

1. 安裝GO : <https://go.dev/dl/>

2. 安裝 GO 1.16.3

2.1 下滑至Archived versions

2.2 找到 go 1.16.3 並打開摺疊

2.3 點選 go 1.16.3.windows-386.msi



Downloads

After downloading a binary release suitable for your system, please follow the [installation instructions](#).

If you are building from source, follow the [source installation instructions](#).

See the [release history](#) for more information about Go releases.

As of Go 1.13, the go command by default downloads and authenticates modules using the Go module mirror and Go checksum database run by Google. See <https://proxy.golang.org/privacy> for privacy information about these services and the go command [documentation](#) for configuration details including how to disable the use of these servers or use different ones.

Featured downloads

Microsoft Windows

Windows 7 or later, Intel 64-bit processor

[go1.19.4.windows-amd64.msi](#)

(135MB)

Apple macOS (ARM64)

macOS 11 or later, Apple 64-bit processor

[go1.19.4.darwin-arm64.pkg](#)

(139MB)

Apple macOS (x86-64)

macOS 10.13 or later, Intel 64-bit processor

[go1.19.4.darwin-amd64.pkg](#)

(145MB)

1. 下載 GO : <https://go.dev/dl/>

2. 下載 GO 1.16.3

2.1 下滑至Archived versions

2.2 找到 go 1.16.3 並打開摺疊

2.3 點選 go 1.16.3.windows-386.msi

 [go1.18.9.windows-arm64.msi](#) Installer Windows ARM64 101MB 843088615326db56ca905c0f9beda55236d723855c3e5291c04ca:

Unstable version

[go1.20rc1](#) ▶

Archived versions ▾

[go1.19.3](#) ▶

[go1.19.2](#) ▶

[go1.19.1](#) ▶

[go1.19](#) ▶

[go1.18.8](#) ▶

[go1.18.7](#) ▶

[go1.18.6](#) ▶

[go1.18.5](#) ▶

3. 安裝 Go 1.16.3

Downloads - The Go Programming Language

go.dev/dl/

Gmail YouTube 地圖 Gmail YouTube 地圖

go1.16.3.darwin-amd64.tar.gz Archive macOS x86-64 124MB 6bb1cf421f8abc2a9a4e39140b7397cdae6aca3e8d36dcff39a1a; 78b1f3b027a8cbc006f6044baaaafc7dd3241f5b55d9e4ab19564da4fad192012f6d7b0b916061eaa41e59b44f02199ccb2442f5e217c; b0f01ba784590c4c760943665fdc4fe0adaa; 7457b11d80e8c17fefef3ef6de423aaba4b1e; c18b1f064dbfc3e265ce4a775a23177ca17e; 883f97d9db74d3d6d80d5fec77455c6ada6c; 5f81486f0df44f3088c3ed47a3bec4099d8ec7dac7f12a63eedc73543e6da0f6017990e214; 1846a4303884cb5a01abb903e3e61e33459a; 6fa75b5288678ef397e76c198cfb26b8ae086f

