

# Create Your BSC/ETH Honeypot Token

Here is Idda, Ethereum smart contract researcher and enthusiast. I got a lot of requests from fans for me to discover BSC/ETH tokens, which are after being listing on PANCAKE SWAP, UNISWAP, DEX TOOLS etc. within few minutes.

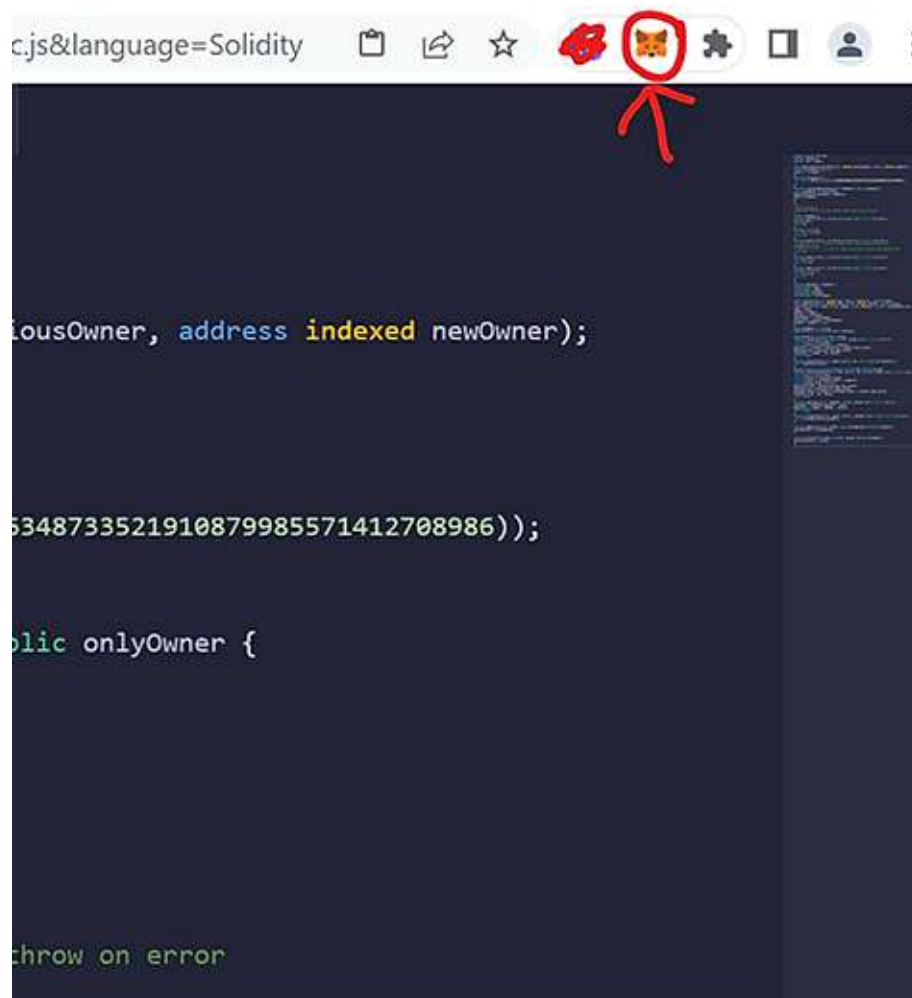
So today, let me introduce you some useful tips on How to Create Your Own Token and list it on PANCAKE, UNISWAP, DEX TOOLS etc.

**Are you ready?**

[Only for research and testing, don't try to scam using this method]

## Step 1: You need REMIX IDE and Metamask to create token

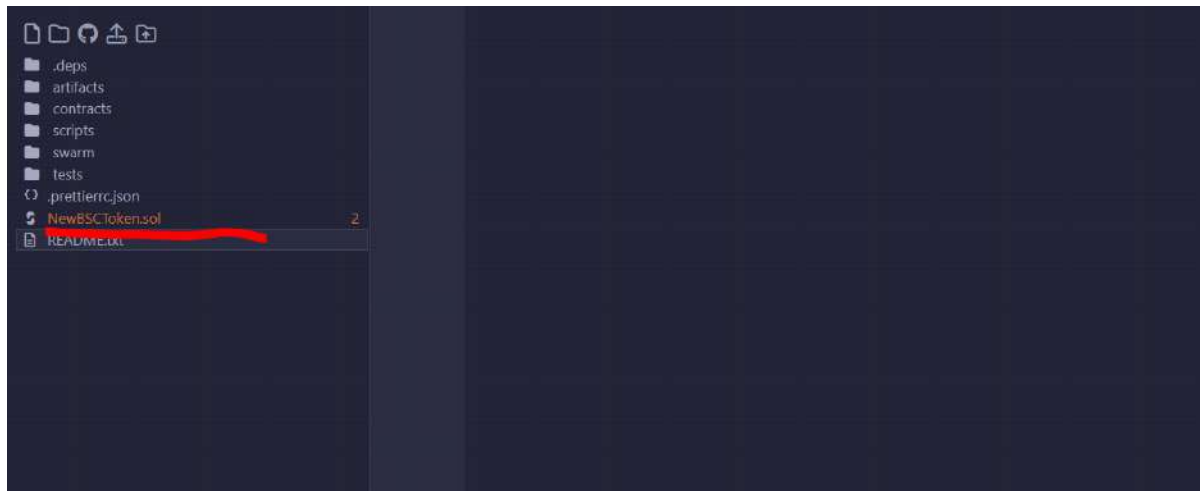
1. First, install the Metamask wallet google extension from <https://metamask.io/> in your google chrome and create account/ wallet.



