

ICT171 Assignment 3

Cloud Server Project & Video Explainer

Nurlankzy Aibike

Student ID: 35766774

DNS: <https://aibikeproject.one/>

IP address: <http://20.174.162.206/>

Table of Contents

1.0. Introduction.....	1
2.0 Server Setup.....	7
3.0 Web Server Installation.....	9
4.0 Domain & DNS Configuration.....	10
5.0 SSL/TLS Setup.....	11
6.0 Health Monitoring Script.....	13
7.0 Version Control with GitHub.....	14
8.0 Reference List	15
9.0 Conclusion.....	17

1.0 Introduction

This project involved installing a virtual machine in the clouds, which is made visible to make available a small public web site. The primary aim was to demonstrate practical skills in configuring an Infrastructure-as-a-Service (IaaS) server and documentation of the steps made in the GitHub and the creation of a video explainer that would explain the entire process of end-to-end deployment. I have used an Ubuntu VM on the cloud (IaaS) in this work, which was interconnected to my MacBook via SSH. The technologies involved include the following: Ubuntu (server), the web server is Nginx, and Certbot to provide TLS certificates, Git and GitHub to manage version control and documentation and simple Bash health-check script that is used to demonstrate the monitoring. The project is a project founded on the assignment brief and learning outcomes that require manual configuration of the VM and reproducible documentation.

2.0 Server Setup

Log in to Microsoft Azure, then click the button “Create a new virtual machine”. After that, we will setup the configurations

The screenshot shows the 'Create a virtual machine' wizard on the Microsoft Azure portal. In the 'Resource group' section, a modal dialog is open for creating a new resource group named 'one-proj'. On the right, the 'Estimated monthly costs' table shows a total cost of \$9.13 for a basic VM with Ubuntu Server 24.04 LTS, Standard_B1s size, and no disks or networking.

Category	Description	Cost
Basics	Virtual machine, Image (Ubuntu Server 24.04 LTS), Size (Standard_B1s)	\$9.13
Disks	None	\$0.00
Networking	None	\$0.00
Estimated monthly cost	Total	\$9.13

- * Create new resource group (example: one-proj)
- * Create a VM name
- * Choose the region where the VM locate (in my case Middle East: UAE North)
- * Image: Ubuntu Server 24.04 LTS - x64 Gen2

The screenshot shows the 'Create a virtual machine' wizard. The 'Virtual machine' configuration section includes fields for 'Virtual machine name' (Aibike-proj), 'Region' (Middle East UAE North), 'Availability options' (Self-selected zone, choosing Zone 1), and 'Image' (Ubuntu Server 24.04 LTS - x64 Gen2). To the right, the 'Estimated monthly cost' is displayed as \$9.13/month.

- * Authentication type: SSH public key
- * Username: can create a new username or remain “azureuser”

* Select inbound ports: choose HTTP (80), HTTPS (443), SSH (22)

* Click Review + create

The screenshot shows the 'Create a virtual machine' wizard on the Microsoft Azure portal. In the 'Inbound port rule' section, under 'Public inbound ports', 'Allow selected ports' is selected. Under 'Select inbound ports', 'HTTP (80)', 'HTTPS (443)', and 'SSH (22)' are checked. To the right, the 'Estimated monthly cost' is listed as '\$9.13 / month'. At the bottom, there are buttons for '< Previous', 'Next : Disks >', and 'Review + create'.

After that, Azure gives us the Public and Private IP. However, the IP address should be static
To configure, in section Virtual Machine click “Networking” → “Network settings” → at the top “Network interface/IP configuration” → “ipconfig4” → in section “Allocation” click “Static”

The screenshot shows the 'Network settings' page for the 'Aibike-Nurlankzyz-proj' VM. On the left, the 'Networking' section is expanded, with 'Network settings' selected. The main pane displays the 'ipconfig1 (primary)' network interface, showing its configuration: Network interface: 'aibike-nurlankzyz-proj627_z1 (primary) / ipconfig1 (primary)', Virtual network / subnet: 'Aibike-Nurlankzyz-proj-vnet / default', Public IP address: '20.174.162.206', Private IP address: '10.0.0.4', Admin security rules: '0 (Configure)', Load balancers: '0 (Configure)', Application security group: 'Aibike-Nurlankzyz-proj-nsg', Network security group: 'Aibike-Nurlankzyz-proj-nsg', Accelerated networking: 'Disabled', and Effective security rules: '0'. Below this, the 'Rules' section shows an inbound port rule for port 22 (SSH) from 'Any' source to 'Any' destination, labeled 'Allow'.

