How to deploy a token and bridge contract

1. Files

There are several files and folders in seasaw token and you will manage them through contracts, scripts folders, and hardhat.config.js file.

In contracts folder, there are solidity files that are directly compiled and deployed to the blockchain.

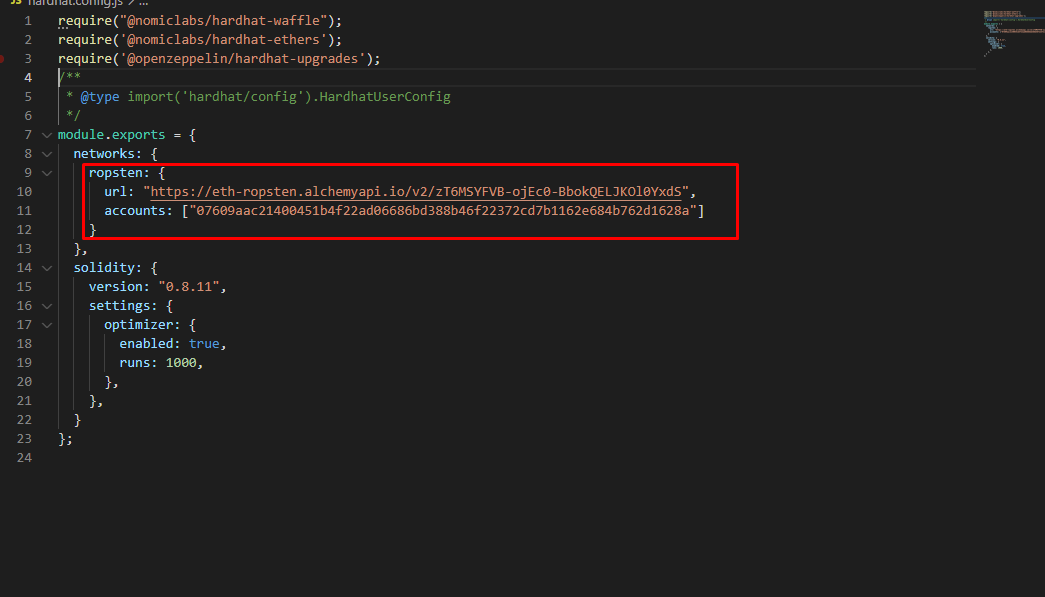
You don’t need to modify them, which are already audited.

In hardhat.config.js, you should configure the network, account, gas, and other things you wanna set for deployment.

In script folder, there are scripts that are used to deploy the contract.

1. How to deploy
2. How to configure the network

Open the hardhat.config.js



There is a networks field.

You can add a network to deploy like above image.

For example, you can add bsc testnet to deploy like below.

bsc\_testnet : {

url: "https://data-seed-prebsc-1-s1.binance.org:8545/",

accounts: ["be331e0f5120038424eeec8cc726e3213ad9038429500a4667f6b57b161d31c0"]

