1. 部署万纳链 Venachain

- Github repo: https://github.com/Venachain/Venachain
- 开发文档: https://venachain-docs.readthedocs.io/zh/latest/
- 部署文档: https://venachain-docs.readthedocs.io/zh/latest/documents/quick/deploy.html

1.1 releases包部署

```
wget
https://github.com/Venachain/Venachain/releases/download/v1.0.1/Venachain_linux_
amd64_v1.0.1.tar.gz
tar -zxvf Venachain_linux_amd64_v1.0.1.tar.gz
```

1.2 容器内部署单节点

```
#新建容器
docker run -it -v ~/install/tmp/:/opt -p 6791:6791 --name venachain ubuntu:20.04
apt update
apt install curl

    #二次进入容器
docker container start venachain
docker exec -it venachain bash

cd /opt/linux
export WORKSPACE=./
cd ${WORKSPACE}/scripts/

# 首次部署并启动节点
./venachainctl.sh one
    # 二次启动节点
./venachainctl.sh start -n 0

# 查看节点状态
./venachainctl.sh status -n 0

# 停止
./venachainctl.sh stop -n 0
# 停止清数据
./venachainctl.sh clear -a
```

1.3 与链交互

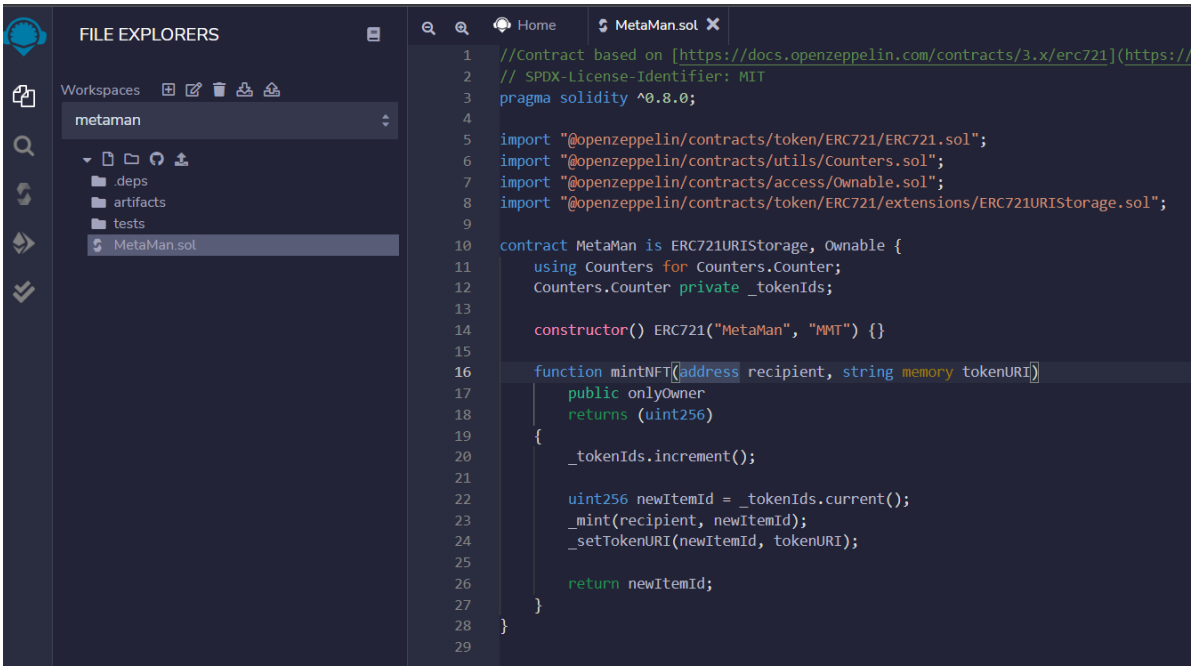
```
cd ${WORKSPACE}/scripts/
./venachainctl.sh console -n 0

eth.accounts
eth.blockNumber
#测试一笔交易
eth.sendTransaction({from:eth.accounts[0],to:eth.accounts[0]})
#查看交易回执
```