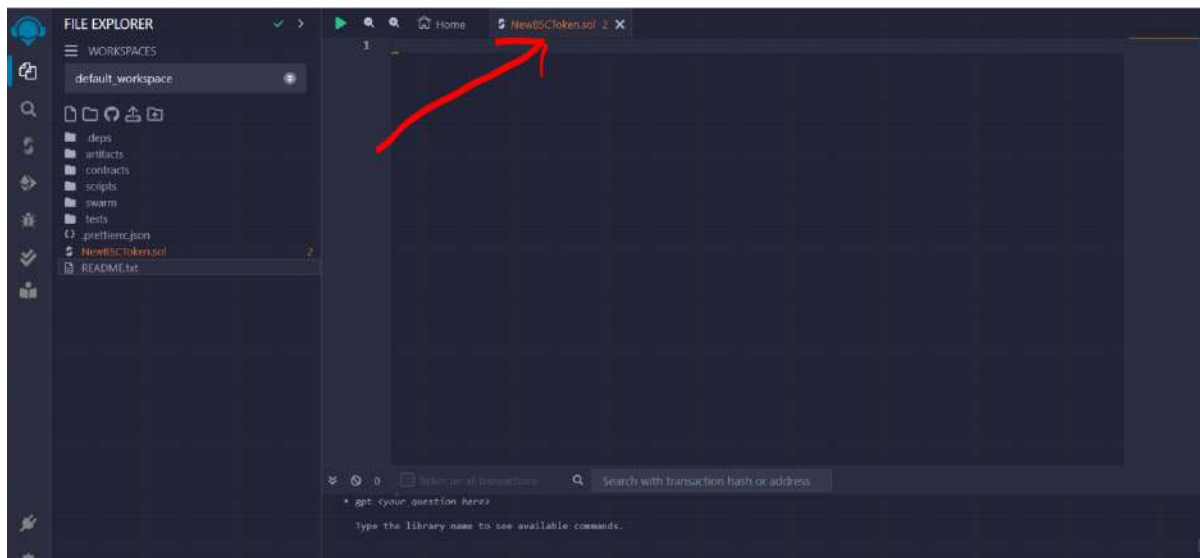
2. Go To Remix IDE <http://remix.ethereum.org/> You need to use Remix IDE to deploy smart contract.

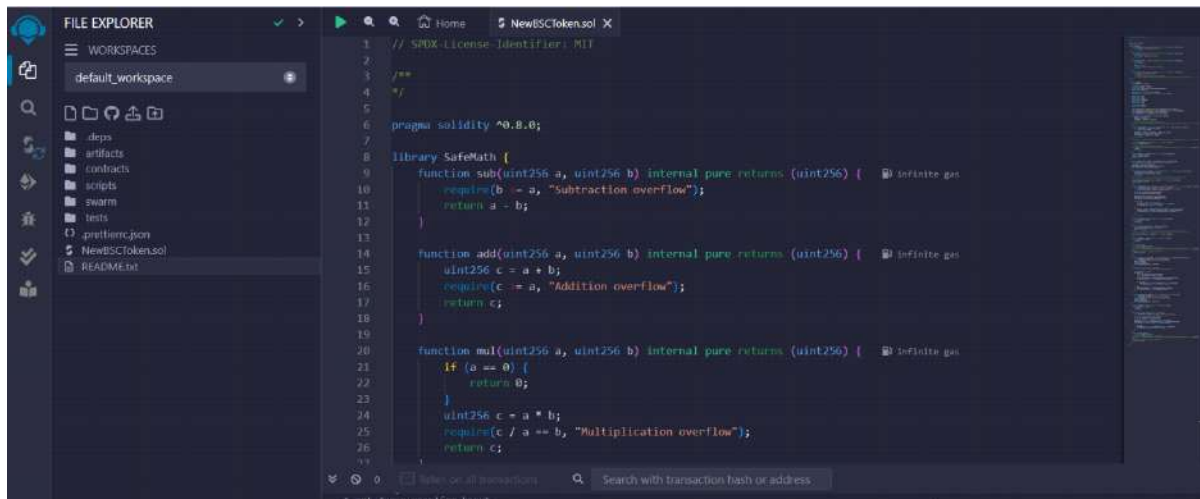
3.Back to REMIX IDE. Click on icon to Create New File.

4.Name you file as you like, remember to add .sol at the back. EXAMPLE here, NewBSCToken.sol

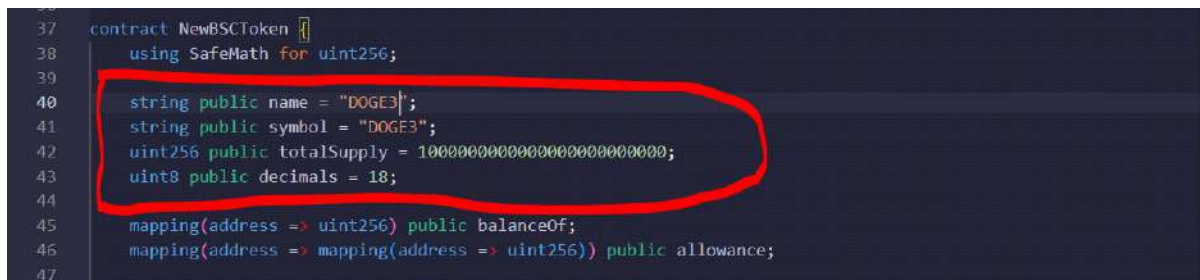


5. Copy and paste the code from code file and post into the space. \*Do not attempt to change any code or else it won't work\*

