ICT171 – Cloud Server Project

Name: Usman Javed
Student Number: 35190771

Date: 6/9/2025
Public IP Address: 48.216.241.199
Domain Name: ict171.online
GitHub Repo: https://github.com/usmanjav-hub/ictusman.git

Video Explainer:

https://drive.google.com/file/d/1Sr_PItmUuAZrIudylksh7JeXy6ym53w4/view?usp=drivesdk

https://drive.google.com/file/d/1thJkczPzqbLKLCYEgc9zDPsV7zSo7OAq/view?usp=drivesdk

https://drive.google.com/file/d/1IKFI7T4HbPK4Ym0gDG5qT0hGkzK_j2c5/view?usp=drivesdk

https://drive.google.com/file/d/1IKFI7T4HbPK4Ym0gDG5qT0hGkzK_j2c5/view?usp=drivesdk

https://drive.google.com/file/d/1S7uw1pK_6VGt4qbTBguduaWGanuly6Qz/view?usp=drivesdk

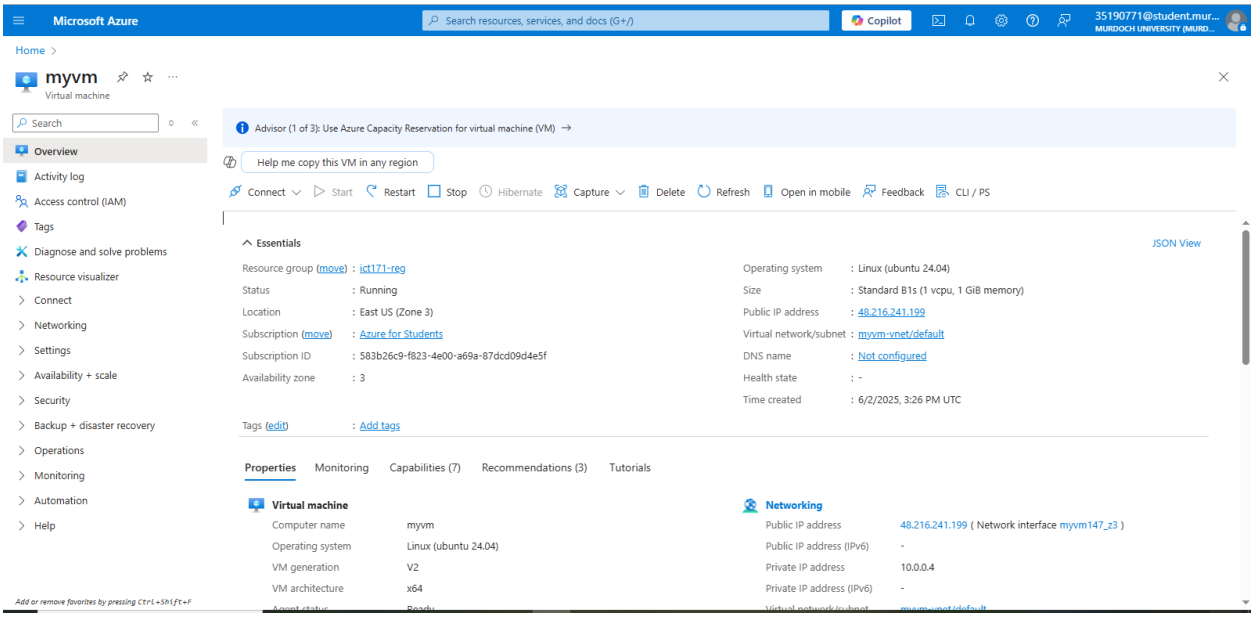
Introduction

This project involves deploying a Linux-based web server in Microsoft Azure using Infrastructure as a Service (IaaS). The purpose is to manually set up and document a server capable of hosting a personal website and demonstrate scripting and system automation.

