目录

[1. Fallback Proxy and How to Upgrade Contract 1](#_Toc16280)

[1.1 What is solidity assembly language 1](#_Toc3999)

[1.2 Fallback Proxy Contract Code Explanation 3](#_Toc7979)

[1.3 how to upgrade smart contracts 5](#_Toc13072)

[2 Docker File:How to dockernize a nodejs webapp 9](#_Toc11121)

[2.1 write a nodejs webapp 9](#_Toc29457)

[2.2 How to build a Docker image from a docker file? 9](#_Toc25363)

[2.3 How to push to the docker hub? 11](#_Toc29399)

[3 Docker-compose: 12](#_Toc18144)

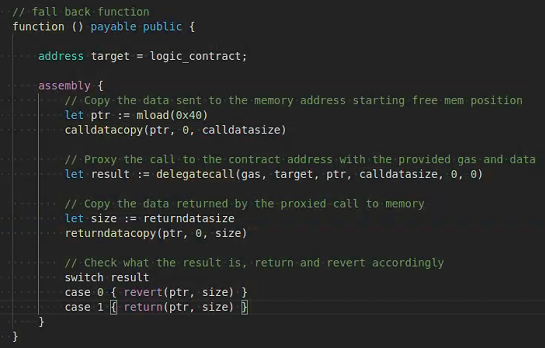
# Fallback Proxy and How to Upgrade Contract

## 1.1 What is solidity assembly language

<https://docs.soliditylang.org/en/v0.5.3/assembly.html>

Code:

<https://github.com/onebit256/Upgradable-Proxy-Smart-Contract>



Solidity manages memory in a very simple way: There is a “free memory pointer” at position 0x40 in memory. If you want to allocate memory, just use the memory starting from where this pointer points at and update it accordingly

So this 0x40 is a “free memory pointer”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| mload(p) |  | F | mem[p…(p+32)) | |
| delegatecall(g, a, in, insize, out, outsize) | |  | H | identical to callcode but also keep caller and callvalue |

|  |  |  |  |
| --- | --- | --- | --- |
| calldatacopy(t, f, s) | - | F | copy s bytes from calldata at position f to mem at position t |

calldatacopy(ptr, 0, calldatasize) here 0 means copy starts from the beginning of the data

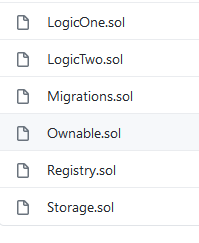
|  |  |  |  |
| --- | --- | --- | --- |
| call(g, a, v, in, insize, out, outsize) |  | F | call contract at address a with input mem[in…(in+insize)) providing g gas and v wei and output area mem[out…(out+outsize)) returning 0 on error (eg. out of gas) and 1 on success |
| callcode(g, a, v, in, insize, out, outsize) |  | F | identical to call but only use the code from a and stay in the context of the current contract otherwise |

|  |  |  |  |
| --- | --- | --- | --- |
| returndatasize |  | B | size of the last returndata |

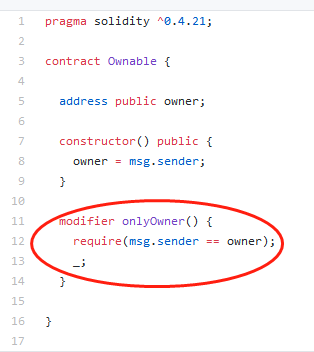
|  |  |  |  |
| --- | --- | --- | --- |
| returndatacopy(t, f, s) | - | B | copy s bytes from returndata at position f to mem at position t |

|  |  |  |  |
| --- | --- | --- | --- |
| return(p, s) | - | F | end execution, return data mem[p…(p+s)) |
| revert(p, s) | - | B | end execution, revert state changes, return data mem[p…(p+s)) |

## 1.2 Fallback Proxy Contract Code Explanation



Ownable.sol



1. Modifier is a decorator function. When we append this as a key word to a function, the function would execute the codes in the decorator function and execute the main function

\_; The function body is inserted where the special symbol "\_;" appears in the definition of a modifier



1. So setLogicContract function would execute the modifier first

setLogicContract is the implementation address registry

1. Function() fallback function, when someone send ETH to this contract without providing data or someone try to call function doesn’t Exit

## 1.3 how to upgrade smart contracts

1. start gananche-cli

Ganache-cli -m “your mnemonic ”

This will always generate the same set of address

1. Type: truffle migrate --network development.

You should be able to see the following output



1. Type: Truffle console

3.1 register log1 address

- Let regi = await Registry.at(Registry.address);

- regi .setLogicContract(LogicOne.address)

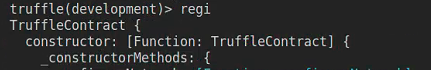
- Let log1 = await LogicOne.at(Registry.address);

- Log1.setValue(2) //the value should be equal to 4

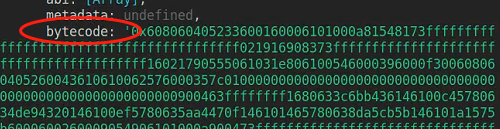
- Let regi = await Registry.at(Registry.address);



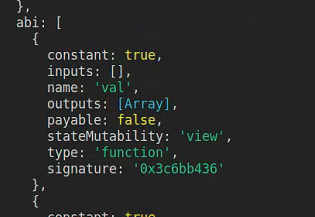
Verify if we manage to instantiate it



Can find the bytecode



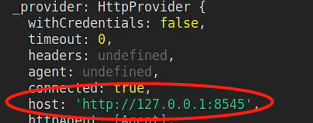
can find abi:



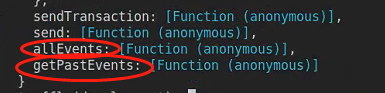
What is abi?

