

Swarm 介绍及挖矿教程

项目基本概况

Swarm 是去中心化的内容存储和分发服务，可以将它视为 CDN，通过互联网在计算机上分发。你可以像运行以太坊节点一样，运行 Swarm 节点并连接到 Swarm 网络上。这与 BitTorrent 相似，也可以类比 IPFS，用 ETH 作为报酬激励。文件被分解成块，分配并被参与的志愿者们储存。那些为存储并为块提供服务的节点，从那些需要储存和检索数据服务的节点得到 ETH 作为补偿。

Swarm 是以太坊项目官方的一部分，主要是由基金会开发，允许矿池存储、带宽和算力资源来支持基于以太坊网络的应用。团队试图创建一个不停机、零故障和防审查的点对点存储和服务解决方案。在 Swarm 内创建一个经济激励的系统将促进资源交换价值的支付和转移。项目使用了以太坊区块链中不同的协议和技术。

团队

Swarm 作为实现分散式网络的三位一体支柱之一的概念和第一个表述出现在 2015 年初以太坊启动之前。由以太坊创始人 Vitalik Buterin, Gavin Wood 和 Jeffrey Wilcke 推动，Swarm 的协议标签 bzz 和 shh 都是 Vitalik 创造的。

Swarm 核心团队共有 32 名员工，并在瑞士成立了 Swarm 协会，除了以太坊基金会的资金支持外，他们还与 Bitcoin Suisse 达成合作，获得了充足的资金，有了自己的资金来源。在过去一年中，该项目几次重组了团队，现在最终拥有 8 个组织良好的子团队：Leet Squad，Bee 团队，Bee-JS 团队，Comms，HR，DevOps，Ops 和知识管理。

Token 经济模型

BZZ Token 是 Swarm 的功能性 (UTILITY) Token，同时也用于带宽和存储奖励。

BZZ 初始供应量：62.5M，最低起拍价（公募预计在主网发布前）：0.32DAI，由 Curve 平台提供流动性。

Token 分配：

筹款：50%

战队：20%

基础：7%

基础设施补助金：10%

DApp 补助金：10%

捐款：3%

项目融资情况

Swarm 作为以太坊项目官方的一部分，长期以来（2015 年开始，进行了五年多的深入研究）由以太坊基金会提供资金支持，除此之外，为保证 Mainnet 的顺利完成，Swarm 已与 Bitcoin Suisse 合作，并在其帮助下完成了 600 万美元的私募，有足够的资金来进一步扩大团队规模，确保 Swarm 1.0 能够按时执行，并于今年二季度发布。

Bitcoin Suisse，成立于 2013 年，是瑞士加密金融和技术的先驱和市场领导者。Bitcoin Suisse 帮助塑造了瑞士的加密货币和区块链生态系统，并一直是“Crypto Valley”和“Crypto Nation Switzerland”发展的推动力。作为受监管的瑞士金融中介机构，并已获得 Swiss and Liechtenstein bank 的许可，Bitcoin Suisse 为私人 and 机构客户提供主要经纪，交易，托管，借贷，抵押和其他加密金融服务。Bitcoin Suisse 在其位于哥本哈根的楚格（Zug,）建立了一支由 200 多名专家组成的团队。



目前，Swarm 已经获得 Alchemy Capital、BitScale Capital、DFG、KR1、HashKey、NGC Venture、了得资本、P2P Capital、Waterdrip Capital（水

滴资本)、White Paper Capital 和 YBB Foundation LTD 等多家知名机构的投资。

Swarm 空投

Swarm 即将空投 1,000,000 个 BZZ。要成为空投的一部分，你必须运行并安装一个 Bee 节点。对网络的贡献越大，你获得 BZZ 代币的机会就越多。在 Swarm 进入主网之前，你需要从 qBee 节点中兑现支票以获得代币。

活动于 2021 年 2 月 14 日开始，Goerli 区块号为 4283397。

官方博客空投说明：

<https://medium.com/ethereum-swarm/important-notice-regarding-the-1-million-bzz-airdrop-d6b52ddd1af3>

为什么要做这个空投？

除了显而易见的原因——奖励官方的早期采用者之外，官方还想在主网上线之前实现一些事情。

1. 对网络（和组织）进行压力测试
2. 增加活跃的 Bee 节点数量

3. 建立良好的技术支持渠道
4. 提高人们对 Swarm 1.0 的认识。
5. 创建高质量的 FAQ 列表和教程
6. 收集社区的反馈意见

现在官方已经把“为什么”说清楚了，让官方深入了解一下你需要做什么才能成为这次空投的一部分。

你需要做什么才能拿到这些 **BZZ** 代币？

这个很简单。

你必须运行并安装你的 Bee 节点。你需要一个工作正常、连接良好、数据丰富的 Bee 节点来增加你获得 BZZ 的机会。你对网络的贡献越大，你获得 BZZ 的机会就越大。

哪些节点可以获得空投资格？

有资格获得 BZZ 代币的节点是那些已经被证明与官方称之为可信的“蜂后节点”（qBZZ 节点）交换过带宽的节点。

最终网络中会确定刚好 35 个 qBZZ 节点，没有人（甚至团队成员）会知道这些节点是哪些。这将在空投最后时刻决定。

你的节点和 qBZZ 节点之间数据交换的证明是你从 qBZZ 节点收到的支票。

在空投结束时，官方只需运行一个脚本，收集所有从 qBZZ 节点证明支票的节点。兑现的 qBZZ 支票越多，你将获得更多的 BZZ 代币。

你会得到多少代币？

所以，官方送出的 BZZ 代币正好是 100 万个。在空投结束后，官方会检查有多少 qBZZ 支票。这将告诉官方有多少节点有权获得 BZZ。qBZZ/BZZ 比率将告诉官方一张支票的价值。

