

# Pool Run Documentation

## 1. clone code

1.1 \$ git clone <https://github.com/swordlet/xdagPool.git>

---

## 2. Configure redis

\$ sudo apt update

\$ sudo apt install redis-server

### 2.1 Modify redis password

\$ sudo nano /etc/redis/redis.conf

Find **#requirepass foobared** in the configuration file and modify it to:

requirepass your\_new\_password

Such as:

```
# If the master is password protected (using the "requirepass" configuration
# directive below) it is possible to tell the replica to authenticate before
# starting the replication synchronization process, otherwise the master will
# refuse the replica request.
requirepass 123456
# masterauth <master-password>
```

### 2.2 Other redis configurations

```
# Note: read only replicas are not designed to be exposed to untrusted clients
# on the internet. It's just a protection layer against misuse of the instance.
# Still a read only replica exports by default all the administrative commands
# such as CONFIG, DEBUG, and so forth. To a limited extent you can improve
# security of read only replicas using 'rename-command' to shadow all the
# administrative / dangerous commands.
replica-read-only no

Replication SYNC strategy: disk or socket.
#
```

Then save and close the file.

---

### 3. Install the environment related to running pool

3.1 Install golang-go ( Requires go-1.20 version or above. Requires go-1.20 version or above. If you already have it, please ignore this step.)

```
$wget https://go.dev/dl/go1.22.0.linux-amd64.tar.gz
```

```
$sudo tar -C /usr/local/ -xzf go1.22.0.linux-amd64.tar.gz
```

```
$export PATH=$PATH:/usr/local/go/bin
```

```
$source ~/.profile
```

```
$go version
```

The output of executing go version should be: go version go1.22.0

```
ubuntu@10-35-195:~$ go version
go version go1.22.0 linux/amd64
```

3.2 \$ sudo apt install cmake make

---

### 4. Configure config.json file

4.1 Enter the xdagPool file

```
$cd xdagPool/
```

```
$cp config.example.json ./config.json
```

The results are as follows:

```
ubuntu@10-35-195:~/xdagPool$ ls
cli  config.json  go.sum  kvstore  logs  payouts  randomx  screenshot.png  stratum  util  ws.txt  xdagj_pool_run_doc.pdf  xdaggo
config.example.json  go.mod  jrpc  LICENSE  main.go  pool  README.md  store.txt  tools  wa  www  xdagj_wallet  xdagpool
```

4.2 Modify the config.json configuration file

```
$ cd tools/
```

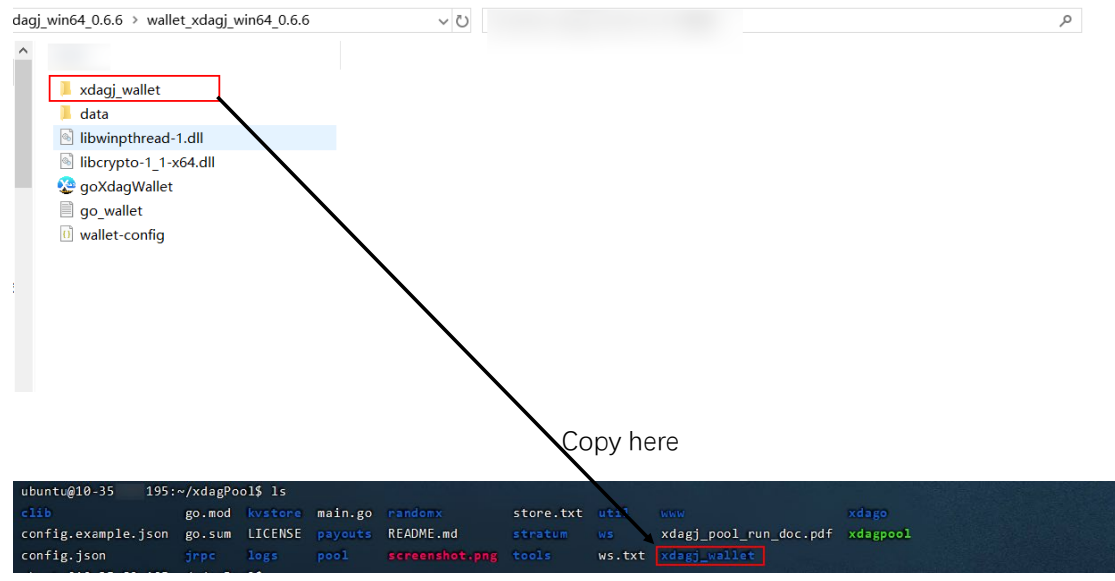
```
$ go build ./encrypt.go
```

```
$ ./encrypt [-p pool password] [-a address] [-w wallet password]
```

