Blockchain & Solidity Lab4 – Voting dApp Development

S2BC

Lab 4: Run a dApp and Consider Next Steps

- BUILD / TEST / INTEGRATE / **RUN**

So far, you've followed the steps in Labs 1 to 3, gaining valuable insights into the core components of blockchain development. Now, in Lab 4, we will discuss crucial considerations for running a dApp in Morpheus.

1. Running the Frontend

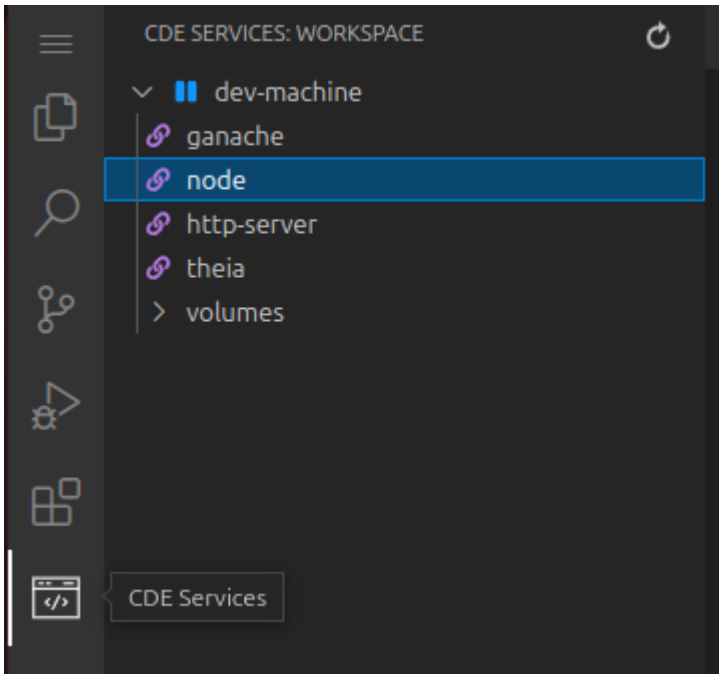
Follow these steps to run the frontend of your voting Dapp:

1. **Start the Frontend Server:**
    - Navigate to the `voting-dapp-2023/frontend` directory.
    - Run the following command:

      ```
      node server.js
      ```

    This will initiate the server for your Dapp's frontend.

2. **Open the Web App in Morpheus:**
    - In your Morpheus IDE interface, locate the CDE menu in the left menu bar.

