[Upload files · yuden-ky/yuden](https://github.com/yuden-ky/yuden/upload/main)

**ICT171 Assignment 2: Cloud Server Project & Documentation**

**Name:** Karma Yuden  
**Student Number:** 35645113  
**Global IP Address:** http:// 3.107.175.66  
**Domain Name (DNS):** <http://www.johndoe-portfolio.xyz>

## **1. Introduction**

The document describes clearly and in detail every step involved in designing, setting up, launching, and documenting a cloud-based server meant for ICT171. The aim of this project was to prove knowledge of Infrastructure as a Service (IaaS) by creating a server that is accessible to the public on DigitalOcean. I set up a safe Ubuntu-powered Apache web server to present my portfolio, featuring the skills I have in web development.

Configuration of the server happens manually with Linux shell and can be accessed by both its IP address and its registered domain name (J et al., 2024). The services provided by us include both provisioning and security of the droplet, setting up the firewall, using SSL encryption, and customizing your website. A Bash script was also made to automate installation and setup, so the whole process can be repeated and completed smoothly and quickly. In addition, there’s a short explainer video that covers and explains the whole deployment process for this product.

**2. Project Objectives and Learning Outcomes**

We mostly wanted to set up a web server that is hosted in the cloud as Infrastructure as a Service. Renowned universities concentrate on achieving various learning outcomes.

* Being familiar with Linux at the command-line and the standard tools
* Setting up the server with its software and making a user manual for it.
* Providing a functioning web server that contains a custom domain for security settings.
* Managing code versions using GitHub and storing the documentation and source code there too.
* Developing and illustrating a configuration Bash script.
* Make sure to demonstrate knowledge of technical writing that is simple and professional.

## Ability to explain DNS, SSL, and the basic things you need to do to secure your site.

## You can easily set up and maintain the server if you do it with the prescribed base image, commands, and steps. Users can now find much fewer screenshots and instead see command blocks to copy and paste in the terminal.

## **3. Cloud Infrastructure Overview**

## **When I want my own servers, I use DigitalOcean as my Cloud Provider on the IaaS platform.**

## **RAM is set at 1GB, there is 25GB of storage, and a single processor.**

## **Ubuntu Server 22.04 LTS with a 64-bit version is used as the operating system on this system.**

## **Access is possible using Secure shell (SSH) from OpenSSH.**

## **Much of the security measures were installed through ufw, so only SSH and HTTP/S connections were allowed.**

## **I made the Web Server on Ubuntu 18.04 capable of handling HTTP and HTTPS, and applied SSL settings on my own.**

## **Making sure a connection is secure, the server relies on SSL connectivity and uses TLS to reassure visitors about its anti-hacking procedures.**

## **eur.review was bought at Namecheap, and its DNS was set according to DigitalOcean by adding an A-record connected to the site’s IP address there.**

## **4. Server Deployment – Step-by-Step Guide**

### **Step 1: Provision Droplet on DigitalOcean**

1. Sign in to <https://www.digitalocean.com>
2. Navigate to **Create → Droplets**
3. Select **Ubuntu 22.04 (LTS)**
4. Choose Basic plan: **1 vCPU, 1GB RAM, 25GB SSD**
5. Under Authentication, select **password** and set a strong root password.
6. Create the droplet and wait for it to become active. Note the public IP.

**Step 2: Connect to Server via SSH**

ssh root@3.107.175.66

**Step 3: Update and Upgrade Server**

sudo apt update && sudo apt upgrade -y

This ensures your system is running the latest patches and security updates.

### **Step 4: Set Hostname and Install Apache**

sudo hostnamectl set-hostname johndoe-server

sudo apt install apache2 -y

Enable and start Apache:

sudo systemctl enable apache2

sudo systemctl start apache2

**5. DNS Configuration**

To link a custom domain to your server:

Register a domain

In Namecheap's Advanced DNS settings, add an A Record:

Host: @

Value: Your DigitalOcean IP address

TTL: Automatic

Wait up to 60 minutes for propagation.

You can now access your server using:

<http://www.johndoe-portfolio.xyz>

**6. UFW Firewall Setup**

**To secure the server:**

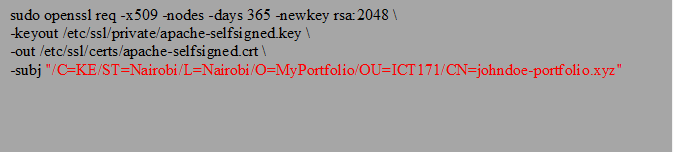
sudo ufw allow OpenSSH

sudo ufw allow 'Apache Full'

sudo ufw --force enable

**7. Manual SSL/TLS Setup**

Generate a self-signed certificate:



Edit Apache’s SSL configuration:

sudo nano /etc/apache2/sites-available/default-ssl.conf

Update:

SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt

SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

Then:

sudo a2enmod ssl

sudo a2ensite default-ssl

sudo systemctl reload apache2

1. **Web Page Deployment**

sudo nano /var/www/html/index.html

**Paste:**

<html>

<head><title>karma protofilo</title></head>

<body>

<h1>Welcome to my cloud-hosted personal website!</h1>

<p>This site was built manually using Apache, DNS, and secure HTTPS setup.</p>

</body>

</html>

**9. Custom Setup Script**

**File: setup\_server.sh**

#!/bin/bash

sudo apt update && sudo apt upgrade -y

sudo apt install apache2 ufw openssl -y

sudo ufw allow OpenSSH

sudo ufw allow 'Apache Full'

sudo ufw --force enable

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \

-keyout /etc/ssl/private/apache-selfsigned.key \

-out /etc/ssl/certs/apache-selfsigned.crt \

-subj "/C=KE/ST=Nairobi/L=Nairobi/O=MyPortfolio/OU=ICT171/CN=<span style=">

sudo a2enmod ssl

sudo a2ensite default-ssl

sudo systemctl reload apache2

**10. GitHub Repository**

**Repository URL:**

**yuden-ky/Ict171-assignment2**

**11. Video Explainer**

**Video URL:**

**<span style=">[REPLACE\_WITH\_YOUR\_VIDEO\_LINK]</span>**

**12. Disaster Recovery Plan**

If the droplet is deleted or corrupted:

Recreate the droplet

SSH into it

Run setup\_server.sh

Re-upload your index.html from GitHub

DNS stays the same unless the IP changes

**Conclusion**

In short, this project gave me the chance to learn how to build, organize, and secure a server through the use of Infrastructure as a Service (IaaS). Thanks to DigitalOcean and Linux, I installed a working Apache web server, carried out my own DNS setup, and used a self-generated SSL certificate to secure the server’s communication (Shrestha, 2024). Apart from that, a Bash script that can be reused made server deployment more efficient and highlighted the role of automation in system administration. GitHub and a video illustration make sure students learn the main topics and experience up-to-date practices used in IT. Working on these projects has helped me learn more about server environments, better document technology, and prepare me for future parts of server and DevOps activities.

**Reference list**

J, P.N., Sameeksha Keshav, K, S.B. and Swetha S (2024). Automating Registration of Executables as Commands on the Ubuntu-Linux Platform. [online] pp.1–5. doi: <https://doi.org/10.1109/incet61516.2024.10593197>.

Shrestha, M. (2024). *Tools for an Automated and Streamlined Deployment to AWS*. [online] www.doria.fi. Available at: <https://www.doria.fi/handle/10024/189018>.