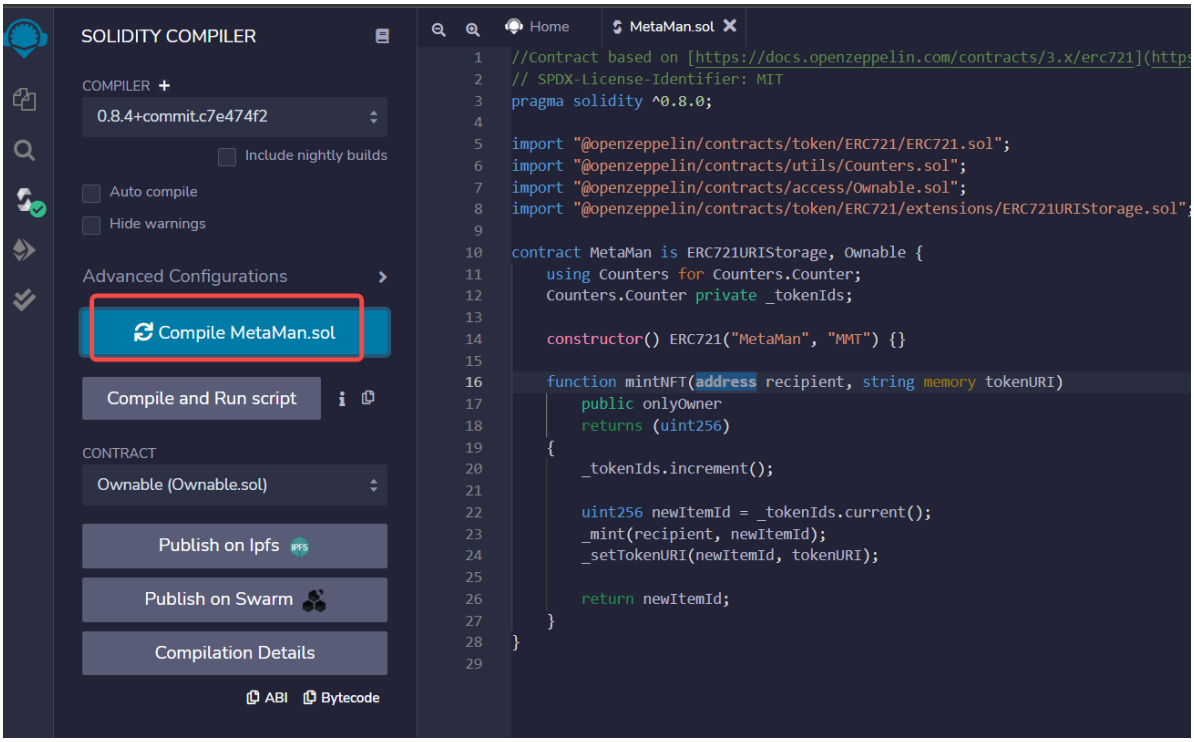
```
eth.getTransactionReceipt("0xcb1d67354698038e23fe736fd0b6cdd51dd4c7bb3927fdc4fd1774aab06b4040")
#查看交易池状态
txpool.status

#解锁 (默认密码 0)
personal.unlockAccount(eth.accounts[0])
personal.unlockAccount(eth.accounts[0], "0")
```

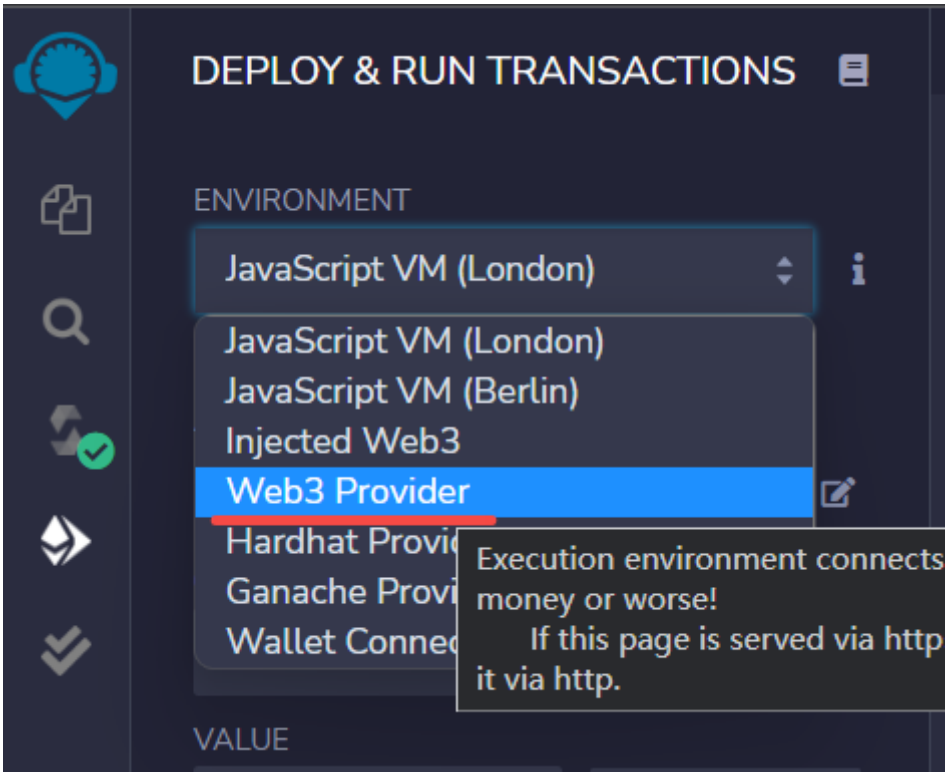
2. 部署NFT合约 MetaMan

注意: 部署前先解锁本地的钱包, 通过命令 `personal.unlockAccount(eth.accounts[0], "0")`

2.1 在remix中编译合约并部署







External node request

Note: To use Geth & https://remix.ethereum.org, configure it to allow requests from Remix:(see Geth Docs on rpc server)

```
geth --http --http.corsdomain https://remix.ethereum.org
```

