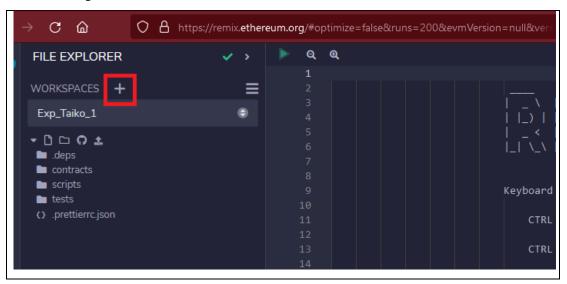
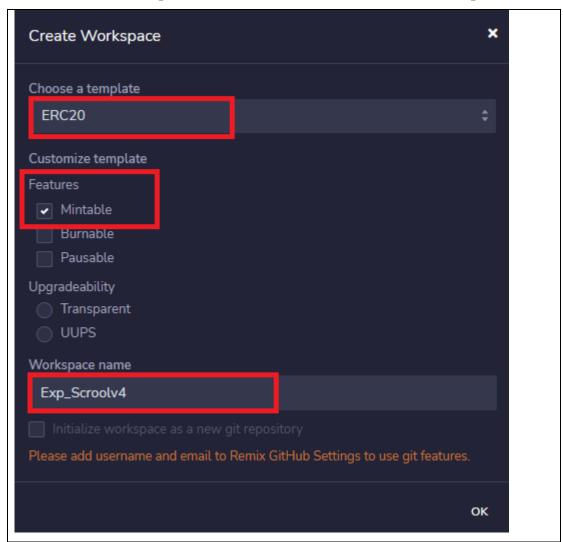
Create and Mint Token on Scroll L2 Network

1. Add Workspaces.



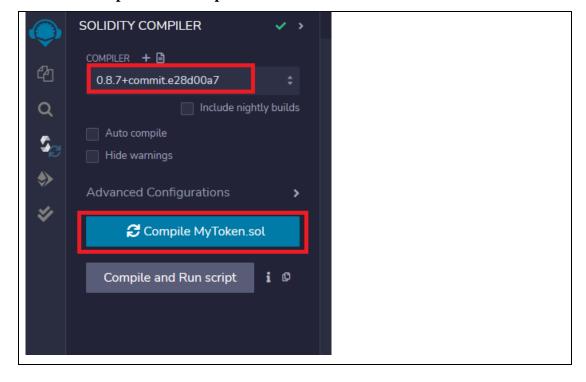
2. Choose ERC20 as template, check Mintable, and rename the Workspace name.



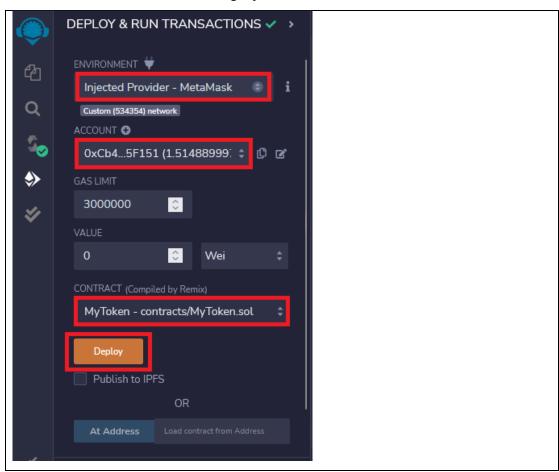
3. Rename the token name and token symbol on MyToken.sol file and MyToken_test.sol.

```
FILE EXPLORER
                                                                                         $ MyToken.sol ★ $ MyToken_test.sol
                                                                 import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
- D 🗅 O 🕹
                                                                  contract MyToken is ERC20, Ownable {
 remix-tests
                                                                      constructor() ERC20("ScrollXHPv4", "ScXHPv4") {}
                                                                     function mint(address to, uint256 amount) public onlyOwner {
    _mint(to, amount);
 rs deploy_with_web3.ts
rs ethers-lib.ts
                                          ✓ → Q Q ♠ Home $ MyToken.sol $ MyToken_test.sol X
FILE EXPLORER
 Exp_Scroolv4
                                             •
- D 🗅 O 🕹
 .deps remix-tests
 S MyToken.sol➡ scripts
  τs deploy_with_web3.ts
τs ethers-lib.ts
                                                                    function testTokenNameAndSymbol () public {
    Assert.equal(s.name(), "ScrollXHPv4", "token name did not match");
    Assert.equal(s.symbol( , "SCXHPv4", token symbol did not match");
  rs web3-lib.ts
  MyToken_test.sol
```

4. Select the compiler and Compile the contract.



5. Select Injected Provider as the Environment and make sure the account and contract are correct before we deploy it.



6. Confirm the transaction on Metamask