go1.16.3.darwin-amd64. **Go Programming Language 386 go1.16.3 Setup**

Welcome to the Go Programming Language 386 go1.16.3 Setup Wizard

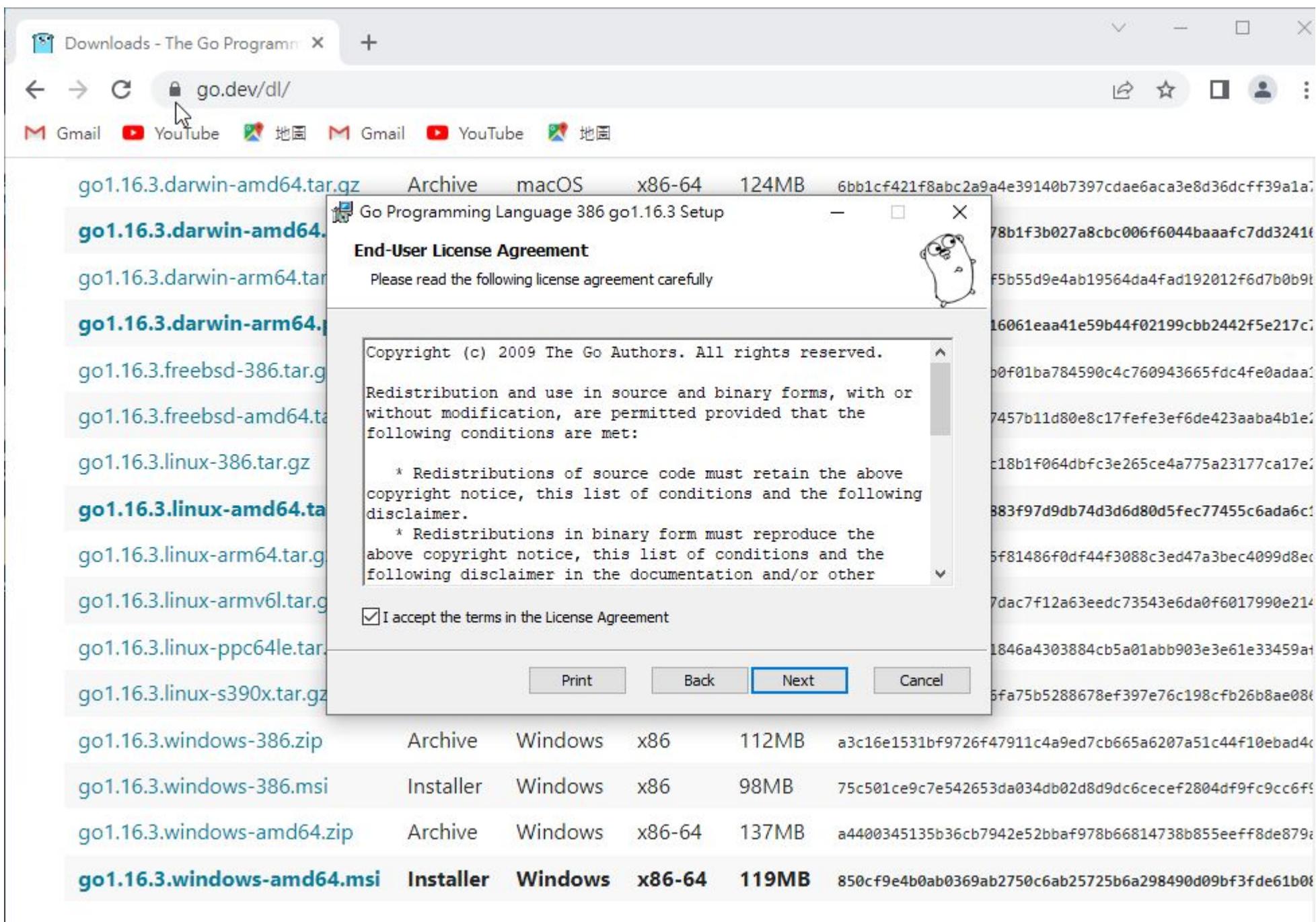
The Setup Wizard will install Go Programming Language 386 go1.16.3 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

http://golang.org

Back **Next** Cancel

go1.16.3.windows-386.zip	Archive	Windows	x86	112MB	a3c16e1531bf9726f47911c4a9ed7cb665a6207a51c44f10ebad4c	
go1.16.3.windows-386.msi	Installer	Windows	x86	98MB	75c501ce9c7e542653da034db02d8d9dc6cecef2804df9fc9cc6f9	
go1.16.3.windows-amd64.zip	Archive	Windows	x86-64	137MB	a4400345135b36cb7942e52bbaf978b66814738b855eeff8de879a	
go1.16.3.windows-amd64.msi	Installer	Windows	x86-64	119MB	850cf9e4b0ab0369ab2750c6ab25725b6a298490d09bf3fde61b08	