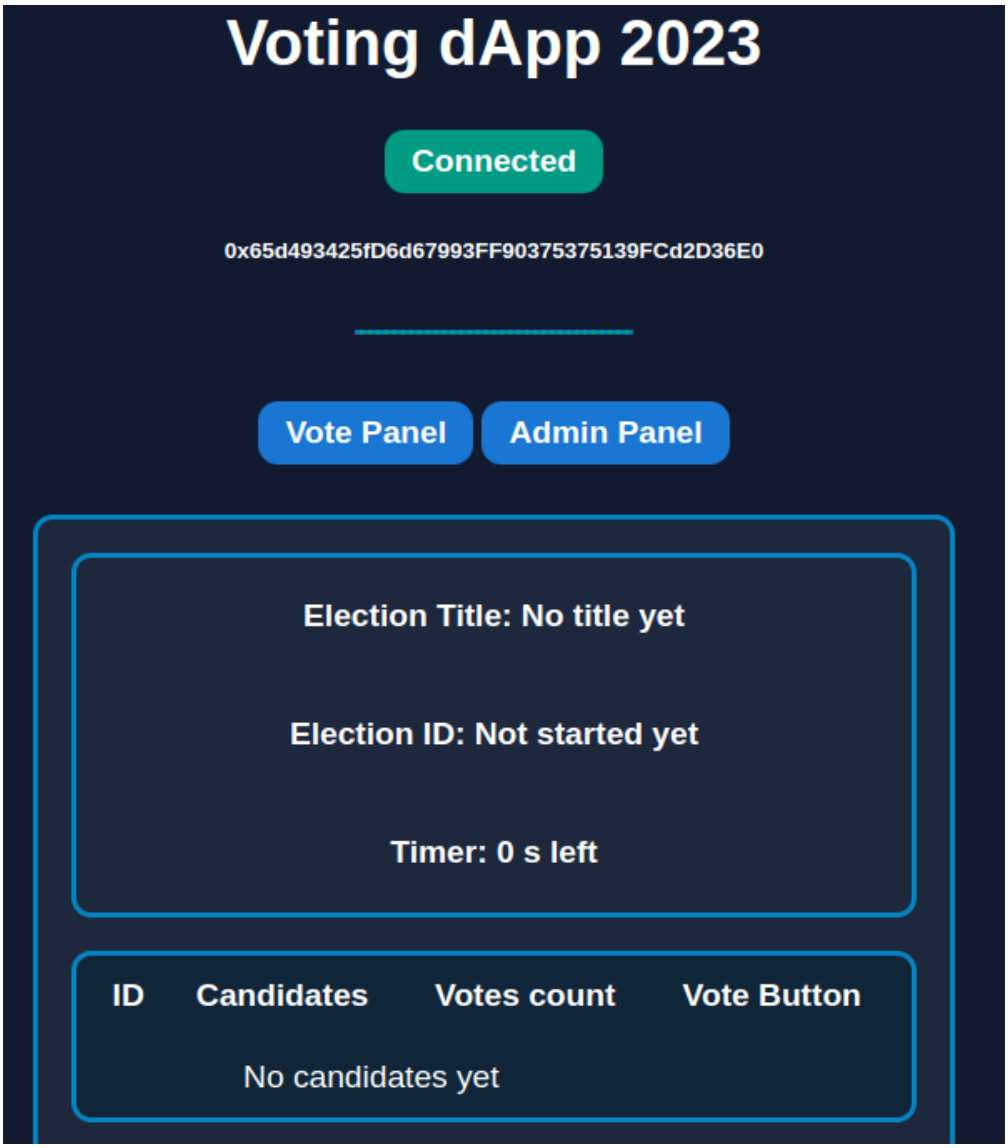


- Click on the node service to open your web app.

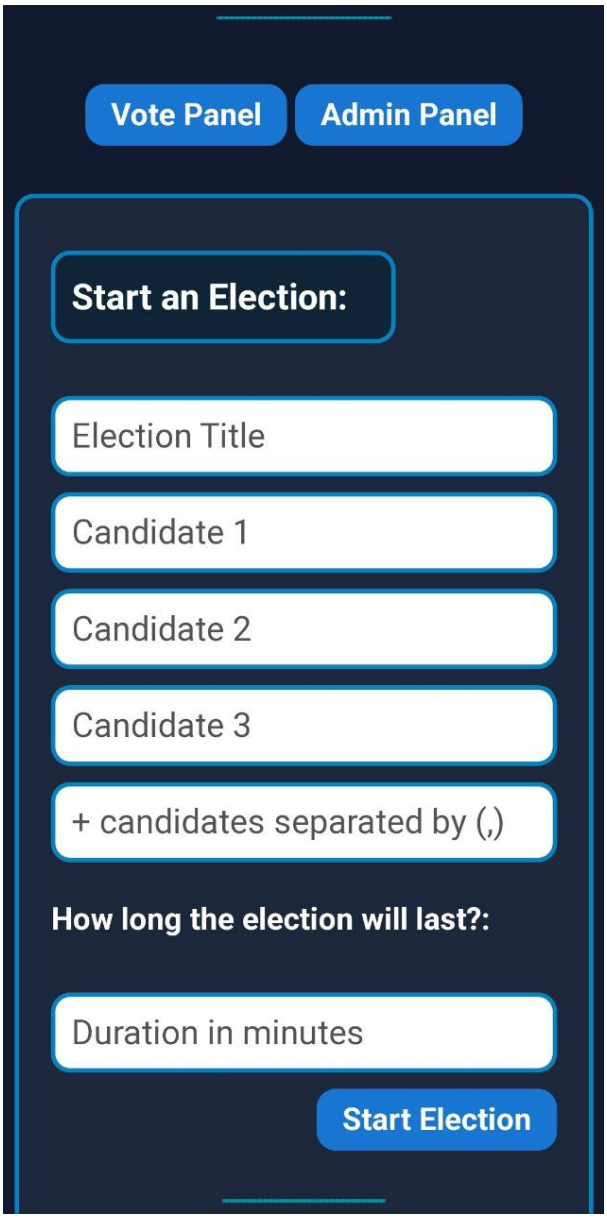
These steps ensure that your frontend server is up and running, and you can access your voting Dapp through Morpheus.

