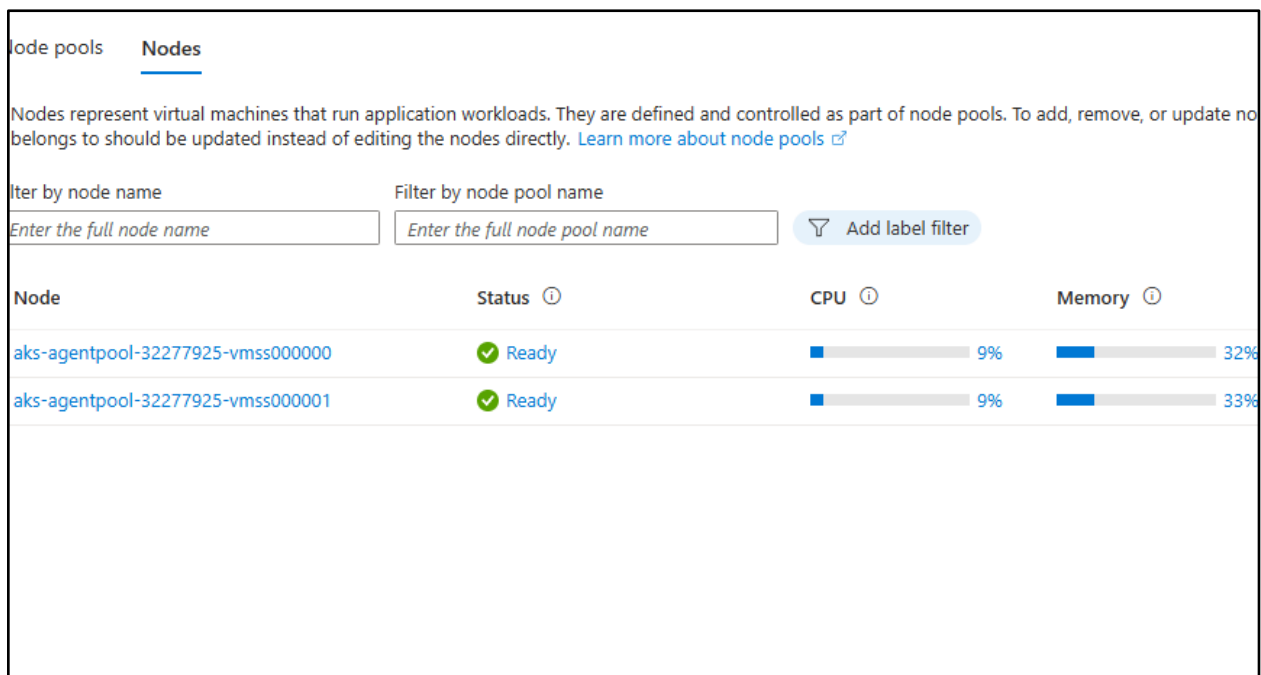


Figure 25: The list of Nodes involved in the Kubernetes Pool count.

3.6. Supporting Migration Code

```
import pandas as pd
import numpy as np
from prophet import Prophet
import matplotlib.pyplot as plt

# Load historical data
cpu_data = pd.read_csv('cpu_usage.csv')
memory_data = pd.read_csv('memory_usage.csv')
transaction_data = pd.read_csv('transaction_rates.csv')
network_data = pd.read_csv('network_usage.csv')

# Preprocess data
def preprocess_data(data, metric_name):
    data['timestamp'] = pd.to_datetime(data['timestamp'])
    data = data[['timestamp', metric_name]]
    data = data.rename(columns={'timestamp': 'ds', metric_name: 'y'})
    return data

cpu_df = preprocess_data(cpu_data, 'cpu_usage')
memory_df = preprocess_data(memory_data, 'memory_usage')
transaction_df = preprocess_data(transaction_data, 'transaction_rate')
network_df = preprocess_data(network_data, 'network_usage')
```

Figure 26: Code to load and preprocess historical data for CPU, memory, transaction, and network metrics in preparation for live migration analysis.

