Cloud Server Project & Video Explainer
ICT171 Assignment 2 - 2025

1. Introduction:

Student Name: Martin MIGNARDOT
Student ID: 35609557
Public IP Address: 52.62.61.188
Domain Name (DNS) : http://ict171-martin.duckdns.org/

In this assignment, I set up a simple web server on a cloud platform. The goal was to host a basic website that allows users to view/add/remove fruits and vegetables to a shopping cart. The site was built using HTML, CSS, and JavaScript.

I used an AWS EC2 instance (Ubuntu) to host the web server. I installed Apache to serve the website, and I uploaded all the necessary files like index.html, style.css, script.js, and images to the correct location.

The website is now available to anyone who visits my server's public IP address. It includes an interactive shopping cart where users can click "Add" to add items, and the total updates automatically.

In this documentation, I explain step by step how I created this server, uploaded the website, and test that it worked. I also explain the JavaScript code I used to manage the shopping cart. I include screenshots and commands so that any students can follow the same process. The aim of this report is to help someone else repeat everything I did, even if they are new to cloud servers.

2. Creating and setting up the EC2 instance

To host my website, I used Amazon Web Service (AWS) to create a virtual server. This type of service is called Infrastructure as a Service (IaaS). I used an EC2 instance running Ubuntu Server 22.04

2.1 - Launching the server

1. I logged into AWS and opened the EC2 dashboard
2. Clicked Launch Instance and chose:

    a. Ubuntu Server 22.04 LTS
    b. t2.micro
    c. My key pair: ict171-key.pem
3. In Network settings, I allowed:
    a. SSH
    b. HTTP
    c. HTTPS
4. I launched the instance, and the public IP is 52.62.61.188

2.2 Connecting via SSH

I used the terminal of my kali-linux to connect :

```
ssh -i ict171-key.pem ubuntu@52.62.61.188
```

2.3 Installing Apache

```
sudo apt update
sudo apt install apache2 -y
```

2.4 Creating the website on the server

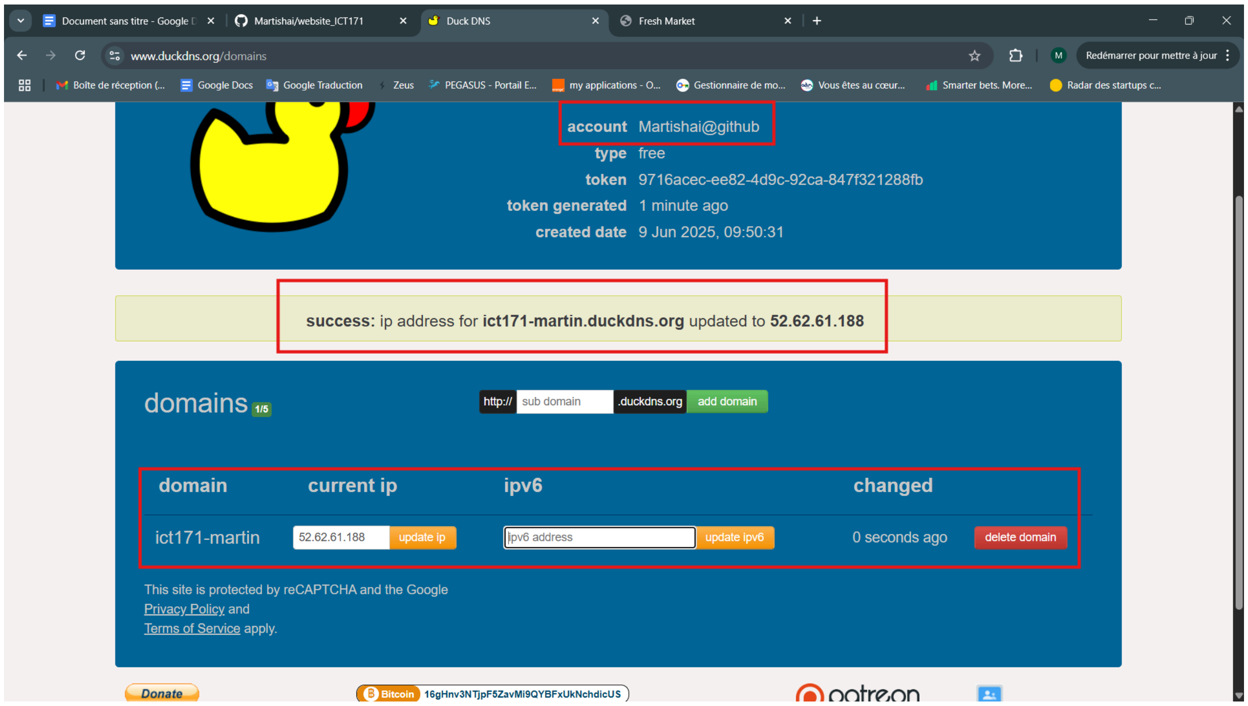
```
sudo mkdir -p /var/www/html/css
sudo mkdir -p /var/www/html/js
sudo mkdir -p /var/www/html/images
sudo touch /var/www/html/index.html
sudo touch /var/www/html/script.js
sudo touch /var/www/html/css/style.css
sudo touch /var/www/html/js/script.js
```

