**FingerNFT部署手册**

2022.06

目录

[一、 整体架构 3](#_Toc109826005)

[二、 IPFS 4](#_Toc109826006)

[1.安装本地节点 4](#_Toc109826007)

[2. 配置pin服务 5](#_Toc109826008)

[三、 Mysql 8](#_Toc109826009)

[3.1 安装Mysql8.0 8](#_Toc109826010)

[3.2 导入初始化脚本 9](#_Toc109826011)

[四、 vue-web 9](#_Toc109826012)

[五、 vue-admin 9](#_Toc109826013)

[六、 fingernft-api 9](#_Toc109826014)

[七、 fingernft-admin 10](#_Toc109826015)

[7.1 部署admin 10](#_Toc109826016)

[7.2 Web配置更改 10](#_Toc109826017)

[7.3 配置更改重启服务 11](#_Toc109826018)

[八、 智能合约 11](#_Toc109826019)

[8.1 rinkeby测试ETH领取 11](#_Toc109826020)

[8.2 Metamask导入rinkeby WETH测试合约地址 11](#_Toc109826021)

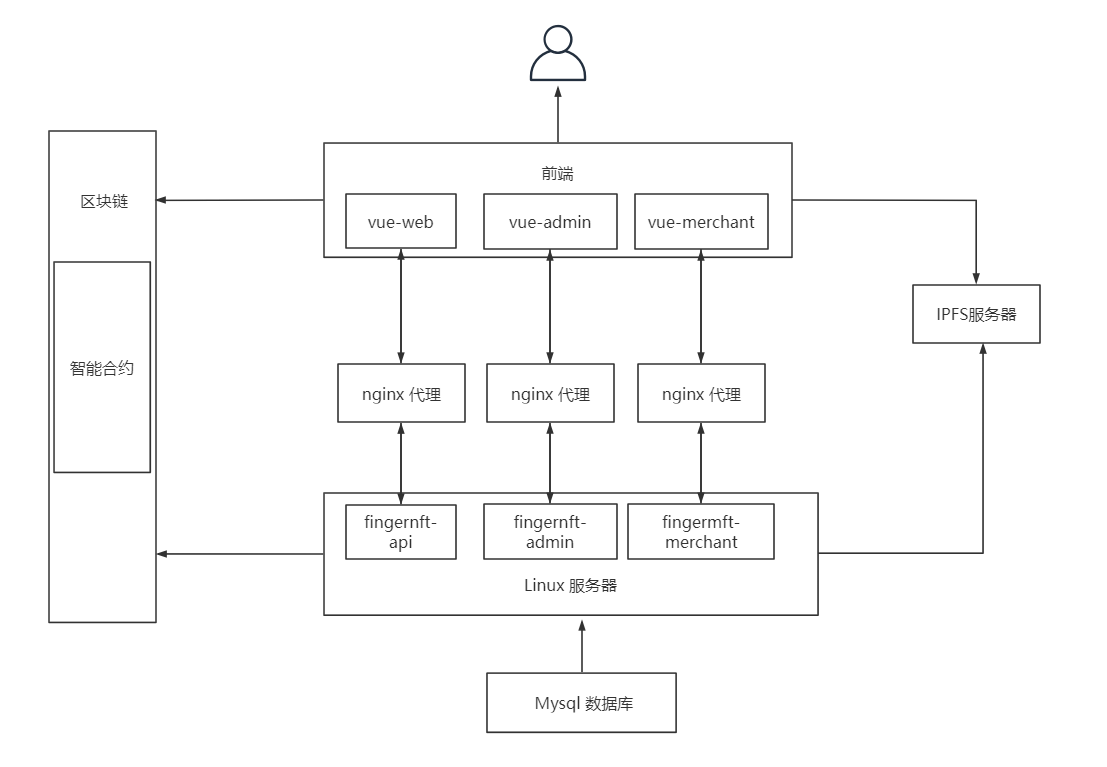
[8.3 rinkeby WETH测试币领取 12](#_Toc109826022)

[8.4 基础合约部署 13](#_Toc109826023)

[8.5 盲盒合约部署 17](#_Toc109826024)

## 整体架构

整个系统部署架构图如下所示：



需要以下7部分的部署：

IPFS: 去中心化的存储服务，需要搭建本地IPFS节点，并且购买IPFS remote Pin服务。

Mysql: 数据库采用Mysql8.0。

vue-web：FingerNFT市场，使用Vue3框架开发，需要使用npm build编译后部署。

vue-admin：FingerNFT市场后台管理，使用Vue3框架开发，需要使用npm build编译后部署。

fingernft-api：FingerNFT市场后端服务，由Java语言开发而成，使用Spring Boot框架，需要使用maven install进行构建部署。

fingernft-admin：FingerNFT市场后台管理后端服务，由Java语言开发而成，使用Spring Boot框架，需要使用maven install进行构建部署。

智能合约：包括市场、盲盒、拍卖合约，使用truffle编译之后进行部署。

## IPFS

### 1.安装本地节点

|  |
| --- |
| # 下载  wget https://dist.ipfs.io/go-ipfs/v0.12.2/go-ipfs\_v0.12.2\_linux-amd64.tar.gz  # 解压安装  tar -zxvf go-ipfs\_v0.12.2\_linux-amd64.tar.gz  cd go-ipfs  bash install.sh  # check  ipfs --version  # 初始化  ipfs init  # 显示配置文件内容  ipfs config show  # 编辑配置文件  export EDITOR=$(which vim)  ipfs config edit |

修改配置文件

|  |
| --- |
| "Addresses": {  "API": "/ip4/127.0.0.1/tcp/5001",  "Announce": [],  "AppendAnnounce": [],  "Gateway": "/ip4/127.0.0.1/tcp/8080",  "NoAnnounce": [],  "Swarm": [  "/ip4/0.0.0.0/tcp/4001",  "/ip6/::/tcp/4001",  "/ip4/0.0.0.0/udp/4001/quic",  "/ip6/::/udp/4001/quic"  ]  }, |

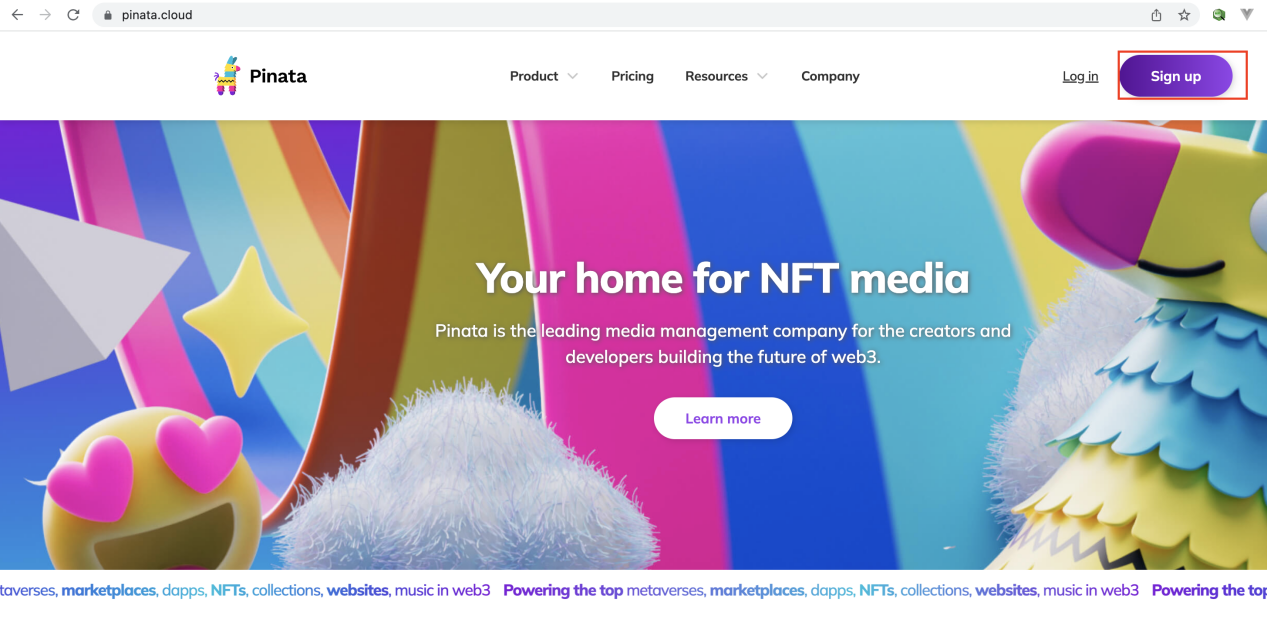
启动

|  |
| --- |
| # 配置跨域  ipfs config --json API.HTTPHeaders.Access-Control-Allow-Methods '["PUT","GET", "POST", "OPTIONS"]'  ipfs config --json API.HTTPHeaders.Access-Control-Allow-Origin '["\*"]'  ipfs config --json API.HTTPHeaders.Access-Control-Allow-Credentials '["true"]'  ipfs config --json API.HTTPHeaders.Access-Control-Allow-Headers '["Authorization"]'  ipfs config --json API.HTTPHeaders.Access-Control-Expose-Headers '["Location"]'  # 启动  nohup ipfs daemon &  # 检查  lsof -i :5001 |