## 5. Import wallet file

## 5.1 Open 0.6.6XDAGJ wallet

## 5.2 Copy the **xdagj\_wallet** folder to the **xdagPool** folder



## 6. Compile project

### 6.1 \$ cd ./xdagPool/clib/randomx

### 6.2 \$ cmake .

```
ubuntu@10-35- 195:~/xdagPool/clib/randomx$ cmake .
```

### 6.3 \$ make

```
ubuntu@10-35- -195:~/xdagPool/clib/randomx$ make
```

### 6.4 \$ cd ./xdagPool

### 6.5 \$ go build

```
ubuntu@10-35- -195:~/xdagPool$ go build
```

### 6.6 After the compilation is successful, you can see a compiled executable file

```
ubuntu@10-35- 195:~/xdagPool$ ls
clib          go.mod      kvstore    main.go    randomx    store.txt  util      www      xdaggo
config.example.json  go.sum     LICENSE   payouts   README.md  stratum    ws        xdagj_pool_run_doc.pdf  xdagpool
config.json      jrpc       logs      pool       screenshot.png  tools      ws.txt    xdagj_wallet
```

## 7. Run your pool

7.1 \$ cd **xdagPool**

7.2 \$ sudo ./xdagpool

```
ubuntu@10-35-195:~/xdagPool$ sudo ./xdagpool
2024/03/06 07:33:23 Loading config: /home/ubuntu/xdagPool/config.json
2024/03/06 07:33:23 logSetLevel: 10
2024/03/06 07:33:23 infoFile: logs/info.log
2024/03/06 07:33:23 errorFile: logs/error.log
2024/03/06 07:33:23 shareFile: logs/share.log
2024/03/06 07:33:23 blockFile: logs/block.log
[I] 2024/03/06 07:33:23.925096 platform_linux.go:13: Rlimit Current: 1048576
Enter Security Password: input your pool password

Enter Security Password:
[I] 2024/03/06 07:35:03.914024 main.go:346: Initializing cryptography...
[I] 2024/03/06 07:35:03.914084 main.go:347: Reading wallet...
[I] 2024/03/06 07:35:04.299502 main.go:319: Backend check reply: PONG
[I] 2024/03/06 07:35:04.303798 client.go:94: Connected to server
[I] 2024/03/06 07:35:04.303840 rpc.go:36: Connected to server
[I] 2024/03/06 07:35:04.303955 main.go:53: Running with 4 threads
[I] 2024/03/06 07:35:04.304063 stratum.go:117: Set purge interval to 3h0m0s
[I] 2024/03/06 07:35:04.304313 stratum.go:185: Stratum listening on 0.0.0.0:3003
[I] 2024/03/06 07:35:28.037644 client.go:146: recv: {
  "msgType": 1,
  "msgContent": {
    "task": {
      "preHash": "1946adca85dc41a5ed9a6186f06c6f8224d1a19ae88b98d9c98cdf5c4fa4019d",
      "taskSeed": "e78f04b557438854898167fd833521cd9222a3b7185dd9440000000000000000"
    },
    "taskTime": 26714227,
    "taskIndex": 2567
  }
}
```

**Started successfully**

Miners can connect to this mining pool and start mining. The default configured pool port is 3003.

---

## 8. Other configurations

```
node_name: "examplePool"
"node_rpc": "http://118.26.111.179:10001", node rpc
"node_ws": "ws://118.26.111.179:7001/", node IP
"ws_ssl": false,
```



```

"addressEncrypted": "eDr11R3kmg/Bbht/K6Qay0bPqdcj4JYd7f9LWmKEs0cfGsXNQ+P9tn1W7jGbKkKW",
"threads": 4,
"coin": "xdag",
"estimationWindow": "15m",
"luckWindow": "24h",
"purgeInterval": "3h",
"purgeWindow": "720h",
"node_name": "example.equal",
"node_rpc": "http://testnet-rpc.xdagj.org:10001",
"node_ws": "ws://118.26.111.179:7001/",
"ws_ssl": false,
"rx_mode": "light",
"walletEncrypted": "WjAtYES+5f+0uzM9cV0R6g==",
"log": {
  "logSetLevel": 10
},

```

**rx\_mode:** fast or light

fast(3G ram), light(300M ram) .

"fast" operates faster, but requires more RAM, "light" is slower, but takes up less RAM.

```

"payout": {
  "poolRation": 5.0,
  "rewardRation": 5.0,
  "directRation": 5.0,
  "threshold": 3,
  "paymentInterval": "10m",
  "mode": "equal",
  "paymentRemark": "http://testpool_1_equal.com"
}

```

**mode:** equal or solo

**threshold:** Indicates the payment threshold for issuing rewards to miners. The default payment threshold here is 3 XDAG.

**paymentInterval:** Payment interval, the default here is 10 minutes, which means that the pool will issue rewards to miners who meet the requirements every 10 minutes.

**paymentRemark:** The remark attached to miner rewards, usually the URL of the pool.