On MacOS I used the Terminal application. As I selected SSH public key as the login method. Azure automatically added my public key to the VM. I checked my keys using the command “ls -la ~/.ssh”

```
Last login: Tue Nov 11 10:00:15 on ttys000
aibike@Aibike-Nurlankzy-proj:~$ ls -la ~/.ssh
ubuntu@20.174.162.206: Permission denied (publickey).
aibike@Aibike-Nurlankzy-proj:~% ls -la ~/.ssh
total 46
drwxr-x--- 7 aibike@Aibike-Nurlankzy 224 Oct 29 13:53
drwxr-x--- 34 aibike@Aibike-Nurlankzy 1988 Nov 11 13:59
-rw-r--r-- 1 aibike@Aibike-Nurlankzy staff 2498 Oct 10 23:08 [aibike-Nurlankzy-proj_key.pem]
-rw-r--r-- 1 aibike@Aibike-Nurlankzy staff 123 Oct 10 03:20 _id_d2d3539.pub
-rw-r--r-- 1 aibike@Aibike-Nurlankzy staff 848 Oct 10 23:26 known_hosts
-rw-r--r-- 1 aibike@Aibike-Nurlankzy staff 96 Oct 9 23:16 known_hosts.old
aibike@Aibike-Nurlankzy-proj:~% ssh -T aibike@20.174.162.206
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.16.0-102-generic #68~)
[...]
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Tue Nov 11 10:00:22 UTC 2025

System load: 0.8          Processes:           117
Usage of /: 12.3% of 28.02GB   Users logged in:      0
Memory usage: 44%          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.
0 updates can be applied immediately.

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

*** System restart required ***
Last login: Tue Nov 11 09:03:27 2025 from 91.72.26.65
aibike@Aibike-Nurlankzy-proj:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://archive.ubuntu.com/ubuntu InRelease
Get:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble InRelease
Fetched 126 kB in 1s (148 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
aibike@aibike-Nurlankzy-proj:~$
```

Then I saw the file “Aibike-Nurlankzy-proj_key.pem” which is the name of my VM. Use the command “ssh -i ~/.ssh/Aibike-Nurlankzy-proj_key.pem aibike@20.174.162.206” to the Terminal, due to which I can enter the server.

1. Update the system: sudo apt update && sudo apt upgrade -y
2. Restart the SSH service: sudo systemctl restart ssh
3. Install UFW to secure the server: sudo apt install ufw -y

4. Configure UFW to allow SSH and HTTP/HTTPS traffic:

```
sudo ufw allow OpenSSH
```

```
sudo ufw allow 'Nginx Full'
```

```
sudo ufw enable
```

```
aibike@aibike-Nurlankyzy-proj:~$ sudo systemctl restart ssh
aibike@aibike-Nurlankyzy-proj:~$ sudo apt install ufw -
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
[ufw is already the newest version (0.36.2-6).
[0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
aibike@aibike-Nurlankyzy-proj:~$ sudo ufw allow OpenSSH
Skipping adding existing rule
Skipping adding existing rule (v6)
aibike@aibike-Nurlankyzy-proj:~$ sudo ufw allow 'Nginx Full'
Skipping adding existing rule
[Skipping adding existing rule (v6)
aibike@aibike-Nurlankyzy-proj:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
[Firewall is active and enabled on system startup
aibike@aibike-Nurlankyzy-proj:~$ █
```

