

# Blockchain & Solidity Lab4 – Crowdfunding 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 front-end

Follow these steps to run the front-end of your crowdfunding Dapp:

#### 1. Start the front-end Server:

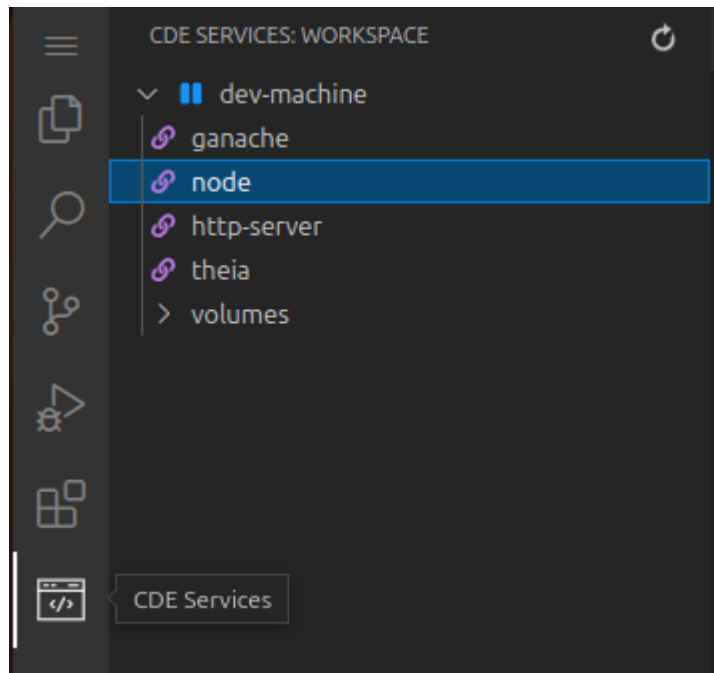
- Navigate to the `crowd-funding-update-2024/front-end` directory.
- Run the following command:

```
npm run dev
```

This will initiate the server for your Dapp's front-end.

#### 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 front-end server is up and running, and you can access your voting Dapp through Morpheus.