3. 安裝 Go 1.16.3

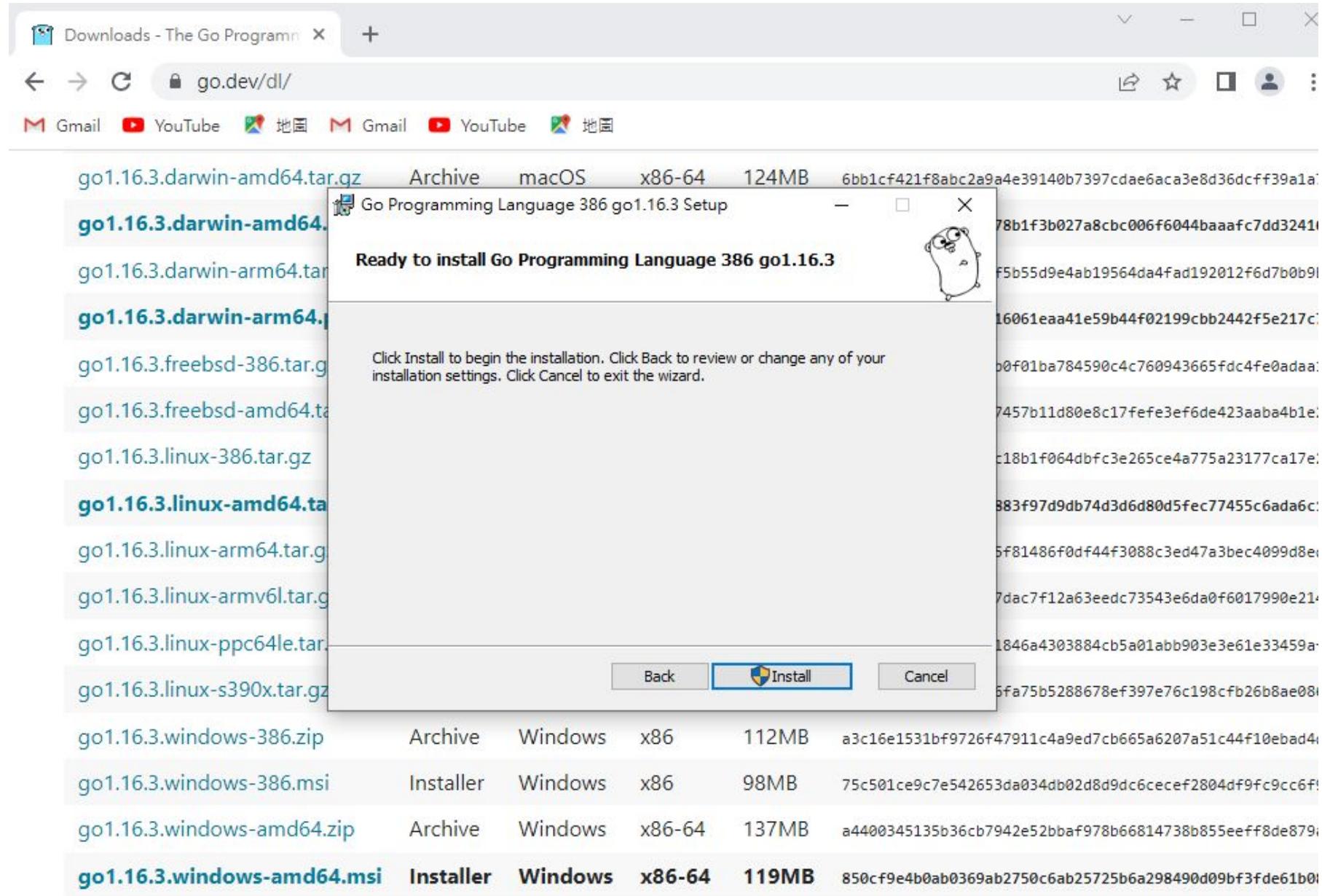


3. 安裝 Go 1.16.3

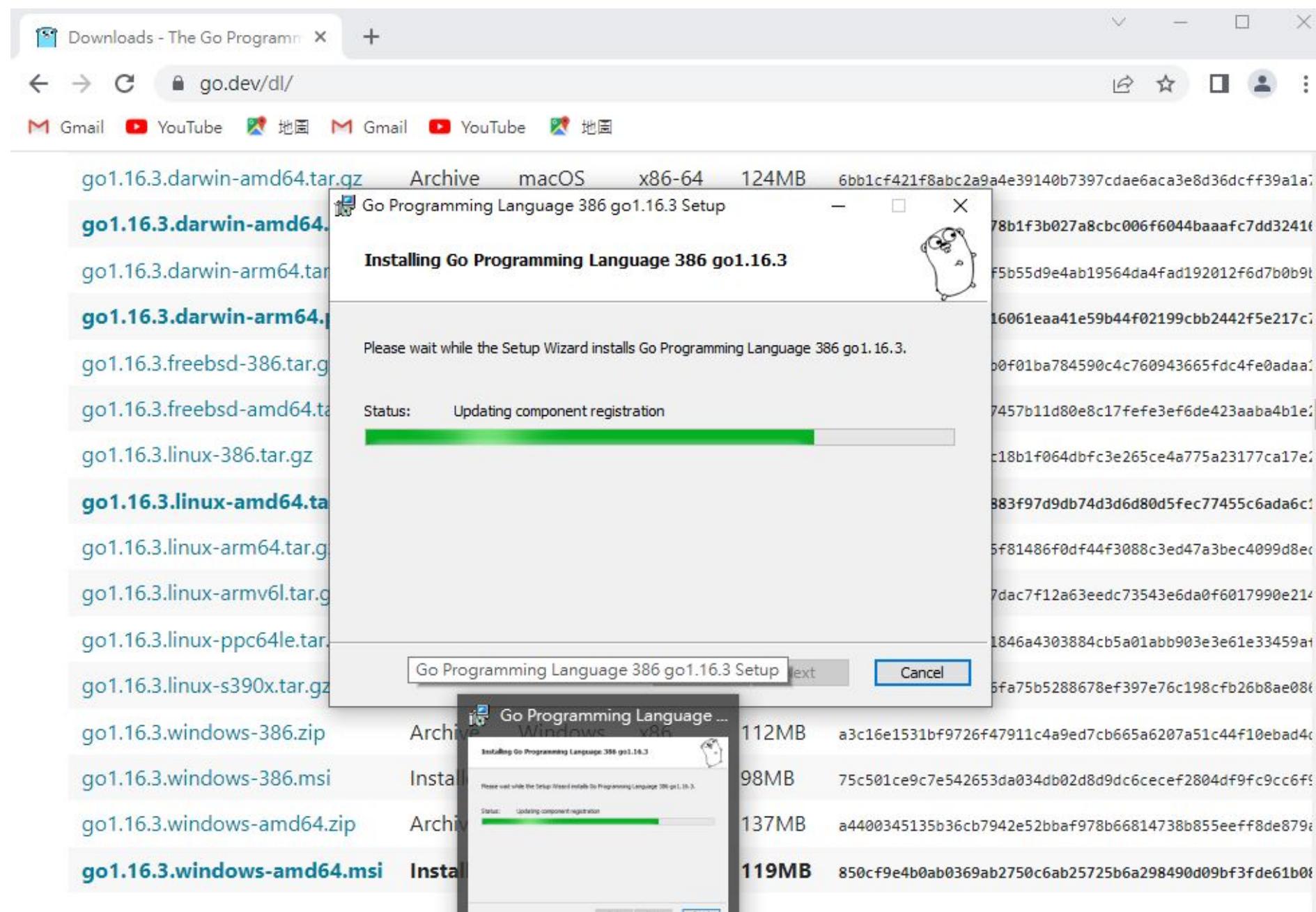
Screenshot of a web browser showing the download page for Go 1.16.3. The page lists various Go binary distributions and their corresponding setup files. A modal window titled "Go Programming Language 386 go1.16.3 Setup" is open, prompting for the destination folder. The user has selected "C:\Program Files (x86)\Go\".

File	Type	Platform	Arch	Size	SHA256
go1.16.3.darwin-amd64.tar.gz	Archive	macOS	x86-64	124MB	6bb1cf421f8abc2a9a4e39140b7397cd...a1a
go1.16.3.darwin-amd64.msi	Install	Windows	x86-64	119MB	850cf9e4b0ab0369ab2750c6ab25725b6a298490d09bf3fde61b0
go1.16.3.darwin-arm64.tar.gz	Archive	macOS	arm64	124MB	78b1f3b027a8cbc006f6044baaa...c7dd3241
go1.16.3.darwin-arm64.msi	Install	Windows	arm64	119MB	850cf9e4b0ab0369ab2750c6ab25725b6a298490d09bf3fde61b0
go1.16.3.freebsd-386.tar.gz	Archive	FreeBSD	x86-64	124MB	f5b55d9e4ab19564da4fad192012f6d7b0b9
go1.16.3.freebsd-amd64.msi	Install	Windows	x86-64	119MB	850cf9e4b0ab0369ab2750c6ab25725b6a298490d09bf3fde61b0
