

ICT171 Assignment 3

Cloud Server Project & Video Explainer

Name: Nurlankazy Aibike

Student ID: 35766774

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

IP address: 20.174.162.206

Link to Github: <https://github.com/aibikekoroleva-sudo/ICT171-cloud-project>

Table of Contents

1.0. Introduction.....	1
2.0 Server Setup.....	2
3.0 Web Server Installation.....	6
4.0 Domain & DNS Configuration.....	8
5.0 SSL/TLS Setup.....	9
6.0 Monitoring Script.....	10
7.0 Conclusion	12
8.0 Reference list.....	13

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 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 connected to my MacBook via SSH. The involved technologies include the following: Ubuntu, the web server Nginx, 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 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' panel displays the following breakdown:

Category	Cost
Basics	\$9.13
Virtual machine	\$9.13
Image	\$0.00
Size	\$9.13
Standard_B1s	
Disks	\$0.00
Networking	\$0.00
Estimated monthly cost	\$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. In the 'Virtual machine' section, the VM name is set to 'Aibike-proj', the region is 'Middle East (UAE North)', and the availability zone is 'Zone 1'. The estimated monthly cost is \$9.13. The 'Image' section shows 'Ubuntu Server 24.04 LTS - x64 Gen2' selected. The 'VM architecture' section has 'x64' selected. At the bottom, there are buttons for 'Next: Disks >' and 'Review + create'.

* 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 in the Azure portal. The 'Review + create' step is active. Configuration details include:

- Username:** azureuser
- SSH public key source:** Generate new key pair
- SSH Key Type:** RSA SSH Format (selected)
- Key pair name:** Aibike-proj_key
- Inbound port rule:** Allow selected ports (selected)
- Select inbound ports:** HTTP (80), HTTPS (443), SSH (22) (all selected)

The estimated monthly cost is \$9.13/month.

After that, Azure gives us the Public and Private IP. However, the IP address should be static. To configure this, 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 VM 'Aibike-Nurlankzyz-proj'. The 'ipconfig1 (primary)' configuration is selected. Key details include:

- 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 groups:** 0 (Configure)
- Network security group:** Aibike-Nurlankzyz-proj-nsg
- Accelerated networking:** Disabled
- Effective security rules:** 0

The 'Inbound port rules' section shows a single rule:

Priority	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	Allow

On MacOS I used the Terminal application. Before, 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”

```
aibike@nurlankzyz:~$ ls -la ~/.ssh
total 46
drwxr-x--- 7 aibikenurlankzyz staff 224 Oct 29 13:53 .
drwxr-x--- 34 aibikenurlankzyz staff 1888 Nov 11 13:59 ..
-rw-r--r-- 1 aibikenurlankzyz staff 2498 Oct 29 23:01 Aibike-Nurlankzyz-proj_key.pem
-rw-r--r-- 1 aibikenurlankzyz staff 123 Oct 10 03:20 ed25589.dub
-rw-r--r-- 1 aibikenurlankzyz staff 840 Oct 9 23:25 known_hosts
-rw-r--r-- 1 aibikenurlankzyz staff 1024 Oct 10 03:20 hosts.old
aibike@nurlankzyz:~$ ssh -i ~/.ssh/Aibike-Nurlankzyz-proj_key.pem aibike@20.174.162.206
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1012-azure x86_64)

 * 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: 0.0 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 91.73.29.65
aibike@Aibike-Nurlankzyz-proj:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://azures.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://azures.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://azures.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://azures.archive.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble InRelease
Fetching 126 kB in 1s (148 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
aibike@Aibike-Nurlankzyz-proj:~$ 
```

Then I saw the file “Aibike-Nurlankzyz-proj_key.pem” which is the name of my VM. Use the command “ssh -i ~/.ssh/Aibike-Nurlankzyz-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

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/aibtikeproject.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

1. 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 website with the search term 'aibikeproject.com' entered. The main result is 'aibikeproject.com' at \$11.28/yr. Below it, a list of suggested domains is shown, each with its price and a 'Add to cart' button. The suggested domains include 'aibikeproject.bike', 'aibikeproject.to', 'aibikeproject.org', 'aibikeproject.net', 'aibikeproject.it.com', 'aibikeproject.inc', and 'aibikeproject.xyz'. Each entry also includes a 'Retail' price and a '40% OFF' discount.