### 配置pin服务

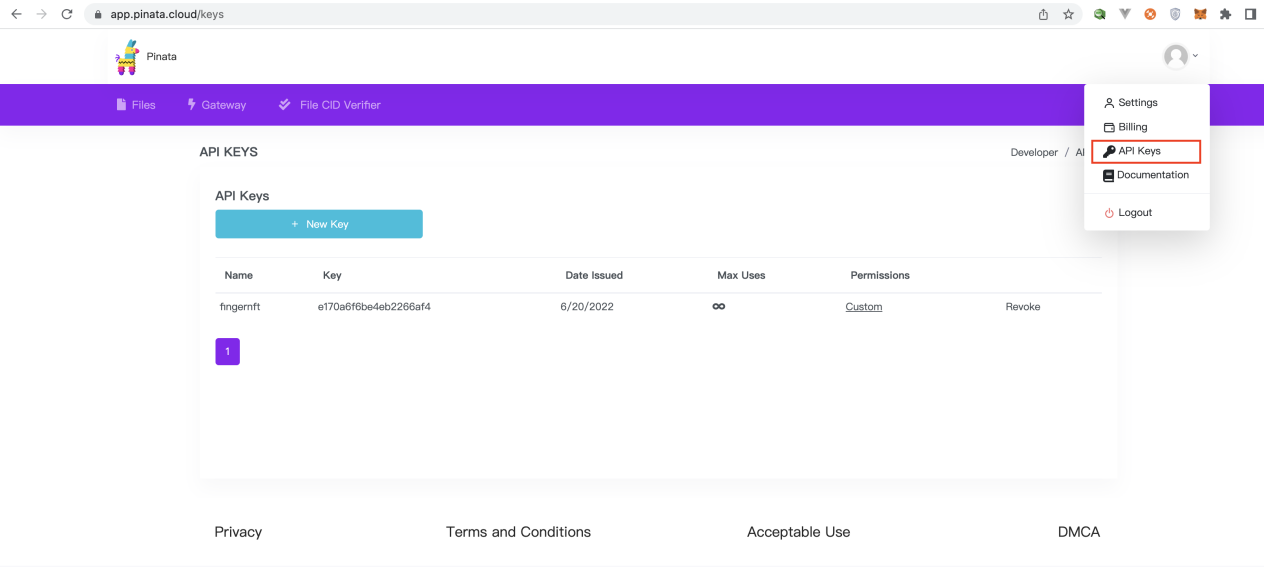
#### 2.1注册pinata

<https://www.pinata.cloud/>

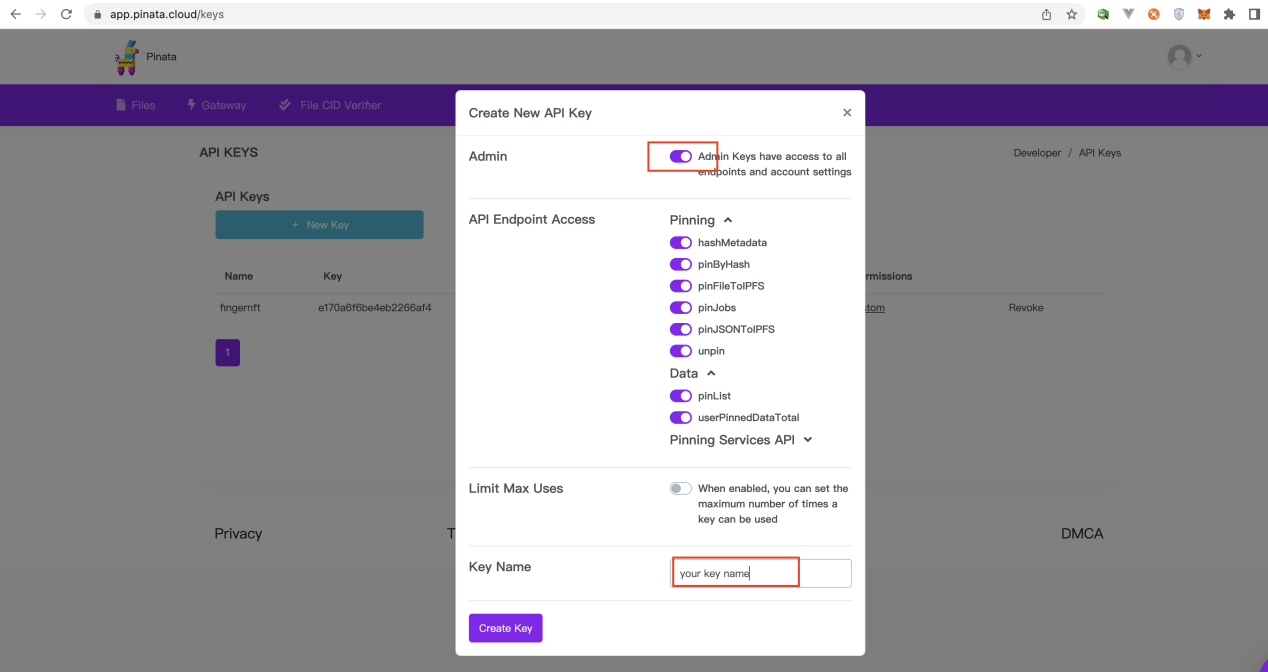


#### 2.2 登录pinata获取

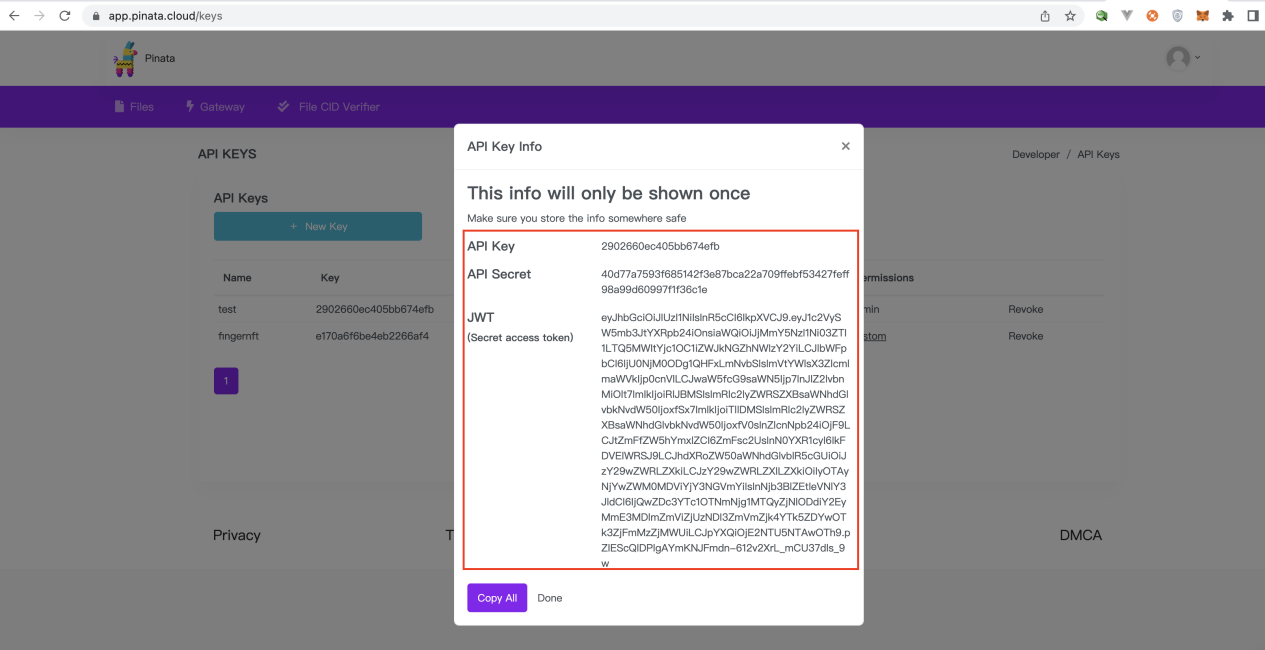
点击头像，下拉菜单中点击API Keys



点击 New Key 按钮



保存key信息



#### 2.3 配置pin服务

添加名为“**pinata**”的remote pin服务

|  |
| --- |
| # 添加pin service  ipfs pin remote service add **pinata** https://api.pinata.cloud/psa YOUR\_JWT  # 显示所有pin service  ipfs pin remote service ls  # 测试，pin一个文件  ipfs pin remote add --service=pinata --name=war-and-peace.txt bafybeib32tuqzs2wrc52rdt56cz73sqe3qu2deqdudssspnu4gbezmhig4  # 显示所有pinned文件  ipfs pin remote ls --service=pinata --status=pinned |

## Mysql

### 3.1 安装Mysql8.0

首先需要在部署的主机上提前正确安装好Mysql8.0, 并将Mysql密码加密方式改为旧的加密方式。

|  |
| --- |
| 1. 连接数据库  [root@localhost ~]# mysql -uroot -p  2. 选择mysql数据库  mysql> use mysql  3. 设置密码加密方式  mysql> alter user 'root'@'%' identified with mysql\_native\_password by 'root'; |