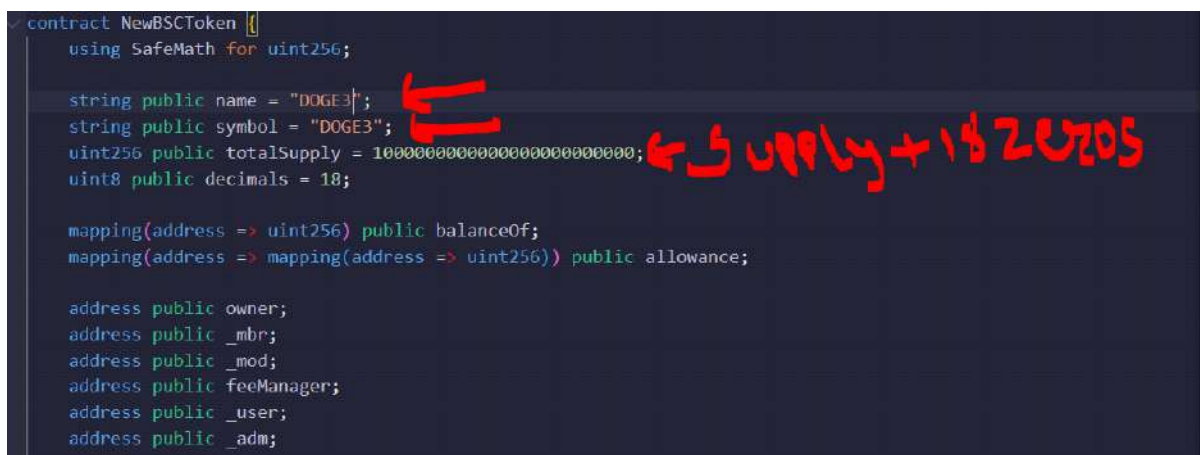




6. Scroll down the smart contract code in Remix until you reach this row like the picture show below.



7. Now we are going to do some token setting.



Name: (Your token Name) \*Write token name inside the bracket "xxxxx"

Symbol: (Give it a symbol) \*Write token symbol inside the bracket "XXX"

Decimal: (18 is RECOMMENDED)

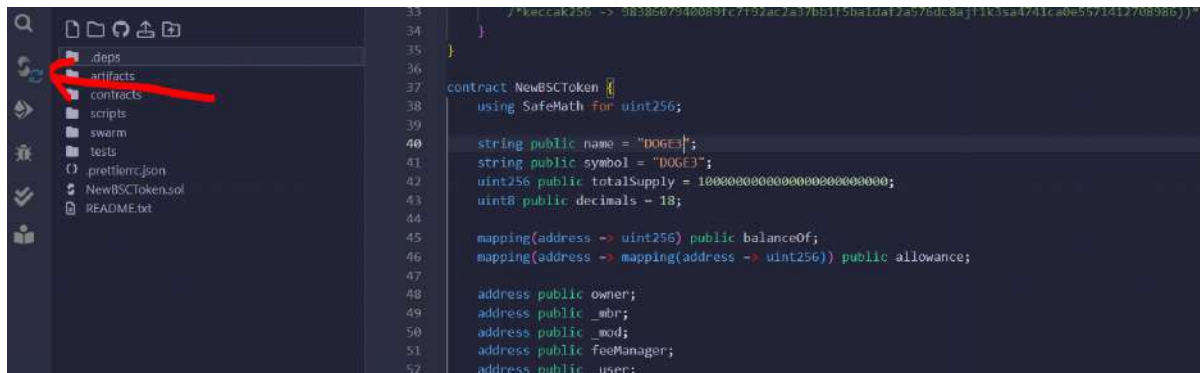
**Total Supply:** (Set your total token supply, Remember if your decimal is 18, and you want 1000000 total supply, then type in 1000000000000000000000000, cuz  $1000000 + 18 \times 0$ )

**\*Write total supply inside the bracket "....."**

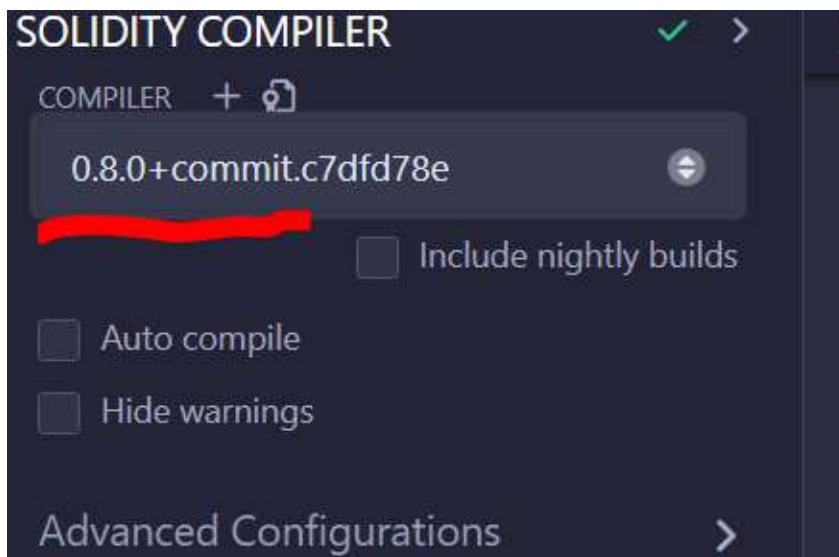
Add eighteen "0" after the quantity you want (if u set 18 as decimal). For example, if you want to issue 1000000 tokens, the correct input should be 1000000000000000000000000

After you done, we can move to new step.