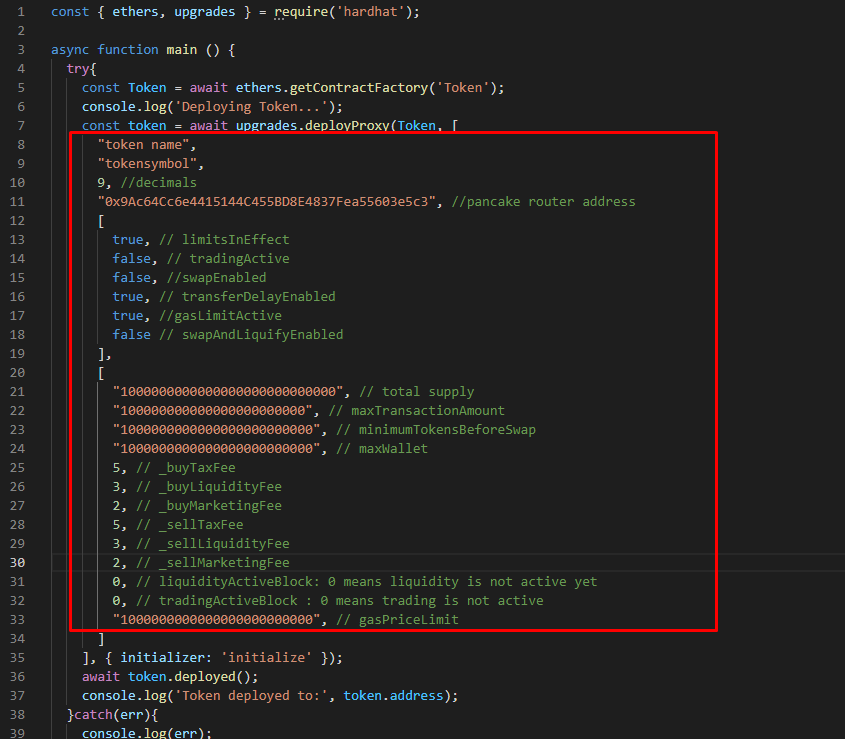
}

url : The endpoint to the network, you can get this info from the metamask.  
account : the wallet private key you want to use to deploy the contract.  
This account will be the owner of the contract.

1. How to deploy the token & bridge contract

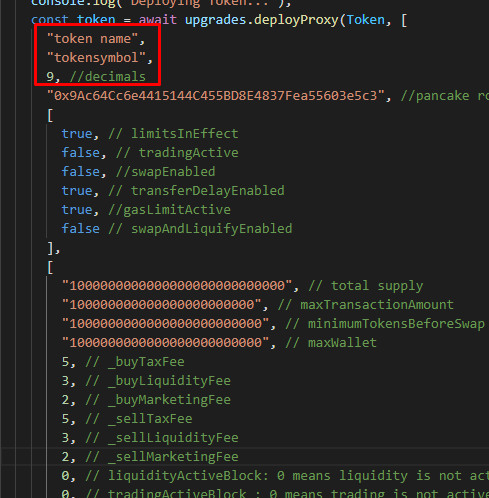
Now you can deploy the contract by using the script.

Let’s see the token deployment script first. (deploy\_upgradeable\_token.js)



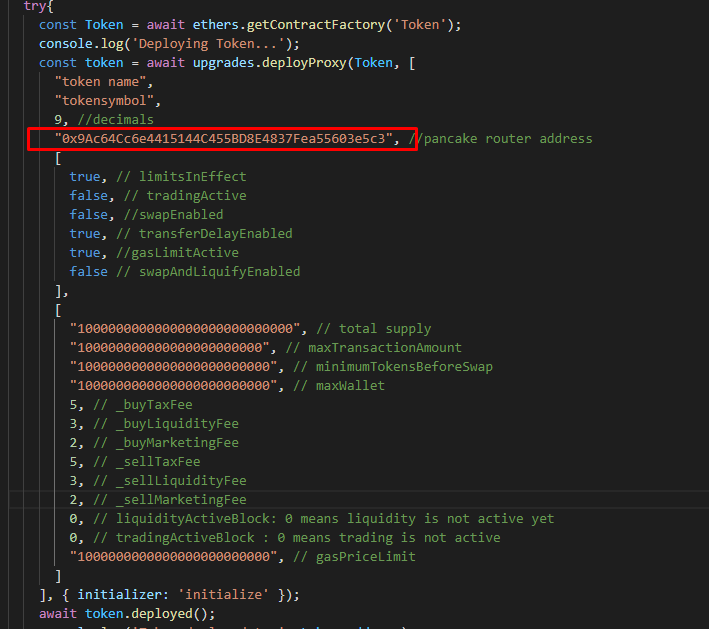
As you can see, there are parameters to be used in the contract that you should input when you deploy.

You should replace all the fields.