### 3.2 导入初始化脚本

初始化脚本在/fingernft\_blindbox/sql/fingernft.sql。

## vue-web

在vue-web运行npm install

安装完依赖后运行npm run build

将dist下所有文件传输到服务器的/work/admin/nginx/html/nft-pc下

## vue-admin

在vue-admin运行npm install

安装完依赖后运行npm run build

将dist下所有文件传输到服务器的/work/admin/nginx/html/nft-admin下

## fingernft-api

安装配置好JDK和maven

在fingerntp-mapi下运行mvn package命令

将fingerntp-api/target/fingerntp-api-1.0.0-exec.jar传输到服务器的/data/java/web/lib下

在服务器的/data/java/web下运行 ./restart.sh

## fingernft-admin

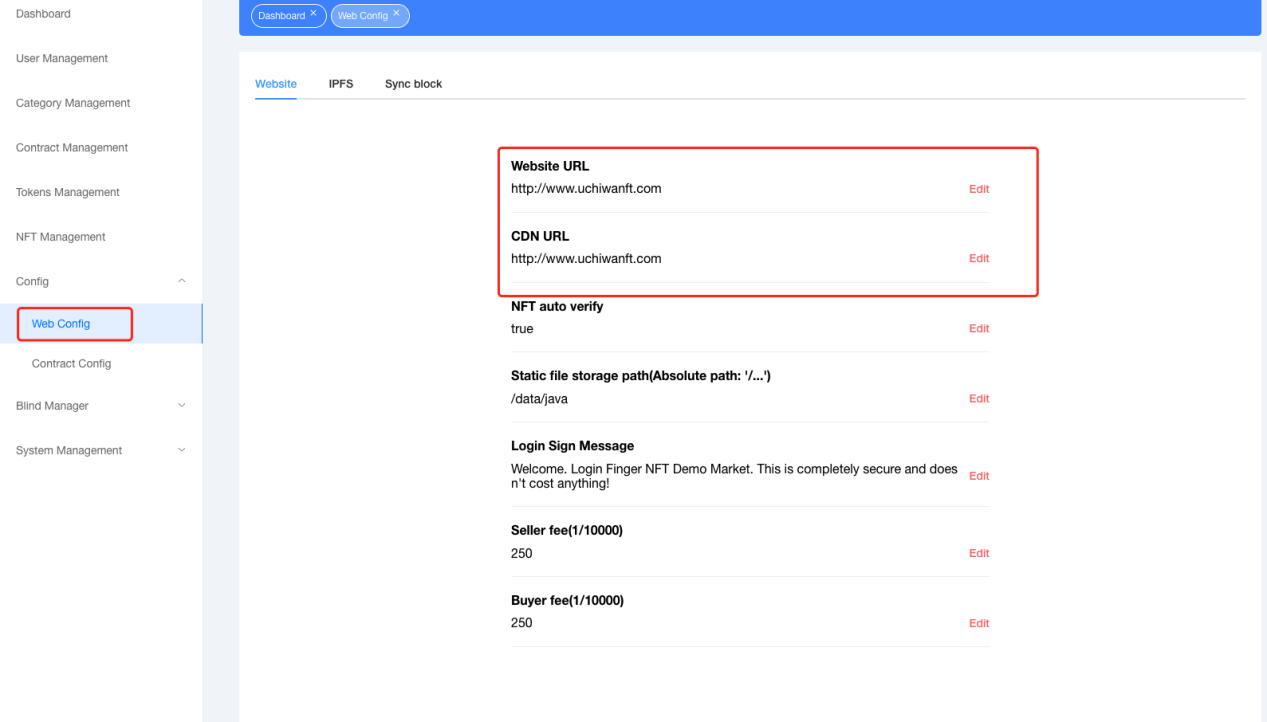
### 7.1 部署admin

将fingerntp-admin/target/fingerntp-admin-1.0.0-exec.jar传输到服务器的/data/java/admin/lib下

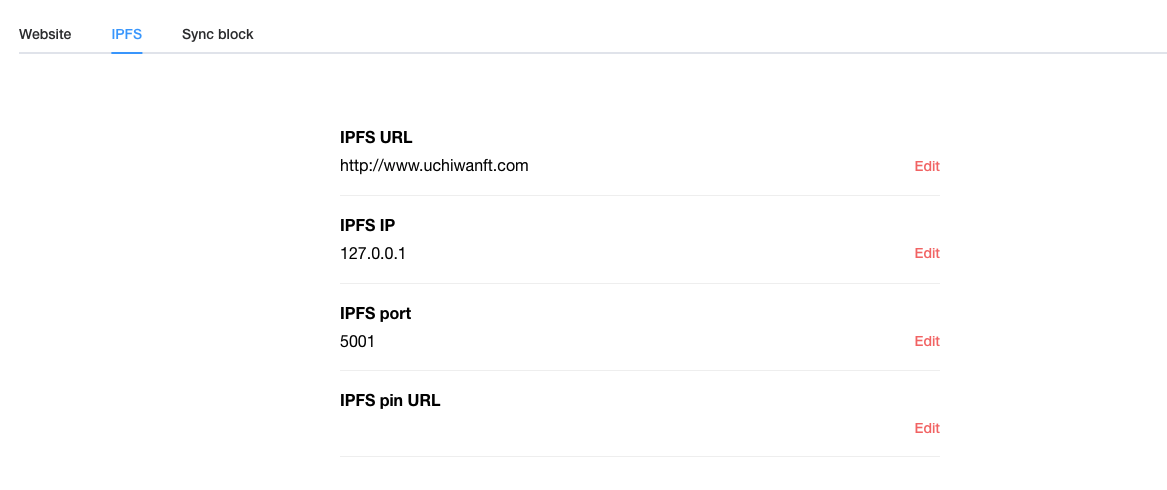
在服务器的/data/java/admin下运行 ./restart.sh

### 7.2 Web配置更改

website配置可以更改网站域名，cdn地址，手续费等



IPFS配置可以更改IPFS访问的地址（图片地址），IPFS IP和端口；如有远程pin服务，IPFS pin URL则需要填写pin服务名称（IPFS服务启动时候指定）