2. Server Setup

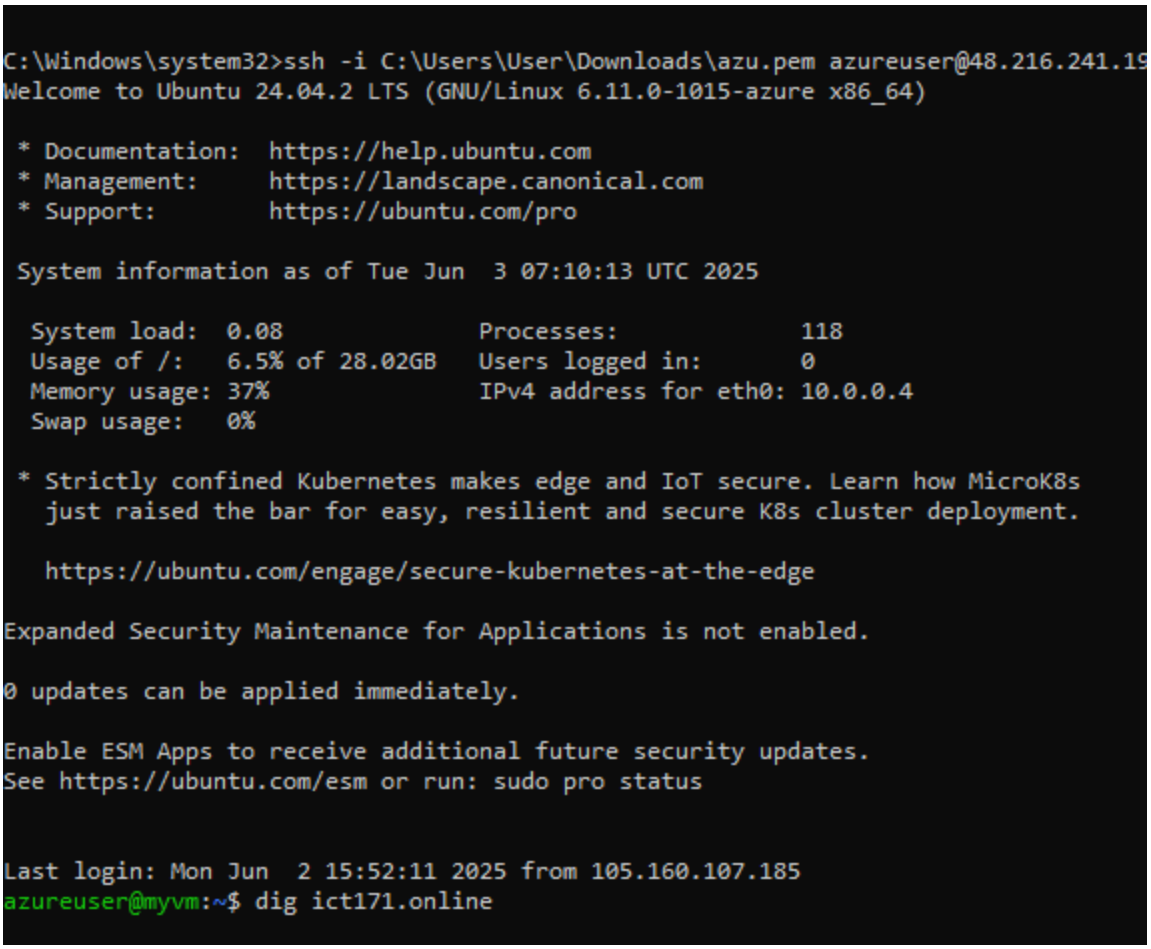
2.1 Azure VM Configuration

- **Resource Group:** ict171-reg
- **VM Name:** myvm
- **Region:** East US (Zone 3)
- **Image:** Ubuntu Server 24.04 LTS
- **Size:** Standard B1s (1 vcpu, 1 GiB memory)
- **Authentication:** SSH Public Key
- **Username:** azureuser
- **Open Ports:** 22 (SSH), 80 (HTTP)



