1. Deploy Bridge and ERC20Handler Contract on Binance, Verify and Configure.

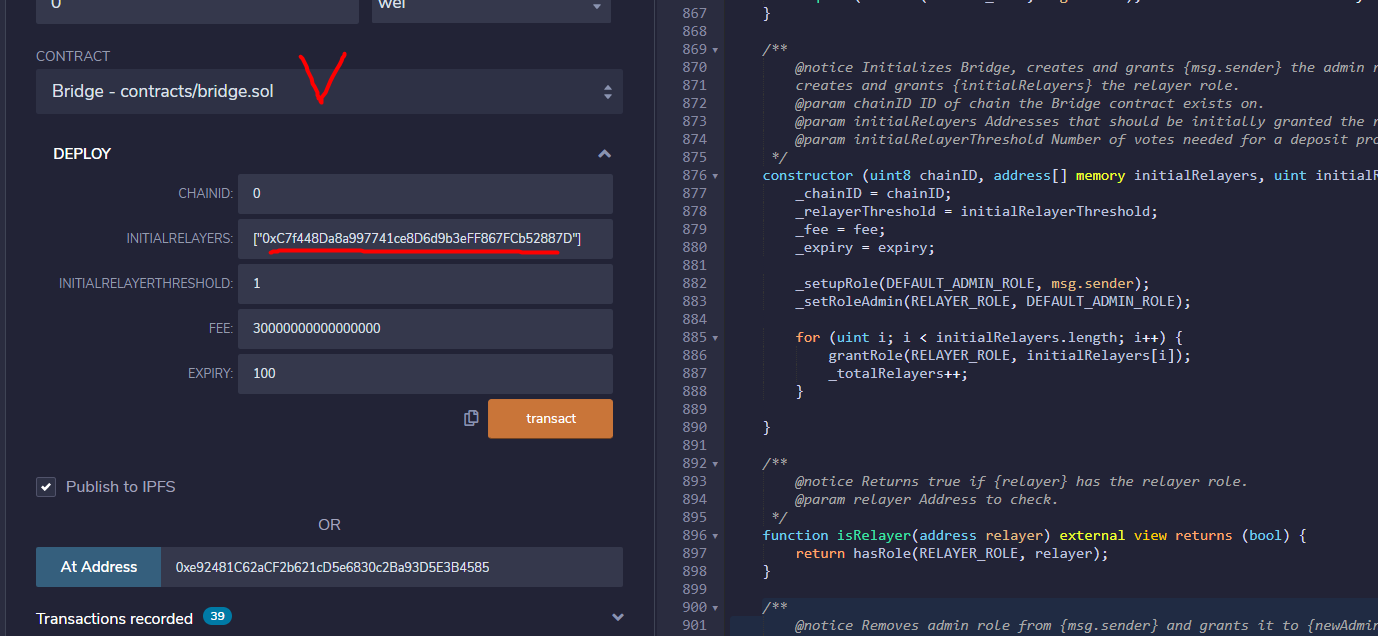
Here, the parameters are currently base on Binance Smart Chain.

1. Bridge Contract

Compiler version: 0.64

Source code:

[https://github.com/vitalikvaraksa/avaBscCross/blob/master/solidity/lockMint-lockMint/Bridge.sol](https://github.com/vitalikvaraksa/avaBscCross/blob/master/solidity/lockMint-burnRelease/Bridge.sol)

Parameter: 

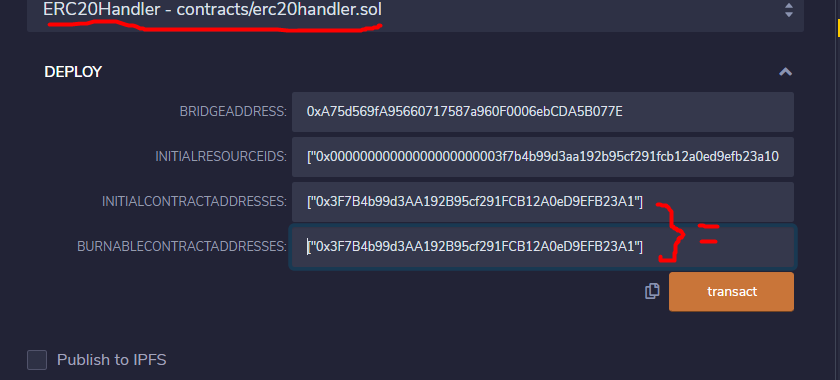
Set the parameters like above. INITIALRELAYER: wallet address that will be used for relayer.