We could think it of as interface to talk with the smart contract on the blockchain

Can find rpc



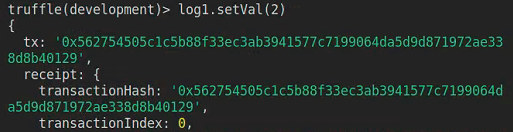
And getPastEvents function



- regi .setLogicContract(LogicOne.address)

- Let log1 = await LogicOne.at(Registry.address);

- Log1.setValue(2)



This means, it is successfully written on the chain.



And the value == 4.

3.2 upgrade logic layer to LogicTwo

- regi.setLogicContract(LogicTwo.address)

- let log2 = await LogicTwo.at(Registry.address)

- log2.setVal(2)

// check value: value should be 6

- LogicTwo.at(Registry.address).val()

- regi.setLogicContract(LogicTwo.address)

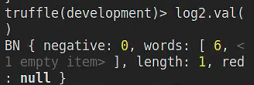


- let log2 = await LogicTwo.at(Registry.address)



- log2.setVal(2)

- log2.val()



3.3 Extra Question

LogicOne should still be able to set the val, when use set val on log1, value is 6. WHY?

Log1.setVal(2)

Log1.val()

# 2 Docker File:How to dockernize a nodejs webapp

## 2.1 write a nodejs webapp

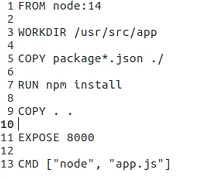
1. Mkdir webapp
2. Cd webapp
3. Npm init
4. Npm install expressjs

Create app.js



## 2.2 How to build a Docker image from a docker file?

1 . create a Dockerfile



1. docker build {docker file path}

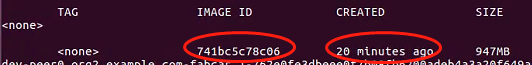
In our case: sudo docker build .





1. Interact with docker container

Sudo docker image list

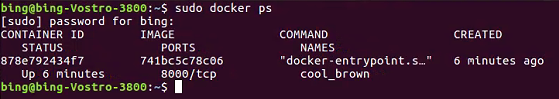


Start docker container :

Sudo docker run -it (image id/ image name)



sudo docker ps



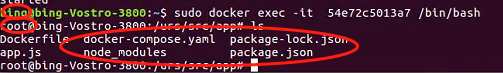
If we open 127.0.0.1:8000, doesn’t work

How to debug?

sudo docker logs --tail 10 xxx



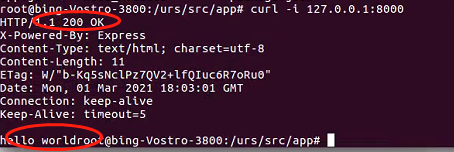
sudo docker exec -it xxx /bin/bash (container interaction)



The user changed, and we are in the working directory of the container

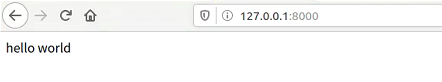
Use curl inside of the container

Curl -i 127.0.0.1:8000



So we have to add localhost’s network to the container:



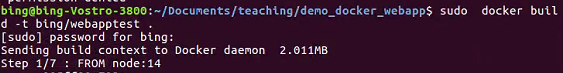


## 2.3 How to push to the docker hub?

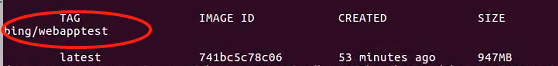
https://phoenixnap.com/kb/create-docker-images-with-dockerfile

docker build -t <your username>/node-web-app .

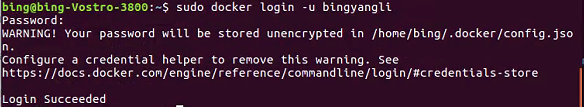
-t tag your image



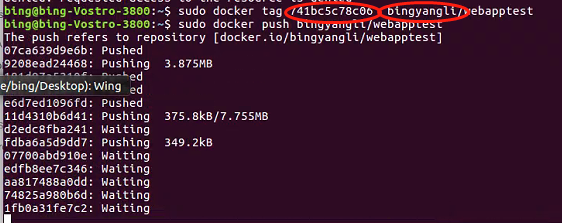
Sudo docker image list



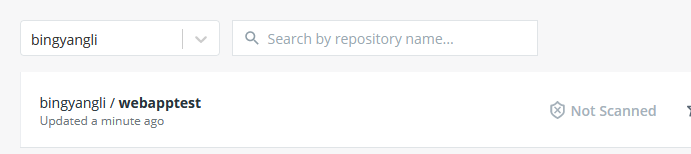
Sudo docker login



Add tag to docker image, the name should be the same as your docker hub user name



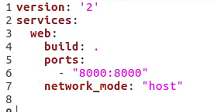




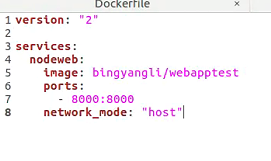
# 3 Docker-compose:

2 ways to use a image in docker-compose.yaml

1. Ask docke-compose.yml to build the image from the local docker file, in this case, we have to put Dockerfile and docker-compose file under the same directory



1. Pull the image from the docker hub



Docker-compose up -d

-d for daemon process

Extra question: Remove network\_mode and try