To run Remix & a local Geth test node, use this command: (see Geth Docs on Dev mode)

```
geth --http --http.corsdomain="https://remix.ethereum.org" --http.api
web3,eth,debug,personal,net --vmdebug --datadir
<path/to/local/folder/for/test/chain> --dev console
```

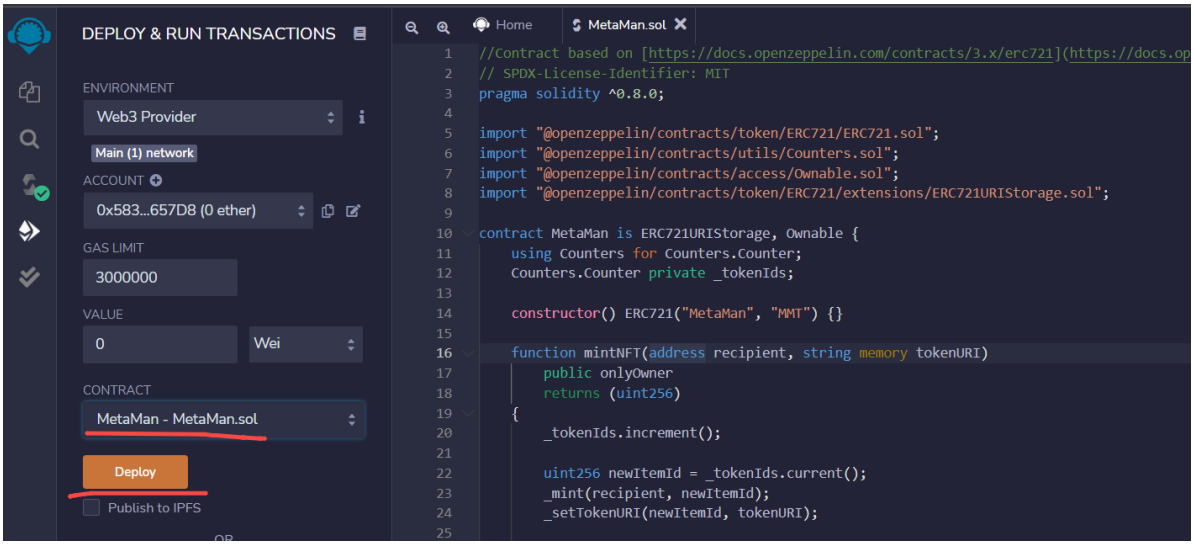
WARNING: It is not safe to use the --http.corsdomain flag with a wildcard: --**http.corsdomain ***

