**Advanced deploy scripts**

**Writing Deploy Scripts**

When we went straight to testing, we left behind a very important element: deploy scripts. Why is this important you ask? Because we need a certain degree of flexibility that we can't obtain in any other way, let's look through the two files FundMe.sol and PriceConverter.sol, we can see that both have an address (0x694AA1769357215DE4FAC081bf1f309aDC325306) hardcoded for the AggregatorV3Interface. This address is valid, it matches the AggregatorV3 on Sepolia but what if we want to test on Anvil? What if we deploy on mainnet or Arbitrum? What then?

The deploy script is the key to overcome this problem!

Create a new file called DeployFundMe.s.sol in script folder. Please use the .s.sol naming convention.

We start with stating the SPDX and pragma:

//SPDX-License-Identifier: MIT

pragma solidity 0.8.18;

After that, we need the imports. We are working on a Foundry Script, thus the next logical step is to import Script.sol

import {Script} from "forge-std/Script.sol";

Another thing that we need for our deploy script to work is (drumroll) to import the contract we want to deploy.

import {FundMe} from "../src/FundMe.sol";

We are ready to define the contract. Remember how we did scripts a couple of lessons ago? Try to do it yourself.

// SPDX-License\_identifier: MIT

pragma solidity ^0.8.18;

import {Script} from "forge-std/Script.sol";

import {FundMe} from "../src/FundMe.sol";

contract DeployFundMe is Script {

function run() external{

vm.startBroadcast();

new FundMe();

vm.stopBroadcast();

}

}

Now let's try it with forge script DeployFundMe.

Everything was ok! Congrats!