3.0 Web Server Installation

1. Install NGINX: sudo apt install -y nginx
2. Verify the service status: sudo systemctl status nginx

```
|firewall is active and enabled on system startup
aibike@aibike-Nurlankyzy-proj:~$ sudo apt install -y nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.24.0-2ubuntu7.5).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
aibike@aibike-Nurlankyzy-proj:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: en>
  Active: active (running) since Mon 2025-11-10 07:58:18 UTC; 1 day 3h ago
    Docs: man:nginx(8)
  Main PID: 391192 (nginx)
    Tasks: 2 (limit: 993)
   Memory: 5.0M (peak: 120.2M)
      CPU: 2.808s
     CGroup: /system.slice/nginx.service
             └─391192 "nginx: master process /usr/sbin/nginx -g daemon on; mast>
               ├ 391193 "nginx: worker process"
Nov 10 07:58:18 Aibike-Nurlankyzy-proj nginx[391191]: 2025/11/10 07:58:18 [warn]>
Nov 10 07:58:18 Aibike-Nurlankyzy-proj systemd[1]: Started nginx.service - A hi>
lines 1-22/22 (END)...skipping...
```

3. Enable to start on boot: sudo systemctl enable --now nginx
4. Create a site configuration file: sudo nano /etc/nginx/sites-available/portfolio
5. Inside the configuration file, specify the server block as follows:

```
GNU nano 7.2
/etc/nginx/sites-available/portfolio *
server {
    listen 80;
    server_name aibikeproject.one www.aibikeproject.one;
    root /var/www/aibikeproject.one/html;
    index index.html;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

Control+o (to save changes) → Enter → Control+x (to exit)

6. Enable the site with: sudo ln -s /etc/nginx/sites-available/portfolio /etc/nginx/sites-enabled
7. Verify the Nginx: sudo nginx -t
8. Reload Nginx: sudo systemctl reload nginx

9. Deploy files into: sudo mkdir -p /var/www/aibikeproject.one/html

10. Set ownership: sudo chown -R www-data:www-data /var/www/aibikeproject.one

```
[aibike@aibike-Nurlankyzy-proj:~$ sudo mkdir -p /var/www/aibikeproject.one/html  
[aibike@aibike-Nurlankyzy-proj:~$ sudo chown -R www-data:www-data /var/www/aibike  
project.one
```

4.0 Domain & DNS Configuration

- Purchase a Domain name (in my case I used NameCheap), log in to the site and choose the appropriate domain that suits to the site:

The screenshot shows the Namecheap domain search results for 'aibikeproject.com'. The main search bar at the top contains 'aibikeproject.com'. Below it, a green checkmark icon and the text '✓ aibikeproject.com' are displayed, along with a price of '\$11.28/yr' and a 'Add to cart' button. A banner at the top indicates a discount: '\$6.49 WITH NEWCOM649'. Below the main search bar, there are several tabs: 'Domains', 'Auctions', 'Premium', 'Generator', 'Beast Mode', and 'Favorites'. The 'Domains' tab is selected. The results section is titled 'Suggested Results' and includes the following domains with their prices and discount information:

Domain	Price	Discount	Add to cart
aibikeproject.bike	\$4.98/yr	88% OFF	Add to cart
aibikeproject.com	\$11.28/yr	\$6.49 WITH NEWCOM649	Add to cart
aibikeproject.to	\$29.98/yr	55% OFF	Add to cart
aibikeproject.org	\$7.48/yr	40% OFF	Add to cart
aibikeproject.net	\$12.98/yr	11% OFF	Add to cart

Below this, there is a 'Results' section with more domain options:

Domain	Price	Discount	Add to cart
aibikeproject.it.com	\$4.98/yr	NEW	Add to cart
aibikeproject.inc	\$498.00/yr	76% OFF	Add to cart
aibikeproject.xyz	\$2.00/yr	90% OFF	Add to cart

- After purchasing, navigate to the DNS settings

Point the domain to the VM's public IP by creating "A Record". In the registrar's DNS panel set '@' → "public IP" → Automatic. Add "CNAME Record" → www → "domain name" → Automatic.

Press save

The screenshot shows the Namecheap Advanced DNS settings for the domain 'aibikeproject.one'. On the left, a sidebar lists various services: Domain List, Hosting List, Private Email, SSL Certificates, Apps, My Offers, and Profile. The 'Advanced DNS' tab is selected in the main header. The interface includes sections for 'DNS TEMPLATES', 'HOST RECORDS', 'DNSSEC', and 'MAIL SETTINGS'. Under 'HOST RECORDS', there are two entries:

Type	Host	Value	TTL
A Record	@	20.174.162.206	Automatic
CNAME Record	www	aibikeproject.one.	Automatic

At the bottom of the 'HOST RECORDS' section, there is a red 'ADD NEW RECORD' button. Below the records, there are sections for 'DNSSEC' and 'MAIL SETTINGS'.

- Verify Domain Link by using an online site: <https://dnschecker.org/>

5.0 SSL/TLS Setup

1. Install Certbot: sudo apt install -y certbot python3-certbot-nginx
2. Install SSL certificate: sudo certbot --nginx -d aibikeproject.one -d www.aibikeproject.one
3. Verify the certificate: sudo certbot renew --dry-run
4. Ensure that the site opens with HTTPS