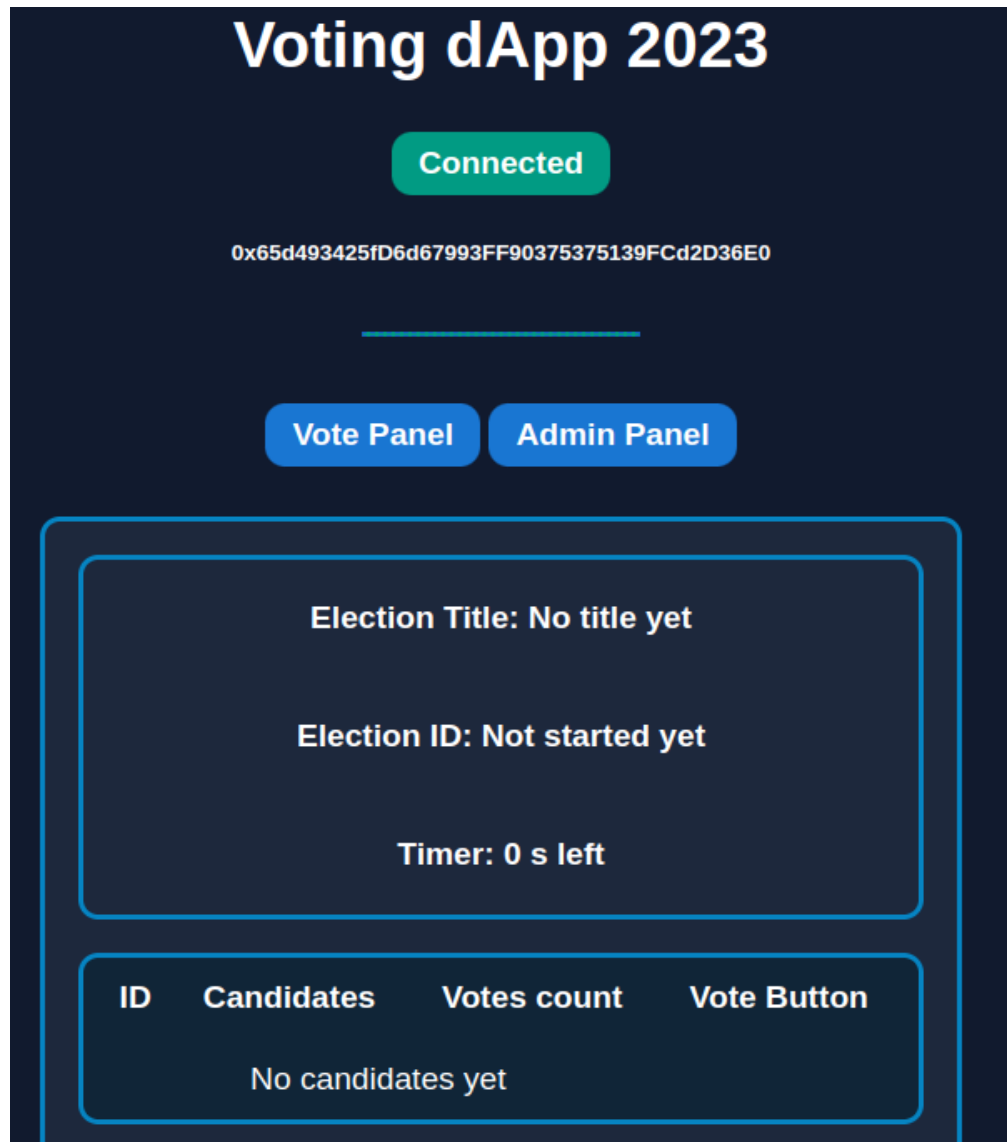
## 2. Trying the Crowdfunding Dapp

### 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 Campaign Count:

- View the Campaign Count on the main dashboard:



### 3. Start a Campaign:

- Create a new Campaign in the main Panel:

The screenshot shows a mobile application interface with a dark blue background. At the top, there are two buttons: 'Vote Panel' and 'Admin Panel'. Below them is a section titled 'Start an Election:' in a rounded rectangle. This section contains several input fields: 'Election Title', 'Candidate 1', 'Candidate 2', 'Candidate 3', and a field with the placeholder text '+ candidates separated by (,)'. Below these fields is the text 'How long the election will last?:' followed by a 'Duration in minutes' input field. At the bottom right of this section is a blue button labeled 'Start Election'.

- Provide campaign details (description and minimum contribution in wei) and click the "Create Campaign" button.

#### 4. **Contribute:**

- Participate in the campaign by contributing. (manager or/and supporter)

The screenshot shows a mobile application interface with a dark blue background. It features a section titled 'Register voters addresses:' in a rounded rectangle. Below this title is an input field with the placeholder text '+ addresses separated by(,)'. At the bottom right of this section is a blue button labeled 'Register Voters'.

#### 5. **Check balance campaign:**

- Return to the Campaign Panel to view ongoing campaign information.

#### 6. Create a request:

- Manager can create a request.

#### 7. Approve request:

- In Campaign Panel, approve requests made by campaign manager.

#### 8. Finalize request:

- Once the campaign is completed, finalize requests to release funds.

#### 9. 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 "poa" 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 `front-end/public/script.js` file, replace all occurrences of "1303" with "11155111" to ensure the front-end connects to the Sepolia chainID.

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

---

### 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 front-end

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

4.2 Copy the contract address and update the variable in `front-end/public/interact-contract.js` as follows:

```
const contractAddress = "campaignCreatorcontractaddress";
```