### 7.3 配置更改重启服务

***Web Config更改后必须重启fingernft-admin和fingernft-api服务***

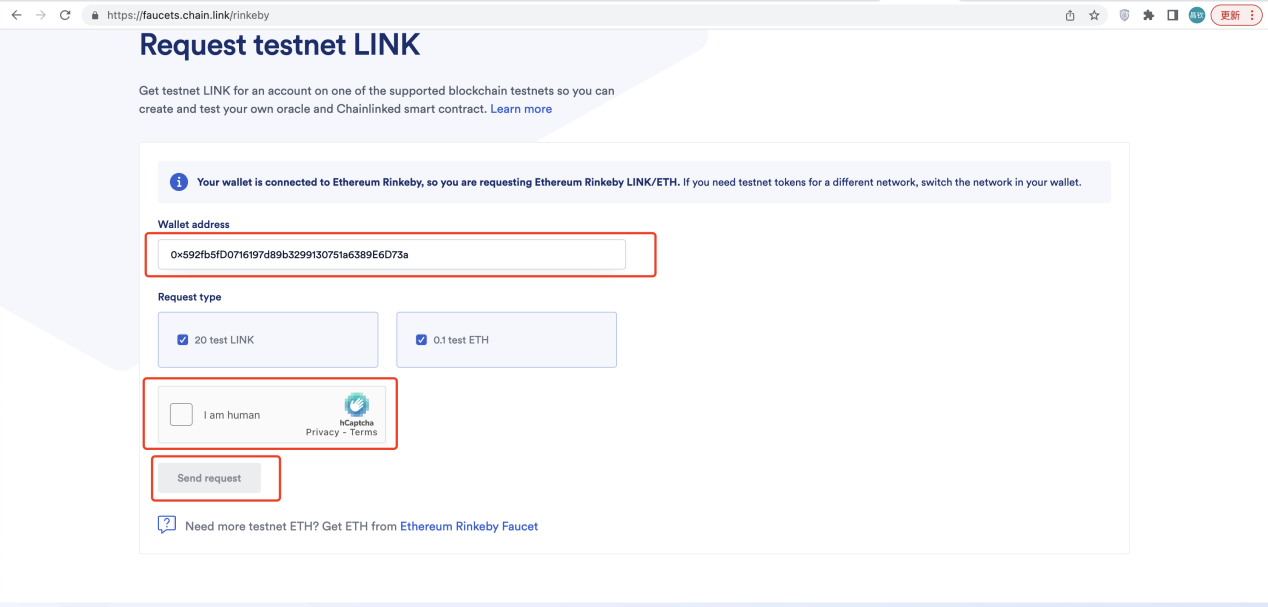
## 智能合约

### 8.1 rinkeby测试ETH领取

首先使用浏览器打开下面rinkeby ETH测试币领取地址。

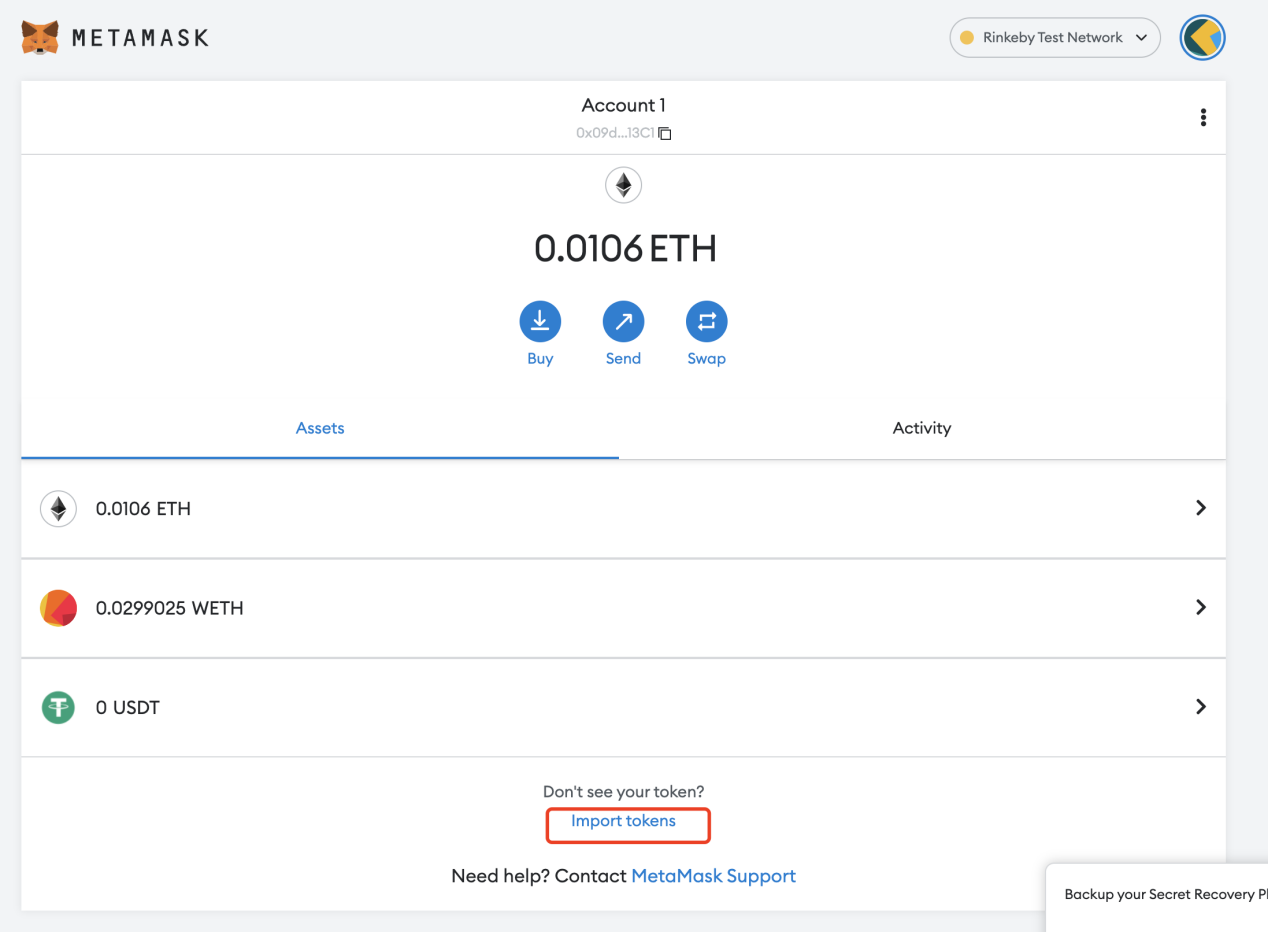
|  |
| --- |
| https://faucets.chain.link/rinkeby |

输入钱包的地址，进行验证码验证之后，点击“Send request”进行测试币领取。



### 8.2 Metamask导入rinkeby WETH测试合约地址

因为默认Metamask默认Swap中WETH测试合约地址是错误的，如果想要兑换测试网WETH代币，需要手动导入WETH测试合约地址，点击“Import Tokens”，如下图所示：



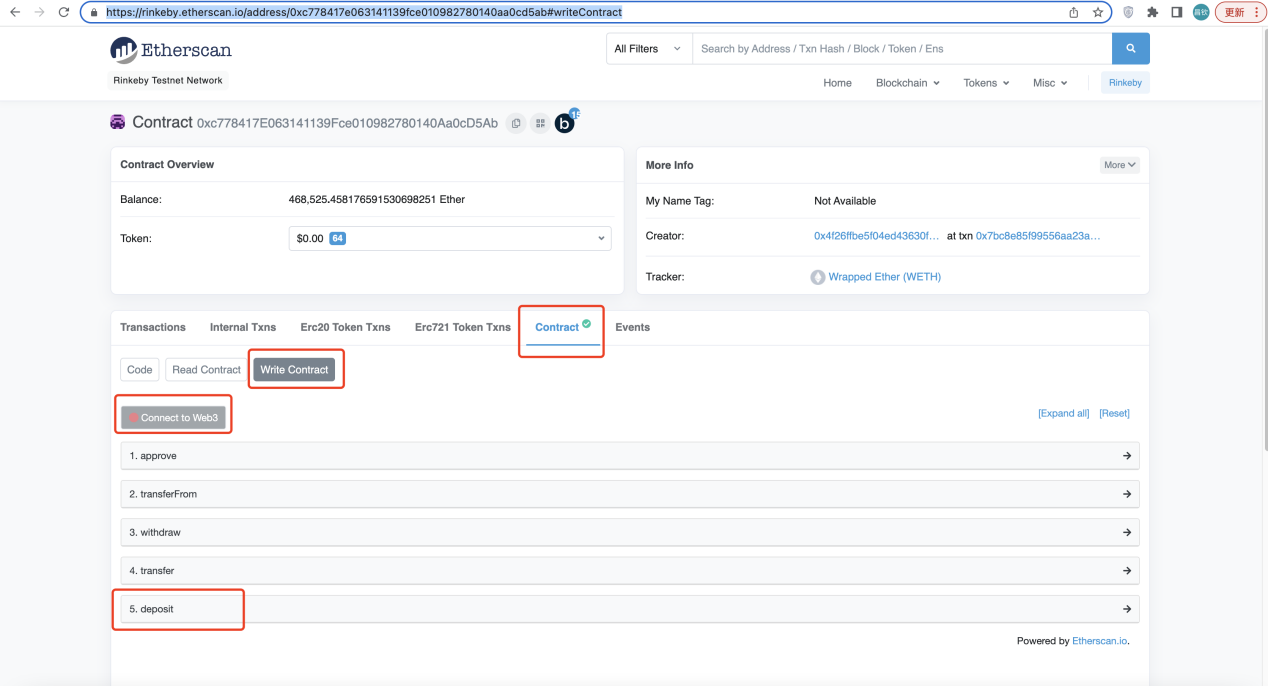
输入合约地址0xc778417e063141139fce010982780140aa0cd5ab，点击“Add Custom Token”,就可以看到WETH的余额。

### 8.3 rinkeby WETH测试币领取

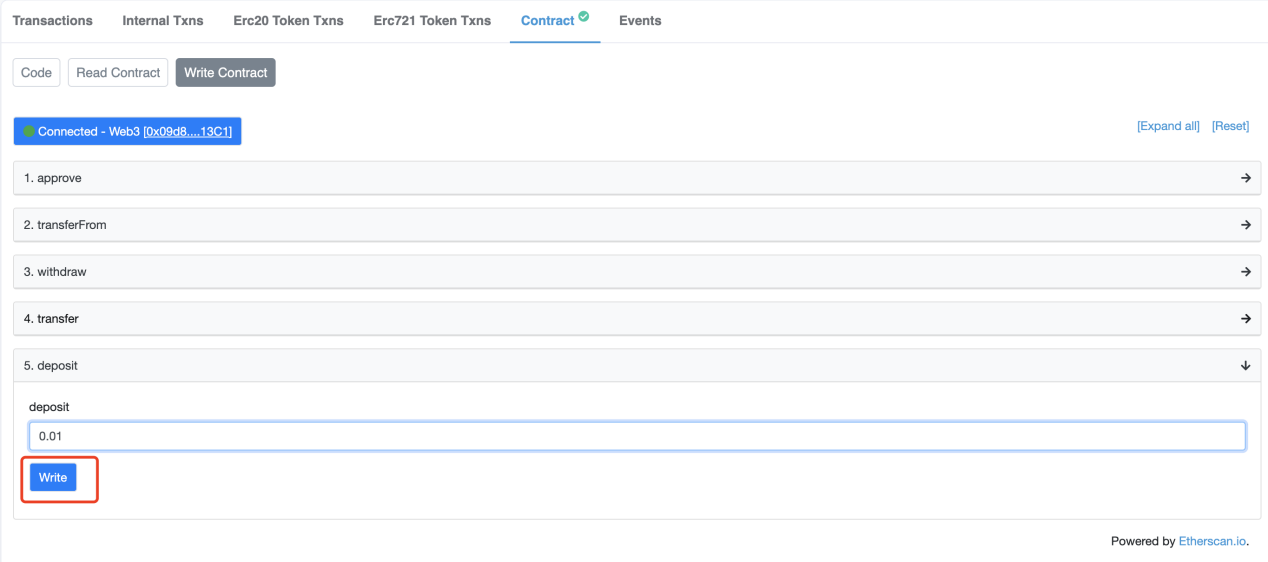
首先使用浏览器打开下面WETH测试网合约地址。

|  |
| --- |
| https://rinkeby.etherscan.io/address/0xc778417e063141139fce010982780140aa0cd5ab#writeContract |