```
aibike@aibike-Nurlankzyz-proj:~$ sudo apt install -y certbot python3-certbot-nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
certbot is already the newest version (2.9.0-1).
python3-certbot-nginx is already the newest version (2.9.0-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
aibike@aibike-Nurlankzyz-proj:~$ sudo certbot --nginx -d aibikeproject.one -d www.aibikeproject.one
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Certificate not yet due for renewal

You have an existing certificate that has exactly the same domains or certificate name you requested and isn't close to expiry.
(ref: /etc/letsencrypt/renewal/aibikeproject.one.conf)

What would you like to do?
-----
1: Attempt to reinstall this existing certificate
2: Renew & replace the certificate (may be subject to CA rate limits)
-----
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 1
Deploying certificate
Successfully deployed certificate for aibikeproject.one to /etc/nginx/sites-enabled/aibikeproject.one
Successfully deployed certificate for www.aibikeproject.one to /etc/nginx/sites-enabled/aibikeproject.one
Congratulations! You have successfully enabled HTTPS on https://aibikeproject.one and https://www.aibikeproject.one
[

-----
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
 * Donating to EFF: https://eff.org/donate-le
-----
aibike@aibike-Nurlankzyz-proj:~$ sudo certbot renew --dry-run
Saving debug log to /var/log/letsencrypt/letsencrypt.log

-----
Processing /etc/letsencrypt/renewal/aibikeproject.one.conf
-----
Account registered.
Simulating renewal of an existing certificate for aibikeproject.one and www.aibikeproject.one

-----
Congratulations, all simulated renewals succeeded:
 /etc/letsencrypt/live/aibikeproject.one/fullchain.pem (success)
-----
aibike@aibike-Nurlankzyz-proj:~$
```

6.0 Health Monitoring Script

I developed a simple Bash script named “aibike_check.sh” to show the monitoring capability of my website. The script sends an HTTP request to the website, records the timestamp with the HTTPS code and can be alerted when the status is not 200.

1. Navigate to project: cd ~/ICT171-cloud-project (I named my repository in GitHub like that)
2. Move the scripts subdirectory: cd ~/ICT171-cloud-project/server/scripts
3. Open a nano text to create a new script file: nano aibike_check.sh
4. Write a simple code:

* URL → target URL for the health check
* TIMESTAMP → captures the exact time and date of the check
* HTTP_CODE → executes the core check
* LOG_FILE → defines the path to the log file
* echo → writes the output to the log file
* if [“\$HTTP_CODE” -ne 200] → checks if the returned HTTP status is not equal to 200
* echo → if the check fails, an alert message

```
UW PICO 5.09                                         File: aibike_check.sh
URL="https://aibikeproject.one"
# Текущая дата и время
TIMESTAMP=$(date +"%Y-%m-%d %H:%M:%S")

# Получаем HTTP-код ответа сайта
HTTP_CODE=$(curl -o /dev/null -s -w "%{http_code}" $URL)

# Записываем результат в лог-файл
LOG_FILE=~/ICT171-cloud-project/server/scripts/health.log
echo "$TIMESTAMP - $URL - $HTTP_CODE" >> $LOG_FILE

# Проверяем, доступен ли сайт (код 200)
if [ "$HTTP_CODE" -ne 200 ]; then
    # Если сайт недоступен - отправляем e-mail
    echo "$TIMESTAMP - ALERT - $URL returned $HTTP_CODE" | mail -s "Site down" you@example.com
fi
```

5. Give a permission: chmod +x aibike_check.sh
6. Go back to the main project: cd ~/ICT171-cloud-project
7. Stage a new script: git add server/scripts/aibike_check.sh
8. Save changes: git commit -m “feat: add aibike-check script”
9. Upload the changes to the repository: git push