### Step 5: Restart the Server

5.1 Start or restart your server using the following command:

```
npm run dev
```

## 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 `<campaignCreatorcontractaddress>` with the actual address of your deployed contract:

```
npx hardhat verify <campaignCreatorcontractaddress> --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.

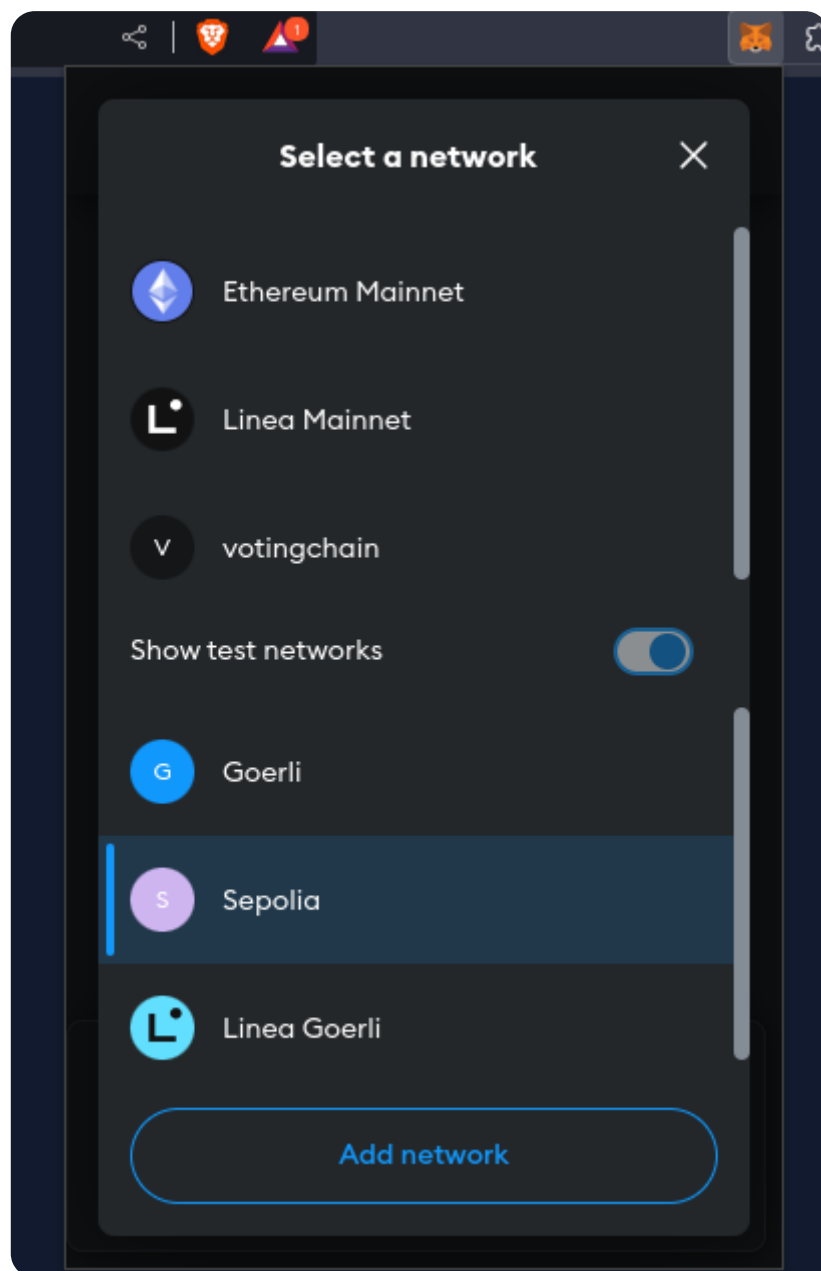
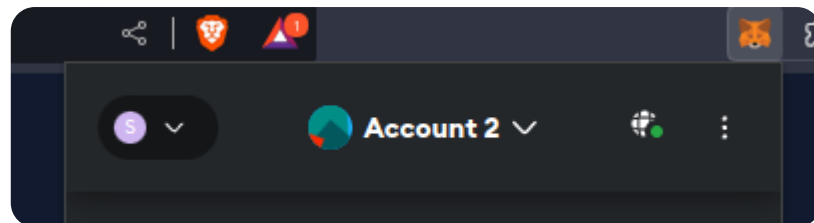
## Step 7: Test the New Setup

7.1 Retry launching a new campaign 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/>





# 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

# Voting dApp 2023

Connected

0x65d493425fD6d67993FF90375375139FCd2D36E0

Vote Panel

Admin Panel

Election Title: What is the best testnet network?

Election ID: 3

Timer: 4 h 59 min 12 s left

ID	Candidates	Votes count	Vote Button
0	Goerli	0	<button>Vote</button>
1	Sepolia	1	<button>Vote</button>
2	Holesky	0	<button>Vote</button>

Refresh voting board

Latest 25 from a total of 798 transactions

Transaction Hash	Method	Block	Age	From	To	Value	Txn Fee
0x13f7e3bab7d7cd41...	Mint Result N...	4933874	2 mins ago	0x65d493...Cd2D36E0	0x46c6a1...B497200e	0 ETH	0.00065613
0x47f09bf28b6cbf706...	End Election	4933873	2 mins ago	0x65d493...Cd2D36E0	0x46c6a1...B497200e	0 ETH	0.00009099
0xae3b068a0d1d6361...	Vote To	4933870	3 mins ago	0x65d493...Cd2D36E0	0x46c6a1...B497200e	0 ETH	0.00032166
0xbf7c3bb3a590e33b2...	Register Voters	4933868	3 mins ago	0x65d493...Cd2D36E0	0x46c6a1...B497200e	0 ETH	0.00024592
0xe9c6f922cd6b2e03b...	Start Election	4933867	4 mins ago	0x65d493...Cd2D36E0	0x46c6a1...B497200e	0 ETH	0.0007729
0x913ee7256c0560fa9...	Reset Election	4916842	2 days 15 hrs ago	0x65d493...Cd2D36E0	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



Home Blockchain ▾ Tokens ▾ NFTs ▾ Misc ▾

## Transaction Details &lt; &gt;

Overview Logs (2) State

More ▾

[ This is a Sepolia Testnet transaction only ]

Transaction Hash: [0x13f76e3bab7d7cd41ce938a179f26003c125b2152020748e78431b123a2bb9da](#)

② Status: Success