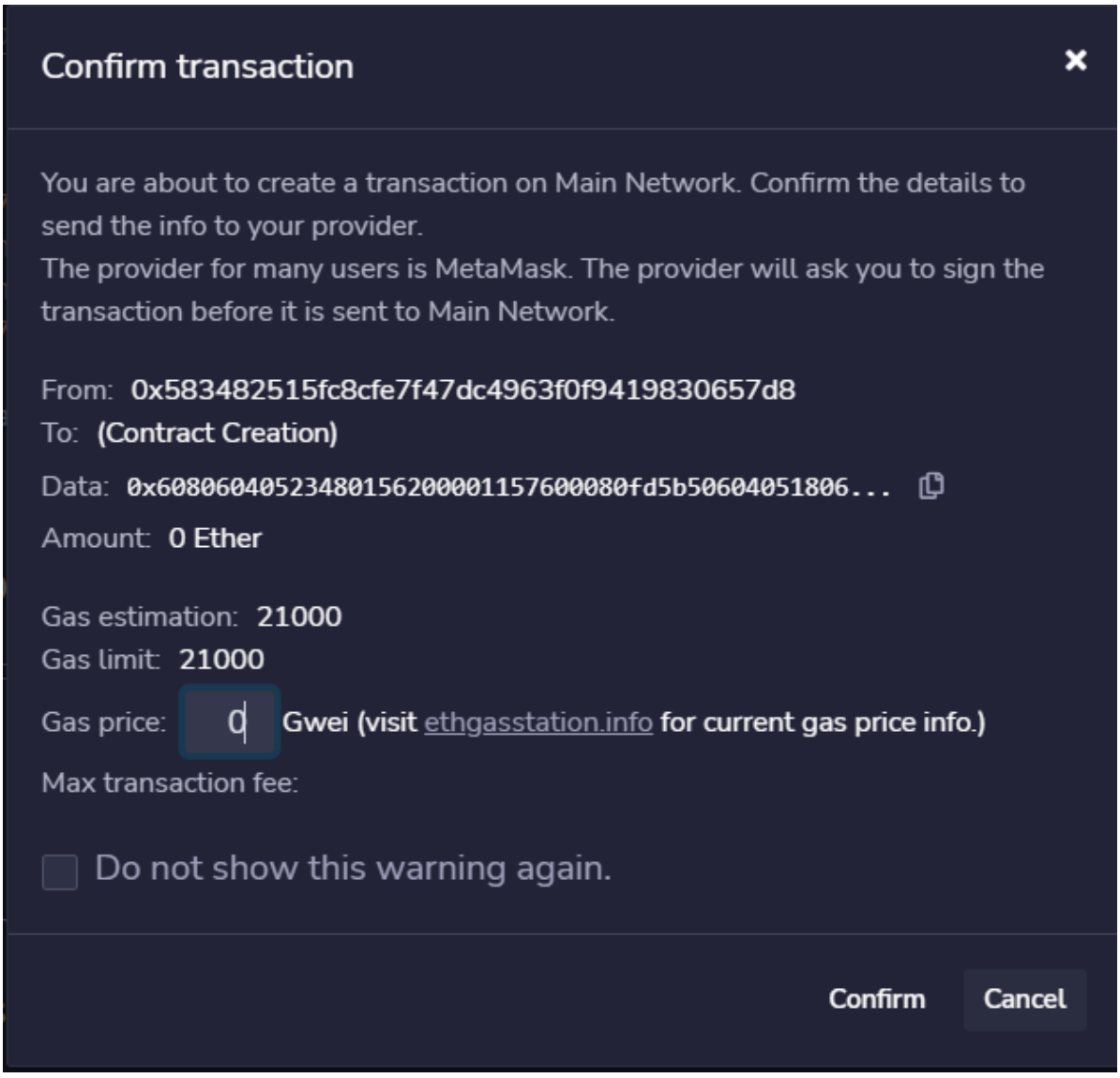
For more info: Remix Docs on Web3 Provider

