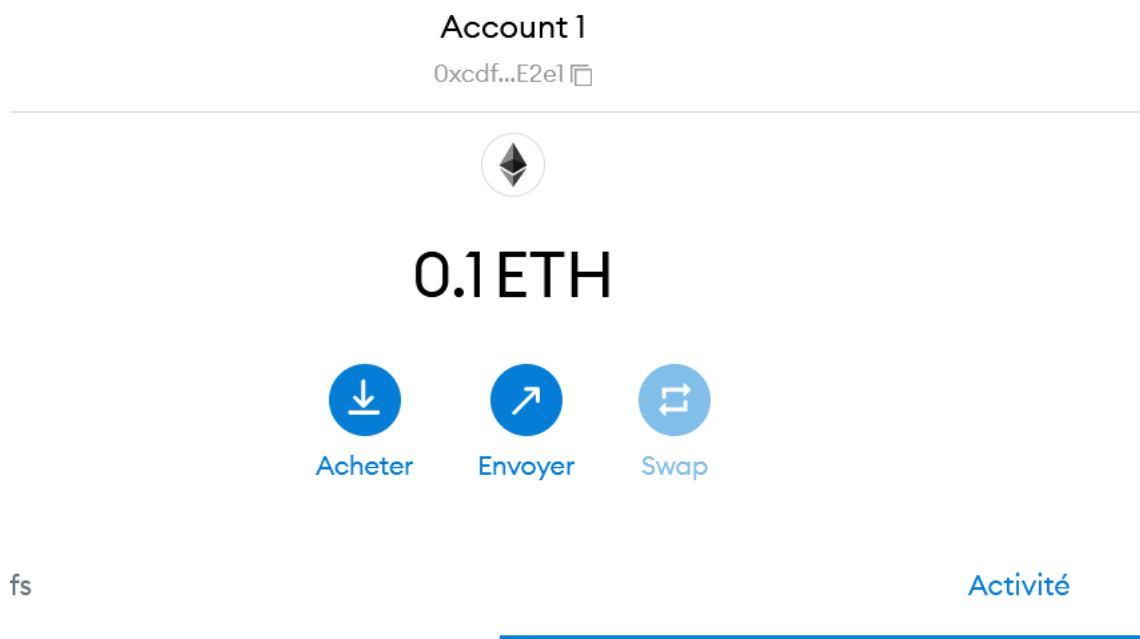
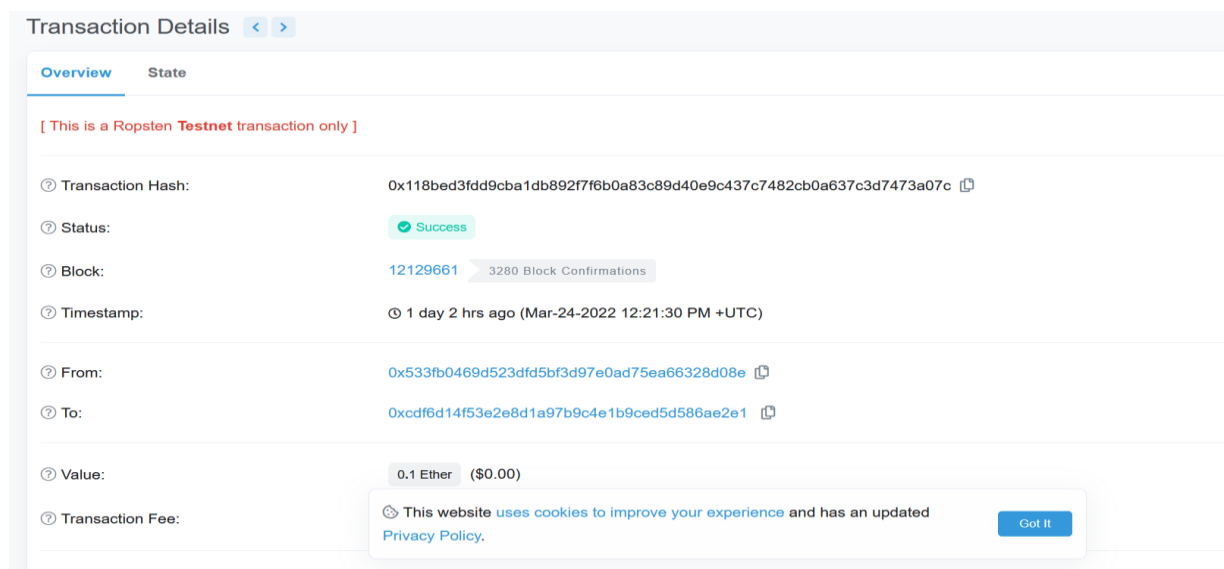


