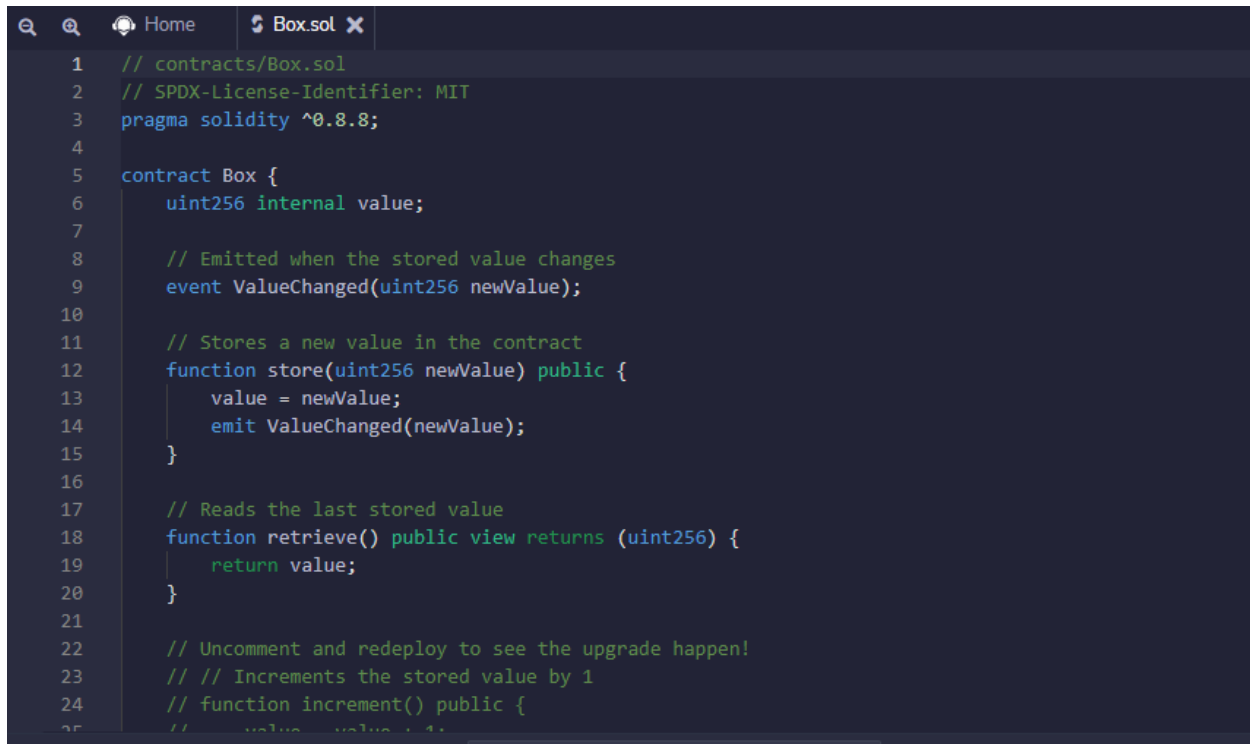


Hardhat Upgrades



```
1 // contracts/Box.sol
2 // SPDX-License-Identifier: MIT
3 pragma solidity ^0.8.8;
4
5 contract Box {
6     uint256 internal value;
7
8     // Emitted when the stored value changes
9     event ValueChanged(uint256 newValue);
10
11     // Stores a new value in the contract
12     function store(uint256 newValue) public {
13         value = newValue;
14         emit ValueChanged(newValue);
15     }
16
17     // Reads the last stored value
18     function retrieve() public view returns (uint256) {
19         return value;
20     }
21
22     // Uncomment and redeploy to see the upgrade happen!
23     // // Increments the stored value by 1
24     // function increment() public {
25     //     value = value + 1;
26 }
```

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.8;
4
5 import "@openzeppelin/contracts/proxy/transparent/ProxyAdmin.sol";
6
7 contract BoxProxyAdmin is ProxyAdmin {
8     constructor(
9         address /* owner */
10     ) ProxyAdmin() {
11         // We just need this for our hardhat tooling right now
12     }
13 }
14
```

```
1 // contracts/BoxV2.sol
2 // SPDX-License-Identifier: MIT
3 pragma solidity ^0.8.8;
4
5 contract BoxV2 {
6     uint256 internal value;
7
8     // Emitted when the stored value changes
9     event ValueChanged(uint256 newValue);
10
11     // Stores a new value in the contract
12     function store(uint256 newValue) public {
13         value = newValue;
14         emit ValueChanged(newValue);
15     }
16
17     // Reads the last stored value
18     function retrieve() public view returns (uint256) {
19         return value;
20     }
21
22     // Increments the stored value by 1
23     function increment() public {
24         value = value + 1;
25         emit ValueChanged(value);
26     }
27 }
```