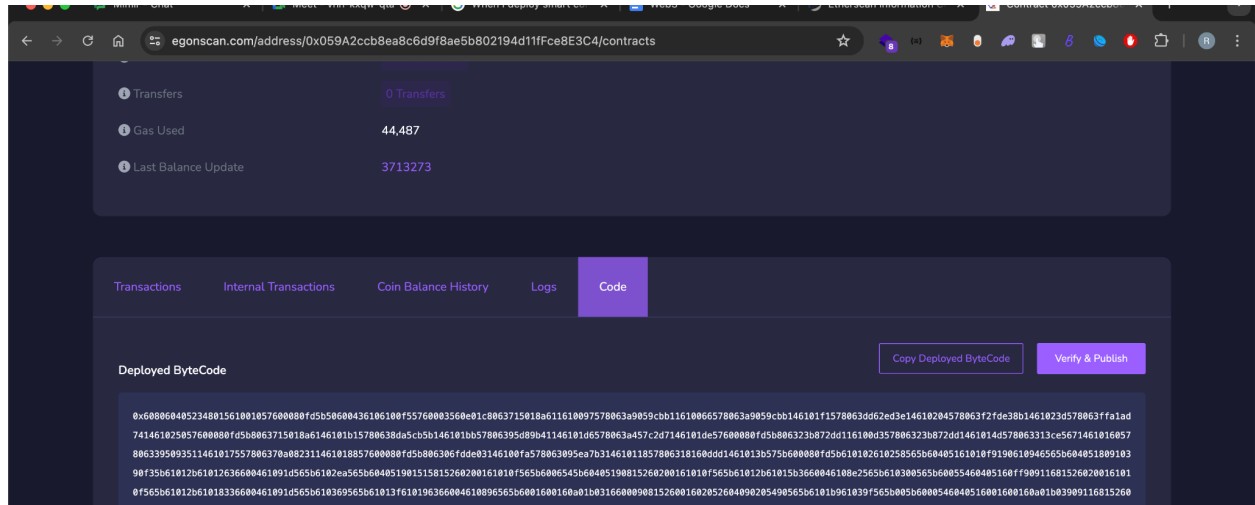


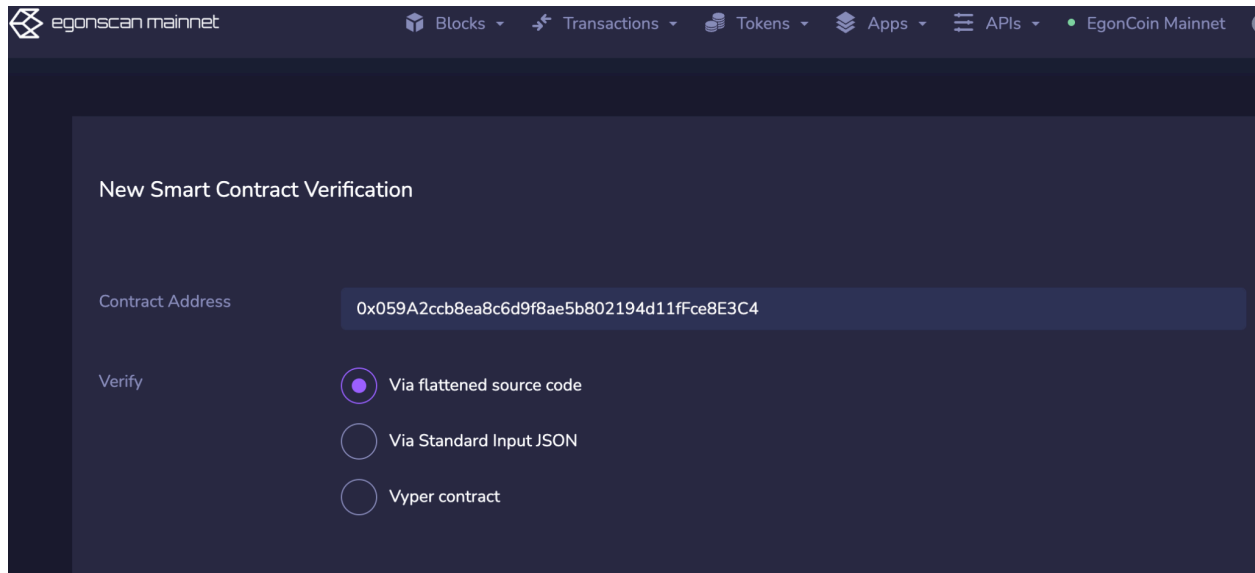
Verifying Contracts

1. Verify and Publish

Under the contract address page, at the bottom, you will be able to find the **“Code”** tab. Then click on **“Verify And Publish.”**