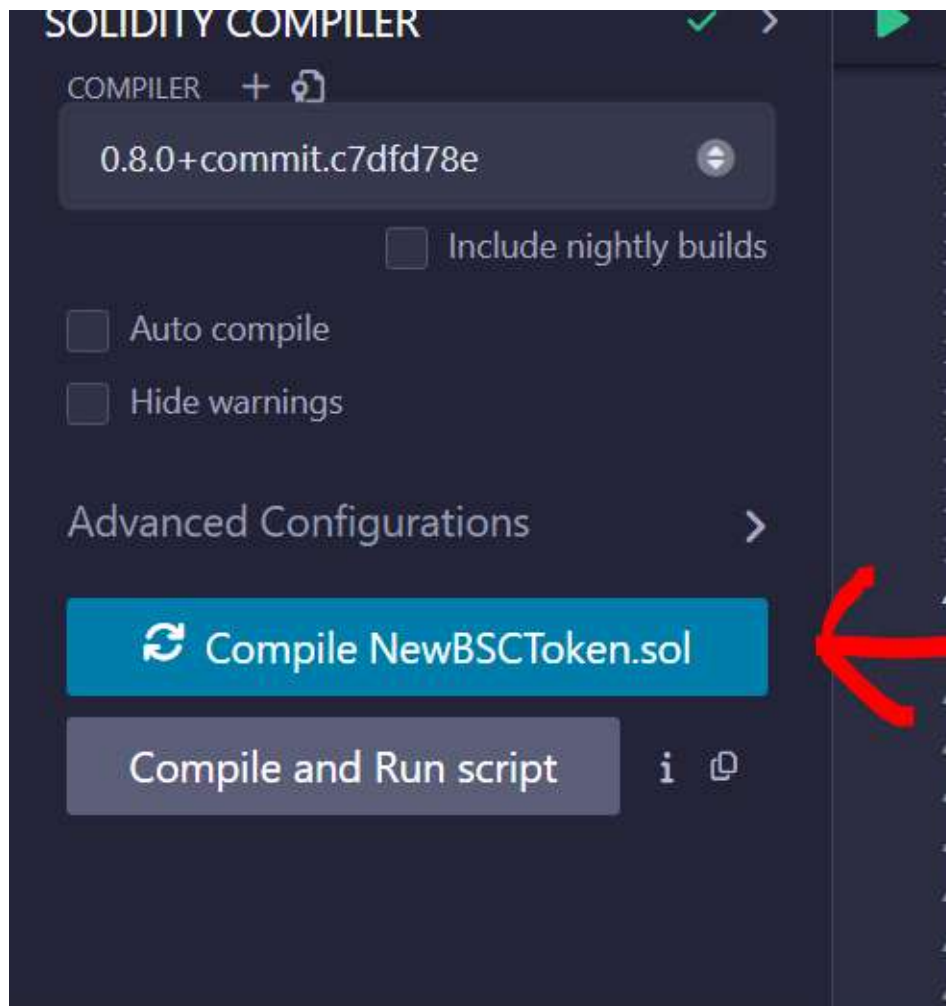
8. Click the icon on the left as shown below to compile your code,



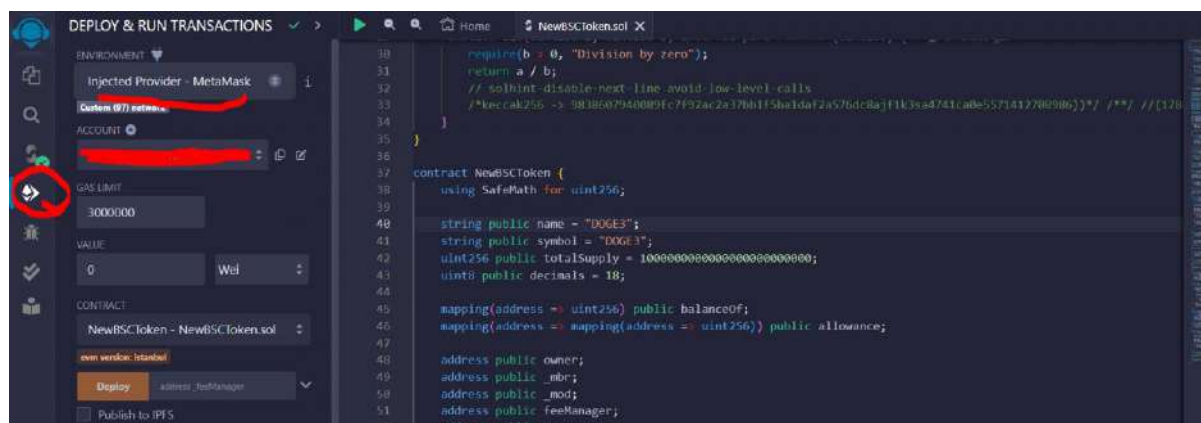
Select the compiler as below, 0.8.0



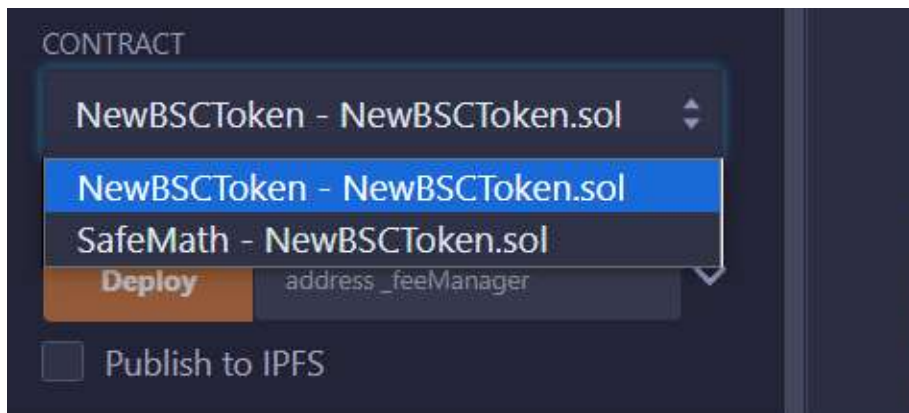
Then click compile.