2. 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 'Advanced DNS' section of the Namecheap control panel for the domain 'aibikeproject.one'. The left sidebar lists various services like Domain List, Hosting List, Private Email, etc. The main interface shows the 'Advanced DNS' tab selected. Under 'HOST RECORDS', there are two entries: an 'A Record' for '@' pointing to 20.174.162.206 with an automatic TTL, and a 'CNAME Record' for 'www' pointing to 'aibikeproject.one.' with an automatic TTL. A red 'ADD NEW RECORD' button is visible. Below the records, sections for 'DNSSEC' and 'MAIL SETTINGS' are present.

3. 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 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

```
aibikenurlankzyy@MacBook-Air-Ajbike ICT171-cloud-project % cd ~/ICT171-cloud-project/server/scripts  
  
aibikenurlankzyy@MacBook-Air-Ajbike scripts % nano aibike_check.sh  
aibikenurlankzyy@MacBook-Air-Ajbike scripts % chmod +x aibike_check.sh  
aibikenurlankzyy@MacBook-Air-Ajbike scripts % cd ~/ICT171-cloud-project  
  
aibikenurlankzyy@MacBook-Air-Ajbike ICT171-cloud-project % git add server/scripts/aibike_check.sh  
aibikenurlankzyy@MacBook-Air-Ajbike ICT171-cloud-project % git commit -m "feat: add aibike_check script"  
[!main f3fecfd] feat: add aibike_check script  
Committer: Айбике Королева <aibikenurlankzyy@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  
aibikenurlankzyy@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/abikerekoroleva-sudo/ICT171-cloud-project.git  
 732a606..f3fecfd main --> main  
aibikenurlankzyy@MacBook-Air-Ajbike ICT171-cloud-project %
```

Then I created a file `README.md` which contains the project's overview.

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

```
sibikenurlankyzy@MacBook-Air-Ajbike ICT171-cloud-project % touch README.md
sibikenurlankyzy@MacBook-Air-Ajbike ICT171-cloud-project % nano README.md

UW PICO 5.09                                         File: README.md

# ICT171 Cloud Project - My Portfolio

Student: Albiye Nurlanlykyzy
Student ID: 35766774

Public IP: 20.174.162.206
Domain: https://sibikeproject.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 work, and innovative designs. The website will serve as a platform for potential clients to learn about their services and explore their portfolio.

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 Project section will feature detailed descriptions of completed and ongoing projects, along with images and links to more information. The Contact section will provide users with multiple ways to get in touch, including email, phone, and social media links.

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.
```

After that, I simply uploaded my report in pdf and video explainer in my repository.

7.0 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.

8.0 Reference list

Electronic Frontier Foundation. (n.d.). *User guide*. Certbot Documentation.

<https://eff-certbot.readthedocs.io/en/stable/using.html>

GitHub Docs. (n.d.). *About Git*. GitHub.

<https://docs.github.com/en/get-started/using-git/about-git>

Jawaid, B. (n.d.). *Building a bash script to monitor server performance*. Medium.

<https://medium.com/@basitjawaid123/building-a-bash-script-to-monitor-server-performance-my-journey-891e68b13a49>

Microsoft. (n.d.). *Quickstart: Create an Azure virtual network*. Microsoft Learn.

<https://learn.microsoft.com/en-us/azure/virtual-network/quickstart-create-virtual-network>

Nginx, Inc. (n.d.). *Beginner's guide*. Nginx Documentation.

https://nginx.org/en/docs/beginners_guide.html

SSL Labs. (2020). *SSL and TLS deployment best practices*. GitHub.

<https://github.com/ssllabs/research/wiki/ssl-and-tls-deployment-best-practices>