2. Trying the Voting Dapp

1. **Connect to Metamask:**
    - Click the connect button. A Metamask popup will appear, asking if you want to connect. Accept using the deployer account.
2. **Check Voting Status:**
    - View the Voting status in the Vote Panel:

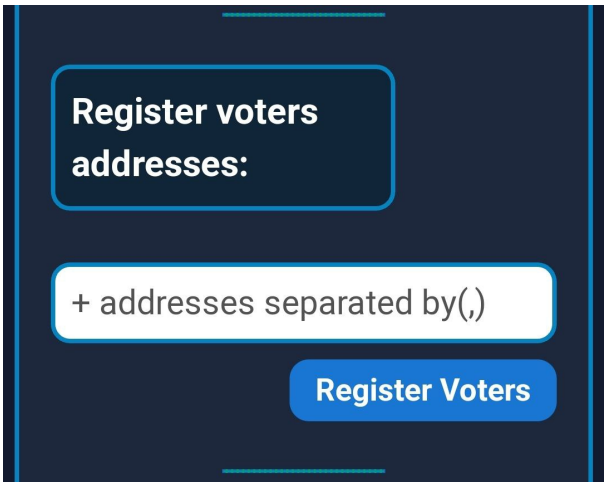


3. **Start an Election:**
    - Visit the Administrator Panel to initiate an election:



- Provide election details (title, candidates, duration) and click the "Start Election" button.

4. **Register Voters:**



5. **Monitor Ongoing Election:**

   - Return to the Vote Panel to view ongoing election information.

6. **Cast a Vote:**
    - Participate in the election by casting your vote.
7. **End the Election:**
    - In the Admin Panel, conclude the election by clicking the "End Election" button.
8. **Mint Results:**
    - Once the election is completed, mint the results using the "Mint Results" button.
9. **Reinitialize Election:**
    - After everything is finished, reinitialize the election by clicking on "Reinitialize" in the Admin Panel.
10. **Console Logs:**

- Open your browser's developer console (F12) to view relevant console logs while navigating the app.

3. Migrating to Sepolia Testnet and Utilizing Etherscan

To successfully migrate your dApp to the Sepolia Testnet and leverage Etherscan for enhanced visibility, follow the steps below:

Step 1: Obtain RPC_URL and Etherscan API Key

1.1 Obtain the RPC_URL for Sepolia from Morpheus, Alkemy's website, or Infura.

1.2 Obtain a free Etherscan API Key from the Etherscan website.

Step 2: Update Configuration Files

2.1 Open your **.env** file and modify the values as follows:

```
RPC_URL="https://eth-sepolia.g.alchemy.com/v2/APIKEY"
PRIVATE_KEY="00000...000"
API_KEY="APIKEYFROMETHERSCAN"
```

Ensure the private key corresponds to the deployer account on Sepolia. You can use any account created with Metamask, and acquire testnet ETH from a faucet like Alkemy faucet.