2. Select first option “Via source code”



3. Enter the contract name of the smart contract here, the smart contract name is "**Standard Token**" and compiler version "**v0.8.4+commit.c7e474f2**", add All solidity source code in "**Enter the solidity source code**" section and click on the "**Verify Publish**" button.

```
contract StandardToken is IERC20, Ownable, BaseToken {
    using SafeMath for uint256;

    uint256 public constant VERSION = 1;

    mapping(address => uint256) private _balances;
    mapping(address => mapping(address => uint256)) private _allowances;

    string private _name;
    string private _symbol;
    uint8 private _decimals;
    uint256 private _totalSupply;

    constructor(
        infinite gas 544200 gas
        string memory name_,
```

4. We can take source code from "<https://github.com/pinkmoonfinance/pinksale-contracts>" and copy and paste "Enter the solidity source code" section

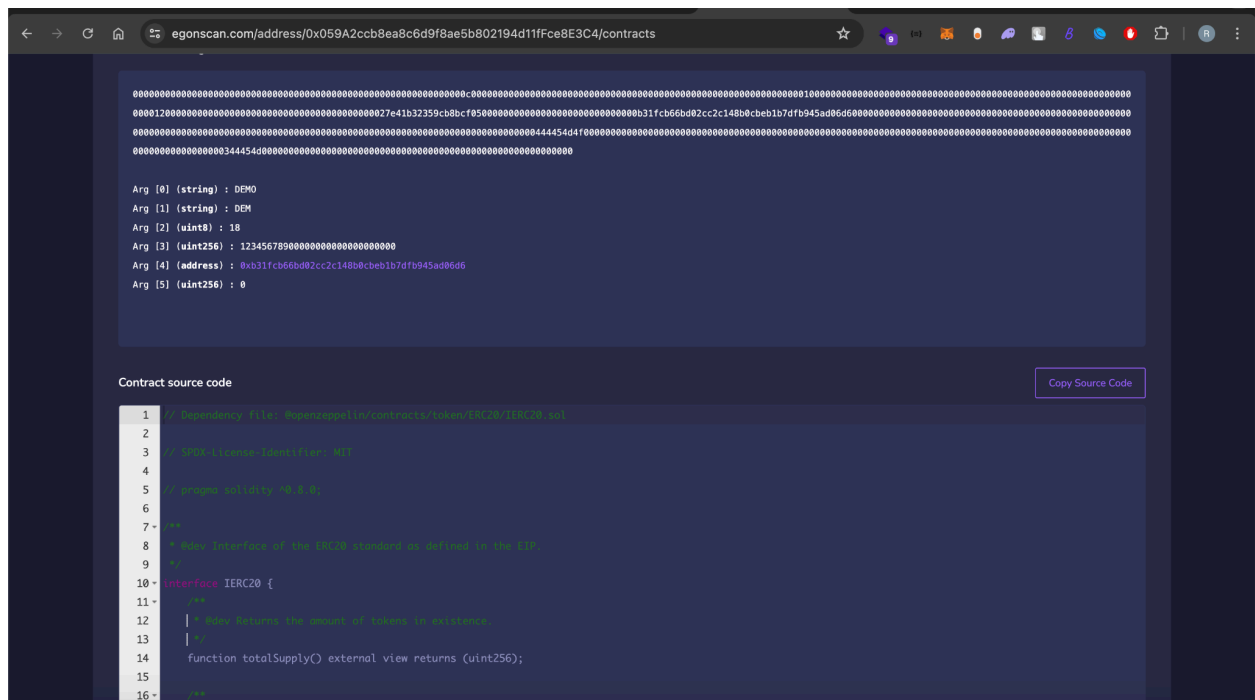
egonscan.com/address/0x059A2ccb8ea8c6d9f8ae5b802194d11fFce8E3C4/verify-via-flattened-code/new

New Solidity Smart Contract Verification

Contract Address	0x059A2ccb8ea8c6d9f8ae5b802194d11fFce8E3C4	The 0x address supplied on contract creation.
Contract Name	DEMO	Must match the name specified in the code. For example, in contract MyContract { } MyContract is the contract name.
Include nightly builds	<input checked="" type="radio"/> No <input type="radio"/> Yes	Select yes if you want to show nightly builds.
Compiler	v0.8.4+commit.c7e474f2	The compiler version is specified in pragma solidity XXX. Use the compiler version rather than the nightly build. If using the Solidity compiler, run solc --version to check.
EVM Version	default	The EVM version the contract is written for. If the bytecode does not match the version, we try to verify using the latest EVM version. EVM version details
Optimization	<input type="radio"/> No <input checked="" type="radio"/> Yes	If you enabled optimization during compilation, select yes.
Optimization runs	200	
Enter the Solidity Contract Code	<pre>/* cannot be the zero address. */ function _mint(address account, uint256 amount) internal virtual { require(account != address(0), "ERC20: mint to the zero address");</pre>	

We recommend using flattened code. This is necessary if your code utilizes a library or inherits dependencies. Use the [POA solidity flattener](#) or the [truffle flattener](#).

5. Once you verify it will take a few minutes to verify and once it is done your contract will be verified and you can see on the **contract address page**.



Here, you can check verify contract:

<https://egonscan.com/address/0x059A2ccb8ea8c6d9f8ae5b802194d11fFce8E3C4/contracts>