go1.16.3.linux-386.tar.gz	Archive	Linux	x86-64	124MB	16061eaa41e59b44f02199ccb2442f5e217c
go1.16.3.linux-amd64.msi	Install	Windows	x86-64	119MB	850cf9e4b0ab0369ab2750c6ab25725b6a298490d09bf3fde61b0
go1.16.3.linux-arm64.tar.gz	Archive	Linux	arm64	124MB	50f01ba784590c4c760943665fdc4fe0adaa
go1.16.3.linux-armv6l.tar.gz	Archive	Linux	armv6l	124MB	7457b11d80e8c17fefef3ef6de423aab...4b1e
go1.16.3.linux-ppc64le.tar.gz	Archive	Linux	ppc64le	124MB	c18b1f064dbfc3e265ce4a775a23177ca17e
go1.16.3.linux-s390x.tar.gz	Archive	Linux	s390x	124MB	883f97d9db74d3d6d80d5fec77455c6ada6c
go1.16.3.windows-386.zip	Archive	Windows	x86	112MB	5f81486f0df44f3088c3ed47a3bec4099d8e
go1.16.3.windows-386.msi	Install	Windows	x86	98MB	75c501ce9c7e542653da034db02d8d9dc6cecef2804df9fc9cc6f
go1.16.3.windows-amd64.zip	Archive	Windows	x86-64	137MB	a4400345135b36cb7942e52bbaf978b66814738b855eeff8de879
go1.16.3.windows-amd64.msi	Install	Windows	x86-64	119MB	850cf9e4b0ab0369ab2750c6ab25725b6a298490d09bf3fde61b0