2.2 Update the `chainID` in your `hardhat.config.js` file from 1303 to 11155111. Then change the network name "votingchain" to "sepolia"

```
require("@nomicfoundation/hardhat-toolbox");

require("dotenv").config();

/** @type import('hardhat/config').HardhatUserConfig */
```

```
module.exports = {
  solidity: "0.8.22",
  networks: {
    sepolia: {
      chainId: 11155111,
      url: process.env.RPC_URL,
      accounts: [process.env.PRIVATE_KEY],
    },
  },
  etherscan: {
    apiKey: process.env.API_KEY,
  },
  paths: {
    artifacts: "./src/artifacts",
    contracts: './src/contracts',
  }
};
```

2.3 In your `frontend/public/script.js` file, replace all occurrences of "1303" with "11155111" to ensure the frontend connects to the Sepolia chainID.

Tip: You can select **1305**, then do Ctrl+D several times to get all occurences selected, and then past 11155111.

Certainly! Here's an organized and clear version of your tutorial:

2.3 Update ChainID in `frontend/public/script.js` File

To ensure that your frontend connects to the Sepolia chainID, follow these steps to replace all occurrences of "1303" with "11155111" in the `frontend/public/script.js` file.

Manual Method:

1. Open your text editor and navigate to the `frontend/public/script.js` file.
2. Locate the first occurrence of "1303" and position your cursor at the beginning of the number.
3. Press `Ctrl + D` (or `Cmd + D` on macOS) to select the current occurrence.
4. Continue pressing `Ctrl + D` until all instances of "1303" are selected.
5. Type "11155111" to replace the selected occurrences.
6. Save the file.

Alternative Method using Find and Replace:

1. Open the `frontend/public/script.js` file in your text editor.
2. Use the find function (`Ctrl + F` or `Cmd + F`) to search for "1303."
3. Click on "Replace" or "Replace All."

4. Enter "11155111" as the replacement and confirm the action.
5. Save the file.

Step 3: Redeploy the Contract on Sepolia

3.1 Navigate to the hardhat folder in your terminal.

3.2 Run the following command to redeploy the contract on Sepolia:

```
npx hardhat run scripts/deploy.js --network sepolia
```

Step 4: Update Contract Address in Frontend

4.1 Once the deployment is complete, locate the voting contract address.

4.2 Copy the contract address and update the variable in `frontend/public/script.js` as follows:

```
const contractAddress = 'votingcontractaddress';
```

Step 5: Restart the Server

5.1 Start or restart your server using the following command:

```
node frontend/server.js
```

Step 6: Verification on Etherscan

6.1 If you have chosen to verify your contract on Etherscan, you have two methods available:

Method 1: Using Hardhat

To verify your contract using Hardhat, follow these steps:

1. Navigate to your Hardhat directory in the terminal.
2. Run the following command, replacing `<votingcontractaddress>` with the actual address of your deployed contract:

```
npx hardhat verify <votingcontractaddress> --network sepolia
```

3. Review the response in the terminal to confirm the success or any output related to the verification process.

4. Check Etherscan to verify if the contract has been successfully verified.

Method 2: Using Etherscan Interface

An alternative method is to use the Etherscan interface directly. Provide the following information to Etherscan:

- Contract Code
- Compiler Version
- ABI (Application Binary Interface) of the contract

This method involves interacting with the Etherscan website to manually input the required details for verification.

Choose the method that best fits your workflow or preference. Successful verification ensures transparency and allows users to explore transactions and events within the voting contract on Etherscan, providing detailed insights at each step of the election.