举个例子：

在空投结束时，有 50 万张 qBZZ 支票从网络中的所有节点收集。在这种情况下，一张支票的价值是：1M BZZ 除以 50 万=每张 qBZZ 支票价值 2 个 BZZ 代币。

因此，如果你的节点收集了 50 张 qBZZ 支票，你将收到 $50 \times 2\text{BZZ} = 100$ 个 BZZ 代币。如果你运行了 3 个节点，他们每个节点收集了 50 张支票，那么就会从空投中获得 300 个 BZZ。

空投何时结束？

空投将在官方上线主网之前结束，估计是 2021 年上半年。

在空投结束前，你应该做的一个重要步骤！

这一点非常重要——当空投结束时，你需要兑现你的 qBZZ 支票。如果没有这一步，一旦官方进入主网，你将不会收到你的 BZZ 代币。

教程：部署节点，获取空投

bee 节点可以部署在 linux、windows 系统上，本文先介绍 Windows 系统上的相关教程。准备条件：

一台配置较高的 Win10 64 位电脑

100G 以上剩余磁盘空间

火狐浏览器

自备梯子，下载相关软件用

软件包里面的压缩包软件，需要解压到当前目录。

1、下载 bee 节点软件。

<https://github.com/ethersphere/bee/releases/download/v0.5.3/bee-windows-amd64.exe>

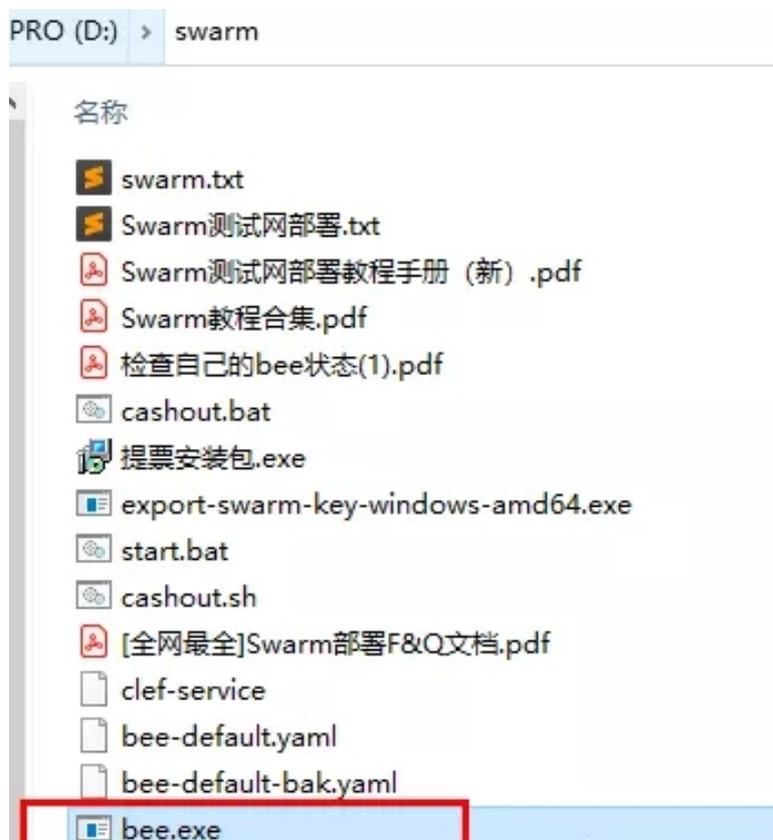
<https://github.com/ethersphere/bee/releases>

▼ Assets 19

-  [bee-darwin-amd64](#)
-  [bee-darwin-amd64.tar.gz](#)
-  [bee-linux-386](#)
-  [bee-linux-amd64](#)
-  [bee-linux-arm64](#)
-  [bee-linux-armv7](#)
-  [bee-windows-386.exe](#)
-  [bee-windows-amd64.exe](#)
-  [bee_0.5.2_386.deb](#)
-  [bee_0.5.2_386.rpm](#)
-  [bee_0.5.2_amd64.deb](#)
-  [bee_0.5.2_amd64.rpm](#)
-  [bee_0.5.2_arm64.deb](#)
-  [bee_0.5.2_arm64.rpm](#)
-  [bee_0.5.2_armv7.deb](#)
-  [bee_0.5.2_armv7.rpm](#)
-  [bee_0.5.2_checksums.txt](#)
-  [Source code \(zip\)](#)
-  [Source code \(tar.gz\)](#)