点击“Contract”tab页，选择“Write Contract”之后，连接metamask钱包，然后执行“deposit”操作，如下图所示：



输入需要兑换的数量之后，点击“Write”，钱包确认交易之后就可以兑换WETH代币。



### 8.4 基础合约部署

#### 8.4.1 区块链网络信息

##### Rinkeby测试网

|  |
| --- |
| apiUrl: <https://rinkeby.infura.io/v3/9aa3d95b3bc440fa88ea12eaa4456161>  chainId: 4  explorer: https://rinkeby.etherscan.io |

##### 以太坊主网

apiUrl需要去<https://infura.io/>申请

|  |
| --- |
| apiUrl: https://mainnet.infura.io/v3/  chainId: 1  explorer: https://etherscan.io |

#### 8.4.2 合约部署

1. **安装nodejs和npm**

2. **进入truffle-contract目录，通过npm安装truffle**

|  |
| --- |
| **$ cd truffle-contract**  **$ npm install truffle** |

3. 进入**truffle-contract/scripts目录，编辑config.js，填写部署信息**

|  |
| --- |
| privateKey:**部署合约的钱包地址私钥，不能和下面的miner是同一个钱包地址**  apiUrl: **区块链rpc地址参考8.4.1节**  chainId: **区块链id地址参考8.4.1节**  NFTName: "FingerNFT"  BoxNFTName: "FingerBoxNFT"  miner: **授权铸造NFT的钱包地址，不能跟部署合约的账号相同，用于铸造NFT时，给予用户授权的，未设置无法铸造NFT**  beneficiary: **收取手续费的钱包地址**  buyerFeeSigner: **用于在交易时，给予用户签名（签名中附带手续费，用户拥有签名才能完成交易）的钱包地址** |

4. 编译合约。

|  |
| --- |
| **$ cd truffle-contract**  **$ truffle compile** |

5. 执行部署脚本，部署合约。

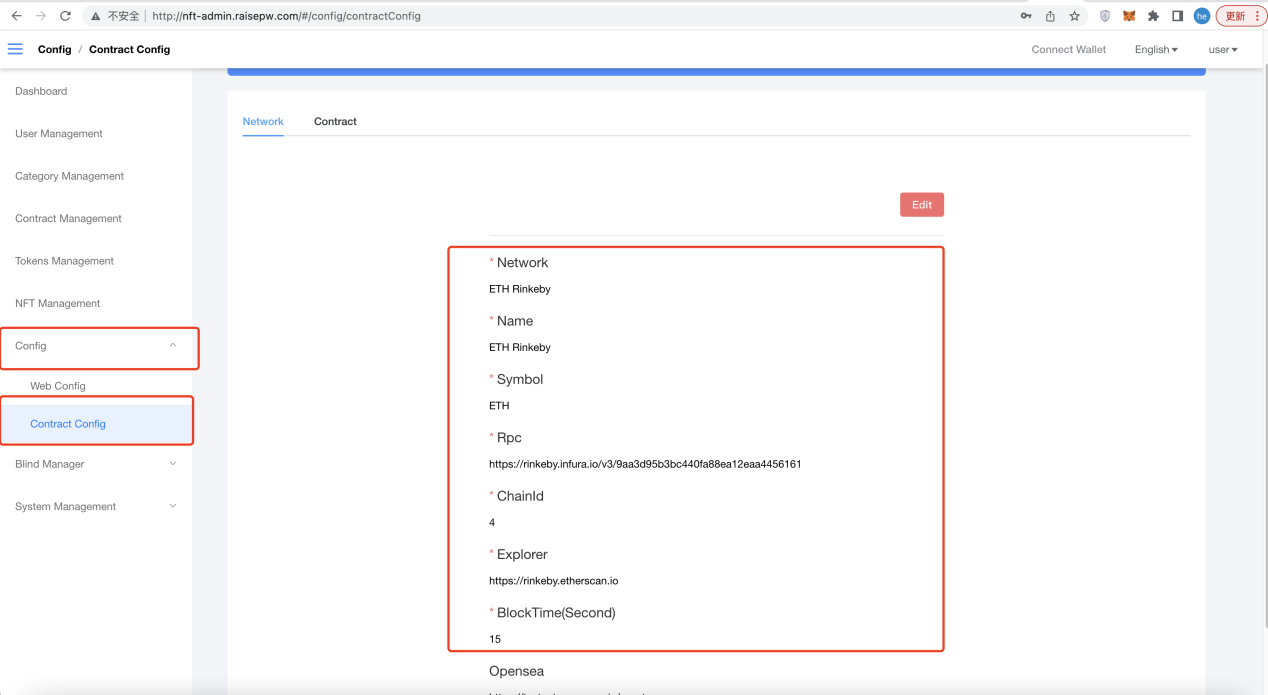
|  |
| --- |
| **$ cd truffle-contract/scripts**  **$ node deploy.js** |

部署成功之后会得到每个基础合约的地址，如下所示：

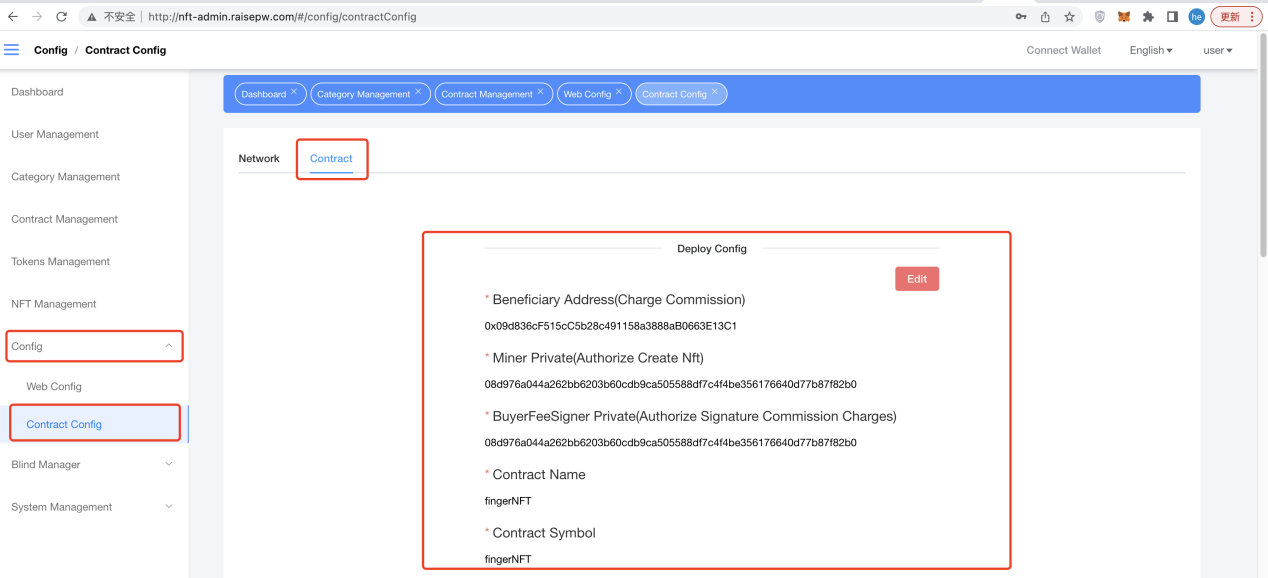
|  |
| --- |
| {  "NFT721": {  "address": "0x761A263e20eB61De1B7fd4e91c9c77dBED8aC5b5"  },  "NFT1155": {  "address": "0x5fb6F0b04FD12E39DF92E6c22d08420D19C6E397"  },  "transferProxy": {  "address": "0x63eA1078f81bB4DFaCd388918Bb72bE43CeA62c6"  },  "transferProxyForDeprecated": {  "address": "0x6F72Ac5323c4B148d2F00ddbb8Fa8f65e9607dd0"  },  "erc20TransferProxy": {  "address": "0x4ADf8aEc54A359E12e9657c3f219Ed2667086494"  },  "exchangeState": {  "address": "0x58b261d667AA7671AA5B5BB19fD9310877C272C7"  },  "exchangeOrderHolder": {  "address": "0xAd308fD89344a04399F0f8c332c0f29fdC12Ee4e"  },  "exchange": {  "address": "0xBC7De12e47c31f18aAAA62d6Df0b305616280F38"  },  "blindState": {  "address": "0x16351d8aa4044E172be109141f6c9F06bef4D040"  },  "copyERC721": {  "address": "0xB625A576d1318109CA8ca623b3cE73bc689DB4c1"  },  "blindBox": {  "address": "0xe7a4c56bC4ae05e3f51482F1571f6a5aaF4c0bb6"  },  "auctionState": {  "address": "0x7125a61403c4fb5C3d6149E85602CD0605E8Dd5A"  },  "auctionExchange": {  "address": "0xE10142eA27E73C6b7A3a264D84578bD5BdAafbdd"  },  "multicall": {  "address": "0x58B6925B654085360FF1f6355E8FddC6311551D5"  }} |