1. ERC20Handler Contract

Compiler version: 0.64

Source code:

https://github.com/vitalikvaraksa/avaBscCross/blob/master/solidity/lockMint-lockMint/ERC20Handler.sol

Parameter: 

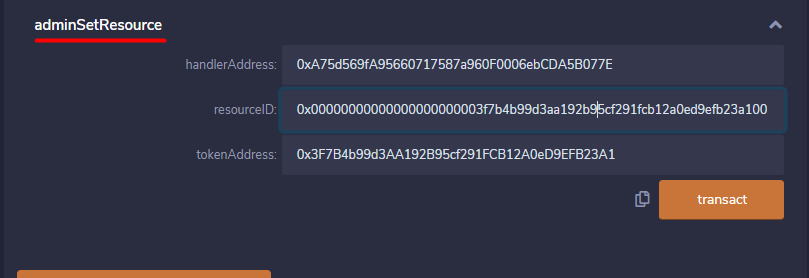
BRIDGEADDRESS: bridge contract address on Binance

INITIALRESOURCEIDS: array of resourceids on Binance, token resource id can be one element of this array.

INITIALCONTRACTADDRESS: array of contract address on Binance, token address can be one element of this array, e.x OPUS token

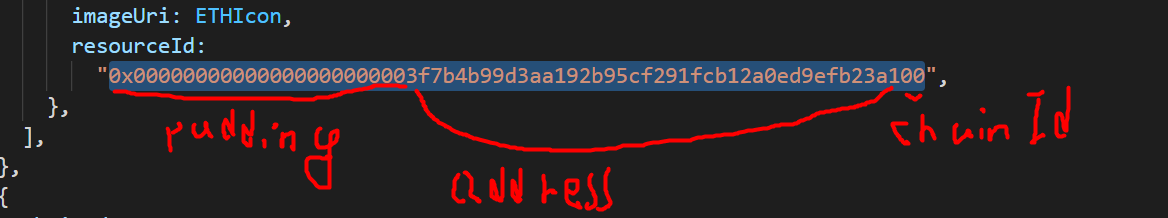
BURNABLECONTRACTADDRESSES: array of contract address that will be burnable

1. Verify them with details of above
2. Configure on Bridge contract



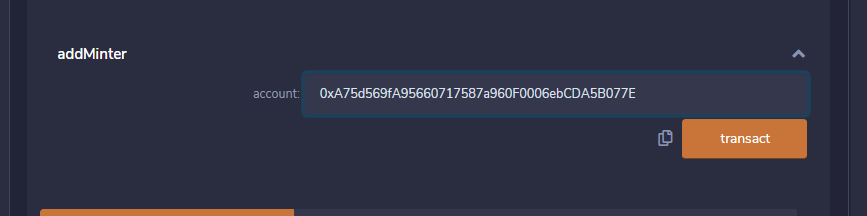
handlerAddress: ERC20Handler Contract address

resourceID: OPUS token resourceID (IT WILL BE SAME ON ALL CHAIN)



tokenAddress: OPUS token address

1. Give mint privilege to ERC20Handler contract



Account: ERC20Handler contract address

1. Deploy Bridge and ERC20Handler Contract on Avalanche, Verify and Configure.

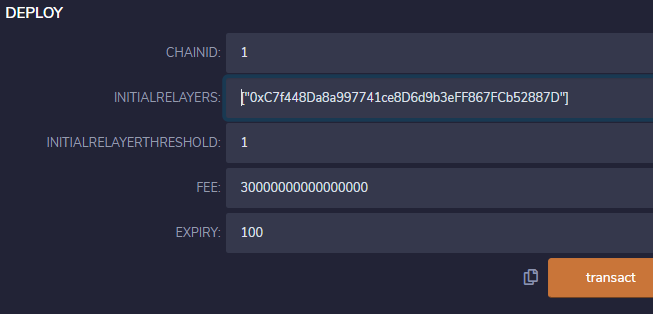
Here, the parameters are currently base on Binance Smart Chain.

1. Bridge Contract

Compiler version: 0.64

Source code:

<https://github.com/vitalikvaraksa/avaBscCross/blob/master/solidity/lockMint-lockMint/Bridge.sol>

Parameter: 

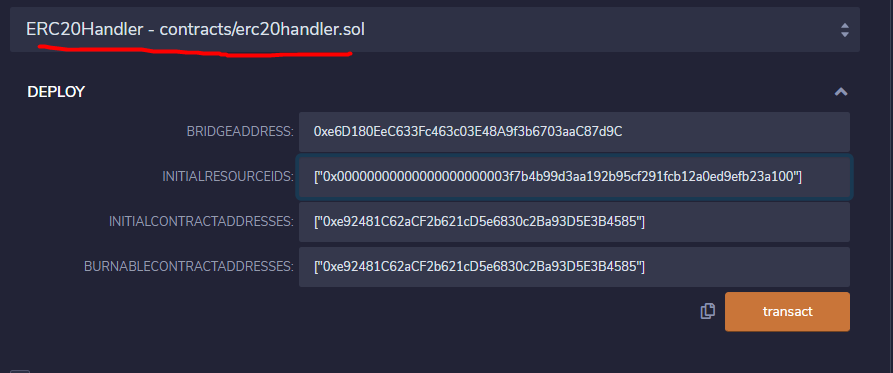
Set the parameters like above. INITIALRELAYER: wallet address that will be used for relayer.