2、把安装包放到你的空间比较大的磁盘中，比如我这里 D 盘。

修改为 bee.exe，方便操作。



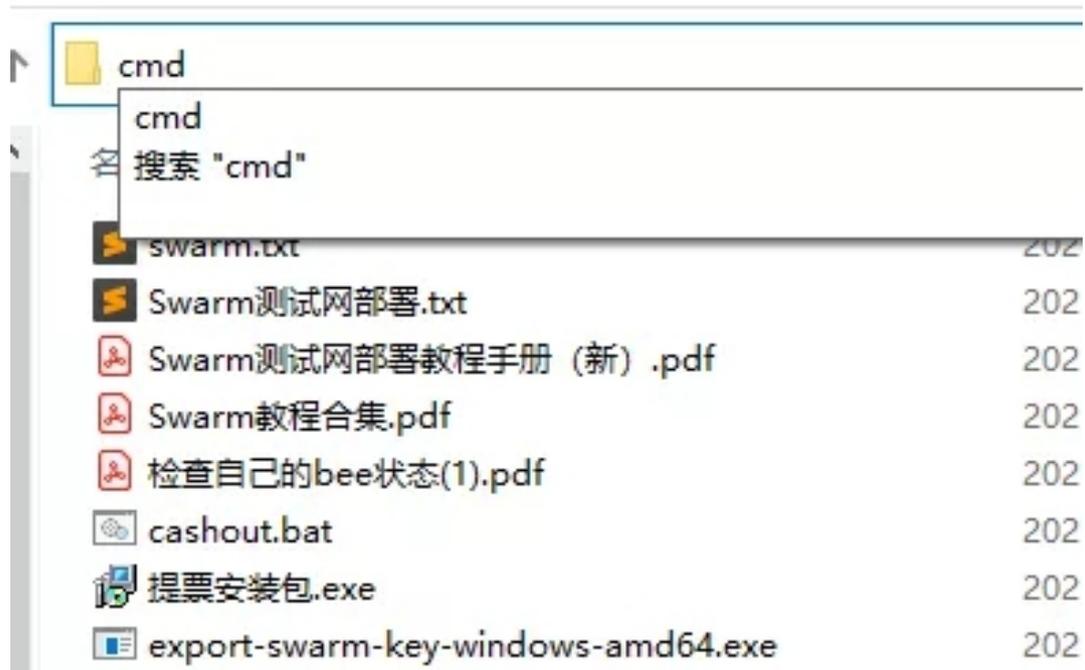
3、用编辑器修改默认配置文件 bee-default.yaml

主要的数据保存目录，一定要是磁盘空间够大的分区。最少 100G 剩余空间。

A screenshot of a Sublime Text editor window titled 'D:\swarm\bee-default.yaml - Sublime Text'. The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. The editor shows the following YAML configuration for 'bee-default.yaml':

```
1 clef-signer-enable: false
2 data-dir: d:/bee
3 swap-enable: true
4 swap-endpoint: https://rpc.slock.it/goerli
5 verbosity: trace
6 welcome-message: "快飞吧, 小蜜蜂!"
7 debug-api-enable: true
8
```

4、在 bee.exe 所在目录地址栏，输入 cmd 回车，打开命令行窗口



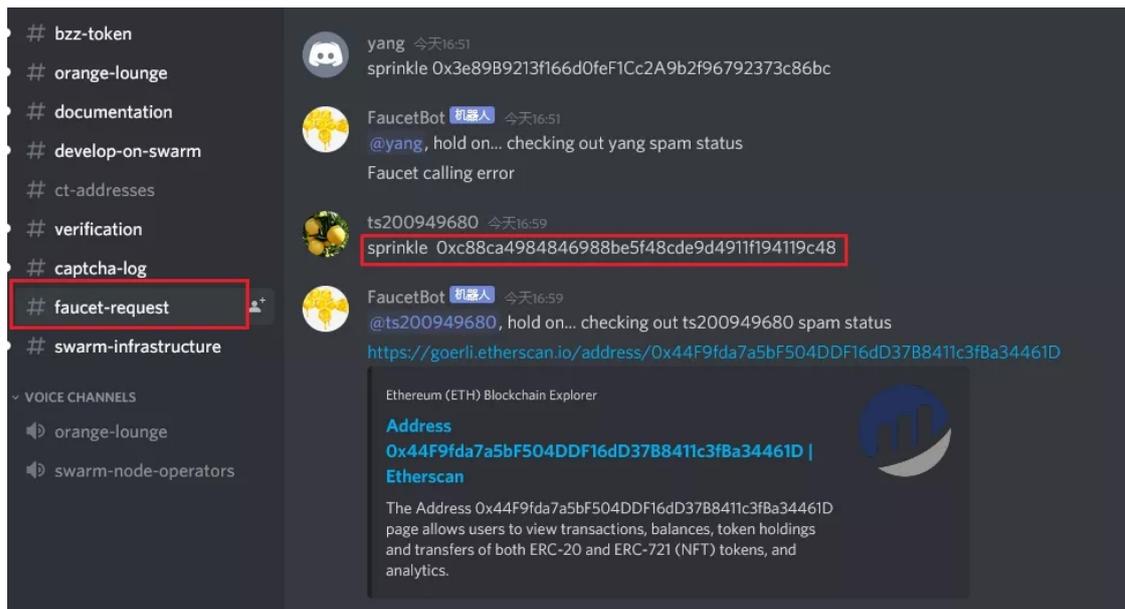
5、在命令行窗口执行以下命令，开启节点。

```
bee.exe start --config bee-default.yaml
```