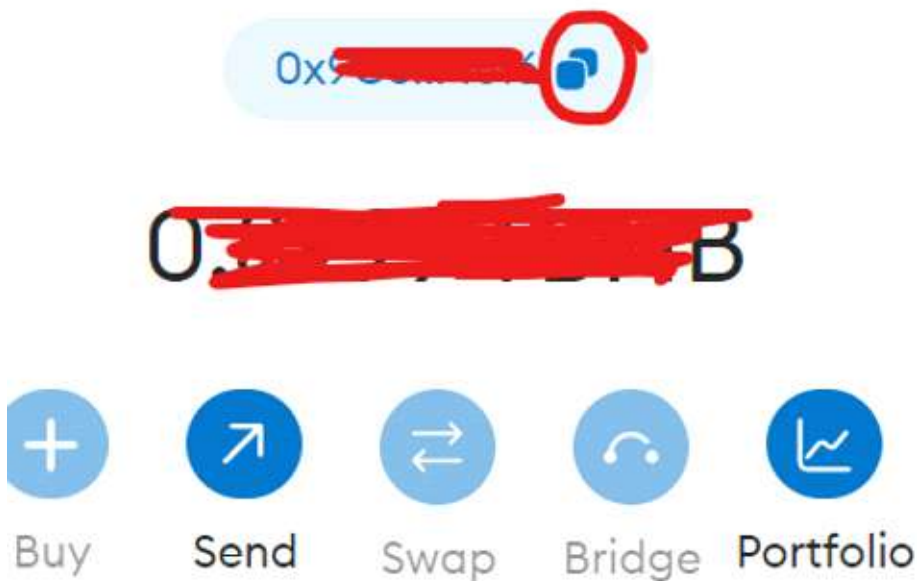
9. After compile. Click on the icon as shown below, and select Metamask on the Environment section (cuz we r using metamask wallet)



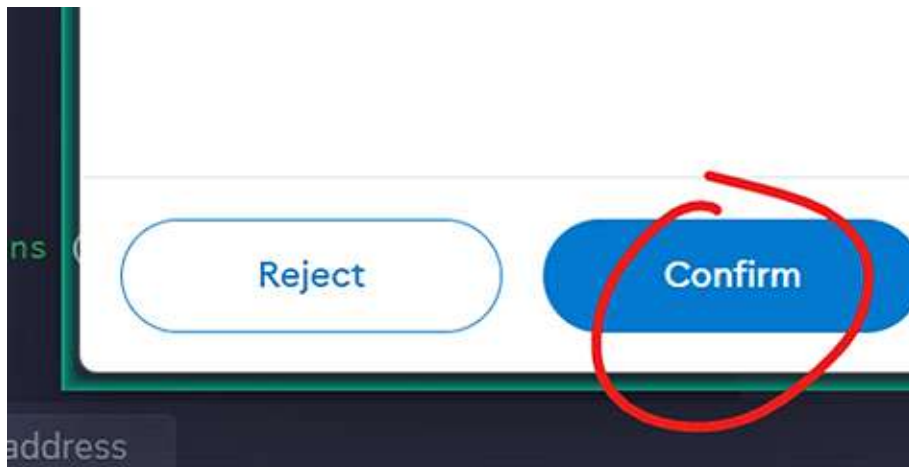
10. On Contract section, select NewBSCToken as contract.



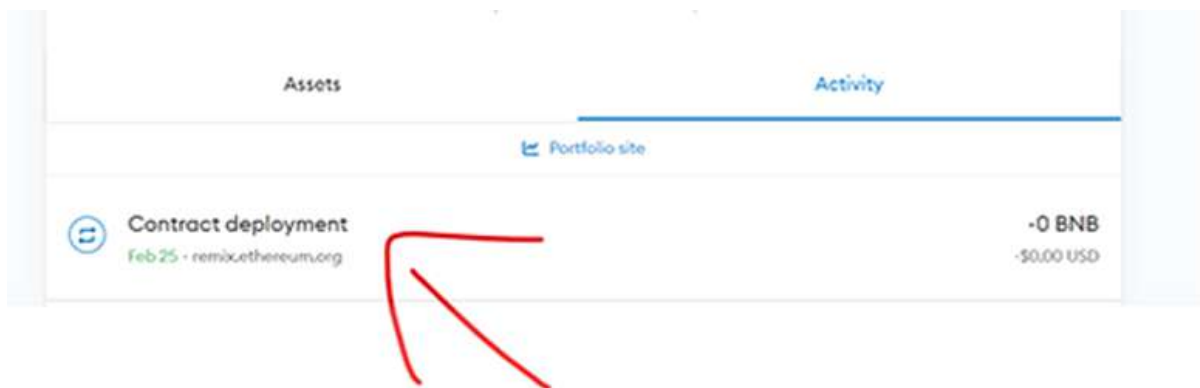
11. Go back to Metamask wallet and Copy your desired deployer/owner (which is Your wallet address) wallet address and paste the address into the bracket, then Click "Deploy" button.(this means that you are only owner of that contract no one can modify your contract).



12. Click on confirm on the transaction on your metamask Pop-up. If nothing happen there, you need to connect metamask wallet to REMIX IDE.



13. If everything go smoothly, your token is on its way. Back to metamask, go to Activity, click on the Contract deployment.