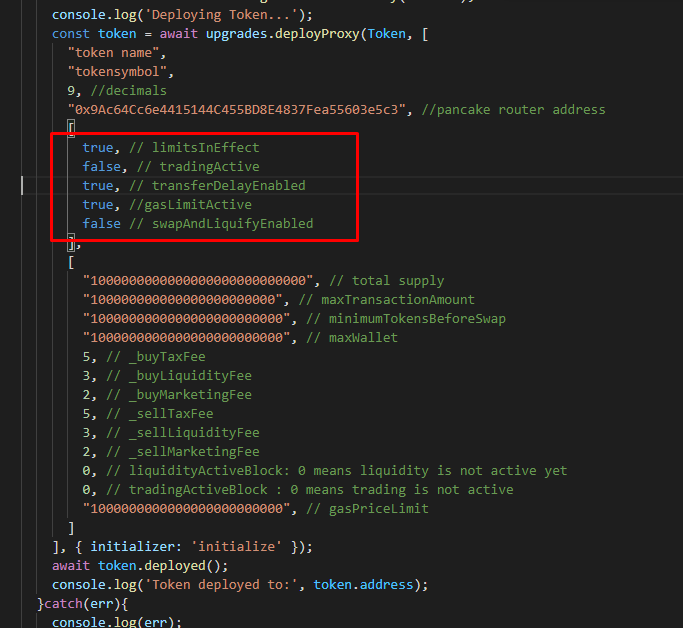
This is a core info for token.

You should replace with the right token name, symbol and decimals.



This is a router address for dex.

If you deploy it on Ethereum, uniswap router address should be input, and if you use binance, pancakeswap router address should be filled up.

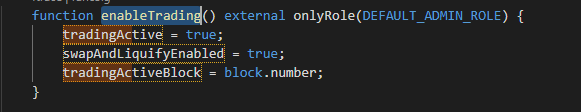


These fields are used for anti sniper and other settings.

Above setting is default.

limitsInEffect : true => anti sniper, maxTransaction amount, max wallet amount… , limits will be turned on.

tradingActive : true => all the accounts can sell or buy tokens from the dex  
you can turn on this switch by running this method – enableTrading().

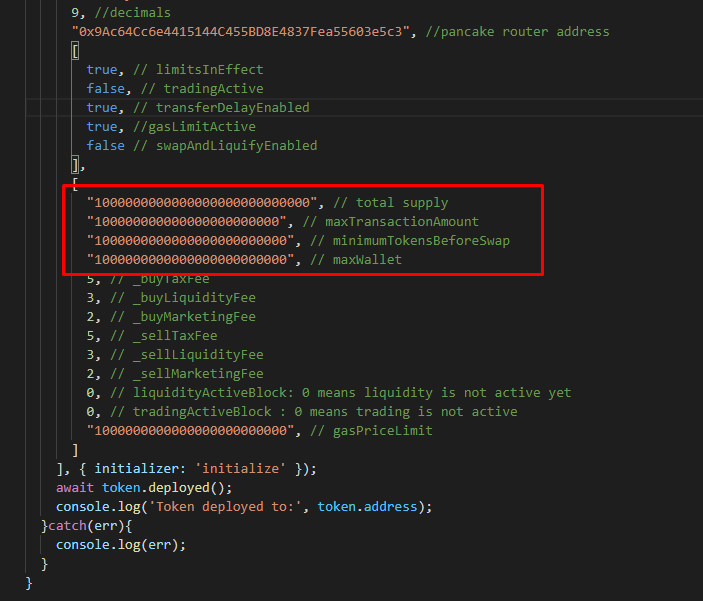


You can see this method in the token contract.

transferDelayEnabled: true => anti sniper feature, an account can buy or sell only one time in a transaction.

gasLimitActive : true => anti sniper feature, only the transaction with below the limit of gas will be accepted.

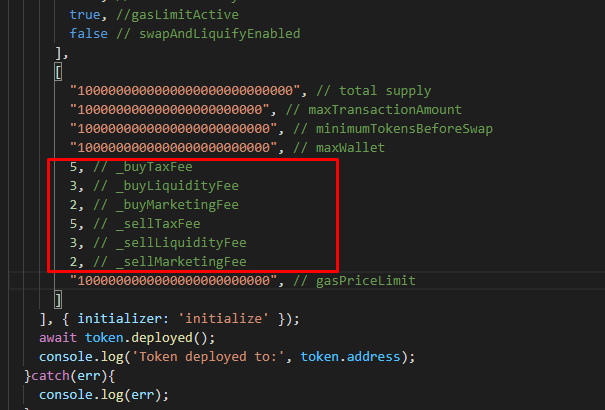
swapAndLiquifyEnabled : true => auto liquidity & marketing fee will be swapped to BNB automatically



Total supply, max transaction amount limit, minimum token amount for swap to BNB to the marketing wallet and auto liq, max wallet amount limit.