小技巧：当你输入 bee 之后按一下键盘的 Tab 按键会自动补全文件名

此时是第一次运行 bee 需要输入密码和确认密码之后会看到一个让你注入水龙头代币的钱包地址。

如果没币，会一直重复提醒，直到自动退出。



测试币到帐户，窗口状态发生变化，提示节点运行成功。

```

C:\Windows\system32\cmd.exe
time="2021-03-22T11:56:08+08:00" level=trace msg="waiting for transaction fdf55636508756a60f3051094cae3b525e70464c76411d7e41287d7f383fec36 to be mined: not found"
time="2021-03-22T11:56:09+08:00" level=trace msg="waiting for transaction fdf55636508756a60f3051094cae3b525e70464c76411d7e41287d7f383fec36 to be mined: not found"
time="2021-03-22T11:56:10+08:00" level=trace msg="waiting for transaction fdf55636508756a60f3051094cae3b525e70464c76411d7e41287d7f383fec36 to be mined: not found"
time="2021-03-22T11:56:11+08:00" level=trace msg="waiting for transaction fdf55636508756a60f3051094cae3b525e70464c76411d7e41287d7f383fec36 to be mined: not found"
time="2021-03-22T11:56:13+08:00" level=trace msg="waiting for transaction fdf55636508756a60f3051094cae3b525e70464c76411d7e41287d7f383fec36 to be mined: not found"
time="2021-03-22T11:56:14+08:00" level=trace msg="waiting for transaction fdf55636508756a60f3051094cae3b525e70464c76411d7e41287d7f383fec36 to be mined: not found"
time="2021-03-22T11:56:15+08:00" level=info msg="successfully deposited to chequebook"
time="2021-03-22T11:56:16+08:00" level=debug msg="initializing NAT manager"
time="2021-03-22T11:56:28+08:00" level=debug msg="NAT manager initialized"
time="2021-03-22T11:56:28+08:00" level=debug msg="p2p address: /ip4/172.16.22.253/tcp/1634/p2p/16UiU2HAMNdwHpDNaFaQaABBSBgCBrL5QcAKsaxpvJScPjdtPQdjs"
time="2021-03-22T11:56:28+08:00" level=debug msg="p2p address: /ip4/127.0.0.1/tcp/1634/p2p/16UiU2HAMNdwHpDNaFaQaABBSBgCBrL5QcAKsaxpvJScPjdtPQdjs"
time="2021-03-22T11:56:28+08:00" level=debug msg="p2p address: /ip6:::1/tcp/1634/p2p/16UiU2HAMNdwHpDNaFaQaABBSBgCBrL5QcAKsaxpvJScPjdtPQdjs"
time="2021-03-22T11:56:28+08:00" level=info msg="database capacity: 5000000 chunks (approximately 19.5GB)"
time="2021-03-22T11:56:28+08:00" level=info msg="name resolver: no name resolution service provided"
time="2021-03-22T11:56:28+08:00" level=info msg="api address: [::]:1633"
time="2021-03-22T11:56:28+08:00" level=info msg="debug api address: [::]:1635"
time="2021-03-22T11:56:28+08:00" level=trace msg="kademlia iterator took 0s to finish"

```

7、查询各种状态命令

浏览器中输入地址

http://localhost:1633

出现以下字样代表 bee 连接成功



Ethereum Swarm Bee

查看连接的节点数，输入：

<http://localhost:1635/peers>

火狐浏览器显示结果比较直观。方便查看。

localhost:1635/peers

JSON 原始数据 头

保存 复制 全部折叠 全部展开 | 过滤 JSON

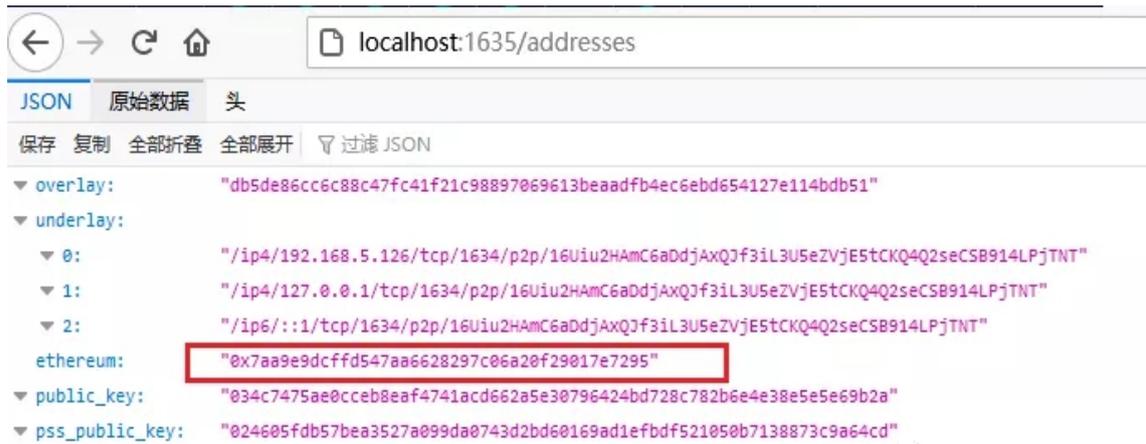
```

peers:
  0:
    address: "0da14616b81c0314e76df5ef86ca393de1d53fa7f44f60c01eca6d8b2b1600e6"
  1:
    address: "25f867e7af0d53fae1de2f11b03f34a3998401f2b8d444d0fea66dc2397fb073"
  2:
    address: "391bbb028580ab6cbcca6f10f8a1167d3b18c51613f4e7de860e925c076d0d7f"
  3:
    address: "86358a243cd9294846c55367e126cb581ca1b4fd7cf31dd530d2ef72b87e6351"
  4:
    address: "9d51cb07ca78578acdeaffa2c15b501f34a2d57b7d0b1d442b96676774ca2fea"
  5:
    address: "a121b60fa5b8a1fe0f93e51464969c7d1d306a863a76e19fee1c046622c4bd7a"
  6:
    address: "a4cd72bf8b7ffd8afffc9f792f71b2016d0753f5e8e4085576eec1c0a5c303a4"
  7:
    address: "c410f0b8f97b76149f76aad808423fef526792a1f76387e3c979425e0459d9d8"
  8:
    address: "c586235b075e5a6c94dfc87394c2b99fa241d228b7a3ccd969e8b793b5d99120"
  9:
    address: "c96f874adeb5276dead777de854c90ceedb18392301b0dd9421c7e0f0e37bb48"
  10:
    address: "ca58f2ac2a6075243a5d2c9eddefd49e01df8f896bd3d4640dafa7a0b67ee3c7"
  11:
    address: "d030ac50caf7fe2844523448f33858a7faf4eaf162f4fa1fc10e17d2356f1a1e"
  12:
    address: "d0c0f281d2c9ed463b83d4c2a11a37c6925ed4d1be0bb36c23b1a2e56f3a250a"
  13:
    address: "d330d804e085ed113315eb0c008ce172fca482d9eb971ce8b5e32964c363482c"
  14:
    address: "d4b152cdac60e0335f21aa1923fa25afe49fe5a73b4acfead20fe3879862b406"
  15:
    address: "d85644a06d49a2dd8d1c6914cf626b9a0c62cc8d355e8c1234053fbf3ce683d4"

