const hre = require("hardhat");

async function main() {

const [deployer] = await hre.ethers.getSigners();

console.log("----------------------------------------------------");

console.log("🚀 Deploying ELMCCoin with the account:", deployer.address);

console.log("----------------------------------------------------");

// --- Configuration: Load from environment variables for production ---

// Ensure these are set in your .env file

const routerAddress = process.env.PANCAKESWAP\_ROUTER\_ADDRESS; // e.g., "0x10ED43C718714eb63d5aA57B78B54704E256024E" for BSC Mainnet

const marketingWallet = process.env.MARKETING\_WALLET\_ADDRESS;

const communityWallet = process.env.COMMUNITY\_WALLET\_ADDRESS;

const teamWallet = process.env.TEAM\_WALLET\_ADDRESS;

// Basic validation for configuration

if (!routerAddress || !marketingWallet || !communityWallet || !teamWallet) {

console.error("ERROR: Please ensure PANCAKESWAP\_ROUTER\_ADDRESS, MARKETING\_WALLET\_ADDRESS, COMMUNITY\_WALLET\_ADDRESS, and TEAM\_WALLET\_ADDRESS are set in your .env file.");

process.exit(1);

)

const ELMCCoin = await hre.ethers.getContractFactory("ELMCCoin");

const elmc = await ELMCCoin.deploy(

routerAddress,

marketingWallet,

communityWallet,

teamWallet

);

// FIX: Replaced deprecated .deployed() with .waitForDeployment()

await elmc.waitForDeployment();

console.log("✅ ELMCCoin deployed to address:", elmc.target);

// Create PancakeSwap Pair

console.log("Creating PancakeSwap pair...");

const txCreatePair = await elmc.createPancakeSwapPair();

await txCreatePair.wait();

const pairAddress = await elmc.pancakeSwapPair();

console.log("✅ PancakeSwap Pair created at address:", pairAddress);

// IMPORTANT: Trading is NOT enabled here.

// For a fair launch, you should manually add liquidity and lock LP tokens FIRST.

// Then, enable trading via a separate script or Defender Action (e.g., scripts/defender/enableTrading.js).

console.log("\n----------------------------------------------------");

console.log("👉 NEXT STEPS AFTER DEPLOYMENT:");

console.log("1. Manually add initial liquidity on PancakeSwap.");

console.log("2. Lock or burn your LP tokens (use scripts/defender/burnLP.js or a locker service).");

console.log("3. Enable trading via Defender Action or scripts/defender/enableTrading.js.");

console.log("4. Transfer ownership to your Gnosis Safe (scripts/transferOwnership.js).");

console.log("----------------------------------------------------");

}

main().catch((error) => {

console.error(error);

process.exitCode = 1;

});