Web3 Provider Endpoint

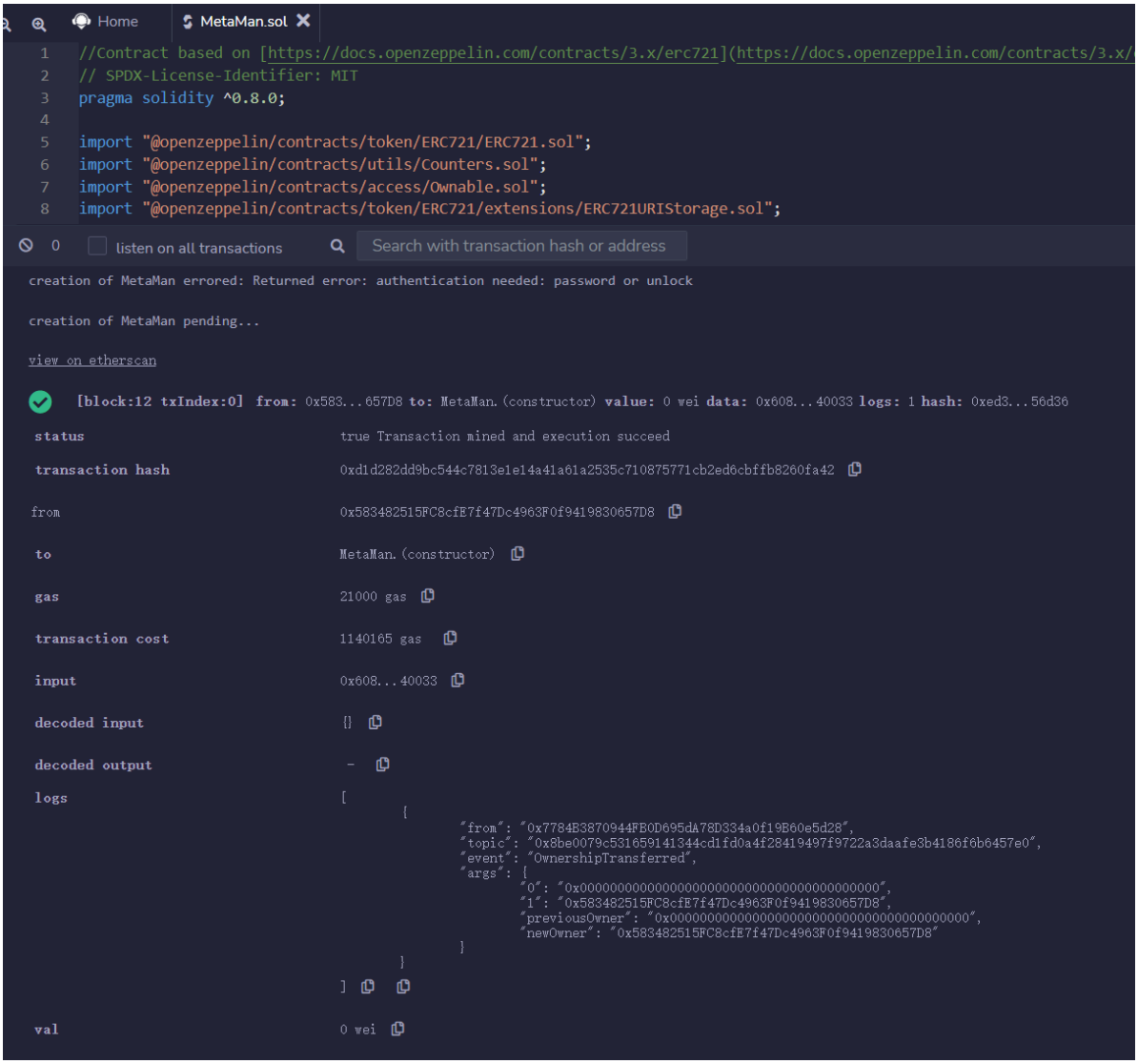
http://127.0.0.1:6791

OK Cancel


DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Web3 Provider
Main (1) network
ACCOUNT
0x583...657D8 (0 ether)
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT
MetaMan - MetaMan.sol
Deploy
Publish to IPFS
Home
MetaMan.sol
//Contract based on [https://docs.openzeppelin.com/contracts/3.x/erc721](https://docs.op
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/utils/Counters.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
contract MetaMan is ERC721URIStorage, Ownable {
using Counters for Counters.Counter;
Counters.Counter private _tokenIds;
constructor() ERC721("MetaMan", "MMT") {}
function mintNFT(address recipient, string memory tokenURI)
public onlyOwner
returns (uint256)
{
_tokenIds.increment();
uint256 newItemId = _tokenIds.current();
_mint(recipient, newItemId);
_setTokenURI(newItemId, tokenURI);


Confirm transaction
You are about to create a transaction on Main Network. Confirm the details to send the info to your provider.
The provider for many users is MetaMask. The provider will ask you to sign the transaction before it is sent to Main Network.
From: 0x583482515fc8cfe7f47dc4963f0f9419830657d8
To: (Contract Creation)
Data: 0x60806040523480156200001157600080fd5b50604051806...
Amount: 0 Ether
Gas estimation: 21000
Gas limit: 21000
Gas price: 0 Gwei (visit ethgasstation.info for current gas price info.)
Max transaction fee:
Do not show this warning again.
Confirm
Cancel

2.2 部署成功



2.3 在命令行中查看

看到部署的**合约地址是:** `0xde5faa436bc5e8ff29cbfe9c48a52fd860260d68`

```
> eth.getTransactionReceipt("0xd1d282dd9bc544c7813e1e14a41a61a2535c710875771cb2ed6cbffb8260fa42")
{
  blockHash: "0xed34da42393e772f06a3b022c775da98644c70fc908d3ef889ec8d9b62d56d36",
  blockNumber: 12,
  contractAddress: "0x7784b3870944fb0d695da78d334a0f19b60e5d28",
  cumulativeGasUsed: 1140165,
  from: "0x583482515fc8cfe7f47dc4963f0f9419830657d8",
  gasUsed: 1140165,
  logs: [{
      address: "0x7784b3870944fb0d695da78d334a0f19b60e5d28",
      blockHash: "0xed34da42393e772f06a3b022c775da98644c70fc908d3ef889ec8d9b62d56d36",
      blockNumber: 12,
      data: "0x",
      logIndex: 0,
      removed: false,
      topics: ["0x8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6b6457e0", "0x0000000000000000000000000000
dc4963f0f9419830657d8"],
      transactionHash: "0xd1d282dd9bc544c7813e1e14a41a61a2535c710875771cb2ed6cbffb8260fa42",
      transactionIndex: 0
  }],
  logsBloom: "0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000020000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000020000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
  status: "0x1",
  to: null,
  transactionHash: "0xd1d282dd9bc544c7813e1e14a41a61a2535c710875771cb2ed6cbffb8260fa42",
  transactionIndex: 0
}
```