#### 8.4.3 NFT Admin区块链合约配置

1. 登录NFT Admin管理控制台，点击**Config – Contract Config**菜单，并且选择**Network**页。按照**8.4.1节**填写区块链网络相关信息，如下图所示。

****

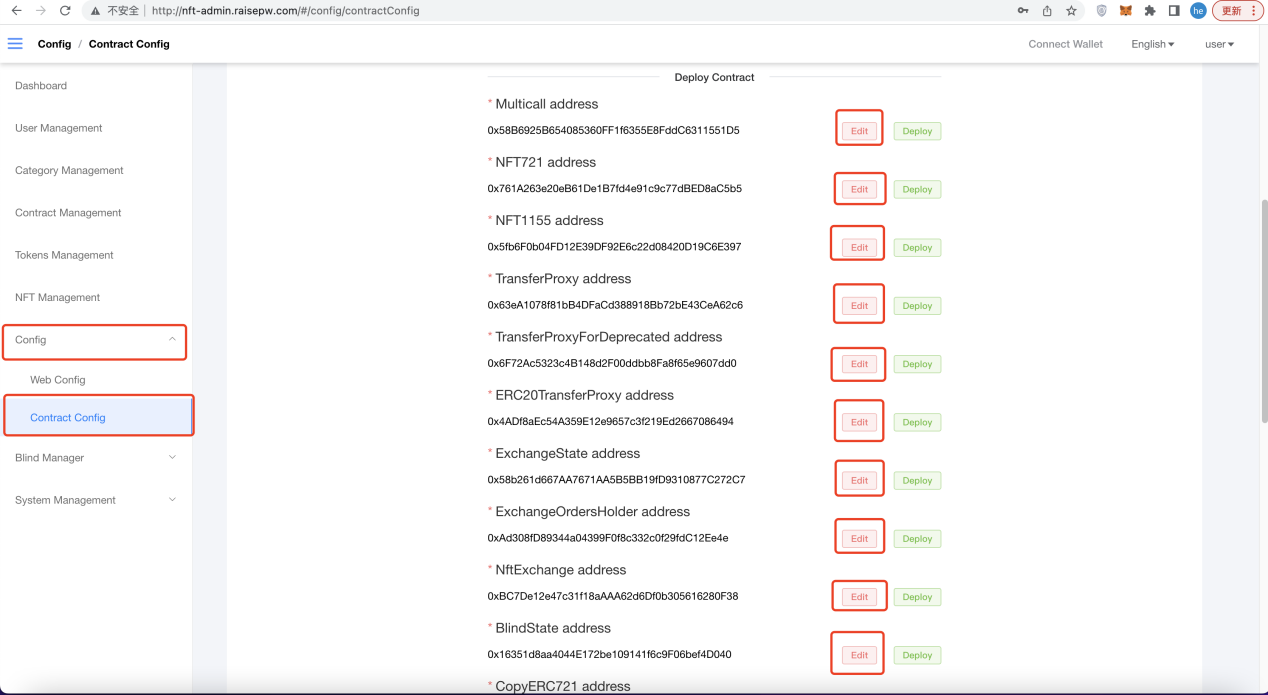
2. 登录NFT Admin管理控制台，点击**Config – Contract Config**菜单，并且选择**Contract**页。首先填写**Deploy Config**，如下图所示。



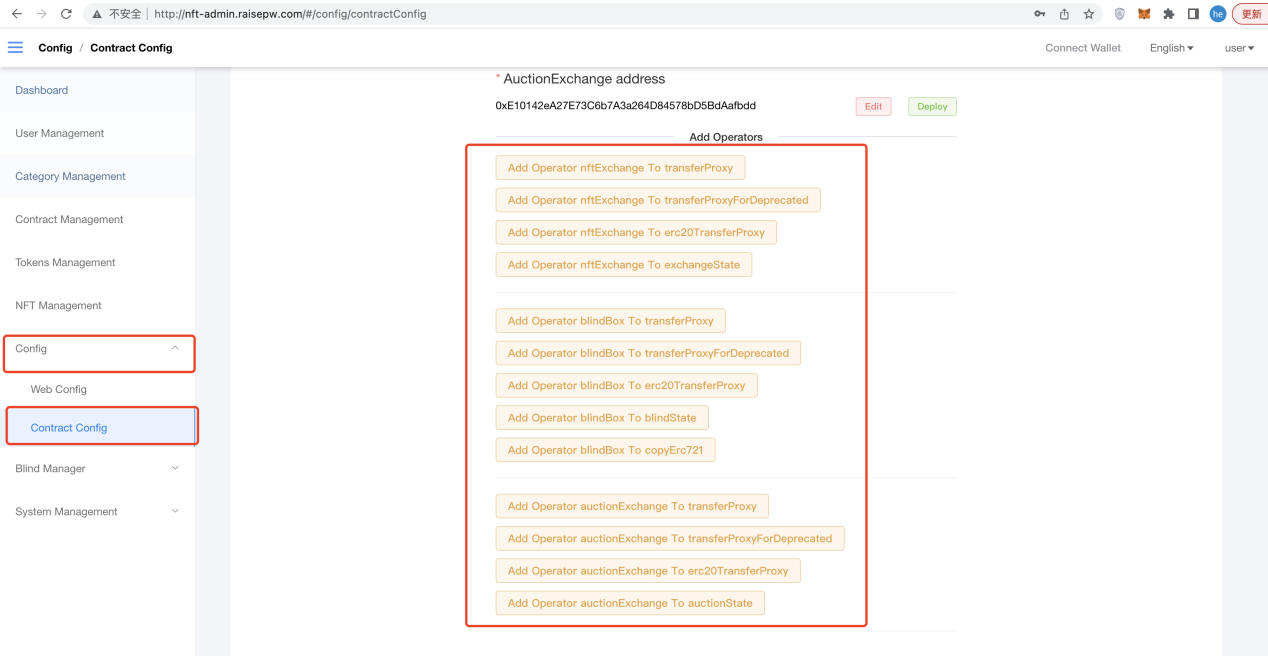
需要填写的信息如下：

|  |
| --- |
| Beneficiary Address：**config.js中的beneficiary**  Miner Private: **config.js中的miner的私钥**  BuyerFeeSigner Private: **config.js中的buyerFeeSigner的私钥** |

3. 然后按照上面部署之后的合约地址分别填写**Deploy Contract**的信息，如下图所示：



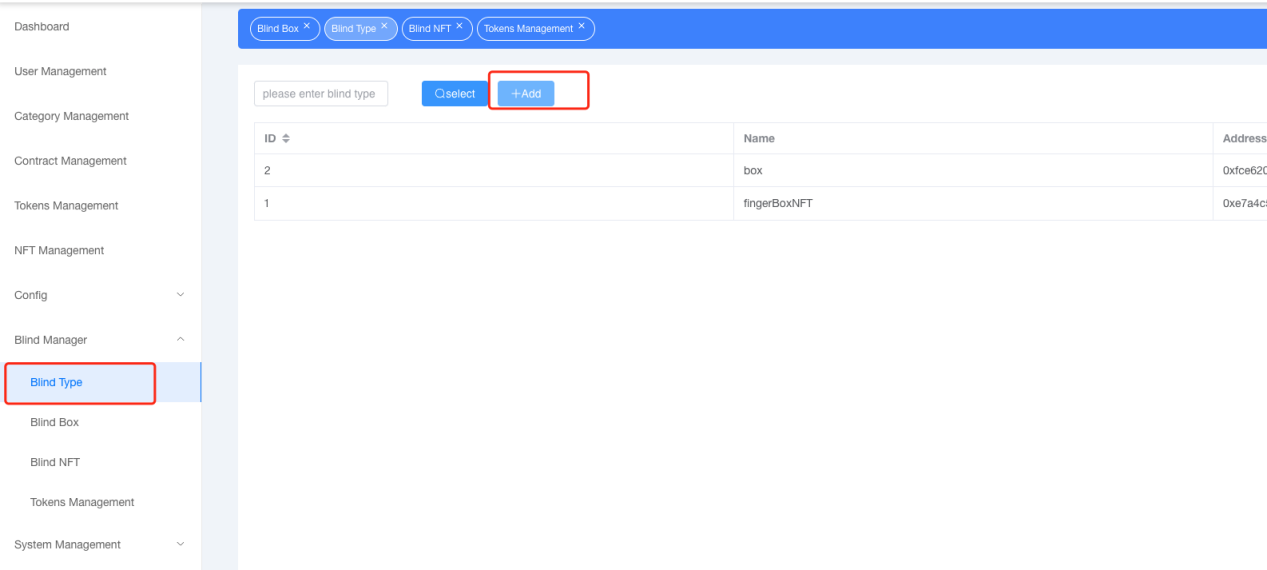
4. 填写完合约地址之后，请分别为每个合约调用**Add Operators**，如下图所示：