## TP BLOCKCHAIN

1. Prise en main des outils Remix et Metamask
  - b. Génération du portefeuille :



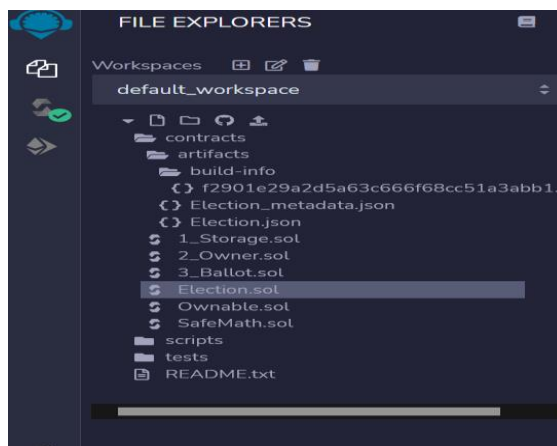
- e. Consulter la transaction générée vers votre compte et prenez en compte les détails de cette dernière. Fournissez également les détails de la transaction



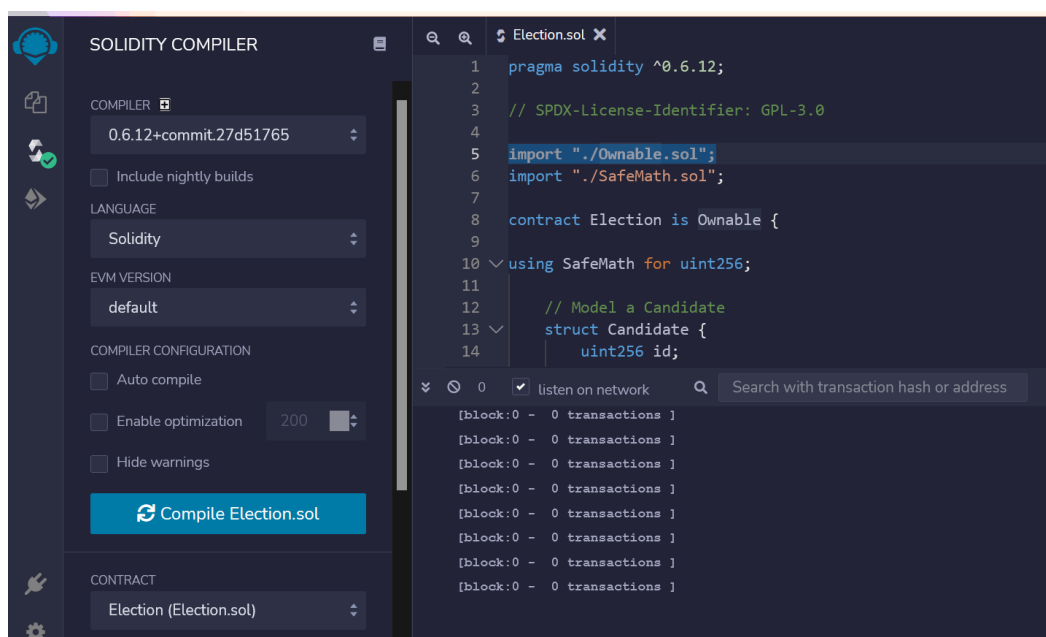
- h. Ouvrir l'IDE Remix

Mariama KANDE

## J'ai exporté les contacts sur Remix



K. Compiler votre smart contract « Election » et fournissez l'ABI ainsi que le Byte code du contrat.



Sur mon compte git j'ai créé un repo GitHub sur lequel j'ai importé mon fichier ABI.