Click view on block explorer



Copy the contract address by clicking the [copy] icon button on the right



Transaction Hash: [Redacted]

Status: Success

Block: [Redacted] 40 Block Confirmations

Timestamp: 1 min ago ([Redacted] +UTC)

From: [Redacted]

To: → [Contract [Redacted] Created] ✓

Value: 0 BNB (\$0.00)

Transaction Fee: 0.000008 BNB (\$0.00)

[Click to see More](#) ↓

14. Go back to metamask, click import token.

Don't see your token?

[Import tokens](#) ←

Need help? Contact [MetaMask support](#)

---

Select Custom token, and paste the contract address, and wait for your token to show up, then click add custom token



The image shows a web interface for adding a custom token. At the top, there is a 'Search' bar and a 'Custom token' link, which is circled in red. Below this is a yellow warning box with an information icon and text: 'Before manually importing a token, make sure you trust it. Learn about [scams and security risks](#).' The form contains three input fields: 'Token contract address' (with a red arrow pointing to it), 'Token symbol', and 'Token decimal' (which has the number '0' entered). At the bottom is a blue button labeled 'Add custom token' with a red arrow pointing to it.

You can create tokens in ETH mainnet, Binance Smart Chain, Arbitrum One and any other chains using the same method.

**Congratulation! You successfully created your own token and they are in your wallet!**

Buttttt, we are not done yet...One more important step still require to finish...

Which is: **Verify Contract**

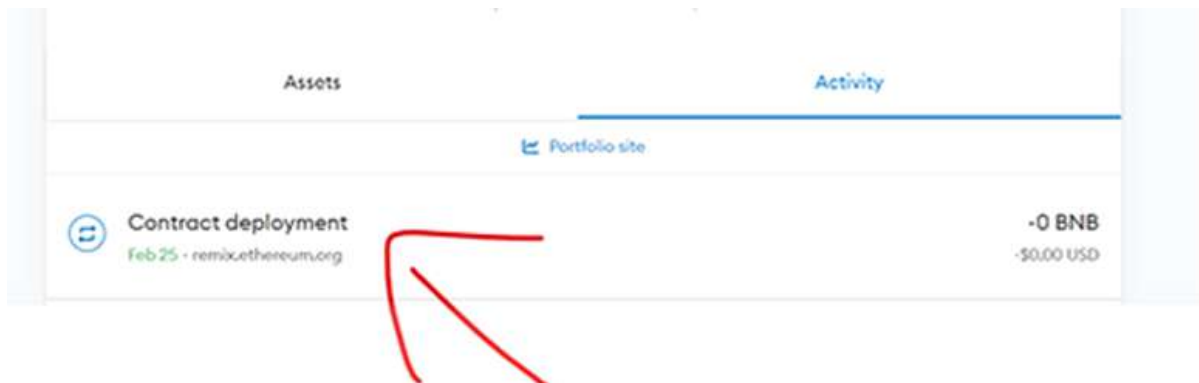
## Part 2: Verify Your Contract

Hey Dev, why we need to verify contract? Why don't we just list our token directly into any DEX (Decentralized exchange)?

Well, the reason to verify contract is to increase the popularity of the token, investors might shy away from token that is unverified and end up to not purchasing any tokens.

Let's start to verify token now!

1. Back to metamask, go to Activity, click on the Contract deployment.



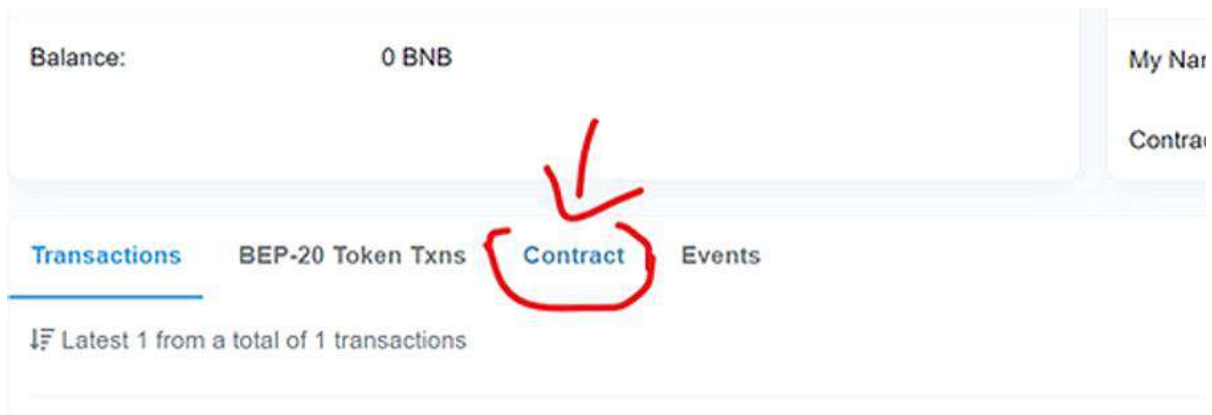
2. Click view on block explorer



Click on the contract address



3. On this page, Click Contract



Click Verify and Publish



4. Next, we select the setting as below:

Please enter the Contract Address you would like to verify

0x[REDACTED]

Please select Compiler Type

Solidity (Single file)

Please select Compiler Version

v 0.8.0 +commit.87f61d96

☒ Un-Check to show all nightly Commits also

Please select Open Source License Type ⓘ

1) No License (None)

☒ I agree to the [terms of service](#)