3. 铸造一个数字人 NFT

3.0 为数字人生成出生证明

未具体列出如果使用*STA*生成出生证明的步骤, 日后再详细说明

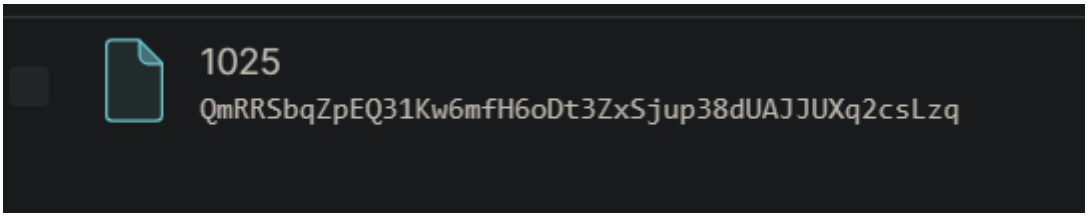

```
{
    "@context": [
      "https://www.w3.org/2018/credentials/v1"
    ],
    "id": "8faba747-8899-4dd6-8925-be061936a90d",
    "issuanceDate": 1652258590771,
    "expirationDate": 1668154143000,
    "credentialSubject": {
      "id": "did:wx::85766e9e-abbc-4c00-b5f4-0ee9ebf6d90a",
      "shortDescription": "string",
      "longDescription": "string",
      "type": "string",
      "data": "{ \"owner\":\"did:vena:aaaaaaaa-452f-4440-aabb-
6ba8b47cd8ef\",\"creator\":\"did:vena:bbbbbbbb-452f-4440-aabb-
6ba8b47cd8ef\",\"metaman\":\"did:vena:cccccccc-452f-4440-aabb-6ba8b47cd8ef\" }"
    },
    "revocation": {
      "type": "SimpleRevocationListV1"
    },
    "proof": {
      "encryptType": "Secp256k1",
      "creator": "did:pnid:cid:7f8ca8982f6cc6e8ea087bd9457ab8024bd2/1",
      "signatureValue":
"8fdfe8dcfa3f306edb088050bd1ed3fe6e392d1502f2fbf1a55d96a02090d273c29fbd25874e1fd5
c7b50b84a3fb05698dbb56ee082c0f5ae3447f17e0a248a1be9a4d47ae5aae688da7755e8901a4d7
364da6fdc91e7c188d8f5711ebfd796e8e04dea591276022c91323b1a05acada9d786751385f5087
d47fd042fa4531b25f56456"
    }
  }
```


3.1 先将数字人的资源传到IPFS上

- 2D形象
- 出生证明 (STA颁发)
- 自我介绍的视频

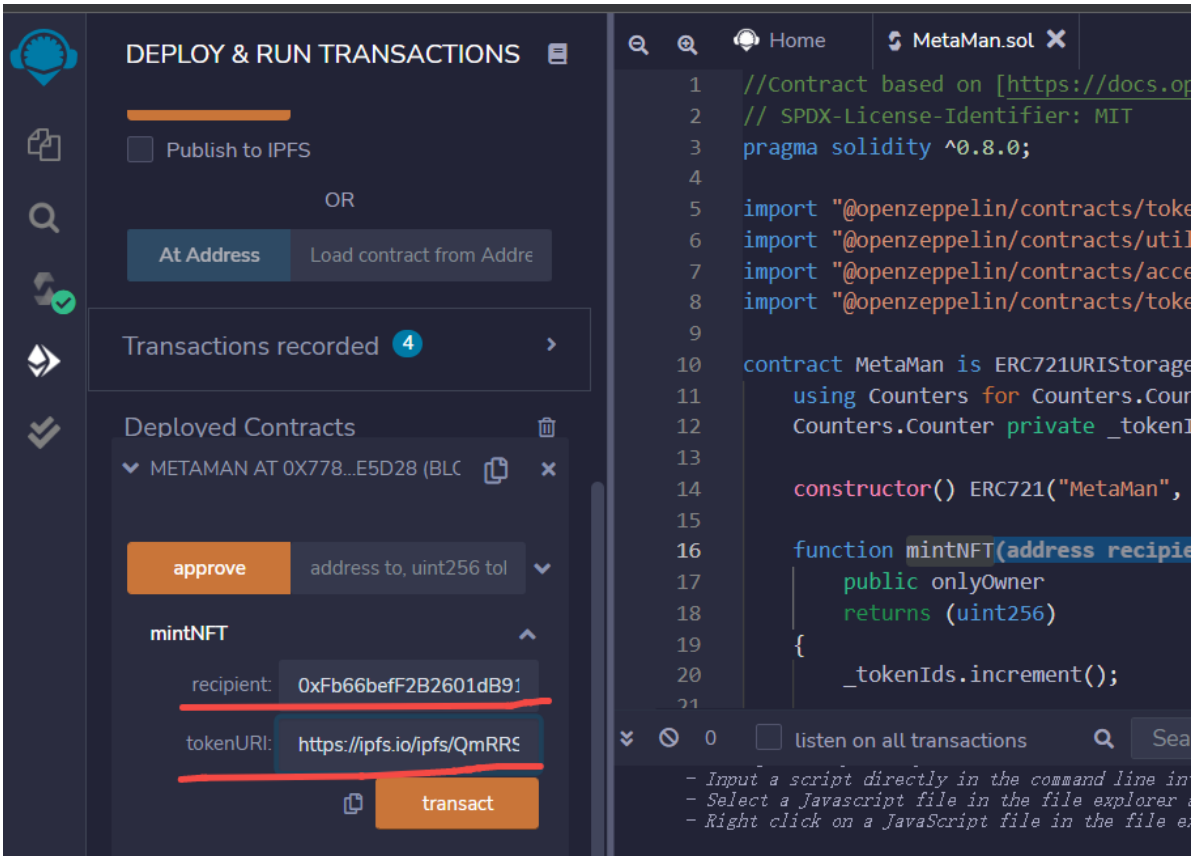
3.2 填写并将NFT对应的元数据文件传到IPFS上


```
{
     "name": "OxJessica",
     "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad
minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id est laborum.",
     "image":
"https://ipfs.io/ipfs/QmYadFKVgcVtuTGBA6KvuVcn4iP58ds5LkYGGiRWHwwmWm",
     "attributes": [
       {
         "display_type": "date",
         "trait_type": "birth_cert",
         "value":
"https://ipfs.io/ipfs/QmefmjSguNLuwXNiYtp7DNENYWbsSd4i5rKtxJbNEx7LFQ"
       },
       {
         "display_type": "date",
         "trait_type": "birthday",
         "value": 1546360800
       },{
         "display_type": "meta",
         "trait_type": "2d",
         "value":
"https://ipfs.io/ipfs/QmefmjSguNLuwXNiYtp7DNENYWbsSd4i5rKtxJbNEx7LFQ"
       },
       {
         "display_type": "meta",
         "trait_type": "3d",
         "value":
"https://ipfs.io/ipfs/QmPAPbhFz5x2CMmNsq9XbguucTtV5vM2znDtn3x1rYyWh4"
       }
     ]
}
```