Mariama KANDE

[Mariama17 / blockchain](#) Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

About

TP

Readme

0 stars

1 watching

0 forks

Releases

No releases published

3f44e4d yesterday 2 commits

ABI.txt ABI yesterday

README.md Initial commit yesterday

README.md

blockchain

I. Déployer le smart contract « Election.sol » sur le réseau Ropsten et fournissez les détails de la transaction

DEPLOY & RUN TRANSACTIONS

ACCOUNT 0x5B3...eddC4 (99.99999)

GAS LIMIT 3000000

VALUE 0 Wei

CONTRACT Election - contracts/Election.sol

Deploy

Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded

Election.sol

```
1 pragma solidity ^0.6.12;
2
3 // SPDX-License-Identifier: GPL-3.0
4
```

ContractDefinition Election 1 reference(s)

listen on network

[vm] from: 0x5B3...eddC4 to: Election.(constructor) value: 0 wei data: 0x608...e0033 logs: 0

hash: 0xb62...07705

status true Transaction mined and execution succeed

transaction hash 0xb622e1c0f2f9597d4e6105db349ff04132fda7e442b9c79c9732595d7cd07705

from 0x5B38Da6a701c568545dcFcb03Fcb875f56beddC4

to Election.(constructor)

gas 80000000 gas

transaction cost 554506 gas

execution cost 554506 gas

input 0x608...e0033

decoded input ()

decoded output -

build-info

{ Election\_metadata.json

{ Election.json

m. Justifier les frais de transactions « Transaction fees » que vous avez payés. Sont-elles identiques à celle de ma transaction ?

transaction cost 554506 gas

execution cost 554506 gas

**La transaction correspond bien à votre transaction**

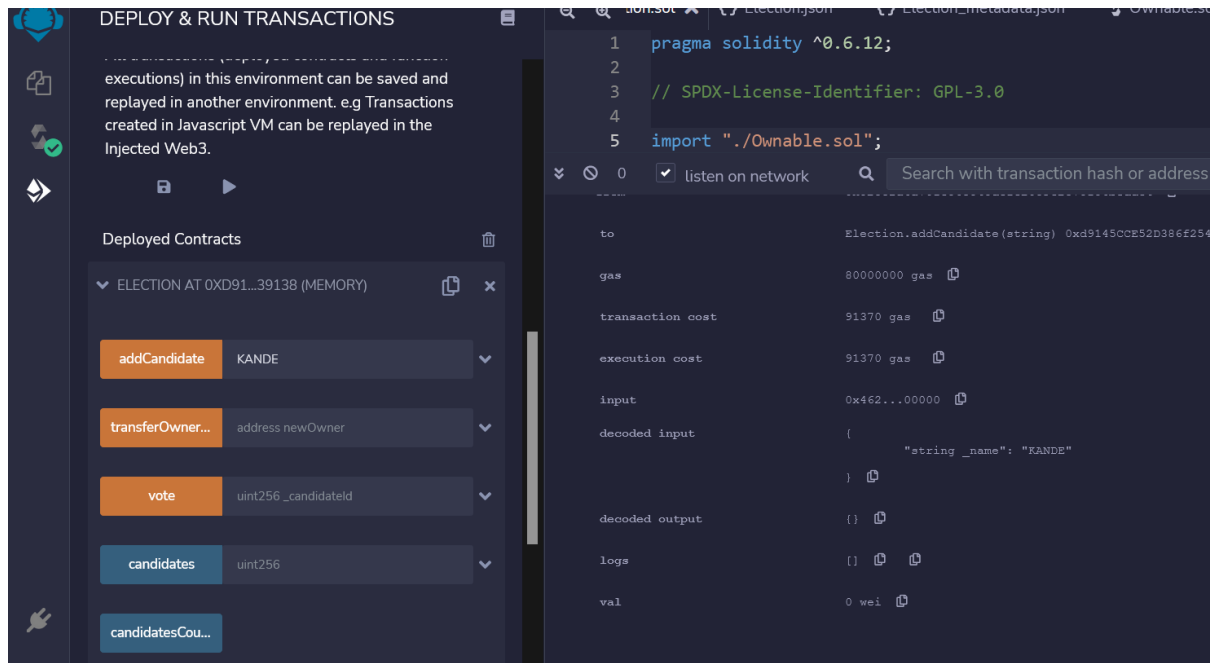
Quelle est l'adresse public de votre smart contract :