1. ERC20Handler Contract

Compiler version: 0.64

Source code:

https://github.com/vitalikvaraksa/avaBscCross/blob/master/solidity/lockMint-lockMint/ERC20Handler.sol

Parameter: 

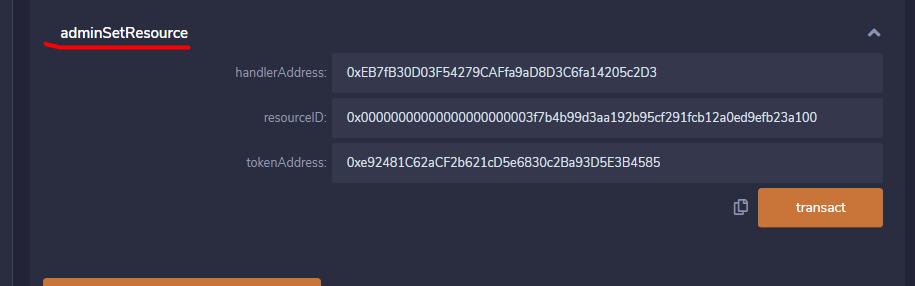
BRIDGEADDRESS: bridge contract address on Avalanche

INITIALRESOURCEIDS: array of resourceids on Avalanche, token resource id can be one element of this array.

INITIALCONTRACTADDRESS: array of contract address on Avalanche, token address can be one element of this array, e.x OPUS token

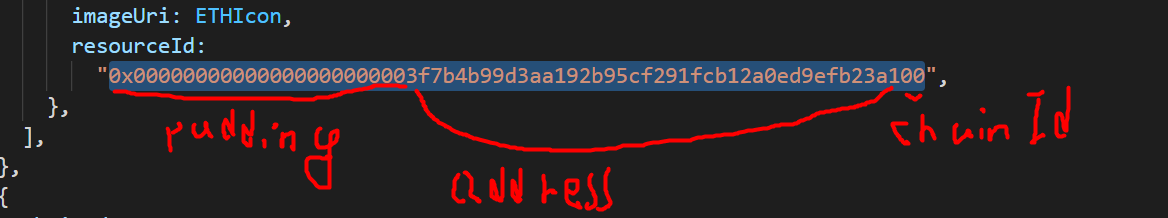
BURNABLECONTRACTADDRESSES: array of contract address that will be burnable

1. Verify them with details of above
2. Configure on Bridge contract



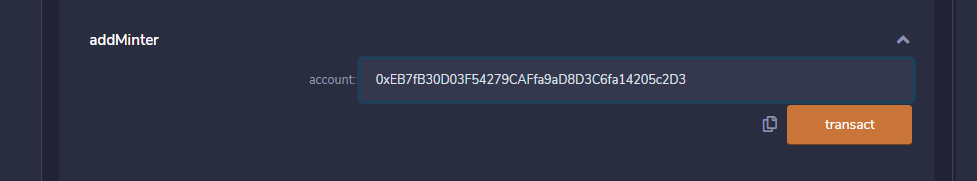
handlerAddress: ERC20Handler Contract address on Avalanche

resourceID: OPUS token resourceID



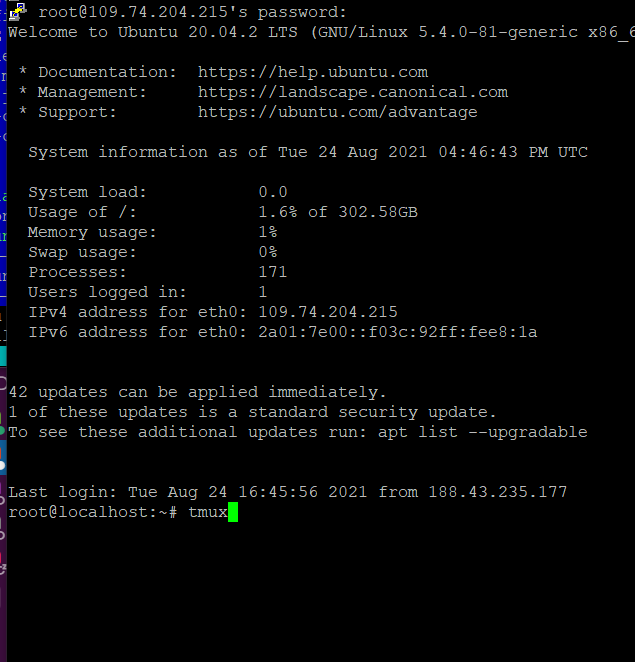
tokenAddress: OPUS token address on Avalanche

1. Give mint privilege to ERC20Handler contract



Account: ERC20Handler contract address on Avalanche

1. Running Relayer
2. First, Launch a Terminal and Run “tmux” on it.



Run “run\_relayer.sh” and input the password for 2 chains.