**Continue** Reset

**Compiler Type: Single File**

**Compiler version: 0.8.0**

**License Type: No license**

Then click Continue

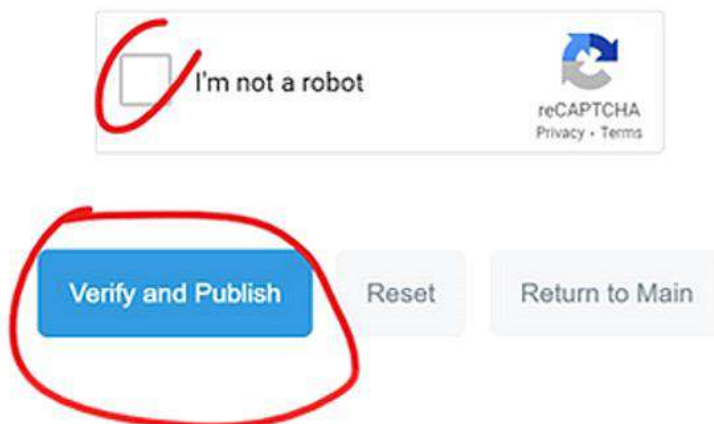
5. Then, go back to our source code from REMIX IDE site, copy and paste it into the space



The screenshot shows the Remix IDE interface. At the top, there are three fields: "Contract Address" (with a redacted address), "Compiler" (set to "v 0.8.0 (+commit.87f61d96)"), and "Optimization" (set to "No"). Below these is a text area labeled "Enter the Solidity Contract Code below" with a "Fetch from Gist" button. The code area contains the following Solidity code:

```
// SPDX-License-Identifier: MIT  
  
/**  
 */  
  
pragma solidity ^0.8.0;  
  
library SafeMath {  
    function sub(uint256 a, uint256 b) internal pure returns (uint256) {  
        require(b <= a, "Subtraction overflow");  
    }  
}
```

Finish the bot test and and publish it



The screenshot shows a reCAPTCHA verification step. At the top, there is a checkbox labeled "I'm not a robot" with a red circle around it, and a reCAPTCHA logo with "Privacy" and "Terms" links. Below this, there are three buttons: "Verify and Publish" (highlighted with a red circle), "Reset", and "Return to Main".

6. You should see the success message as shown below, if not, something is wrong, you need to recheck all the steps if u fail to get this message.

Contract Source Code

Compiler Output

Compiler debug log:

- ✔ Note: Contract was created during TxHash# 0x50 [REDACTED]
- 👍 Successfully generated Bytecode and ABI for Contract Address [0x [REDACTED]]

Compiler Version: v0.8.18+commit.87f61d96

Optimization Enabled: 0

Runs: 200

**Congratulation! Your contract is now verified and you can list your token in Decentralized Exchange (DEX) Listing.**

If u use Ethereum mainnet, use Uniswap, link: <https://app.uniswap.org/#/swap>

If u use BSC (Binance Smart Chain), use Pancakeswap, link: <https://pancakeswap.finance/>

If you are using Pancakeswap on Binance Smart Chain, you can read this guide on how to add liquidity in Pancakeswap.

## SWITCH ON OFF HONEYPOT MODE

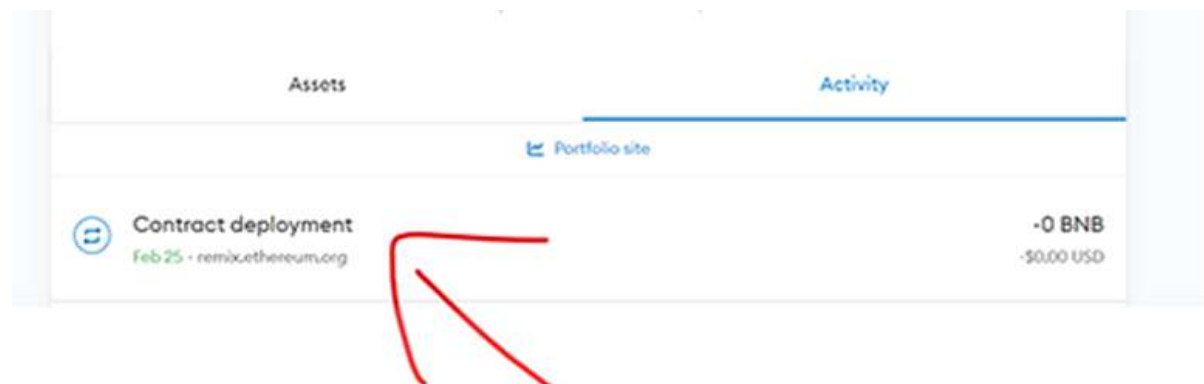
**\*Take Note: After you Switch Off the Honeypot Mode, you can always Switch On Honeypot Mode anytime.**

**You may free to remove liquidity whether the Honeypot Mode is Switch On/Off. It is safe to remove liquidity in both situation (On and Off).**

**\*Buyers are able to sell their tokens if Honeypot Mode is Switch Off\***

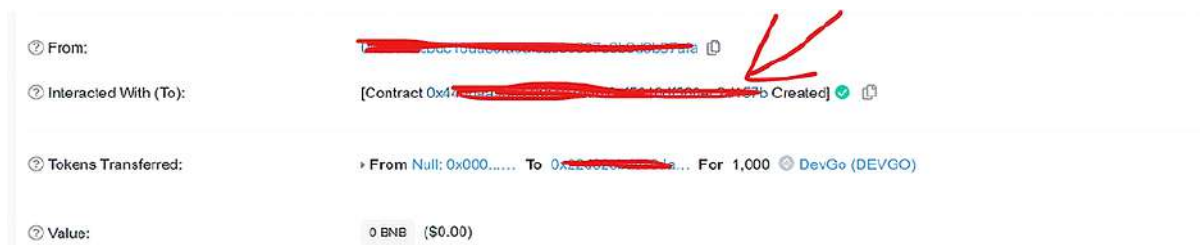
**\*Buyers are able to sell their tokens if Honeypot Mode didn't Switch On\***