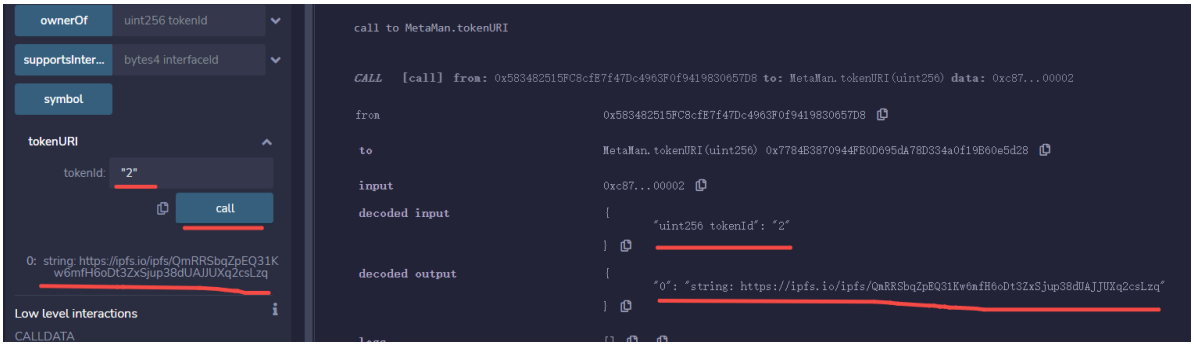
3.3 铸造一个数字人NFT

为我的测试地址 `0xFb66befF2B2601dB91Af45cd3c77706a5D4D13f7` 铸造一个数字人NFT, tokenURI使用ipfs上的元数据文件 `https://ipfs.io/ipfs/QmRRSbqZpEQ31Kw6mfH6oDt3ZxSjup38dUAJJUXq2csLzq`