3. 安裝 Go 1.16.3



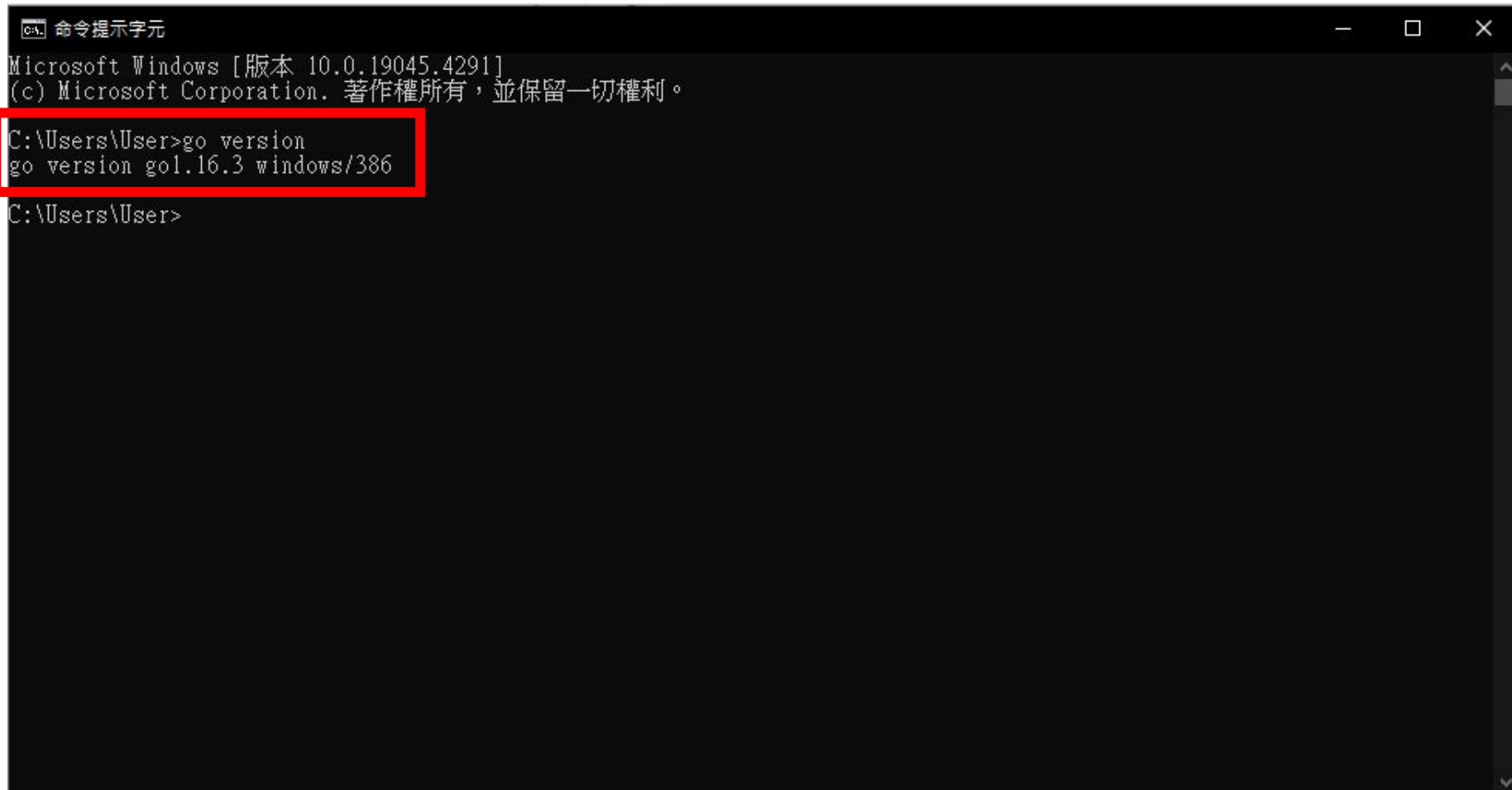
3. 安裝 Go 1.16.3



4. 測試安裝是否成功

4.1 打開CMD

