Deploy Uniswap-V3-Core

Step 1: Setting up the Environment:

git clone https://github.com/Uniswap/uniswap-v3-core.git

- This command clones the Uniswap V3 core repository from GitHub to your local machine.

cd uniswap-v3-core

- Change directory into the cloned repository.

npm install --save-dev hardhat

- Install Hardhat as a development dependency for the project.

npx hardhat init:

- Initialize Hardhat in your project directory.

npx hardhat compile

- Compile the contracts using Hardhat.

Step 2: Creating Deployment Script:

mkdir scripts && touch scripts/deploy.ts

- mkdir scripts && touch scripts/deploy.ts. This command creates a directory named "scripts" and a TypeScript file named "deploy.ts" inside it. This file will contain the deployment script for the Uniswap V3 contracts.

Past this code in it:

```
import { ethers } from "hardhat";
import FACTORY_JSON from
"../artifacts/contracts/UniswapV3Factory.sol/UniswapV3Factory.json";


const FACTORY_ABI = FACTORY_JSON.abi;
const FACTORY_BYTECODE = FACTORY_JSON.bytecode;

async function main() {
  const [deployer] = await ethers.getSigners();

  console.log("Deploying contracts with the account:", deployer.address);
  const UniswapV3Factory = await ethers.getContractFactory("UniswapV3Factory");
  const deployedFactory = await UniswapV3Factory.deploy();

  console.log("Uniswap V3 Factory address:", deployedFactory.address);
}

main()
  .then(() => process.exit(0))
  .catch((error) => {
    console.error(error);
    process.exit(1);
  });
```

Step 3: Make changes in hardhat.config.js file

Past this code in it:

```
import 'hardhat-typechain'
import '@nomiclabs/hardhat-ethers'
import '@nomiclabs/hardhat-waffle'
import '@nomiclabs/hardhat-etherscan'
import 'dotenv/config' //

const PRIVATE_KEY = [process.env.ACCOUNT_PRIVATE_KEY]; //
const ETHERSCAN_API_KEY = [process.env.ETHERSCAN_API_KEY];//
const OKLINK_API_KEY = [process.env.OKLINK_API_KEY];//
const ALCHEMY_API_KEY = [process.env.ALCHEMY_API_KEY];//


export default {
  networks: {
    hardhat: {
      allowUnlimitedContractSize: false,
    },
    mainnet: {
      url: `https://mainnet.infura.io/v3/${process.env.INFURA_API_KEY}`,
    },
    ropsten: {
      url: `https://ropsten.infura.io/v3/${process.env.INFURA_API_KEY}`,
    },
    rinkeby: {
      url: `https://rinkeby.infura.io/v3/${process.env.INFURA_API_KEY}`,
    },
    goerli: {
      url: `https://goerli.infura.io/v3/${process.env.INFURA_API_KEY}`,
    },
    kovan: {
      url: `https://kovan.infura.io/v3/${process.env.INFURA_API_KEY}`,
    },
    arbitrumRinkeby: {
      url: `https://arbitrum-rinkeby.infura.io/v3/${process.env.INFURA_API_KEY}`,
    },
    arbitrum: {
      url: `https://arbitrum-mainnet.infura.io/v3/${process.env.INFURA_API_KEY}`,
    },
    optimismKovan: {
      url: `https://optimism-kovan.infura.io/v3/${process.env.INFURA_API_KEY}`,
    },
    optimism: {
      url: `https://optimism-mainnet.infura.io/v3/${process.env.INFURA_API_KEY}`,
    },
    mumbai: {
      url: `https://polygon-mumbai.infura.io/v3/${process.env.INFURA_API_KEY}`,
    },
    polygon: {
      url: `https://polygon-mainnet.infura.io/v3/${process.env.INFURA_API_KEY}`,
    },
    bnb: {
      url: `https://bsc-dataseed.binance.org/`,
    },
    polygon_amoy: {
      url: `https://polygon-amoy.g.alchemy.com/v2/${ALCHEMY_API_KEY}`,
      chainId: 80002,
      accounts: PRIVATE_KEY,
    },
  },
  etherscan: {
    apiKey: {
      polygon_amoy: ETHERSCAN_API_KEY,
    },
    customChains: [
      {
        network: "polygon_amoy",
        chainId: 80002,
        urls: {
          apiURL: "https://rpc-amoy.polygon.technology/",
          // apiURL: `https://polygon-amoy.g.alchemy.com/v2/${ALCHEMY_API_KEY}`,
          browserURL: "https://polygon-mumbai.g.alchemy.com/v2/demo"
        }
      }
    ]
  },
  solidity: {
    version: '0.7.6',
    settings: {
      optimizer: {
        enabled: true,
        runs: 800,
      },
      metadata: {
        bytecodeHash: 'none',
      },
    },
  },
}
```