2.2 SSH Access

ssh -i azu.pem azureuser@48.216.241.199



3. Web Server Installation

3.1 Apache Setup

sudo apt update && sudo apt upgrade -y

sudo apt install apache2 -y

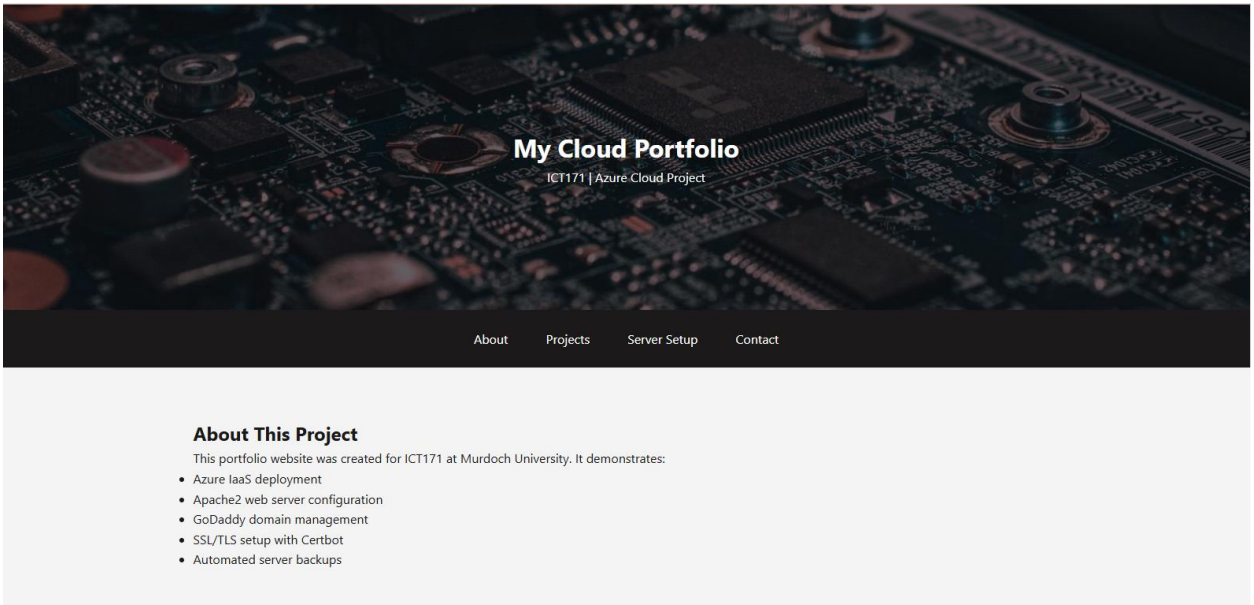
sudo systemctl enable apache2

sudo systemctl start apache2

To verify:

- Visit: http:// 48.216.241.199 or https://ict171.online
- Apache default page should appear.

New updated page



4. DNS Configuration

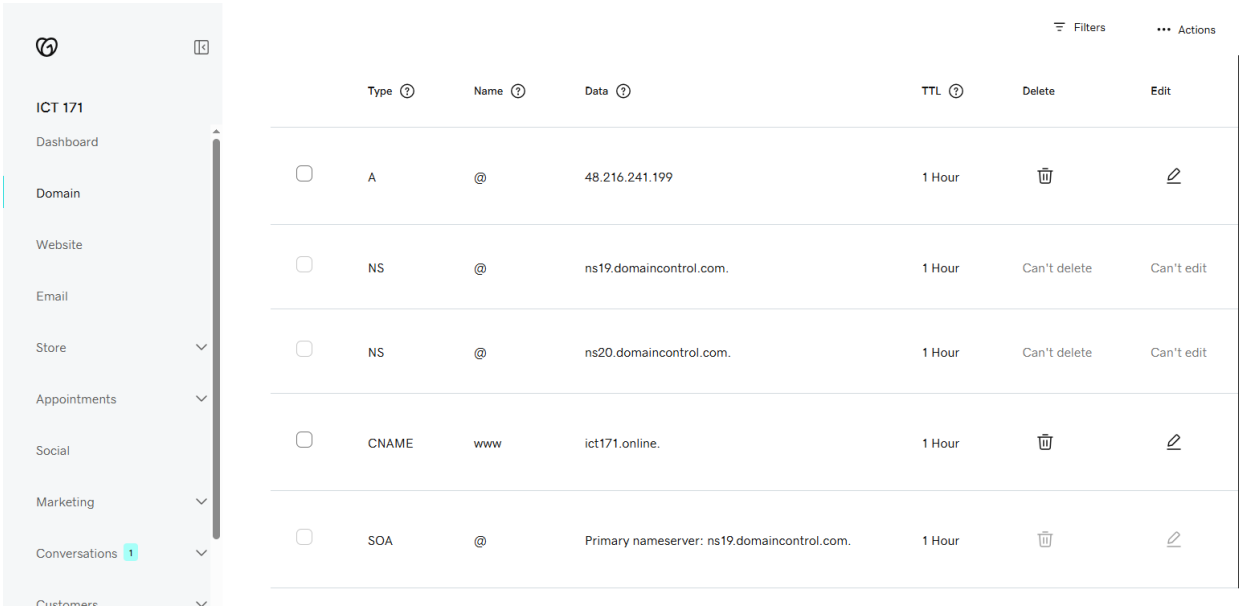
4.1 Domain Used

- Domain Registrar: Go Daddy
- Domain: ict171.online

4.2 DNS Settings

Type	Name	Value
A	@	48.216.241.199

DNS propagation may take up to 30 minutes.



