**CLOUD SERVER PROJECT**

ICT171

Introduction to Server Environments and Architectures

Muhammad Naseef  
35099165

Global IP address : 13.51.219.83

Domain**:** [**https://adventureticket.click**](https://adventureticket.click)

GitHub Repository link: <https://github.com/naseefyasarpc/ICT171assignment2>

**STEP 1: SETTING UP CLOUD SERVER (IAAS)**

1. **Create an Account in AWS**

* Visit the AWS website and sign up for an account.

1. **Access EC2 Dashboard**

* In the AWS Management Console, use the search bar to type “EC2” and select the EC2 Dashboard from the results.

1. **Launch an Instance**

* Click the “Launch Instance” button to create a new virtual server.

1. **Choose an Amazon Machine Image (AMI)**

* Select “Ubuntu Server 22.04 LTS (HVM), SSD Volume Type” which is eligible for the free tier.

1. **Select Instance Type**

* Choose the instance type “t2.micro”, which provides 1 vCPU and 1 GiB RAM and is free tier eligible.

1. **Create a Key Pair**

* Create a new key pair for secure SSH access.

1. **Configure Network Settings (Security Group)**

* Create a new security group with the following rules:
* Allow SSH (Port 22) for remote access.
* Allow HTTP (Port 80) for regular web traffic.
* Allow HTTPS (Port 443) for secure connections (SSL/TLS).
* During development, set the source to Anywhere (0.0.0.0/0).

1. **Assign an Elastic IP**

* Assign and associate a public elastic IP address to your instance for external access.

1. **Launch the Instance**

* After completing the configuration, proceed to launch the instance, which will now be running and accessible.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 2: Connect to the Created EC2 Instance**

1. **Connect using PuTTY**:

* Open PuTTY and enter the public IP address of your EC2 instance in the "Host Name (or IP address)" field.
* Navigate to Connection > SSH > Auth in the left menu.
* Click on Browse, and select your converted Key Pair file.
* After this setup, click Open to initiate the connection. When prompted for a username, enter ec2-user (or ubuntu, depending on your instance).

1. **Install Apache**:

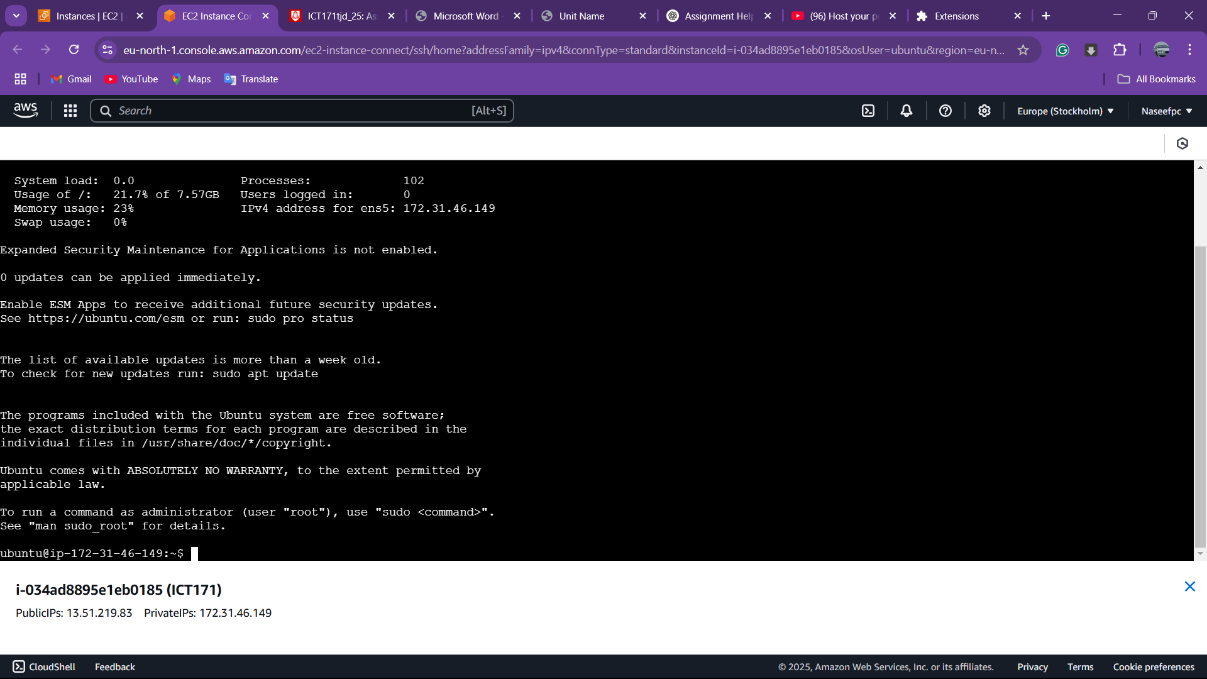
* Once connected, run the following commands to install Apache:

sudo apt update

sudo apt install apache2 -y

sudo systemctl enable apache2

sudo systemctl start apache2



**Step 3: Domain Name Registration and DNS Entry**

1. **Register Domain**:

* In **Route 53**, search for and purchase domain.

1. **Create DNS Record**:

* Navigate to **Hosted Zones** and create a new record set:
* Type: **A - IPv4 address**
* Value: Enter the public IP address of your EC2 instance.
* Click **Create** to finalize.

1. **Add CNAME for www**:

* Create a **CNAME** record:
* Name: www
* Type: **Alias** to your A record.
* Click **Create** to save.

1. **Access Your Domain**:

* Once DNS propagation is complete, access the site by entering the domain

A screenshot of a computer

AI-generated content may be incorrect.

**Step 4: Installing WordPress on EC2**

1. Connect to Your EC2 Instance.
2. Install LAMP Stack (Apache, MySQL, and PHP):

* Run the following commands:

sudo apt update

sudo apt install apache2 mysql-server php libapache2-mod-php php-mysql -y

1. Download and Install WordPress:

* Change to the web directory:

cd /var/www/html

* Download the latest WordPress package:

sudo wget <https://wordpress.org/latest.tar.gz>

* Extract the downloaded archive:

sudo tar -xvzf latest.tar.gz

* Move the WordPress files to the current directory:

sudo mv wordpress/\* .

* Remove the WordPress directory and the tar file:

sudo rm -r wordpress latest.tar.gz

1. Set File Permissions:

* Adjust the ownership and permissions with the following commands:

sudo chown -R www-data:www-data /var/www/html

sudo chmod -R 755 /var/www/html

1. **Create a MySQL Database for WordPress**:

* Log into MySQL:

sudo mysql -u root -p

* In the MySQL shell, run the following commands:

CREATE DATABASE wordpress;

CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'yourpassword';

GRANT ALL PRIVILEGES ON wordpress.\* TO 'wpuser'@'localhost';

FLUSH PRIVILEGES;

EXIT;

1. **Configure WordPress**:

* Copy the sample configuration file:

sudo cp wp-config-sample.php wp-config.php

* Open the configuration file in a text editor:

sudo nano wp-config.php

* Update the database details in the configuration file:

define('DB\_NAME', 'wordpress');

define('DB\_USER', 'wpuser');

define('DB\_PASSWORD', 'yourpassword');

define('DB\_HOST', 'localhost');

* Save and exit the editor.

1. **Restart Apache**:

* Restart the Apache server to apply the changes:

sudo systemctl restart apache2

1. **Access the WordPress Setup Wizard**:

* Open a web browser and go to your public domain
* Complete the installation by following the on-screen instructions, such as selecting your language and entering your website details  
  A screenshot of a computer

  AI-generated content may be incorrect.

**Step 5: Securing Your Website with SSL**

1. **Install Certbot**:

* Connect to your AWS EC2 instance via SSH.
* Update your package list with:

sudo apt update

1. **Install Certbot**:

* Run the following command to install Certbot:

sudo apt install certbot python3-certbot-apache -y

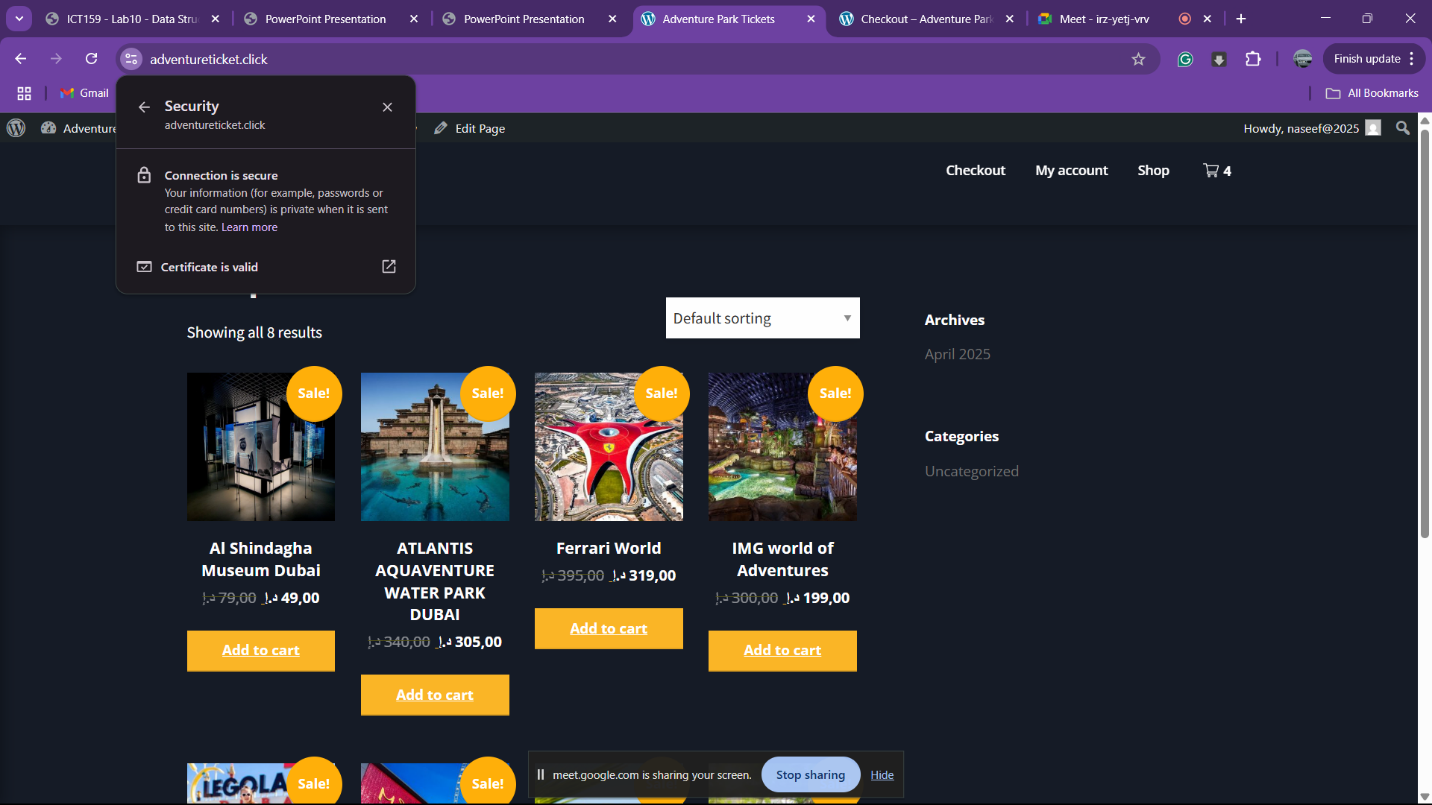
1. **Obtain an SSL Certificate**:

* Execute the Certbot command to get your SSL certificate:

sudo certbot --apache -d yourdomain.com -d www.yourdomain.com

* Follow the prompts to configure your SSL settings.

1. **Verify SSL Installation**:

* After installation, check your website using [https://yourdomain.com](https://yourdomain.com/) to confirm the SSL certificate is active.  
    
  

**Step 6: Setting Up Your Adventure Ticket Selling Website**

1. **Login to Your WordPress Admin**:

* Go to <https://yourdomain.com/wp-admin> and log in with your credentials.  
   (ex : <https://adventureticket.click/wp-admin>)

1. **Choose and Activate a Theme**:

* Navigate to **Appearance** → **Themes**.
* Select a theme suitable for ticket sales or events (e.g., a business or e-commerce theme) and click **Activate**.

1. **Install Necessary Plugins**:

* Go to **Plugins** → **Add New**.
* Install and activate:
* **WooCommerce**: For managing ticket sales.

1. **Configure WooCommerce**:

* Follow the WooCommerce setup wizard to set your currency and payment options.

1. **Create Event Categories**:

* Navigate to **Products** → **Categories**.
* Add categories like **Outdoor Adventures**, **Cultural Tours**, etc.

1. **Add Your Adventure Tickets**:

* Go to **Products** → **Add New**.
* Input details for each ticket (name, description, price, etc.), assign them to the appropriate categories, and click **Publish**.

1. **Set Your Homepage**:

* Go to **Settings** → **Reading**.
* Set your homepage to display upcoming events or ticket sales prominently.

A screenshot of a computer

AI-generated content may be incorrect.