4.2 輸入go version



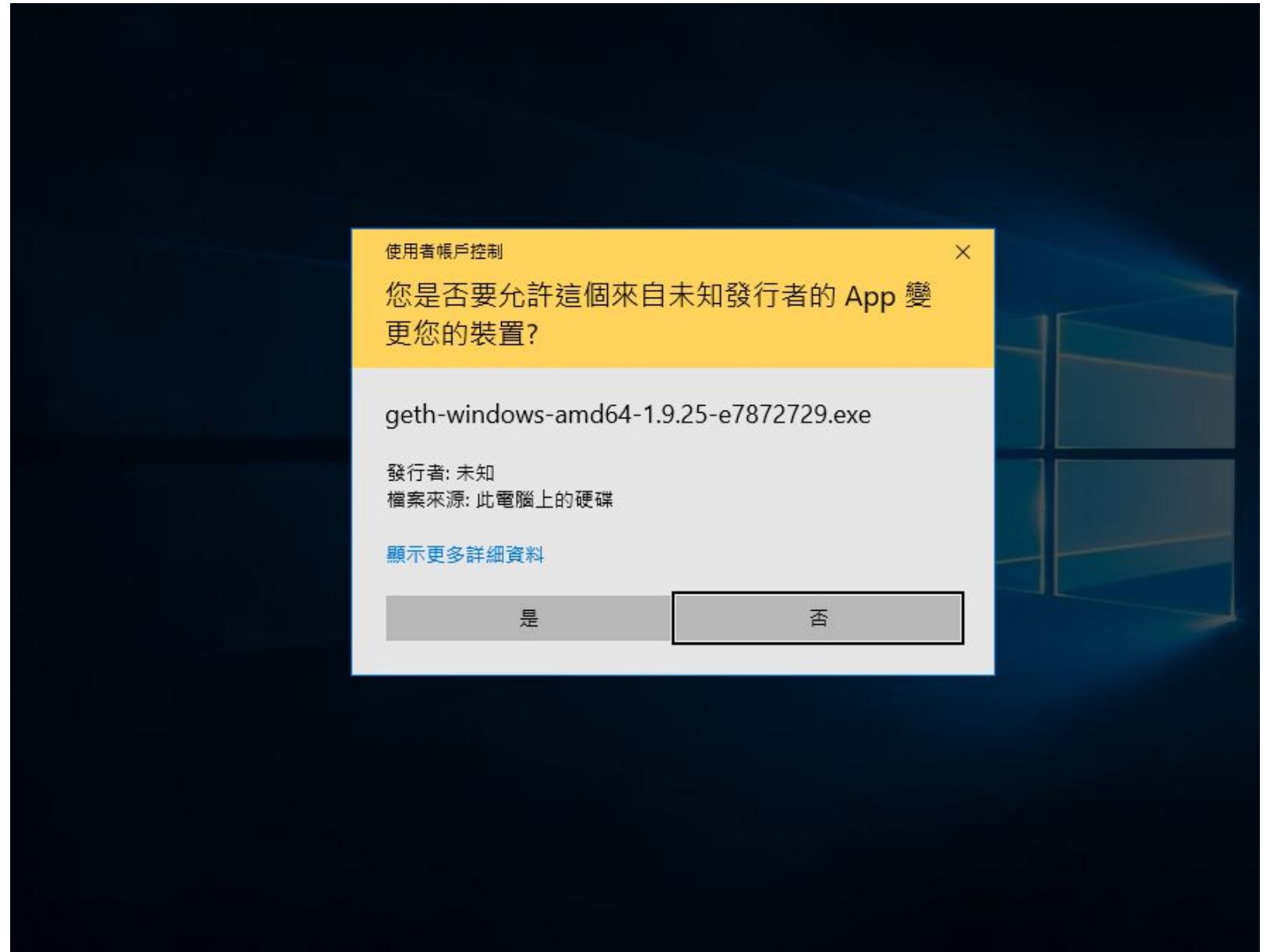
5. 安裝 Geth <https://geth.ethereum.org/downloads/>