Block: 4933874 2 Block Confirmations

🕒 Timestamp: 🕒 25 secs ago (Dec-22-2023 02:48:00 AM +UTC)

Transaction Action: [Call](#) [Mint Result NF Ts](#) 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: Burnt: 0.0009358989846373 ETH (30.06) Txn Savings: 0.000073274615218 ETH (36.46)

[illegible] 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         }



ERC-721 Token ID [4] Election NFT... (ENFT...)

From 0x000000...00000000 To 0x65d493...Cd2D36E0

Home Blockchain Tokens NFTs Misc

**Election NFT #4**  
 Election NFT
 [Chat with Owner](#)

**Details**

- Owner: 0x65d493425fd6d67993ff90375375139fcd2d36e0
- Contract Address: 0x382D80Cb8f4f07472d2f8d18C70A3812870A66e
- Creator: 0x65d493425fd6d67993ff90375375139fcd2d36e0
- Token ID: 4
- Token Standard: ERC-721
- [Affiliate Disclosure](#)

**Item Activity**  
 A total of 1 record found
 

First < Page 1 of 1 > Last

	Txn Hash	Age	Action	Price	From	To
	0x13f76e3bab7d7cd41...	1 min ago	Mint		0x000000...00000000	0x65d493...Cd2D36E0

Show rows: 25 <div>First < Page 1 of 1 > Last</div>

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

Latest 13 Contract Events  
 Tip: Logs are used by developers/external UI providers for keeping track of contract actions and for auditing

Txn Hash	Block	Age	Method	Logs
0x47f09bf28b6cbf706...	4933873	3 mins ago	0x59f78468 endElection()	ElectionFinished (index_topic_1 address owner) [topic0] 0xdbecddf642d2e5303d30402a5b34dd15059be4afdda8926ead7343c24ad210f6 [topic1] 0x00
0xae3b068a0d1d6361...	4933870	4 mins ago	0x3d7b995c voteTo(uint256)	VoteCast (index_topic_1 address voter, uint256 candidateId) [topic0] 0xa36cc2bebb74db33e9f88110a07ef56e1b31b24b4c4f51b54b1664266e29f45b [topic1] 0x00 Hex → 0001
0xe9c6f922cd6b2e03b...	4933867	5 mins ago	0x856f6842 startElection(string,stri...	ElectionStarted (index_topic_1 address owner, uint256 startTimeStamp, uint256 endTimeStamp, string title) [topic0] 0x2203c6991ccdd511f009b8abe763b9cfadc82f4849da493c72b15071012764e [topic1] 0x00 Hex → 00 Hex → 00 Hex → 00

## 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>