```

查看你的 bee 钱包地址。就是上面提到的那个，输入：

<http://localhost:1635/addresses>



输入: <http://localhost:1635/chequebook/address> 这是你的账本合约地址，
提取支票的记录全在这里包括 IN 和 OUT:



当你有支票后，并且提取之后，拿着这个支票合约地址，到区块浏览器查
你的支票收支和余额状态（红色部分替换成你自己的）直接访问：

<https://goerli.etherscan.io/address/0x480F1b5719519363414F642de0F098c1C7069E16#tokentxns>

查询支票：

<http://localhost:1635/chequebook/cheque>

如下图，lastreceived: null 表示还没收到支票

localhost:1635/chequebook/cheque

JSON 原始数据 头

保存 复制 全部折叠 全部展开 过滤 JSON

```
▼ lastcheques:
  ▼ 0:
    ▼ peer: "d8b8f620c7c4c164f298d1481245dfd0fdb84ba912122dc987fce11323d5cc13"
    lastreceived: null
    ▼ lastsent:
      beneficiary: "0x8226be080bd193cC9344baE76d4389849C46ba15"
      chequebook: "0x480F1b5719519363414F642de0F098c1C7069E16"
      payout: 896200000000
```

有支票的记录:

```
▼ lastcheques:
  ▼ 0:
    ▼ peer: "960605e6dfe1f384c40471ce61c5985d3294e9cd17c89044aee9c221873e715e"
    ▼ lastreceived:
      beneficiary: "0xd59611947a27Ff3DC52D5825d9bb7Eb1c511a84a"
      chequebook: "0xC547dfA1FA3989a145Eb8efE7D251Ef1cEF423Ef"
      payout: 1798000000000
    lastsent: null
  ▼ 1:
    ▼ peer: "96650dc2f1ec194b6cce4dea376f5ec76e5d408399e32879d2b6ed14b9d657ca"
    ▼ lastreceived:
      beneficiary: "0xd59611947a27Ff3DC52D5825d9bb7Eb1c511a84a"
      chequebook: "0x5fB348c9dbBfb184C38fd3888A17806dCf8ab5fd"
      payout: 8993000000000
    lastsent: null
  ▼ 2:
    ▼ peer: "96489c62191eb06ba5bac6086bbb6c8f41857456551d3b950fd7ecb41827ca79"
    lastreceived: null
    ▼ lastsent:
      beneficiary: "0x51cd2E4D080FC3b897900ED8802c28d0a55f8E6a"
      chequebook: "0xF90827af92366b60bb373811C418C7ddC5E67C2d"
      payout: 44912000000000
  ▼ 3:
    ▼ peer: "965630de0bd60efd5dd449a892d3fac47f379bd08e6e6b263bd4a2b72864a7a6"
    lastreceived: null
    ▼ lastsent:
      beneficiary: "0x7633bd0978ceE4dA512841308DfEDa0ffB3845cE"
      chequebook: "0xF90827af92366b60bb373811C418C7ddC5E67C2d"
      payout: 48252000000000
  ▼ 4:
    ▼ peer: "9651106ac19d9b7d9309206eac67072d1a9f25fbca610e673202bbf3f55f67bd"
    lastreceived: null
    ▼ lastsent:
      beneficiary: "0xbB47204811c5b145E60F611ddBA859091374Da50"
      chequebook: "0xF90827af92366b60bb373811C418C7ddC5E67C2d"
      payout: 22420000000000
```

至此，完成全部部署，并正常运行！

Windows 一键部署常见问题：

- 1、需要将默认磁盘改成自己电脑里可用空间最大的磁盘；
- 2、cmd 窗口内输入密码不会有任何显示，输入完成按回车；
- 3、如果闪退就是密码错了，如果等几分钟自己退了就是没有测试币；
- 4、官方水龙头（领测试币）：