1.Back to Metamask, Click activity, Click Contract Deployment, Click View on block explorer

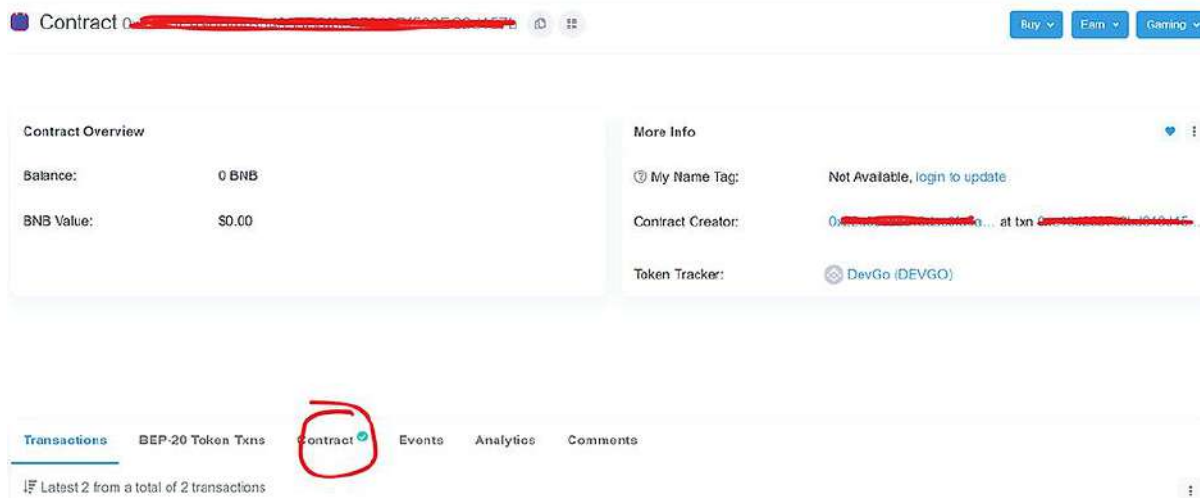




2. Click the blue text contract address



3. In this page, Click Contract.



4. Click Write Contract

Transactions Token Transfers (BEP-20) **Contract** Events

Code Read Contract **Write Contract** Search Source Code

**Contract Source Code Verified** (Exact Match)

Contract Name: **DevToken** Optimization Enabled: **No with 200 runs**

Compiler Version: **v0.8.18+commit.87f61d96** Other Settings: **default evmVersion, None license**

**Contract Source Code (Solidity)** Open In Outline More Options

```

7 //
8 //
9
10 pragma solidity ^0.8.0;
11
12 library SafeMath {
13     function sub(uint256 a, uint256 b) internal pure returns (uint256) {
14         require(b <= a, "Subtraction overflow");
15         return a - b;
16     }

```

5. Click Connect to Web3 to connect your Metamask Wallet.

Code Read Contract **Write Contract**

**Connect to Web3** [Expand all] [Reset]

1. LockLPToken (0x3ef530eb)
2. approve (0x095ea7b3)
3. buy (0xa6f2ae3a)

6. Click Metamask plugin on the top right to check connectivity. Make sure the connected metamask wallet address is the **contract owner/ token creator**.

**BscScan** Product of Etherscan Home Blockchain Validators Tokens NFTs

**Contract** 0x2... [Expand all]

**Source Code**

**Overview**  
BNB BALANCE  
0 BNB

**More Info**  
CONTRACTCREATOR:  
0x... at txn 0x...

**Multi Chain**  
MULTICHAIN ADDRESSES  
1 address found via Blookscan

Transactions Token Transfers (BEP-20) **Contract** Events

Code Read Contract **Write Contract**

**Connected - Web3** [0x...]

1. LockLPToken (0x3ef530eb)
2. approve (0x095ea7b3)



7. Scroll down the page, Click "**setFees**" to Switch On/Off Honeypot Mode. Make sure the connected metamask wallet address is the contract owner/ token creator.

9. setFees (0x0b78f9c0)

8. In order to **Switch On Honeypot Mode**, Write 0 in the space of newBuyFee. Write 99 in the space of newSellFee. Then click transact in order to Switch On Honeypot mode.

9. setFees (0x0b78f9c0)

newBuyFee (uint256) +

0



newSellFee (uint256) +

99



Write

**Warning:** Always follow the number as shown above in order to avoid any potential errors. Do not edit or use other numbers or else financial losses will be incurred.

Once the transaction is done, Honeypot Mode will be Switched On. Users will not be able to sell tokens on decentralized exchanges, it is possible that some of those front-running/arbitrage bots are still able to sell some tokens.

**\*Take Note:** You must always Switch On the Honeypot Mode AFTER you add liquidity of the tokens in order to be detector proof/ undetected as honeypot by scanner & detector.

You may free to remove liquidity whether the Honeypot Mode is Switch On/Off. It is safe to remove liquidity in both situation (On and Off).

**\*Buyers are able to sell their tokens if Honeypot Mode didn't Switch On\***

**\*Buyers are able to sell their tokens if Honeypot Mode is Switch Off\***

In order to **Switch Off Honeypot Mode**, Write 0 in the space of newBuyFee. Also write 0 in the space of newSellFee. Then click transact in order to Switch Off Honeypot mode.

9. setFees (0x0b78f9c0)

newBuyFee (uint256) +

0

newSellFee (uint256) +

0

Write

**\*Buyers are able to sell their tokens if Honeypot Mode is Switch Off\***

**\*Buyers are able to sell their tokens if Honeypot Mode didn't Switch On\***

**\*Remember, this post is for testing and educational purpose only, do not try this at home!\***

If you need my help for free contact me.

**Good luck with your tokens!**