```
aibikenurlankzy@MacBook-Air-Ajbike ICT171-cloud-project % cd ~/ICT171-cloud-project/server/scripts  
aibikenurlankzy@MacBook-Air-Ajbike scripts % nano aibike_check.sh  
aibikenurlankzy@MacBook-Air-Ajbike scripts % chmod +x aibike_check.sh  
aibikenurlankzy@MacBook-Air-Ajbike scripts % cd ~/ICT171-cloud-project  
  
aibikenurlankzy@MacBook-Air-Ajbike ICT171-cloud-project % git add server/scripts/aibike_check.sh  
aibikenurlankzy@MacBook-Air-Ajbike ICT171-cloud-project % git commit -m "feat: add aibike_check script"  
[main f3fecfd] feat: add aibike_check script  
[ Committer: Айбике Королева <aibikenurlankzy@MacBook-Air-Ajbike.local>  
Your name and email address were configured automatically based  
on your username and hostname. Please check that they are accurate.  
[You can suppress this message by setting them explicitly. Run the  
following command and follow the instructions in your editor to edit  
your configuration file:  
  
    git config --global --edit  
  
After doing this, you may fix the identity used for this commit with:  
  
    git commit --amend --reset-author  
  
1 file changed, 23 insertions(+)  
create mode 100755 server/scripts/aibike_check.sh  
aibikenurlankzy@MacBook-Air-Ajbike ICT171-cloud-project % git push  
Enumerating objects: 6, done.  
Counting objects: 100% (6/6), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (5/5), 1.02 KiB | 1.02 MiB/s, done.  
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/aibikekoroleva-sudo/ICT171-cloud-project.git  
 [ 732a686..f3fecfd main -> main  
aibikenurlankzy@MacBook-Air-Ajbike ICT171-cloud-project % ]
```

Then I created a file README.md which contains the project's overview, student_info.txt where the report pdf is located and video_link.txt where my video explainer is provided.

1. Navigate to project: cd ~/ICT171-cloud-project
 2. Create a file: touch README.md
 3. Open a nano text to write: nano README.md (I wrote my Student ID, DNS, IP address and overview)
 4. Click Control+o → Enter → Control+x

So people can see my project's portfolio due to README.md

```
aibikenurlankzy@MacBook-Air-Ajbike ICI171-cloud-project % touch README.md
aibikenurlankzy@MacBook-Air-Ajbike ICI171-cloud-project % nano README.md

UW PICO 5.09                                         File: README.md

# ICI171 Cloud Project - My Portfolio

Student: Alibike Nurlankzy
Student ID: 35766774

Public IP: 20.174.162.206
Domain: https://ajbikeproject.one

## Overview
One Development Company is a leading construction developer based in Dubai with a reputation of growth and residential developments. This company was chosen for its ambitious projects, focus on the quality of the work, and commitment to innovation. The company's mission is to create sustainable and eco-friendly living spaces that meet the highest standards of safety and comfort.

The site will present One Development as a professional developer of the highest standards. It will have a clear homepage with an overview of the corporation and highlighted projects. The Projects section will feature a grid of thumbnail images representing completed and ongoing projects. Each project page will provide detailed information about the scope, location, and features of the development.

The site will be hosted on an Infrastructure as a Service (IaaS) model based on Microsoft Azure. A virtual machine will be configured with a static public IP address: 20.174.162.206. Security measures include SSL encryption and regular backups to ensure data integrity and availability.

The final deliverable will be a concise marketing website providing a visual presentation of the projects and corporate image of One Development. It will give potential clients and investors easy access to information about the company's capabilities and track record.
```

Reference list

Conclusion

In conclusion, I successfully deployed an Ubuntu Virtual Machine, installed and configured NGINX, wrote a monitoring script, and provided a video-explainer of the process in GitHub. Throughout the execution, I gained practical experience with using keys on macOS, integrating NGINX with system-managed applications and the importance of maintaining reproducible server documentation. The issues were the delay in the propagation of DNS and minor syntax errors in NGINX configuration.