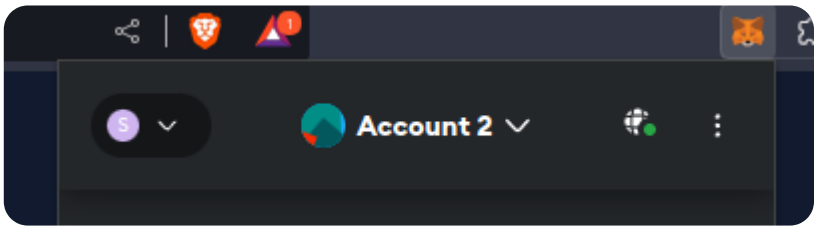
Step 7: Test the New Setup

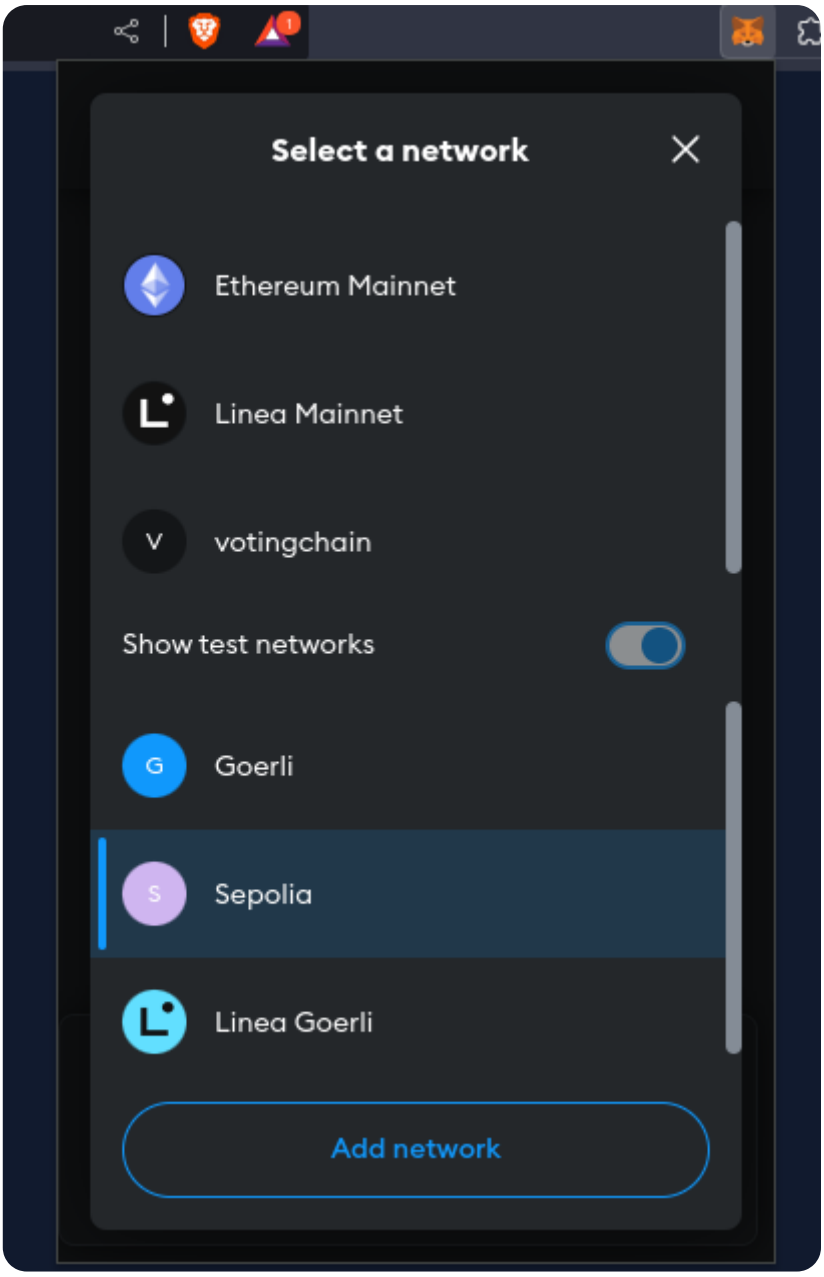
7.1 Retry launching a new election on this updated setup to ensure seamless functionality.

By following these steps, your dApp should now be successfully migrated to the Sepolia Testnet, utilizing the specified RPC_URL and providing enhanced insights through Etherscan verification.

