# BLOCKCHAIN STORAGE: READ ME

## Setup

### Dependencies

Before running the setup scripts for the project, please make sure all of the required dependencies are installed.

NPM:

- Express
- Body-parser
- Web3
- @openzeppelin/merkle-tree
- Fs
- @google-cloud/storage

Python

- Web3
- Google-cloud-storage

### Infrastructure

Make sure that there is at least one Google account. There will need to be a VM-instance hosting a Linux virtual machine and at least one bucket set up. A service account needs to be added for a bucket, and the private key to access that bucket will need to be imported into the VM. Make sure that there are at least two MetaMask accounts, one for the server and one for the user. The Sepolia test network is used. The fileImport folder contains the temp_key for the server owner.

### Scripts

Assuming that dependencies are satisfied, and the infrastructure is ready, separate the script files into two parts:

Client-side

- Bridge.py
- Testdriver.py

Server-side

- The rest of the files

Run setup.py on the server to generate metatree.json, bucketlist.json, and keyfiles folder.

Please move all bucket access keys into the keyfiles folder.

Check to make sure that metatree.json only contains an empty array, i.e., [].

Make sure that bucketlist.json only contains an empty array, i.e., [].

Next, run providers.py to set up bucket providers. The syntax to do this can be found as a comment in the file. For instance,

```
python3 providers.py append 0xdeadbeef google-cloud-project bucket-1
/home/server/keyfiles/bucket-key.json
```

Once this is done, deploy the smart contract of the server owner.

The deployment process is as follow:

- Remix IDE was used for deployment
- Metamask was used as the wallet for deploying the contract to the account stored in the wallet
- In the remix IDE compile the code, and the deploy the contract
- Pay for the deployment of the contract via metamask
- Wait for the transaction to complete, then the contract is deployed to a certain random address
- Connect to the contract via the address of the contract

Start the server using

```
node index.js
```

## Testing

If using testdriver.py to test, change the ip and port to the corresponding machine. The server by default is set to port 3125. The IP will depend on the machine. Change the pub_addr_id variable to that of the metamask public address and the private key to the account private key.

The test driver can be run with a command on the client, for instance, to add a file run:

```
python3 testdriver.py 2 “InTheoryThisShouldWork.txt” “But in reality
it rarely does”
```