### 8.5 盲盒合约部署

1. 创建盲盒类型

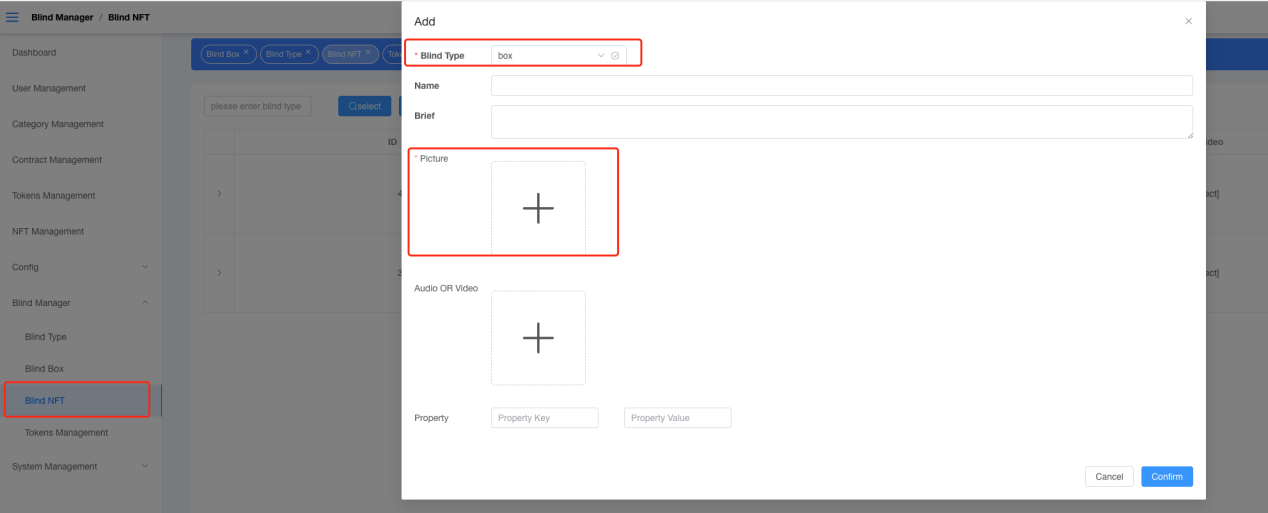
点击创建盲盒类型



连接钱包后，点击deploy进行部署，部署成功后点击confirm确定添加

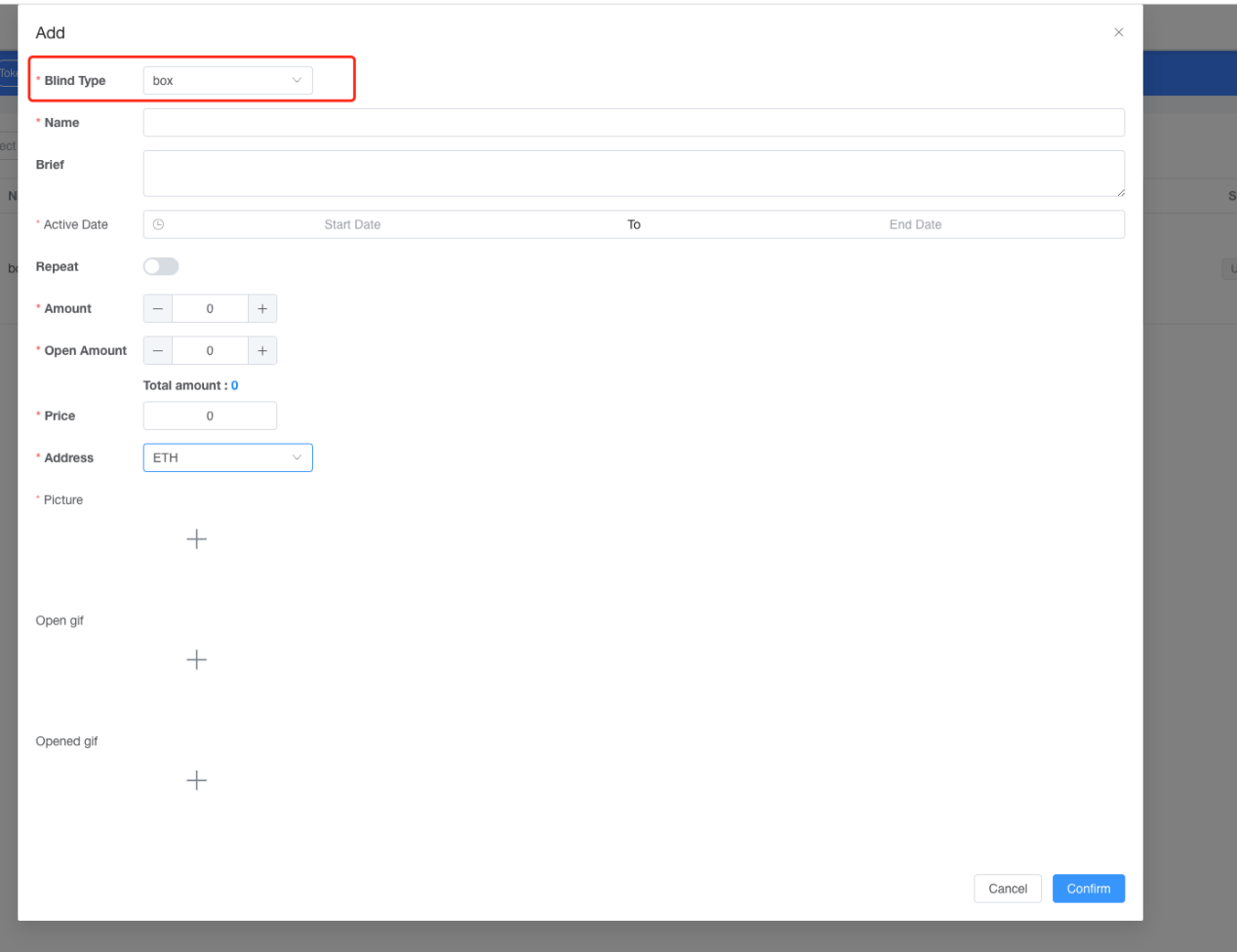
1. 创建盲盒NFT

选择上一步中创建的盲盒类型，并填写NFT相关信息，上传对应图片