Explore the following Etherscan screenshots for a visual confirmation:

https://sepolia.etherscan.io/




Select a network
Ethereum Mainnet
Linea Mainnet
votingchain
Show test networks
Goerli
Sepolia
Linea Goerli
Add network

Voting dApp 2023

Connected

0x65d493425fD6d67993FF90375375139FCd2D36E0

Vote Panel | Admin Panel

Start an Election:

What is the best testnet netw

Goerli

Sepolia

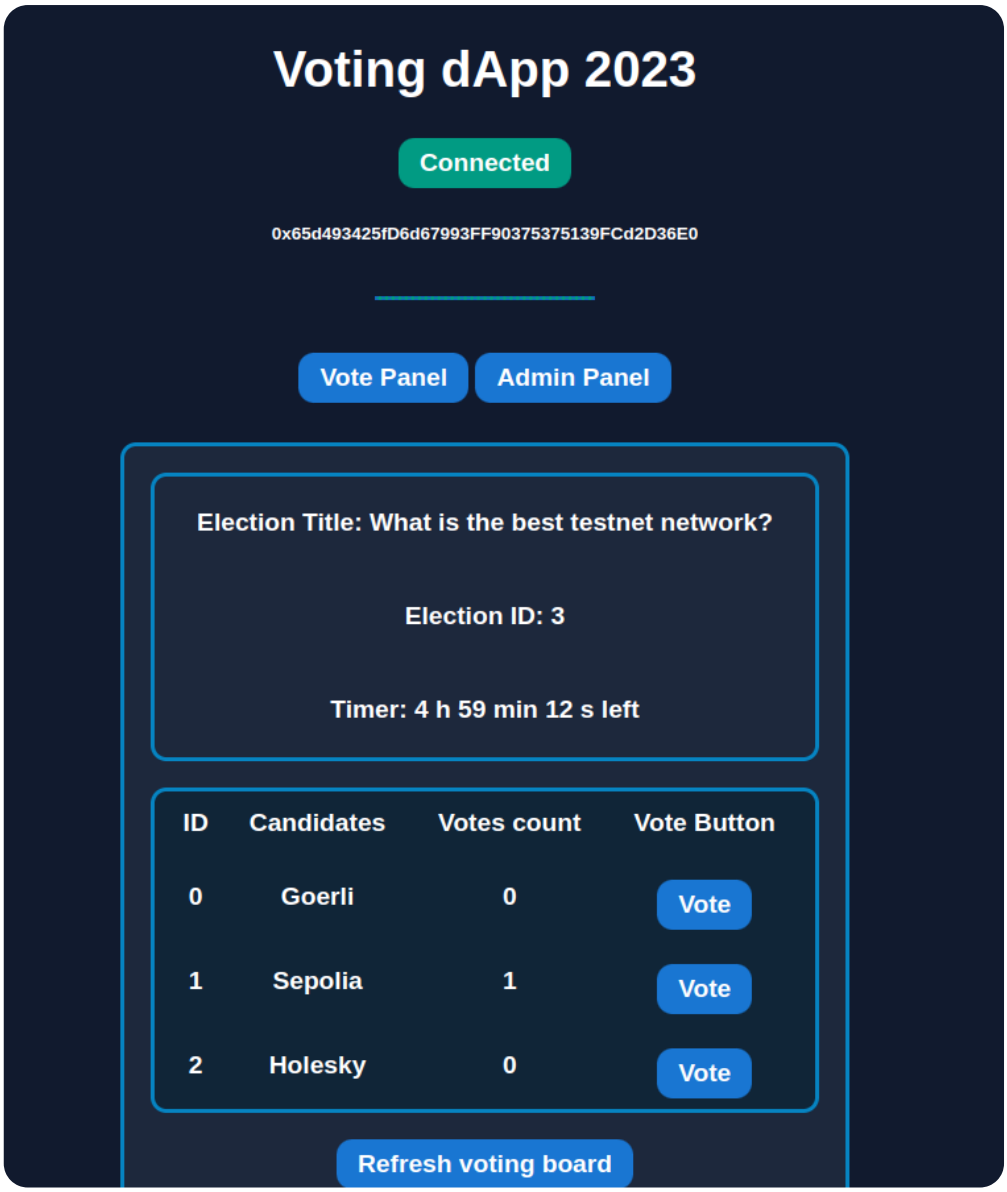
Holesky

+ candidates separated by (,)