All the values includes decimals.  
so if you want to make the total supply 1000 and the decimals is 9, you should input like below.

1000000000000



Fee percentage



Gas price limit to prevent snipers that has also decimals.

You can deploy the token contract by using the below command in the script folder.

npx hardhat run --network bsc\_testnet scripts/deploy\_upgradeable\_token.js

you should change the network parameter with yours that is filled up in the hardhat.config.js.

If you didn’t install the npm modules before run the above cmd, it will be failed.

You should run this cmd in the token folder then.

npm install

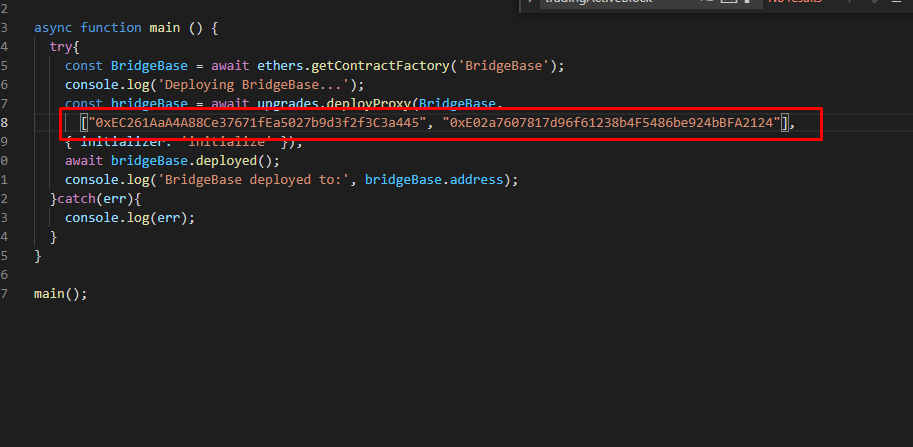
After the token contract deployed, you should save the token address.

Now you can deploy the bridge contract.( deploy\_upgradeable\_bridge.js)

There are 2 parameters.

First one is a account for admin.

2nd one is a token address you want to use in this bridge.



You should replace them with your account and the token address.

Now you should run this cmd in the script folder.

npx hardhat run --network bsc\_testnet scripts/deploy\_upgradeable\_bridge.js

don’t forget to save the address to be deployed.

1. How to upgrade the contract

You can upgrade the token contract by running the scripts below.

upgrade\_token.js

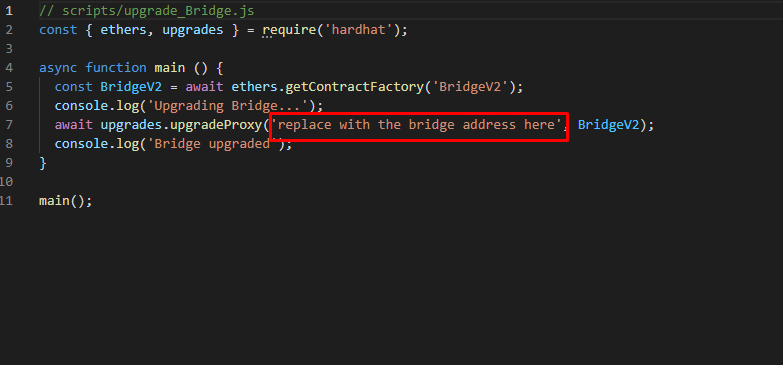
upgrade\_bridge.js

you should fill up the token and bridge address you wanna upgrade before run the cmd.



In the token upgrade script, I used TokenV2. In this case, there should be a TokenV2.sol file in contract folder.

Everything is the same in bridge upgrade script.



Now you can run the cmd below to upgrade.

npx hardhat run --network bsc\_testnet scripts/upgrade\_bridge.js

npx hardhat run --network bsc\_testnet scripts/upgrade\_token.js