5. Scripting: Web Content Backup Script

1. Script Explanation

This Bash script (backup_website.sh) automates daily backups of your portfolio website's content and Apache configuration. It:

- Creates timestamped backups
- Compresses files into .tar.gz format
- Stores backups in /var/backups/website/
- Keeps only the 7 most recent backups
- Logs all operations

```
#!/bin/bash

# Website Backup Script for ICT171 Assignment

# Purpose: Automatically backs up web content and Apache config
```

```
# Configuration
BACKUP_DIR="/var/backups/website"
LOG_FILE="/var/log/website_backup.log"
WEB_ROOT="/var/www/html"
APACHE_CONF="/etc/apache2"
KEEP_BACKUPS=7

# Create backup directory if missing
mkdir -p "$BACKUP_DIR"

# Generate timestamp
TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
BACKUP_FILE="$BACKUP_DIR/backup_$TIMESTAMP.tar.gz"

# Perform backup
echo "$(date) - Starting backup" >> "$LOG_FILE"
tar -czf "$BACKUP_FILE" "$WEB_ROOT" "$APACHE_CONF" 2>> "$LOG_FILE"

# Clean old backups
ls -t "$BACKUP_DIR"/backup_*.tar.gz | tail -n +$(($KEEP_BACKUPS + 1)) | xargs rm -f

# Verify and log
if [ -f "$BACKUP_FILE" ]; then
   echo "$(date) - Backup successful: $BACKUP_FILE" >> "$LOG_FILE"
else
```

```
    echo "$(date) - Backup failed!" >> "$LOG_FILE"
fi
```

2. How to Use

Installation

1. **Save the script:**

```
sudo nano /usr/local/bin/backup_website.sh
```

2. **Make it executable:**

```
sudo chmod +x /usr/local/bin/backup_website.sh
```

Manual Execution

```
sudo /usr/local/bin/backup_website.sh
```

Automate with Cron (Daily Backups)

1. **Open crontab:**

```
sudo crontab -e
```

2. **Add this line (runs daily at 2 AM):**

```
0 2 * * * /usr/local/bin/backup_website.sh
```

3. How to Verify

Check Backup Files

```
ls -lh /var/backups/website/
```

Sample output:

```
-rw-r--r-- 1 root root 12M Jul 10 02:00 backup_20250710_020001.tar.gz
-rw-r--r-- 1 root root 11M Jul 9 02:00 backup_20250709_020001.tar.gz
```

Verify Backup Contents

```
tar -tzf /var/backups/website/latest_backup.tar.gz | head -10
```

Check Logs

```
tail -5 /var/log/website_backup.log
```

Sample output:

Mon Jul 10 02:00:01 UTC 2025 - Starting backup

Mon Jul 10 02:00:12 UTC 2025 - Backup successful: /var/backups/website/backup_20250710_020001.tar.gz

```
azureuser@myvm:~$ sudo nano /usr/local/bin/backup_website.sh
azureuser@myvm:~$ sudo chmod +x /usr/local/bin/backup_website.sh
azureuser@myvm:~$ sudo nano /usr/local/bin/backup_website.sh
azureuser@myvm:~$ sudo /usr/local/bin/backup_website.sh
azureuser@myvm:~$ sudo crontab -e
no crontab for root - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny
  4. /bin/ed

Choose 1-4 [1]: 1

crontab: installing new crontab
azureuser@myvm:~$ ls -lh /var/backups/website/
total 40K
-rw-r--r-- 1 root root 38K Jun  3 09:07 backup_20250603_090738.tar.gz
```

6. GitHub Repository

📁 GitHub: https://github.com/usmanjav-hub/cloud-server-ict171

Contents:

- backup_web_content.sh
- README.md
- PDF documentation
- Screenshots

7. Video Explainer

Covers:

- VM setup
- SSH access
- Apache install
- DNS setup

8. Troubleshooting and Lessons Learned

- **Issue:** DNS A record delay
  Solution: Waited 30–60 minutes before retrying
- **Issue:** SSH timed out
  Solution: Checked Network Security Group settings to allow port 22
- **Lesson Learned:** Azure makes it easy to configure but still requires careful handling of firewall rules and scripts.

9. References

- DigitalOcean Docs: https://www.digitalocean.com/community/tutorials
- Apache Ubuntu Setup: https://ubuntu.com/server/docs/web-servers-apache
- Azure VM Docs: https://learn.microsoft.com/en-us/azure/virtual-machines/
- Crontab Guru: https://crontab.guru