3. DNS configuration

To make my server easier to access and test, I set up a custom domain name using DuckDNS, which is a free dynamic DNS service.

I registered the subdomain “ict171-martin.duckdns.org” and linked it to the public IP address of my AWS EC2 instance.
This allows my website to be accessed using a readable and memorable address instead of the raw IP.

DuckDNS was simple to configure and works well for cloud projects and student assignments. Since my EC2 IP is static, I did not need to install any update scripts. I only had to manually register the IP on the DuckDNS dashboard.



account Martishai@github
type free
token 9716acec-ee82-4d9c-92ca-847f321288fb
token generated 1 minute ago
created date 9 Jun 2025, 09:50:31

success: ip address for **ict171-martin.duckdns.org** updated to **52.62.61.188**

domains 1/5

http:// sub domain .duckdns.org add domain

domain	current ip	ipv6	changed	
ict171-martin	52.62.61.188 update ip	ipv6 address update ipv6	0 seconds ago	delete domain

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

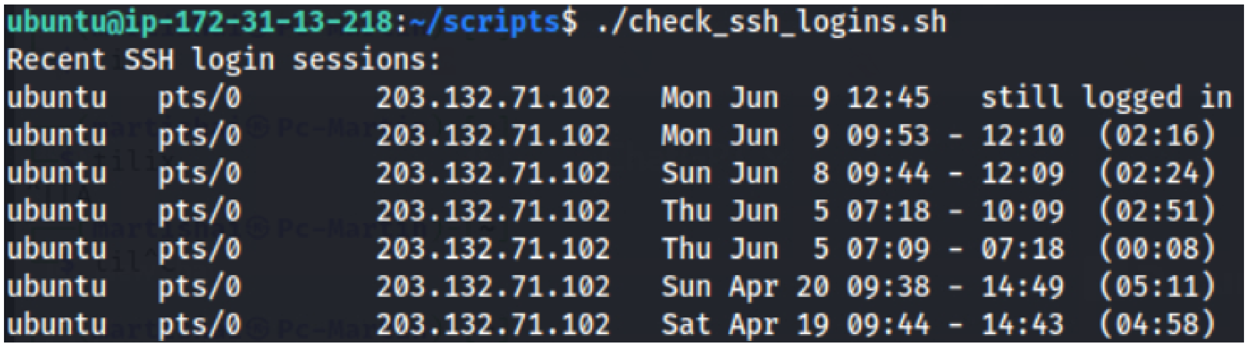
4. Script explanation

At first, I was not really sure whether the assignment required a web-based script like JavaScript or a system-level script like bash. So, to be safe I prepared both:

- A JavaScript script that controls the shopping cart on the website
- A Bash script that detects recent SSH logins to the server