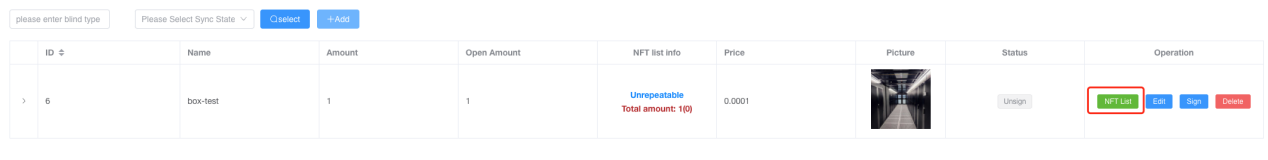


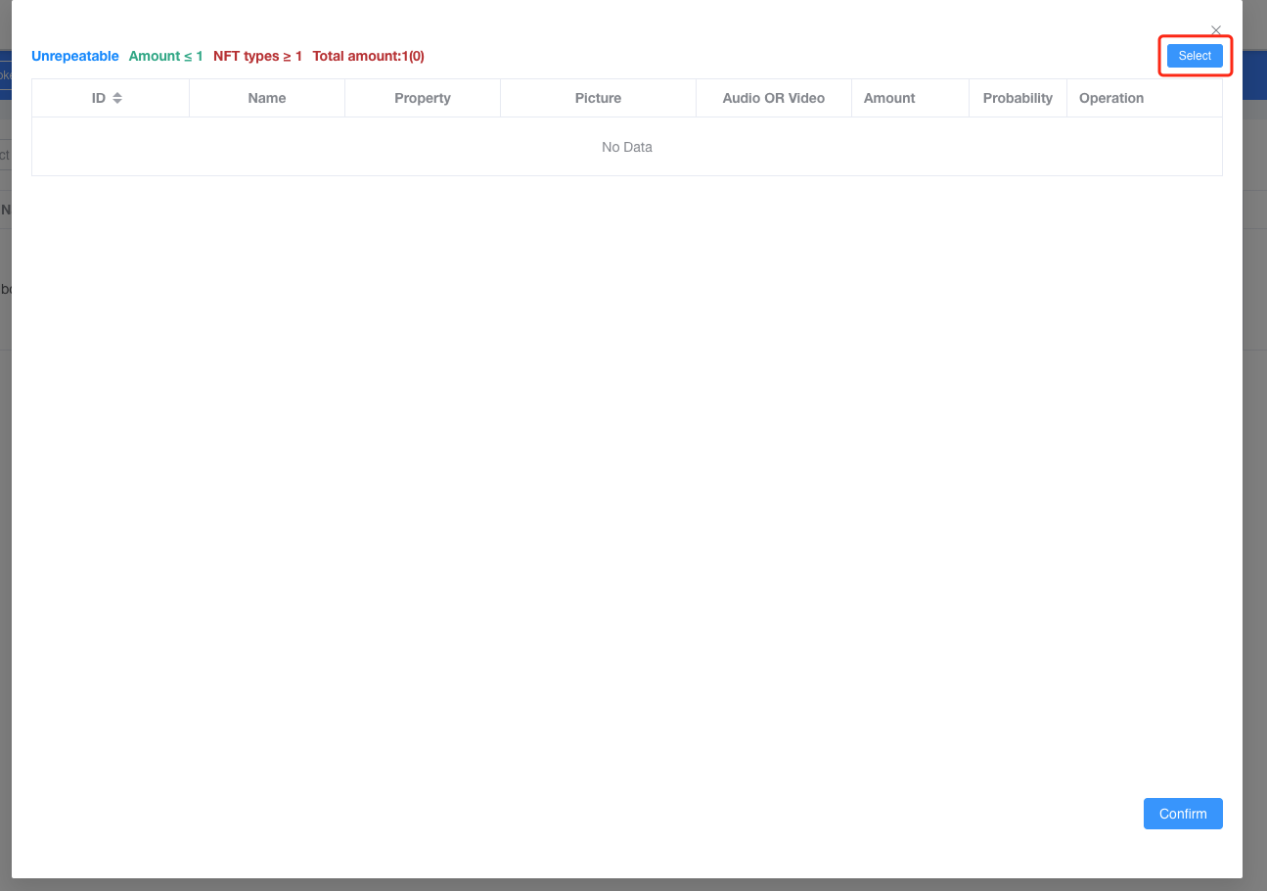
1. 创建盲盒

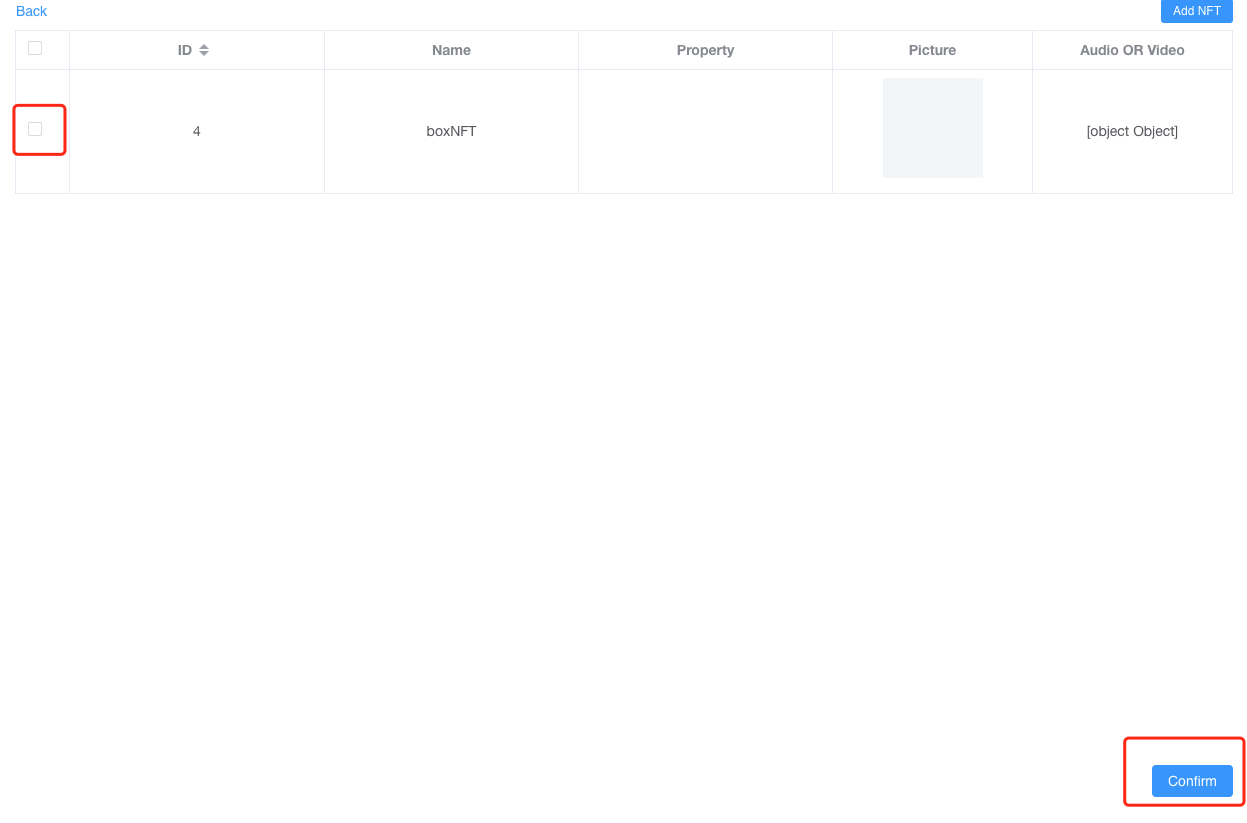
同样选择创建的盲盒类型，并填写所有必填信息



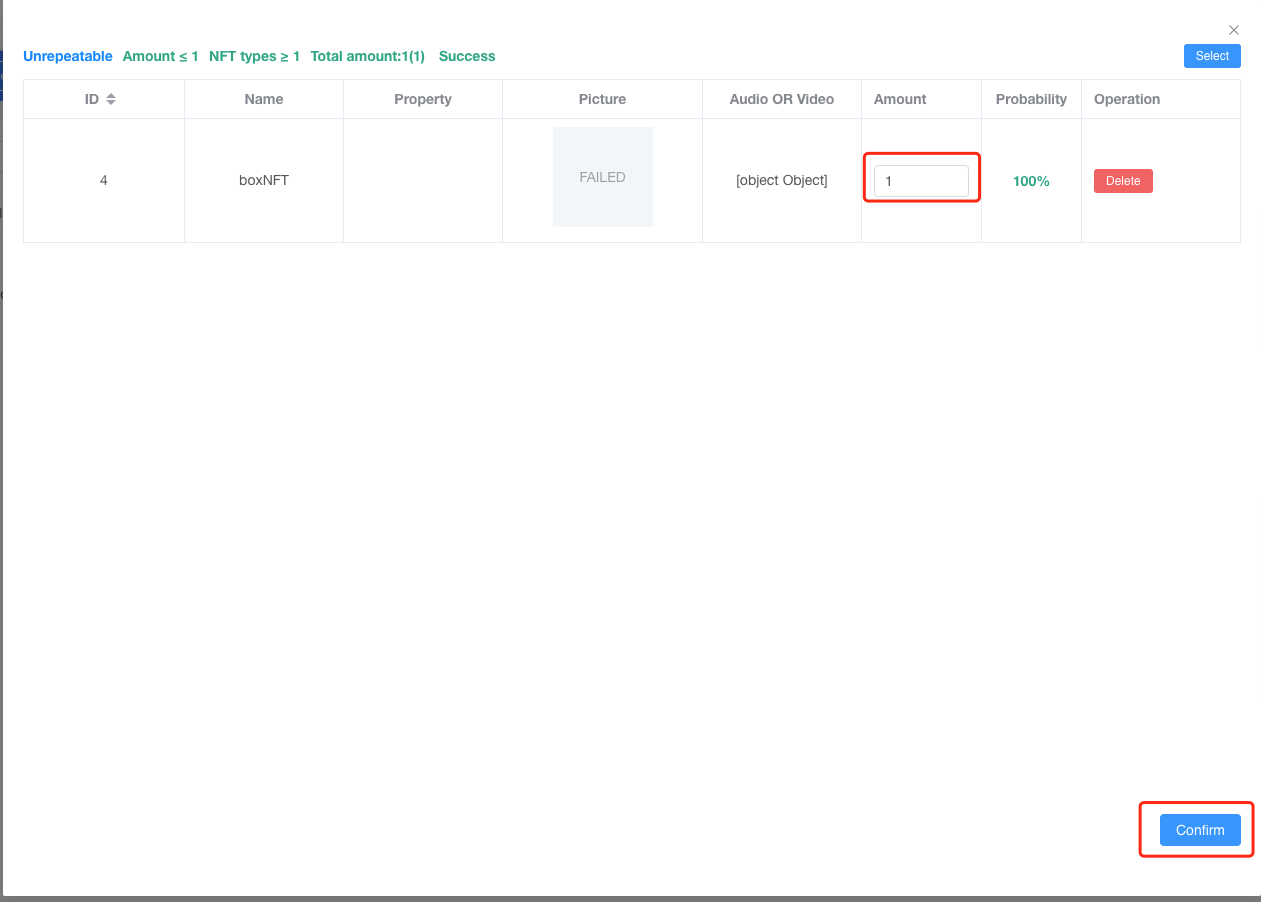
1. 选择NFT列表签名发布







配置NFT列表并确定



签名发布