6. 下滑安裝 Geth 1.9.25 Installer

Go Ethereum								Search site...	🔍
			Install	Downloads	Documentation				
Geth 1.9.25	e7872729...	Installer	32-bit	64.17 MB	12/11/2020	Signature	a737cf1ae84a7171ce8111f		
Geth 1.9.25	e7872729...	Archive	32-bit	19.12 MB	12/11/2020	Signature	790a610c77d3e1fe9e53e41		
Geth 1.9.25	e7872729...	Installer	64-bit	65.08 MB	12/11/2020	Signature	9b36ff48b7f4ecf6f6d2277		
Geth 1.9.25	e7872729...	Archive	64-bit	19.46 MB	12/11/2020	Signature	4d3607f66a700308863a08c		
Geth & Tools 1.9.25	e7872729...	Archive	32-bit	88.1 MB	12/11/2020	Signature	2c93eb722170d726f5586e5		
Geth & Tools 1.9.25	e7872729...	Archive	64-bit	89.59 MB	12/11/2020	Signature	5cc72efa60bed241859d6b1		
Geth 1.9.24	cc05b050...	Installer	32-bit	64.13 MB	11/13/2020	Signature	4e8cd524fdbb5f051b9d6d8		
Geth 1.9.24	cc05b050...	Archive	32-bit	19.14 MB	11/13/2020	Signature	92e3cdc1743e9de9e23bb3a		

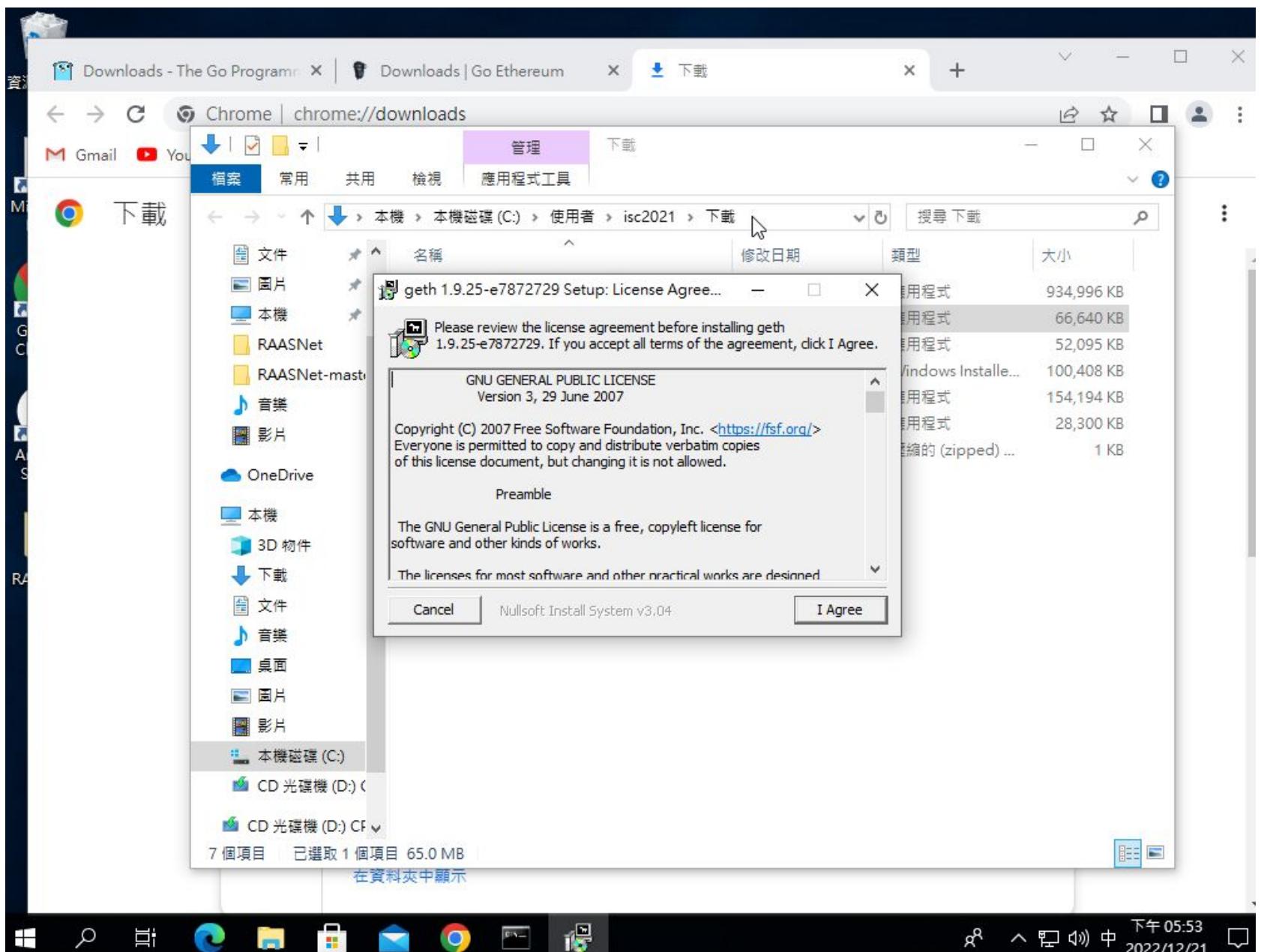
5. 安裝 Geth <https://geth.ethereum.org/downloads/>