How long the election will last?:

300

Start Election



Latest 25 from a total of 798 transactions

Transaction Hash	Method	Block	Age	From		To	Value	Txn Fee
0x13f76e3bab7d7cd41...	Mint Result N...	4933874	2 mins ago	0x65d493...Cd2D36E0	OUT	0x46c6a1...B497200e	0 ETH	0.00065613
0x47f09bf28b6cbf706...	End Election	4933873	2 mins ago	0x65d493...Cd2D36E0	OUT	0x46c6a1...B497200e	0 ETH	0.00009099
0xae3b068a0d1d6361...	Vote To	4933870	3 mins ago	0x65d493...Cd2D36E0	OUT	0x46c6a1...B497200e	0 ETH	0.00032166
0xbf7c3bb3a590e33b2...	Register Voters	4933868	3 mins ago	0x65d493...Cd2D36E0	OUT	0x46c6a1...B497200e	0 ETH	0.00024592
0xe9c6f922cd6b2e03b...	Start Election	4933867	4 mins ago	0x65d493...Cd2D36E0	OUT	0x46c6a1...B497200e	0 ETH	0.0007729
0x913ee7256c0560fa9...	Reset Election	4916842	2 days 15 hrs ago	0x65d493...Cd2D36E0	OUT	0x46c6a1...B497200e	0 ETH	0.00637536

Method

Mint Result N...

End Election

Vote To

Register Voters

Start Election

Reset Election

Transactions | Token Transfers (ERC-20) | Contract | Events

Sepolia Testnet

Search by Address / Txn Hash / Block / Token

Etherscan

Home | Blockchain | Tokens | NFTs | Misc

Transaction Details

Overview | Logs (2) | State | More

[This is a Sepolia **Testnet** transaction only]

Transaction Hash:	0x13f76e3bab7d7cd41ce938a179f26003c125b2152020748e78431b123a2bb9da
Status:	Success
Block:	4933874 2 Block Confirmations
Timestamp:	25 secs ago (Dec-22-2023 02:48:00 AM +UTC)
Transaction Action:	Call Mint Result NFTs Function by 0x65d493...Cd2D36E0 on 0x46c6a1...B497200e
From:	0x65d493425fD6d67993FF90375375139FCd2D36E0
Interacted With (To):	0x46c6a1AE9A084013eb5345e104Ee8AFEB497200e
ERC-721 Tokens Transferred:	ERC-721 Token ID [4] Election NFT... (ENFT...) From 0x000000...00000000 To 0x65d493...Cd2D36E0
Value:	0 ETH ($0.00)
Transaction Fee:	0.00065613149846073 ETH ($0.00)
Gas Price:	2.714708614 Gwei (0.000000002714708614 ETH)

More Details: + Click to show more

Gas Limit & Usage by Txn: 244,651 | 241,695 (98.79%)

Gas Fees: Base: 1.214708614 Gwei | Max: 3.018303854 Gwei | Max Priority: 1.5 Gwei

Burnt & Txn Savings Fees:

Other Attributes: Txn Type: 2 (EIP-1559) | Nonce: 773 | Position in Block: 79

Input Data:

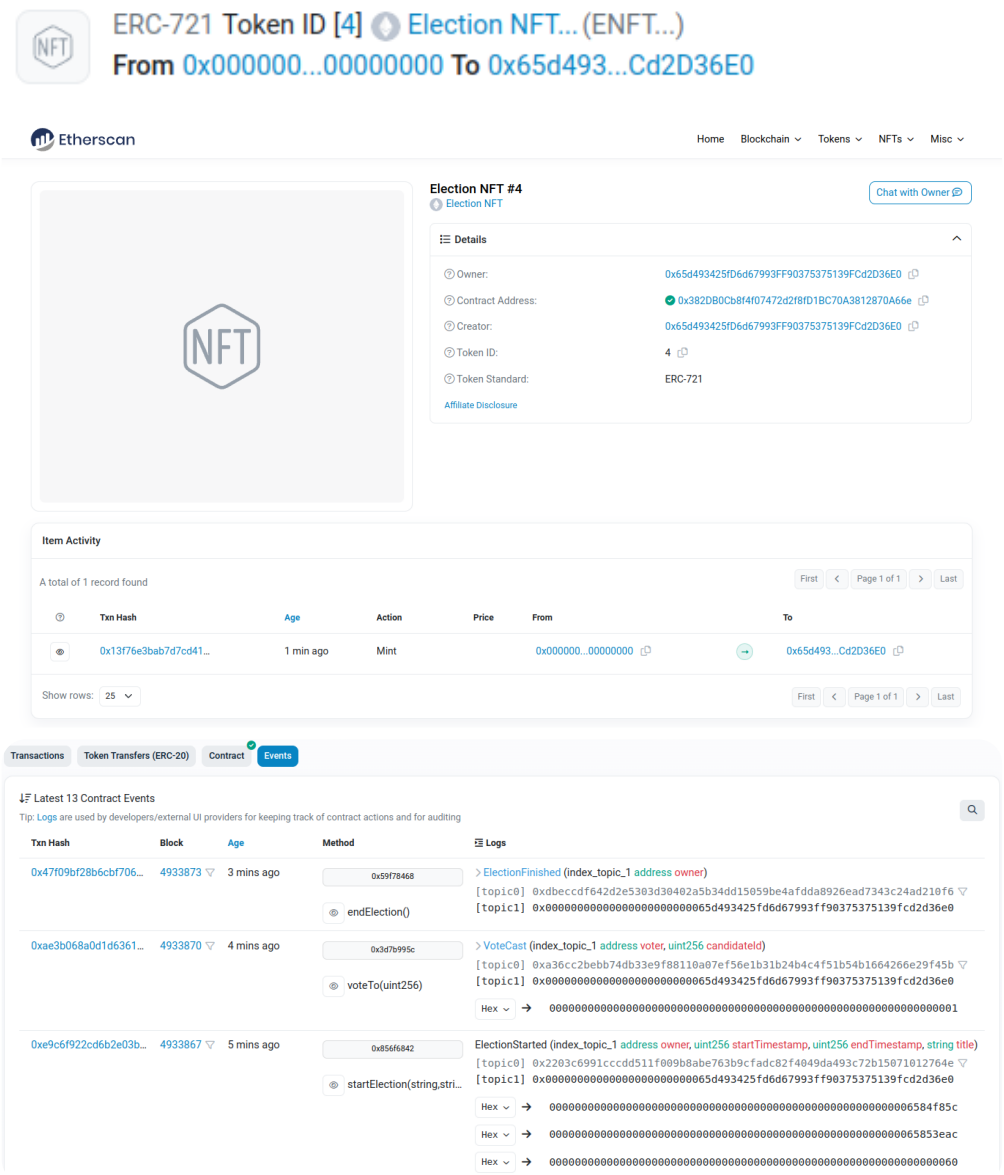
```
Function: mintResultNFTs(string _tokenURI) ***

MethodID: 0x128c14ba
[0]:  0000000000000000000000000000000000000000000000000000000000000020
[1]:  0000000000000000000000000000000000000000000000000000000000000046
[2]:  7b22656c65637469766f6e5469746c65223a225768617420697320746865206265
[3]:  7374207465737420e6574206e657477f726b3f222c22656c65637469766f6e4944
[4]:  223a332c2277696e6e657249422231c2277696e6e65726e6165225a2253
```

View Input As | Decode Input Data

Decode Input Data

#	Name	Type	Data
0	_tokenURI	string	{"electionTitle":"What is the best testnet network?" "electionID":3 "winnerID":1 "winnerName":"Sepolia" "numberOfVotes":1 "startTime":1703213148 "endTime":1703213256}



4. Uploading Your dApp on Morpheus app library to share with community

For detailed steps on uploading your dApp, refer to the documentation.

https://docs.morpheuslabs.io/docs/submit-app-to-the-app-store