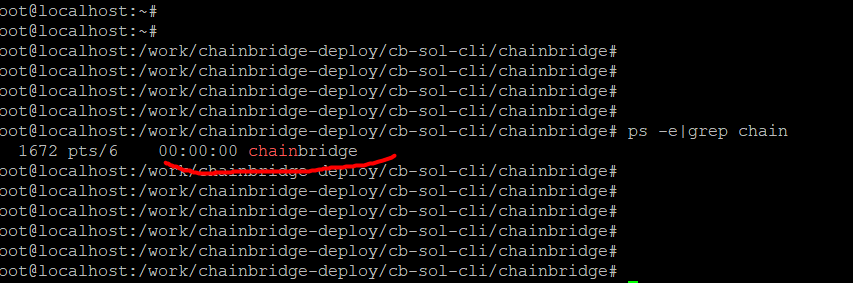
Now, Once you launch the Relayer, it will be running after putty session is disabled.

By using following method, you can check if it is running.

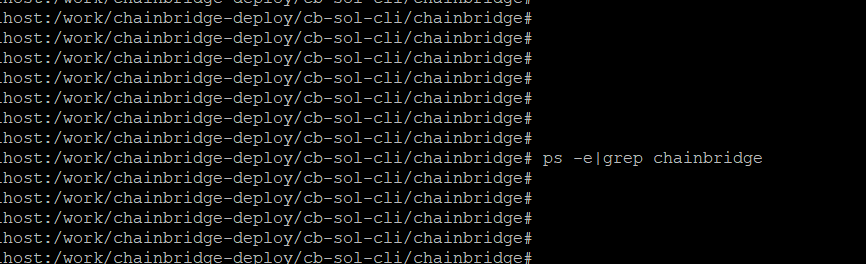
1. Check if the Relayer is running now

Run “ps -e|grep chainbridge” on terminal

If there is a letter of “run\_relayer.sh”, it is running now.(as following picture)



If nothing, it’s not running now.(as following picture)



You can use this command to check if it is running now.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Create Relayer and Run it\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Relay wallet address should be one that was used to deploy contracts

1. Make /work/chainbridge-deploy/cb-sol-cli/chainbridge/run.sh
2. Edit it like following and save it

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*run.sh\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#!/bin/sh

rm config.json

RELAY\_ADDR="<Relayer wallet address>"

SRC\_BRIDGE="<Bridge Contract Address on Binance>"

SRC\_HANDLER="<ERC20Handler Contract Address on Binance>"

DST\_BRIDGE="<Bridge Contract Address on Avalanche c-chain>"

DST\_HANDLER="<ERC20Handler Contract Address on Avalanche c-chain>"

echo "{

\"chains\": [

{

\"name\": \"Binance\",

\"type\": \"ethereum\",

\"id\": \"0\",

\"endpoint\": \"wss://speedy-nodes-nyc.moralis.io/60ca04c0cc0a62bb55e61fe9/bsc/mainnet/ws\",

\"from\": \"$RELAY\_ADDR\",

\"opts\": {

\"bridge\": \"$SRC\_BRIDGE\",

\"erc20Handler\": \"$SRC\_HANDLER\",

\"genericHandler\": \"$SRC\_HANDLER\",

\"gasLimit\": \"1000000\",

\"maxGasPrice\": \"23000000000\"

}

},

{

\"name\": \"Avalanche\",

\"type\": \"ethereum\",

\"id\": \"1\",

\"endpoint\": \"wss://api.avax.network/ext/bc/C/ws\",

\"from\": \"$RELAY\_ADDR\",

\"opts\": {

\"bridge\": \"$DST\_BRIDGE\",

\"erc20Handler\": \"$DST\_HANDLER\",

\"genericHandler\": \"$DST\_HANDLER\",

\"gasLimit\": \"1000000\",

\"maxGasPrice\": \"23000000000\"

}

}

]

}" >> config.json

1. Run “chmod 777 run.sh” and launch “run.sh” on “/work/chainbridge-deploy/cb-sol-cli/chainbridge”
2. On terminal, run following command

PK=”<Relayer wallet Private key>”

./build/chainbridge accounts import --privateKey $PK

./build/chainbridge --config config.json --verbosity trace –latest &