<https://faucet.ethswarm.org/>

<http://goerli-faucet.slock.it/>

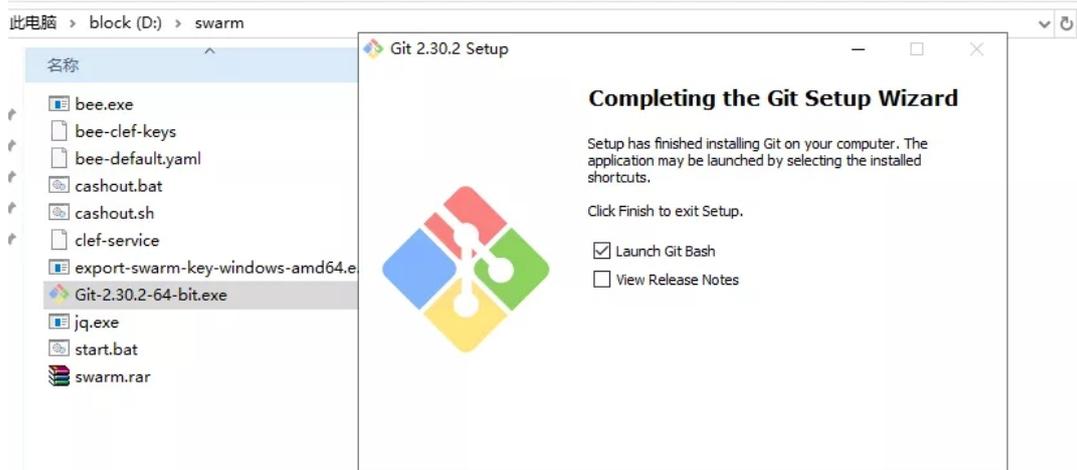
- 5、本地部署不可断电断网和关机；

提取支票

如果获取到支票，需要用相应的软件提取。

安装 git 软件

双击“Git-2.30.2-64-bit.exe”，默认安装 Git

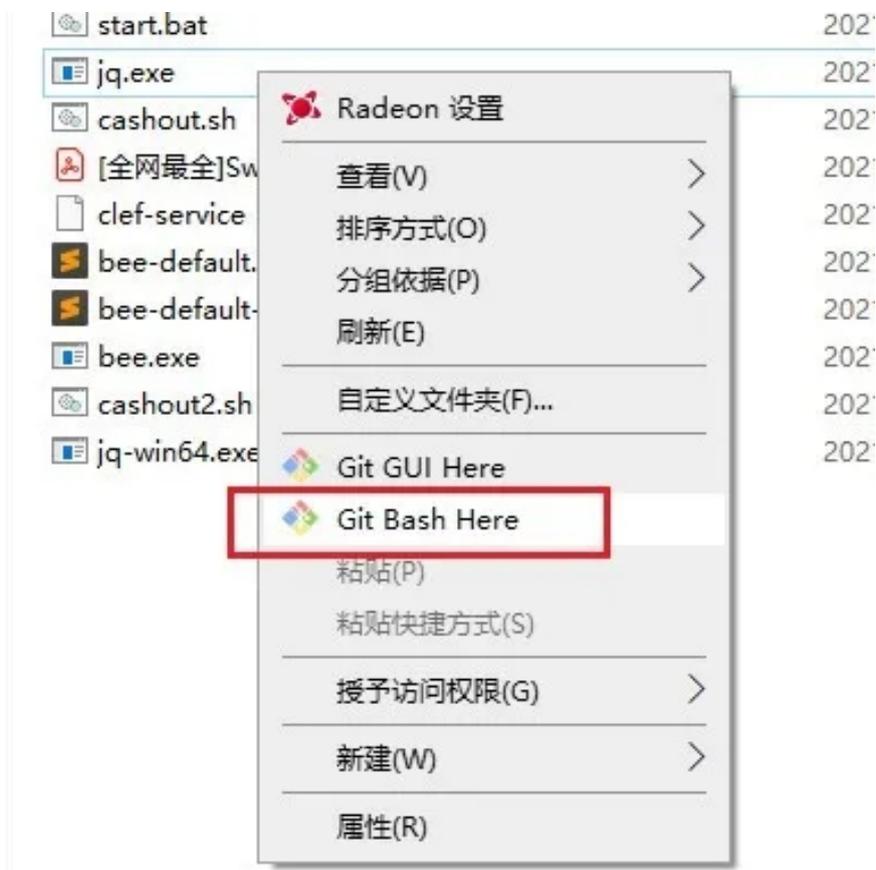


复制软件包里面的 jq.exe 执行文件到

C:\Program Files\Git\usr\bin

目录下

在 swarm 目录空白处，右键点击，运行 gitbash 窗口。

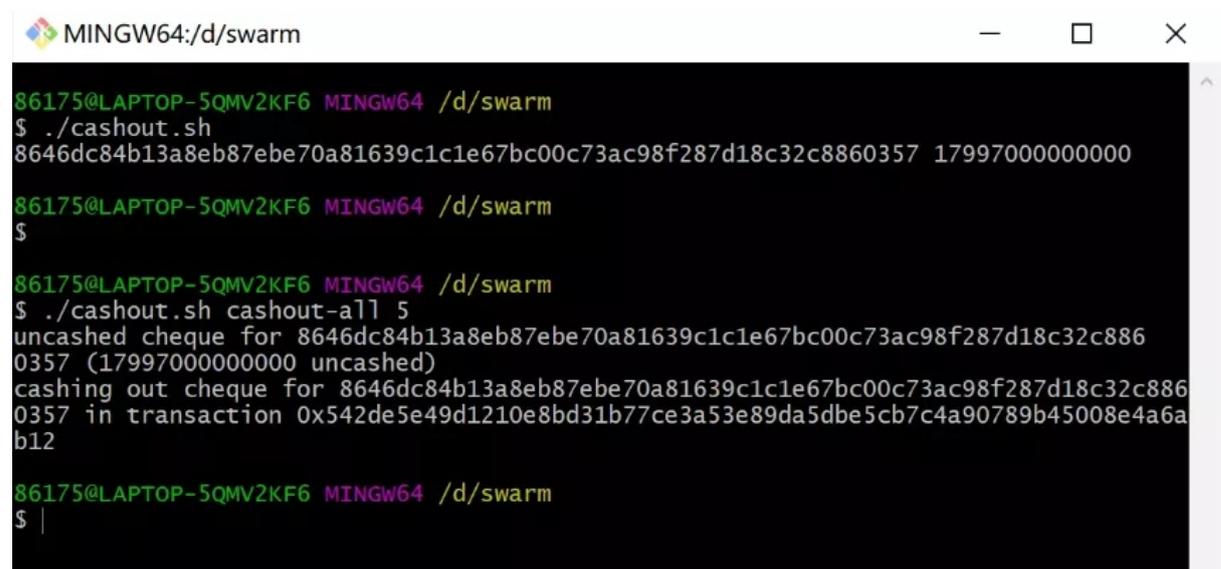


查询支票

```
./cashout.sh
```

领取支票

./cashout.sh cashout-all 5 或者，用目录中的 bat 批处理执行文件，执行命令。



```
MINGW64:/d/swarm
86175@LAPTOP-5QMV2KF6 MINGW64 /d/swarm
$ ./cashout.sh
8646dc84b13a8eb87ebe70a81639c1c1e67bc00c73ac98f287d18c32c8860357 17997000000000

86175@LAPTOP-5QMV2KF6 MINGW64 /d/swarm
$

86175@LAPTOP-5QMV2KF6 MINGW64 /d/swarm
$ ./cashout.sh cashout-all 5
uncashed cheque for 8646dc84b13a8eb87ebe70a81639c1c1e67bc00c73ac98f287d18c32c886
0357 (17997000000000 uncashed)