from 0x5B38Da6a701c568545dcFcb03Fcb875f56beddC4

Mariama KANDE

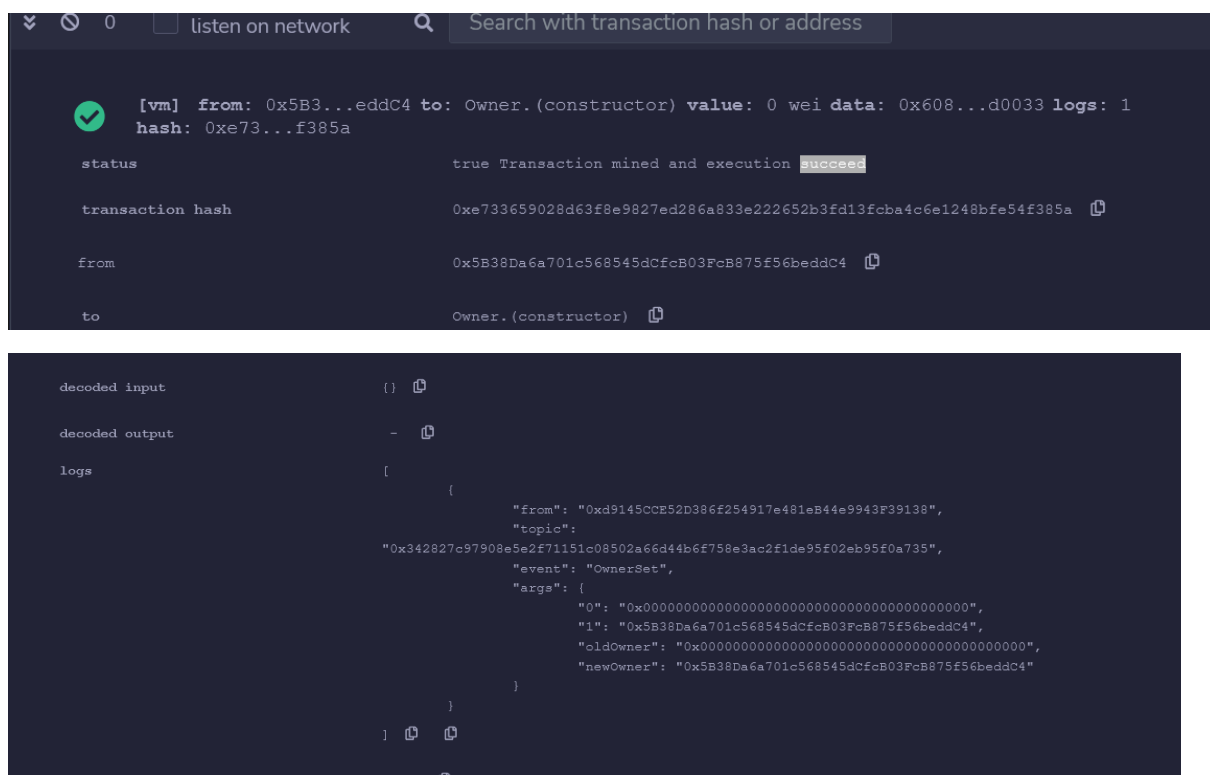
**0x5B38Da6a701c568545dCfcB03FcB875f56beddC4**

n. Interagissez avec votre smart contract après l'avoir déployé en ajoutant le nom du premier candidat qui sera votre « Nom de famille »



