Cloud Server Project Documentation

**Student Name:** Mahammad Naif Tonse  
**Student ID:** 35473548  
**Domain Name:** [*https://naifsballgame.online*](https://naifsballgame.online/)  
**Public IP Address:** 13.48.121.106

**Project Overview**

The aim of this project is to deploy a simple browser based ping pong game on a cloud server using Amazon AWS EC2 infrastructure. The goal was to create a website with a custom domain name and HTTPS support in a server, which can be setup, configured and secured. The game showcases real-time interactivity using JavaScript canvas drawing, simulating paddle-ball dynamics and basic AI opponent behavior.

**Server Setup**

* **EC2 Instance Configuration:**
* **Cloud Provider:** Amazon AWS
* **Instance Type:** t2.micro (Free Tier)
* **Operating System:** Ubuntu Server 22.04 LTS
* **Web Server:** Apache2
* **Steps to Launch EC2 and Configure Apache**

# Update packages

sudo apt update && sudo apt upgrade -y

# Install Apache

sudo apt install apache2 -y

# Enable Apache to start on boot

sudo systemctl enable apache2

# Start Apache

sudo systemctl start apache2

* **Uploading Game Files**

The HTML, CSS, and JavaScript files were transferred to the EC2 instance by:

scp -i "C:\Users\Maham\Downloads\PASSKNIFE.pem" ballgame.html ec2-ubuntu@13.48.121.106 :/var/www/html/

* **Domain and DNS Configuration**
* **Domain Name:**

The domain **naifsballgame.online** was purchased via Godaddy

* **4. Enabling HTTPS**

**Certbot Installation**

sudo apt install certbot python3-certbot-apache -y

**SSL Certificate Setup**

sudo certbot --apache

Certbot automatically edited the Apache configuration to redirect HTTP traffic to HTTPS and installed the certificate.

* **Firewall and Security Group Settings**
* Port 80 (HTTP) and 443 (HTTPS) were opened in the AWS EC2 Security Group.
* SSH (port 22) was restricted to my IP address for security.
* **Code Snippet and Explanation**

HTML + JavaScript Snippet:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Ping Pong Game</title>

<style>

body { text-align: center; background: #222; color: white; }

canvas { background: black; display: block; margin: auto; }

</style>

</head>

<body>

<h1>Ping Pong Game</h1>

<canvas id="gameCanvas" width="800" height="400"></canvas>

<script>

const canvas = document.getElementById("gameCanvas");

const ctx = canvas.getContext("2d");

const paddleWidth = 10, paddleHeight = 80;

let playerY = canvas.height / 2 - paddleHeight / 2;

let aiY = canvas.height / 2 - paddleHeight / 2;

let ballX = canvas.width / 2, ballY = canvas.height / 2;

let ballSpeedX = 5, ballSpeedY = 3;

let gamePaused = false;

function drawRect(x, y, width, height, color) {

ctx.fillStyle = color;

ctx.fillRect(x, y, width, height);

}

function drawCircle(x, y, radius, color) {

ctx.fillStyle = color;

ctx.beginPath();

ctx.arc(x, y, radius, 0, Math.PI \* 2);

ctx.fill();

}

function draw() {

drawRect(0, 0, canvas.width, canvas.height, "black");

drawRect(0, playerY, paddleWidth, paddleHeight, "white");

drawRect(canvas.width - paddleWidth, aiY, paddleWidth, paddleHeight, "white");

drawCircle(ballX, ballY, 8, "white");

}

function update() {

if (gamePaused) return;

ballX += ballSpeedX;

ballY += ballSpeedY;

if (ballY < 0 || ballY > canvas.height) ballSpeedY \*= -1;

if (ballX < paddleWidth && ballY > playerY && ballY < playerY + paddleHeight) {

ballSpeedX \*= -1;

}

if (ballX > canvas.width - paddleWidth && ballY > aiY && ballY < aiY + paddleHeight) {

ballSpeedX \*= -1;

}

if (ballX < 0 || ballX > canvas.width) {

gamePaused = true;

}

aiY += (ballY - (aiY + paddleHeight / 2)) \* 0.1;

}

canvas.addEventListener("mousemove", (event) => {

const rect = canvas.getBoundingClientRect();

playerY = event.clientY - rect.top - paddleHeight / 2;

});

document.addEventListener("keydown", (event) => {

if (event.code === "Space" && gamePaused) {

ballX = canvas.width / 2;

ballY = canvas.height / 2;

ballSpeedX = 5 \* (Math.random() > 0.5 ? 1 : -1);

ballSpeedY = 3 \* (Math.random() > 0.5 ? 1 : -1);

gamePaused = false;

}

});

function gameLoop() {

update();

draw();

requestAnimationFrame(gameLoop);

}

gameLoop();

</script>

</body>

</html>

**Explanation:**

This code creates a basic ping pong game in which the player controls a paddle on the left side with their mouse, and the computer controls the paddle on the right using simple AI logic. The ball bounces off the paddles and top/bottom boundaries, and the game pauses when a point is scored. Pressing the spacebar restarts the game.

**Rebuild Instructions**

If the EC2 server is deleted, follow these steps:

1. Launch a new EC2 instance (Ubuntu)
2. SSH into the instance
3. Install Apache
4. Upload the HTML file to /var/www/html/
5. Reconfigure DNS A record if IP has changed
6. Reinstall Certbot and run certbot --apache

**Additional Info**

* **GitHub Repo:** https://github.com/naiftonse/BallGame
* **Video Explainer Link:** https://imagekit.io/tools/asset-public-link?detail=%7B%22name%22%3A%22VideoExplainer\_35473548\_NaifTonse.mp4%22%2C%22type%22%3A%22video%2Fmp4%22%2C%22signedurl\_expire%22%3A%222028-04-09T20%3A49%3A06.217Z%22%2C%22signedUrl%22%3A%22https%3A%2F%2Fmedia-hosting.imagekit.io%2F455ac7ed1ca24008%2FVideoExplainer\_35473548\_NaifTonse.mp4%3FExpires%3D1838926146%26Key-Pair-Id%3DK2ZIVPTIP2VGHC%26Signature%3DFFXtRo2v8Y1yF2vRzwM~e~x4sxCigxUA9p2oKz0yB5IhX-SeZQnSWuv6qCkrQB4cY-WwQVluPynEgl1HWZD49xPENwerO~BMth1FDfQ1DaLAwW9-IsKu6ajC3jsXXwTZ0~bpza-mDbVyMN-s3zuqRLsmOxmiEVqV7G2AOlmDQ9z~gPiGjXoA1iasxETT1y49Kf41TLIcRI2allJv7p2VSG3cgB~GeMhAlN2CJdCJ5Lcf~1bNwYyMyDrR6z-nb8qcnPrt~WYyVSxWWXrxUtnlSF4WOmAYv8UcxEepte2tB6ymsaCTe8jhQQokKF8ADg7Isw8BevAlPZKVv0La36qF1g\_\_%22%7D