cashing out cheque for 8646dc84b13a8eb87ebe70a81639c1c1e67bc00c73ac98f287d18c32c886
0357 in transaction 0x542de5e49d1210e8bd31b77ce3a53e89da5dbe5cb7c4a90789b45008e4a6a
b12

86175@LAPTOP-5QMV2KF6 MINGW64 /d/swarm
$ |
```

修改 cashout.bat

```
start C:\"Program Files"\Git\git-bash.exe -c
"/d/swarm/cashout.sh cashout-all 5"
```

参数第一部分为 git-bash 安装路径

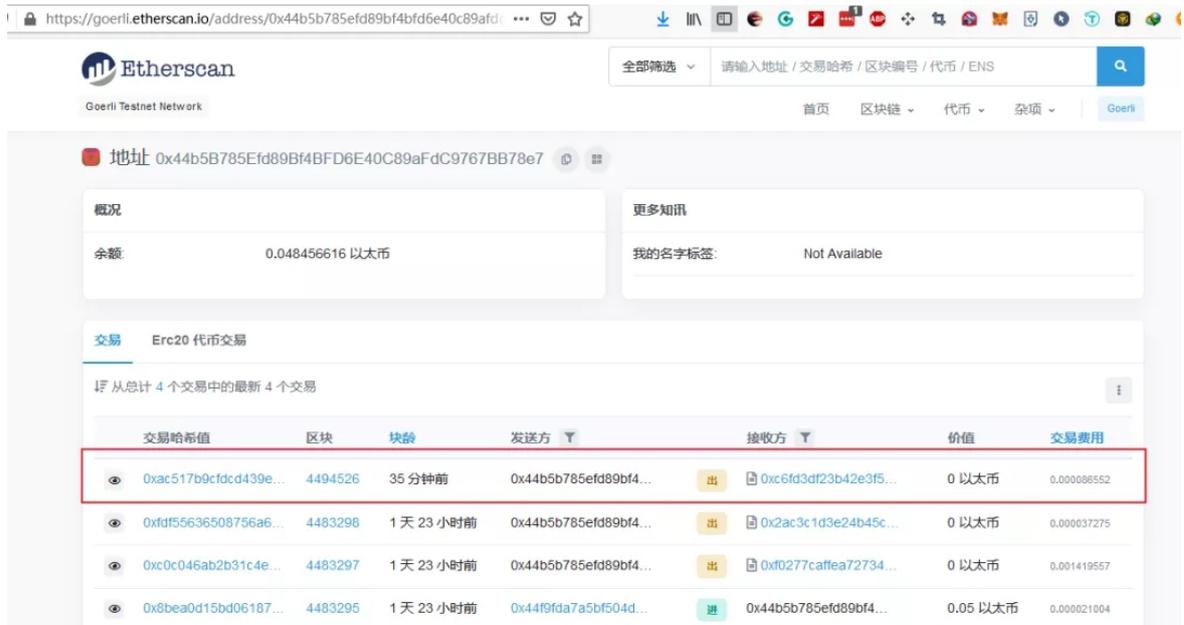
第二部分为 cashout.sh 的保存路径

改好后运行，即可领取支票，正常弹出 git-bash 小窗口，然后显示 cashing out xxx 就行了

据说这个支票是有有效期的，貌似一周，所以，两三天想起来就领一次吧。

也可以自己设置计划任务，定时跑，省的忘了

支票领取成功，可以在浏览器上看到一笔 out 的合约交互记录



The screenshot shows the Etherscan.io interface for a wallet address: 0x44b5b785efd89bf4bfd6e40c89afd... The balance is 0.048456616 ETH. The transaction history shows four transactions, with the first one highlighted in red. The first transaction is an outgoing transaction (out) of 0 ETH to address 0xc6fd3df23b42e3f5... with a transaction fee of 0.000086552 ETH.