The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is visible, showing a list of deployed contracts. The 'ELECTION AT 0XD91...39138 (MEMORY)' contract is selected. Below the list, there are buttons for 'addCandidate', 'transferOwner...', 'vote', 'candidates', and 'candidatesCou...'. The 'addCandidate' button is highlighted with the input 'KANDE'. On the right, the 'console' panel shows the execution of the 'addCandidate' function. The input is 'KANDE' and the output is 'string \_name: "KANDE"'. The console also shows the transaction cost and execution cost.

O. Générer la transaction ensuite l'ajout du premier candidat et fournissez les détails de la transaction



The screenshot shows the Remix IDE console. The top part of the console shows the transaction details for the 'addCandidate' function. The transaction is successful, with the status 'true Transaction mined and execution succeed'. The console also shows the decoded input and output. The decoded input is an empty array, and the decoded output is an empty array. The logs show the transaction details, including the 'from' address, the 'to' address, and the 'value'.

Mariama KANDE

**Avec le déploiement je suis maintenant propriétaire (newOwner 0x5) du contact.**

**Cela s'explique grâce à ce code ci-dessous :**

```

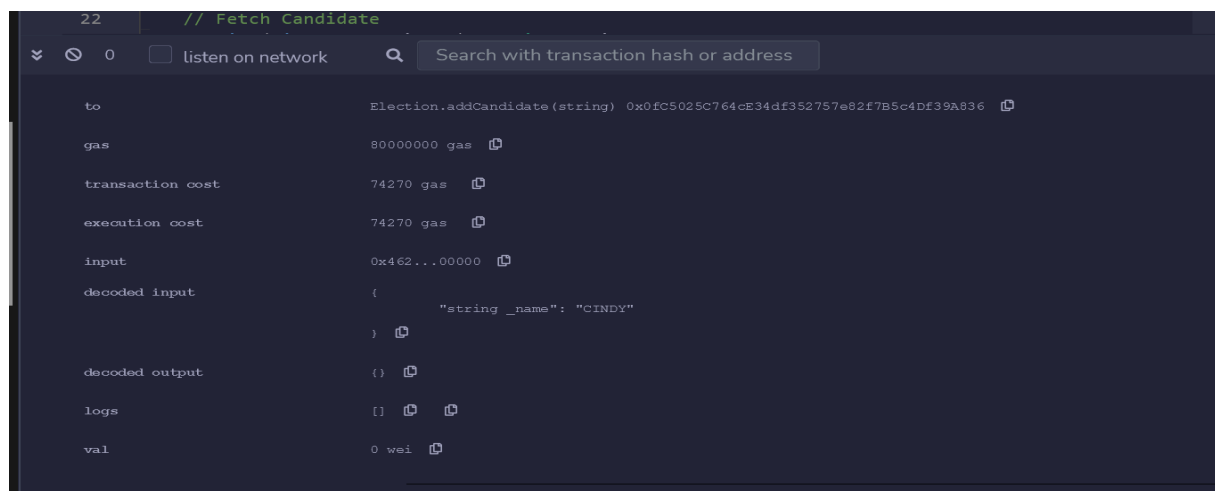
37      * @param newOwner The address to transfer ownership to.
38      */
39      function transferOwnership(address newOwner) public onlyOwner {
40          require(newOwner != address(0), "Address shouldn't be zero");
41          emit OwnershipTransferred(owner, newOwner);
42          owner = newOwner;
43      }
44

```

p. Consulter la valeur de votre CandidateID à l'aide de Remix et fournissez le détail.