4.1 Bash Script - check_ssh_logins.sh

```
#!/bin/bash
echo "Recent SSH login sessions:"
last -i | grep -E "pts|tty" | head -n 10
```



```
ubuntu@ip-172-31-13-218:~/scripts$ ./check_ssh_logins.sh
Recent SSH login sessions:
ubuntu   pts/0        203.132.71.102   Mon Jun  9 12:45   still logged in
ubuntu   pts/0        203.132.71.102   Mon Jun  9 09:53 - 12:10  (02:16)
ubuntu   pts/0        203.132.71.102   Sun Jun  8 09:44 - 12:09  (02:24)
ubuntu   pts/0        203.132.71.102   Thu Jun  5 07:18 - 10:09  (02:51)
ubuntu   pts/0        203.132.71.102   Thu Jun  5 07:09 - 07:18  (00:08)
ubuntu   pts/0        203.132.71.102   Sun Apr 20 09:38 - 14:49  (05:11)
ubuntu   pts/0        203.132.71.102   Sat Apr 19 09:44 - 14:43  (04:58)
```

This Bash script displays the last 10 login sessions to the server using SSH.
It works by checking all terminal logins (pts/tty), which typically correspond to remote SSH access.
This is useful for system monitoring and gives a basic layer of security awareness for server administrators.

The script was created directly on the EC2 server and made executable with chmod +x. It can be run anytime to check who has recently logged in.

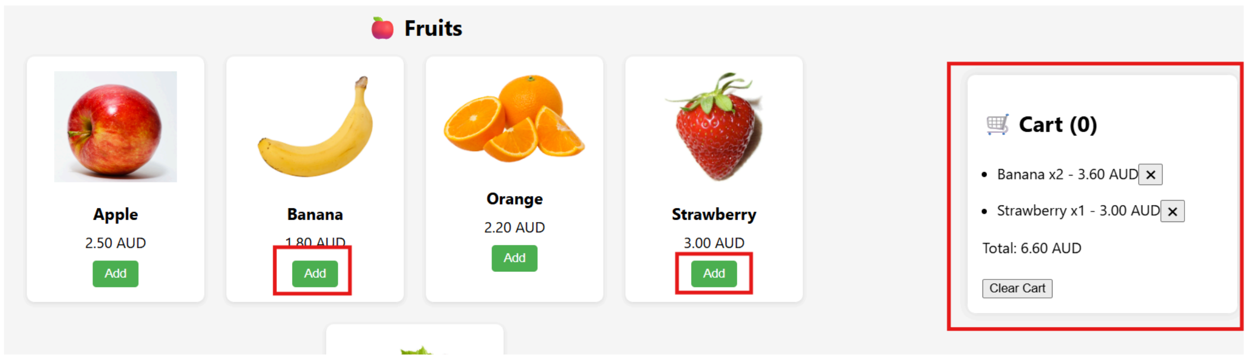
4.2 JavaScript Script - Shopping Cart

```
function addToCart(id) {
  const product = products.find(p => p.id === id);
  cart.push(product);
  updateCart();
}

function updateCart() {
  const list = document.getElementById('cart-items');
  const total = document.getElementById('cart-total');
  const count = document.getElementById('cart-count');
  list.innerHTML = '';
  let sum = 0;
  const groupedItems = {};

  cart.forEach(item => {
    if (!groupedItems[item.id]) {
      groupedItems[item.id] = { ...item, quantity: 1 };
    } else {
      groupedItems[item.id].quantity++;
    }
  });
  Object.values(groupedItems).forEach(item => {
    const li = document.createElement('li');
    li.textContent = `${item.name} x${item.quantity} - ${(item.price * item.quantity).toFixed(2)}
AUD`;
    const removeBtn = document.createElement('button');
    removeBtn.textContent = '✖';
    removeBtn.onclick = () => removeOneFromCart(item.id);
    li.appendChild(removeBtn);
    list.appendChild(li);
    sum += item.price * item.quantity;
  });

  total.textContent = sum.toFixed(2);
  count.textContent = cart.length;
}
```



- Banana x2 - 3.60 AUD ✕
- Strawberry x1 - 3.00 AUD ✕

Total: 6.60 AUD

This script manages the shopping cart logic on the website. When a user clicks “Add”, the item is stored in a cart array. The updateCart() function displays grouped items (e.g. “Banana x3”), updates the total price, and allows removing items.

This code shows how JavaScript can dynamically interact with the DOM and provide real-time interactivity without reloading the page.