交易哈希值	区块	块龄	发送方	接收方	价值	交易费用
0xac517b9cfdc439e...	4494526	35 分钟前	0x44b5b785efd89bf4...	出 0xc6fd3df23b42e3f5...	0 以太币	0.000086552
0xfdf5636508756a6...	4483298	1 天 23 小时前	0x44b5b785efd89bf4...	出 0x2ac3c1d3e24b45c...	0 以太币	0.000037275
0xc0c046ab2b31c4e...	4483297	1 天 23 小时前	0x44b5b785efd89bf4...	出 0xf0277cffee72734...	0 以太币	0.001419557
0x8bea0d15bd06187...	4483295	1 天 23 小时前	0x44f9fda7a5b504d...	进 0x44b5b785efd89bf4...	0.05 以太币	0.000021004

如果关闭节点软件执行窗口后，可以执行目录内 `start.bat` 批处理命令，重新上线节点。

导出私钥

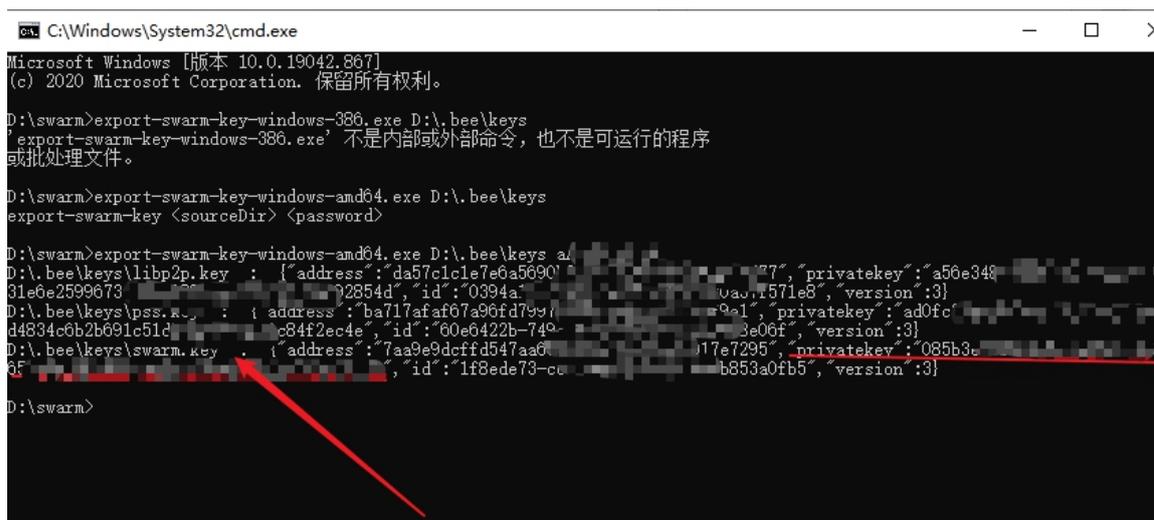
节点的钱包地址需要软件导出对应的私钥，可以导入小狐狸 `metamask` 钱包。

下载官方的 windows 私钥导出软件：

<https://github.com/ethersphere/exportSwarmKey/releases/download/v0.1.0/export-swarm-key-windows-amd64.exe>

在命令行窗口执行命令，swarm.key 后面的引号内文本就是私钥，整段文本复制保存好。

```
export-swarm-key-windows-amd64.exe D:\.bee\keys 你的密码
```



```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19042.867]
(c) 2020 Microsoft Corporation. 保留所有权利。

D:\swarm>export-swarm-key-windows-386.exe D:\.bee\keys
'export-swarm-key-windows-386.exe' 不是内部或外部命令，也不是可运行的程序
或批处理文件。

D:\swarm>export-swarm-key-windows-amd64.exe D:\.bee\keys
export-swarm-key <sourceDir> <password>

D:\swarm>export-swarm-key-windows-amd64.exe D:\.bee\keys a
D:\.bee\keys\libp2p.key : {"address":"da57c1c1e7e0a56901", "privatekey":"a56e344
31e6e2599673", "id":"0394a", "version":3}
D:\.bee\keys\pss.key : {"address":"ba17afaf07a96fd7997", "privatekey":"ad0fc
d4834c6b2b691c51d", "id":"60e6422b-749", "version":3}
D:\.bee\keys\swarm.key : {"address":"7aa9e9dcffd547aa6", "privatekey":"085b3
3f", "id":"1f8ede73-c", "version":3}

D:\swarm>
```

相关网站

官网：<https://swarm.ethereum.org/>

推特：<https://twitter.com/ethswarm>

博客：<https://ethswarm.medium.com/>

Github：<https://github.com/ethersphere>

discord：<https://discord.gg/ykCupZMuww>

电报：https://t.me/joinchat/AAAAAE_IOhnTT94O-vNyFQ