



ARASHTAD

Title	HARMONY BLOCKCHAIN GETTING FAMILIAR WITH IT USING PYTHON
Description	Intro
Date	June 3rd, 2022
Author	Arashtad
Author URI	<a href="https://Arashtad.com">https://Arashtad.com</a>



In this article, we are going to get familiar with the Harmony blockchain. and install the dependencies such as Docker to be able to connect to the Harmony blockchain. After installing everything and connecting to the blockchain, using the Python scripts, we will get some data from the shard of the Testnet.

## WHAT IS HARMONY BLOCKCHAIN?

Harmony is a powerful blockchain platform designed to ease and facilitate the development of Dapps. The innovation of Harmony in decentralized applications is based on random state sharding. The random state sharding allows creating blocks faster. This blockchain is EVM compatible with staking features as well. The good news for Python developers is that all the smart contracts in Harmony are written in Python. Developing on Harmony looks very familiar for Ethereum developers because it is fully Ethereum compatible.

## INSTALLING THE DEPENDENCIES FOR HARMONY BLOCKCHAIN

To get started with the harmony blockchain, follow the instructions below:  
For Windows and Linux:

```
pip install pyhmy
```

For Mac Os and Linux:

```
sudo pip3 install pathlib sudo pip3 install pyhmy
```

## DOCKER

Docker is the most important dependency for Harmony blockchain. However, it's different to install it based on the operating system and we are going through on Linux.

## INSTALLATION GUIDE ON LINUX

### 1. Depedencies

Enter the following commands one by one in the terminal to install the docker dependencies:

```
sudo apt update sudo apt -y install apt-transport-https ca-  
certificates curl software-properties-common sudo apt -y remove  
docker docker-engine docker.io containerd runc
```

### 2. Add Docker's official GPG key

Enter the following 2 commands one by one in the terminal:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg
--dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

### 3. Add the Docker repository to Linux:

To add the docker repository to Linux, enter the 2 following commands separately in the terminal:

```
echo "deb [arch=$(dpkg --print-architecture)
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu bionic stable" | sudo
tee /etc/apt/sources.list.d/docker.list > /dev/null deb [arch=amd64
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu bionic stable
```

### 4. Install Docker Engine and Docker Compose:

The 2 separate commands should install the Docker engine and Docker Compose for you:

```
sudo apt update sudo apt install docker-ce docker-ce-cli
containerd.io
```

### 5. Checking the installation:

To check the installation, first install the 2 following commands in the terminal:

```
sudo usermod -aG docker USER newgrp docker
```

And then, check the version of Docker:

```
docker version
```

Result:

```
Client: Docker Engine - Community Version: 20.10.14 API version:
1.41 Go version: go1.16.15 Git commit: a224086 Built: Thu Mar 24
01:47:57 2022 OS/Arch: linux/amd64 Context: default Experimental:
true Server: Engine: Version: 20.10.14 API version: 1.41 (minimum
version 1.12) Go version: go1.16.15 Git commit: 87a90dc Built: Thu
Mar 24 17:15:03 2022 OS/Arch: linux/amd64 Experimental: false
containerd: Version: v1.5.11 GitCommit:
3df54a852345ae127d1fa3092b95168e4a88e2f8 runc: Version: 1.0.3
GitCommit: docker-init: Version: 0.19.0 GitCommit: de40ad0
```

After installation for every use, you need to first sign in to docker and run the following commands in terminal:

```
sudo usermod -aG docker docker login
```

Now, let's git clone the harmony blockchain repository by using the following command in the terminal:

```
git clone https://github.com/harmony-one/pyhmy
```

Then, install it using this command in the terminal of the same directory:

```
make install
```

Finally, the whole set of the following commands altogether will provide you the connection to the Harmony blockchain:

```
mkdir -p $(go env GOPATH)/src/github.com/harmony-one cd $(go env
GOPATH)/src/github.com/harmony-one git clone
https://github.com/harmony-one/mcl.git git clone
https://github.com/harmony-one/bls.git git clone
https://github.com/harmony-one/harmony.git cd harmony make test-rpc
```

Once you faced the “=== FINISHED RPC TESTS ===” message and also passed all tests, you can be sure that everything has gone the right way.

## CONNECTING TO HARMONY TESTNET

Using the below scripts, we will at first to a Testnet of shard0 in the Harmony blockchain and then use a test address to check the balance, balance by shard, latest balance, and the account nonce:

```
from pyhmy import account

test_net = 'https://api.s0.b.hmny.io'      # this is shard 0
test_address = 'one18t4yj4fuutj83uwqckkvxp9gfa0568uc48ggj7'

balance = account.get_balance(test_address, endpoint=test_net)

total_balance = account.get_total_balance(test_address,
endpoint=test_net)
balance_by_shard = account.get_balance_on_all_shards(test_address,
endpoint=test_net)

genesis_balance = account.get_balance_by_block(test_address,
block_num=0, endpoint=test_net)
latest_balance = account.get_balance_by_block(test_address,
block_num='latest', endpoint=test_net)
account_nonce = account.get_account_nonce(test_address,
block_num='latest', endpoint=test_net)

print(balance)
print(total_balance)
print(balance_by_shard)
print(genesis_balance)
print(latest_balance)
print(account_nonce)
```

Result:

```
10000000000000000000 1010000000000000000000 [{"shard": 0, 'balance':
10000000000000000000}, {'shard': 1, 'balance': 0}, {'shard': 2,
'balance': 10000000000000000000}, {'shard': 3, 'balance': 0}] 0
10000000000000000000 10169
```