**Nous avons les valeurs de 0 et 1.**

q. Ajouter un second candidat de votre choix dans le smart contract et fournissez le détail de la transaction



Field	Value
to	Election.addCandidate(string) 0x0fc5025c764cE34df352757e82f7B5c4Df39A836
gas	80000000 gas
transaction cost	74270 gas
execution cost	74270 gas
input	0x462...00000
decoded input	{ "string_name": "CINDY" }
decoded output	{}
logs	[]
val	0 wei

s. Fournissez l'adresse du propriétaire du contrat

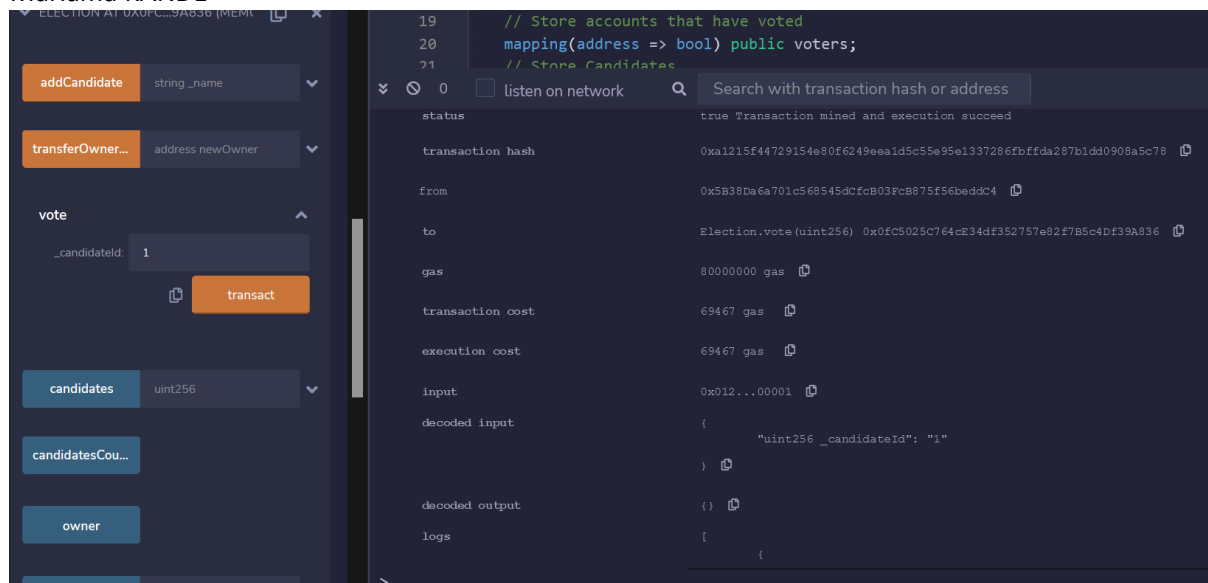

```

from      0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

```

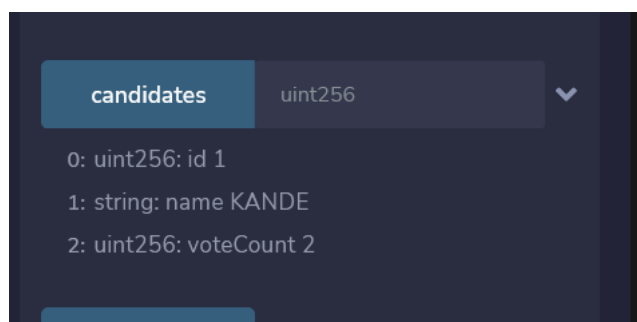
t. Réaliser le premier vote pour l'un des candidats à travers Remix et fournissez le détail de la transaction

Mariama KANDE

v. Demander à votre camarade proche de vous d'interagir avec votre contrat et de voter pour l'un des deux candidats en lui fournissant l'adresse publique de votre smart contract.

**J'ai changé de compte j'ai voté puis suis revenue sur mon compte**



w. Réaliser ensuite le transfert de la propriété à votre camarade en lui demandant son adresse publique

x. A votre avis comment pourrions-nous sécurisé l'appel de la fonction addCandidate afin que vous soyez le seul à pouvoir gérer les candidats ?

**Il suffit de vérifier le détenteur du contrat actuel**