Code Section	Description
import statements	Import required libraries: pandas, numpy, Prophet, and matplotlib.
pd.read_csv()	Load data from CSV files for CPU, memory, transactions, and network usage.
preprocess_data()	Define a function to prepare data by renaming columns and parsing timestamps.
cpu_df to network_df	Preprocess each dataset using preprocess_data function for analysis.

```

# Combine metrics into a single DataFrame
combined_df = cpu_df.copy()
combined_df['y'] = cpu_df['y'] + memory_df['y'] + transaction_df['y'] + network_df['y']
combined_df['y'] = combined_df['y'] / 4 # Average the metrics

#fit the Prophet model
model = Prophet(daily_seasonality=True, weekly_seasonality=True, yearly_seasonality=False)
model.fit(combined_df)

# Create a future DataFrame
future = model.make_future_dataframe(periods=48, freq='H')
forecast = model.predict(future)

#optimal migration time
forecast['yhat_scaled'] = (forecast['yhat'] - forecast['yhat'].min()) / (forecast['yhat'].max() - forecast['yhat'].min())
optimal_time = forecast.loc[forecast['yhat_scaled'].idxmin(), 'ds']

print(f"Optimal migration time: {optimal_time}")

# Plot
model.plot(forecast)
plt.title('Predicted System Activity')
plt.xlabel('Timestamp')
plt.ylabel('Predicted Activity Level')
plt.show()

```

Figure 27: Code to predict system activity levels and determine the optimal migration time for non-web applications using Prophet forecasting.

Code Section	Description
combined_df	Combine CPU, memory, transaction, and network metrics into a single DataFrame.
Averaging metrics	Compute the average of all metrics for balanced analysis.
Prophet model	Initialize and fit the Prophet model with daily and weekly seasonality settings.
Future DataFrame	Create a future DataFrame for predictions over a specified period (48 hours).
Optimal migration time	Calculate the time with the lowest predicted activity level (yhat_scaled).
Forecast plot	Plot the predicted system activity over time.

3.7. Migration Sample Files of Database

```
CREATE TABLE actor (  
  actor_id SERIAL PRIMARY KEY,  
  first_name VARCHAR(45) NOT NULL,  
  last_name VARCHAR(45) NOT NULL,  
  last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP  
);  
  
CREATE TABLE film (  
  film_id SERIAL PRIMARY KEY,  
  title VARCHAR(255) NOT NULL,  
  description TEXT,  
  release_year INT,  
  language_id INT NOT NULL,  
  rental_duration INT NOT NULL DEFAULT 3,  
  rental_rate NUMERIC(4,2) NOT NULL DEFAULT 4.99,  
  length INT,  
  replacement_cost NUMERIC(5,2) NOT NULL DEFAULT 19.99,  
  last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP  
);  
  
CREATE TABLE film_actor (  
  actor_id INT NOT NULL,  
  film_id INT NOT NULL,  
  last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (actor_id, film_id)  
);
```

Figure 28: Schema for managing actor-film relationships, including actor details, film metadata, and many-to-many associations.

Schema Element	Description
actor Table	Stores actor information with fields for ID, first name, last name, and timestamp.
film Table	Stores film details including title, description, release year, language, and pricing.
film_actor Table	Junction table linking actors to films, enabling a many-to-many relationship.
last_update Columns	Tracks the last modification timestamp for all tables.

```

INSERT INTO actor (first_name, last_name) VALUES
('PENELOPE', 'GUINNESS'),
('NICK', 'WAHLBERG'),
('ED', 'CHASE'),
('JENNIFER', 'DAVIS');

INSERT INTO film (title, description, release_year, language_id, length) VALUES
('ACADEMY DINOSAUR', 'A Epic Drama of a Feminist and a Mad Scientist', 2006, 1, 86),
('ACE GOLDFINGER', 'A Astounding Epistle of a Database Administrator', 2006, 1, 48),
('ADAPTATION HOLES', 'A Astounding Reflection of a Lumberjack', 2006, 1, 50);

INSERT INTO film_actor (actor_id, film_id) VALUES
(1, 1),
(1, 2),
(2, 2),
(3, 3),
(4, 1);

```

Figure 29: Sample data insertion for actors, films, and their associations to demonstrate the Pagila schema functionality.

Code Section	Description
actor Table Inserts	Adds sample actors with their first and last names.
film Table Inserts	Adds sample films with details like title, description, year, language, and length.
film_actor Table Inserts	Links actors to films through actor IDs and film IDs for a many-to-many relationship.

```

CREATE OR REPLACE FUNCTION get_film_count() RETURNS INT AS $$
|   SELECT COUNT(*) FROM film;
$$ LANGUAGE SQL;

```

Figure 30: Demonstration function to retrieve the total number of films in the database using a SQL query.

Code Section	Description
CREATE OR REPLACE FUNCTION	Defines or updates a function named get_film_count.
RETURNS INT	Specifies the function returns an integer value.
SELECT COUNT(*) FROM film	Counts the total number of entries (films) in the film table.
LANGUAGE SQL	Declares that the function uses SQL as its programming language.

References

1. Pagila Sample Database found at <https://github.com/devrimgunduz/pagila>
2. PostgreSQL used in both the Azure and the AWS cloud cli and installed using <https://www.postgresql.org/>
3. CRIU usage and installation guide for the tar file https://criu.org/Main_Page
4. Docker installation using the ubuntu from the official site <https://www.docker.com/>