And by using the following scripts, we can get the data of the number out of transactions that have been sent and received on shard 0 of the Testnet.

```
from pyhmy import account

test_net = 'https://api.s0.b.hmny.io'      # this is shard 0
test_address = 'one18t4yj4fuutj83uwqckkvxp9gfa0568uc48ggj7'

tx_count = account.get_transactions_count(test_address,
tx_type='ALL', endpoint=test_net)

sent_tx_count = account.get_transactions_count(test_address,
tx_type='SENT', endpoint=test_net)
received_tx_count = account.get_transactions_count(test_address,
tx_type='RECEIVED', end-point=test_net)

legacy_tx_count = account.get_transaction_count(test_address,
block_num='latest', endpoint=test_net)
# API is legacy
legacy_tx_count_pending =
account.get_transaction_count(test_address, block_num='pending',
end-point =
    test_net)

print(tx_count)
print(sent_tx_count)
print(received_tx_count)
print(legacy_tx_count)
print(legacy_tx_count_pending)
```

Result:

```
10211 10162 49 10169 10169
```

Using the code below, we can get the number of staking transactions in total, sent and received on shard 0 of the Harmony Testnet:

```
from pyhmy import account

test_net = 'https://api.s0.b.hmny.io'      # this is shard 0
test_net_shard_1 = 'https://api.s1.b.hmny.io'
test_address = 'one18t4yj4fuutj83uwqckkvxp9gfa0568uc48ggj7'

stx_count = account.get_staking_transactions_count(test_address,
tx_type='ALL', end-point=test_net)
sent_stx_count =
account.get_staking_transactions_count(test_address, tx_type='SENT',
end-point=test_net)
received_stx_count =
account.get_staking_transactions_count(test_address,
tx_type='RECEIVED',
        endpoint=test_net)

print(stx_count)
print(sent_stx_count)
print(received_stx_count)
```

Result:

```
0 0 0
```

Using the following code, we can get the first 100 transaction hashes of the shard 0:

```
from pyhmy import account

test_net = 'https://api.s0.b.hmny.io'      # this is shard 0
test_net_shard_1 = 'https://api.s1.b.hmny.io'
test_address = 'one18t4yj4fuutj83uwqckkvxp9gfa0568uc48ggj7'

first_100_tx_hashes = account.get_transaction_history(test_address,
page=0, page_size=20,
        include_full_tx=False, endpoint=test_net)

print(first_100_tx_hashes)
```

Result:

```
[ '0xa5bf23cbaf63fa16cebc18d13fef77172c9e49fd2eaf866e7fb2c50529a9af42'  
,  
'0xedd0dc457976b853de8434c6dbf47be465debe4f5cbd8c9ff1f5e0f920044dfb'  
,  
'0x16bb8b5df5e19ac35c1f908f48915899c5b02ce97bdc030400b31444e296093b'  
,  
'0x128dc419a805f8147ad827f4f4da2f5f5eaaf5a42f8515ba3001519f4a4b1758'  
,  
'0xf36dfee42ccefae3977536b25eaff426fa8e2fc0513ee41e01547320db03e015'  
,  
'0x9a570499ce34c02bde2b124f1fd75c0aded1e84ec9960477ab0f1854bd8ecef'  
,  
'0x4e982c8f493a15f0edfe88221b1ff6d359650cd4273397a5ee057ba1df847551'  
,  
'0x8a5ed5f6e183da30da74dee5f373b9fdea6e0c3c8b0dbbf4e0c218c637aaf0cf'  
,  
'0x04b76e4f75bc51607b283fd1e0e75df999e7aac679e7b4a2aadfb96bb7ac39b2'  
,  
'0x512003037125cbde3ae7af5495eca93c335ce84e181692612853d0f014819e45'  
,  
'0x113efecb62251356b4c3620c2f79ffdfef5905a8616ea7094edd428dbf4f326ad'  
,  
'0x09dcf5b49ef74c6f5f159c4168dd9573afcc927e1b45d26c22ee7e4d33e1b152'  
,  
'0x11bb90e10da359246fe8a026f919b5a6ec6641093c1026967313a843b9bb2b64'  
,  
'0xd8a792e8e918ff307c2234cef0327fc64977143101f6b20c52b4eb1d7ec80deb'  
,  
'0xf44a7a632f27709dc0faaa4f9fa59a65d575cd11b904f61515d2e33e1f9f8360'  
,  
'0xc004838024c304de93e67a930a4667e59a9d011e2049d11bf0bbab3ca1972d9b'  
,  
'0x95c149c43ed6ca4a206798901289f56832575b953b3fd82c41d65215a6807b46'  
,  
'0x298b8d589d2e78c55e30f699d2617aa25dab647bbfd442f7b6d313140be90089'  
,  
'0x3ad6e7bef84a7b8cb6d5322e4f13e9f39e40d1536843388251f32825734ef56d'  
,  
'0xf49b9467a6cc689a389ccf60b1e69e4688e0d2d84e3238eea9dfbc5aaf1aa2b2'  
,  
' ]
```





## WRAPPING UP

In this tutorial, we have managed to install the dependencies for connecting to the Harmony blockchain. Then, by using docker, we connected to the Harmony blockchain. In addition to that, by using some Python script and harmony library, we got some data from the shard 0 of the Testnet.