Step 4: Installing Required Packages

npm install ethers hardhat @nomiclabs/hardhat-waffle ethereum-waffle chai @nomiclabs/hardhat-ethers dotenv

- Install necessary packages required for deployment, including Ethereum libraries, testing tools, and dotenv for managing environment variables.

Step 5: Configure Environment Variables

Create a .env file in your project directory and add the necessary private keys and other environment variables.

Past this code in it and add Your private keys:

```
/* How to export an account's private key */
/* https://support.metamask.io/hc/en-us/articles/360015289632-How-to-export-an-
account-s-private-
key#:~:text=On%20the%20account%20page%2C%20click,click%20%E2%80%9CConfirm%E2%80%9
D%20to%20proceed. */

ACCOUNT_PRIVATE_KEY="************************....**"
POLYGONSCAN_API_KEY="************************....**"
ETHERSCAN_API_KEY="************************....**"
INFURA_API_KEY="************************....**"
ALCHEMY_API_KEY="************************....**"
```

Step 6: Deploying Contracts:

npx hardhat run --network hardhat scripts/deploy.ts

- This command deploys the Uniswap V3 contracts on the local Hardhat network.

npx hardhat run scripts/deploy.ts --network polygon_amoy:

- This command deploys the Uniswap V3 contracts on the Polygon network.

Step 7: ???:

Congratulations!!! You are done!

Deploy WETH9

Step 1: Setting up the Environment:

mkdir WETH9 && cd WETH9/

- Makes directory WETH9 and relocates to it.

npm install --save-dev hardhat

- Installing Hardhat as a development dependency for your project.

npx hardhat init

- Initializes Hardhat in your project directory.

Step 2: Creating Smart contract:

touch contracts/weth9.sol

- Creating a new Solidity file named `weth9.sol` inside the `contracts` directory.

Past this code in it:

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.4;

contract WETH9 {
    string public name     = "Wrapped Ether";
    string public symbol   = "WETH";
    uint8  public decimals = 18;

    event  Approval(address indexed src, address indexed guy, uint wad);
    event  Transfer(address indexed src, address indexed dst, uint wad);
    event  Deposit(address indexed dst, uint wad);
    event  Withdrawal(address indexed src, uint wad);

    mapping (address => uint)                       public  balanceOf;
    mapping (address => mapping (address => uint))  public  allowance;

    receive() external payable {
        deposit();
    }

    function deposit() public payable {
        balanceOf[msg.sender] += msg.value;
        emit Deposit(msg.sender, msg.value);
    }
    function withdraw(uint wad) public {
        require(balanceOf[msg.sender] >= wad);
        balanceOf[msg.sender] -= wad;
        payable(msg.sender).transfer(wad);
        emit Withdrawal(msg.sender, wad);
    }

    function totalSupply() public view returns (uint) {
        return address(this).balance;
    }

    function approve(address guy, uint wad) public returns (bool) {
        allowance[msg.sender][guy] = wad;
        emit Approval(msg.sender, guy, wad);
        return true;
    }

    function transfer(address dst, uint wad) public returns (bool) {
        return transferFrom(msg.sender, dst, wad);
    }

    function transferFrom(address src, address dst, uint wad) public returns
(bool){
        require(balanceOf[src] >= wad);

        if (src != msg.sender) {
            require(allowance[src][msg.sender] >= wad);
            allowance[src][msg.sender] -= wad;
        }

        balanceOf[src] -= wad;
        balanceOf[dst] += wad;

        emit Transfer(src, dst, wad);

        return true;
    }
}
```

Step 3: Creating Deployment script:

touch ignition/modules/weth9.js

- Creating a new JavaScript file named `weth9.js` inside the `ignition/modules` directory.

Past this code in it:

```
const { buildModule } = require("@nomicfoundation/hardhat-ignition/modules");
```