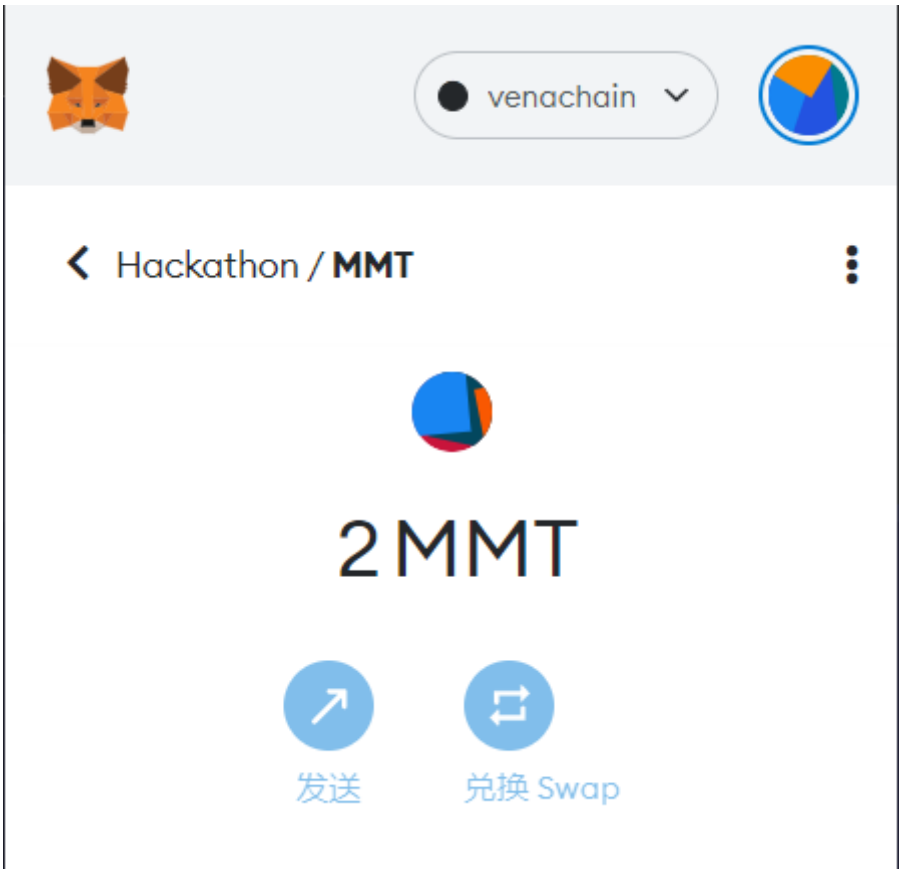


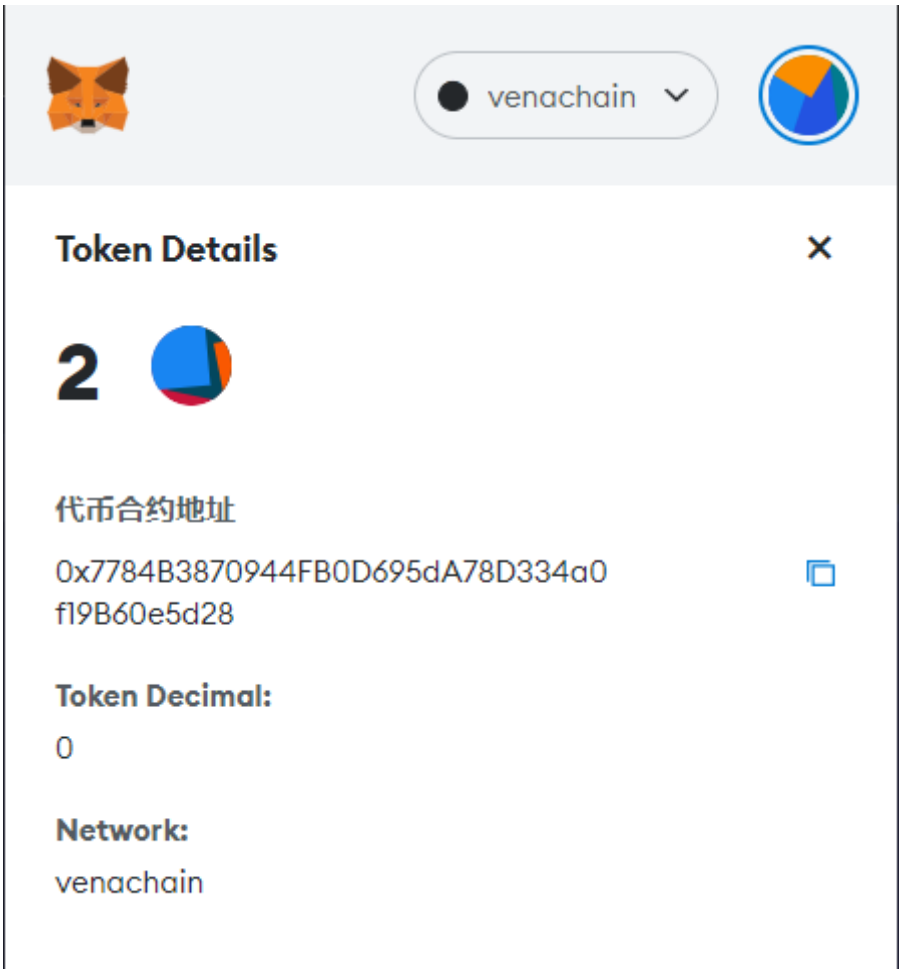
铸造成功 (我铸造了两次)

3.4 查询NFT token的URI



3.5 MetaMask中查看





4. 在浏览器中查看

4.1 看元数据

https://ipfs.io/ipfs/QmRRSbqZpEQ31Kw6mfH6oDt3ZxSjup38dUAJJUXq2csLzq

4.2 看虚拟人的资源

- 2D形象

  https://ipfs.io/ipfs/QmYadFKVgcVtuTGBA6KvuVcn4iP58ds5LkYGGiRWHwwmWm
- 出生证明 (STA颁发)

  https://ipfs.io/ipfs/QmefmjSguNLuwXNiYtp7DNENYWbsSd4i5rKtxJbNEx7LFQ
- 自我介绍的视频

  https://ipfs.io/ipfs/QmPAPbhFz5x2CMmNsq9XbguucTtV5vM2znDtn3x1rYyWh4