```
module.exports = buildModule("Weth9Module", (m) => {

  const weth9 = m.contract("WETH9");
  return { weth9 };
});
```

Step 4: Make changes in hardhat.config.js file

Past this code in it:

```
require("@nomicfoundation/hardhat-toolbox");
require("dotenv").config();
require("@nomicfoundation/hardhat-verify");




const PRIVATE_KEY = [process.env.ACCOUNT_PRIVATE_KEY]; //
const ETHERSCAN_API_KEY = [process.env.ETHERSCAN_API_KEY];//
const OKLINK_API_KEY = [process.env.OKLINK_API_KEY];//
const ALCHEMY_API_KEY = [process.env.ALCHEMY_API_KEY];//

/** @type import('hardhat/config').HardhatUserConfig */
module.exports = {
  solidity: "0.8.24",
  networks: {
    polygon_amoy: {
      // url: `https://polygon-amoy.infura.io/v3/${process.env.INFURA_API_KEY}`,
      url: `https://polygon-amoy.g.alchemy.com/v2/${ALCHEMY_API_KEY}`,
      // url: "https://rpc-amoy.polygon.technology/",
      chainId: 80002, // Set the chain ID for Polygon Amoy testnet
      accounts: PRIVATE_KEY,
    },
  },
  etherscan: {
    apiKey: {
      polygon_amoy: ETHERSCAN_API_KEY,
    },
    customChains: [
      {
        network: "polygon_amoy",
        chainId: 80002,
        urls: {
          apiURL: "https://rpc-amoy.polygon.technology/",
          // apiURL: `https://polygon-amoy.g.alchemy.com/v2/${ALCHEMY_API_KEY}`,
          browserURL: "https://polygon-mumbai.g.alchemy.com/v2/demo"
        }
      }
    ]
  },
  sourcify: {
    // Disabled by default
    // Doesn't need an API key
    enabled: true
  }
};
```

Step 5: Configure Environment Variables

touch .env

- Creating a new file named `.env` in your project directory.

Add the necessary private keys and other environment variables.

Past this code in it and add Your private keys:

```
/* How to export an account's private key */
/* https://support.metamask.io/hc/en-us/articles/360015289632-How-to-export-an-
account-s-private-
key#:~:text=On%20the%20account%20page%2C%20click,click%20%E2%80%9CConfirm%E2%80%9
D%20to%20proceed. */

ACCOUNT_PRIVATE_KEY="************************....**"
POLYGONSCAN_API_KEY="************************....**"
ETHERSCAN_API_KEY="************************....**"
INFURA_API_KEY="************************....**"
ALCHEMY_API_KEY="************************....**"
```

Step 6: Install npm dependencies and Deploy

npm install @nomicfoundation/hardhat-web3 @nomicfoundation/hardhat-toolbox dotenv

- Installing these packages in your project.

npx hardhat ignition deploy ignition/modules/weth9.js --network polygon_amoy

- Deploys the WETH9 module using the Ignition deployment script on the Polygon network.

Detailed gide to this step You can find in Hardhat official web-page.:

https://hardhat.org/hardhat-runner/docs/guides/project-setup

Uniswap V3 Deploy Script

Step 1: Setting up the Environment

git clone https://github.com/Uniswap/deploy-v3.git

- Cloning Uniswap deploy-v3 repo.

cd deploy-v3/

- Change directory to deploy-v3

yarn install

- Install yarn dependencies

yarn add @vercel/ncc@latest –dev

- Add the "@vercel/ncc" package as a development dependency to the project using yarn package manager.

yarn build

- Compile project source code.

Step2: Run The script to deploy contracts:

node dist/index.js\

**--private-key <YOUR_PRIVATE_KEY> **

**--json-rpc <JSON_RPC_URL> **

**--weth9-address <WETH9_CONTRACT_ADDRESS> **

**--native-currency-label <NATIVE_CURRENCY_LABEL> **

**--owner-address <OWNER_ADDRESS> **

**--state <STATE_JSON_FILE_PATH> **

**--v2-core-factory-address <V2_CORE_FACTORY_ADDRESS> **

**--gas-price <GAS_PRICE_IN_GWEI> **

--confirmations <NUMBER_OF_CONFIRMATIONS>

- in this last command please add “0x” before your private key to succeed.

Final notes:

Detailed description of last step is in this Link:`

https://github.com/Uniswap/deploy-v3?tab=readme-ov-file