6. 下滑安裝 Geth 1.9.25 Installer



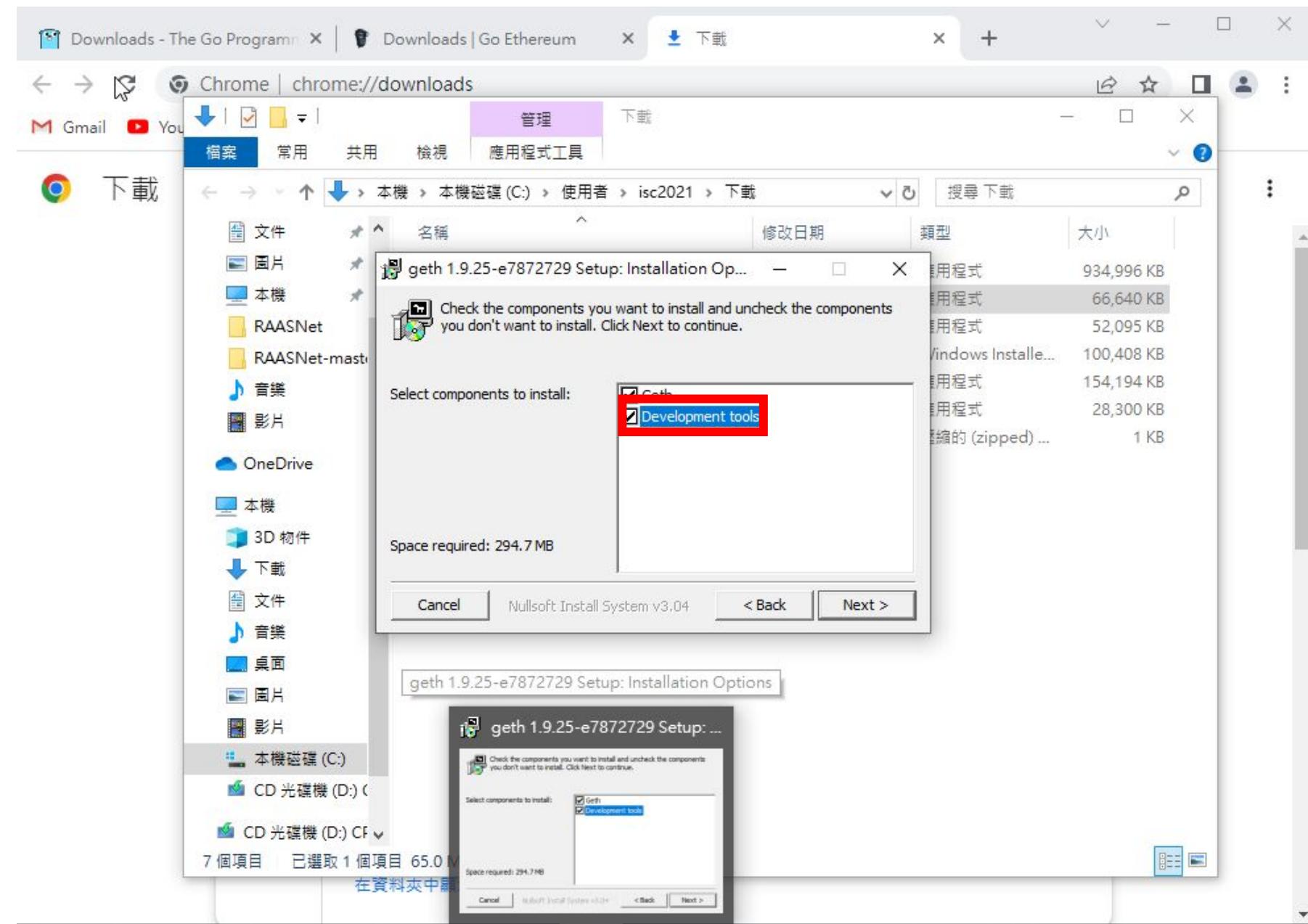
5. 安裝 Geth <https://geth.ethereum.org/downloads/>

6. 下滑安裝 Geth 1.9.25 Installer



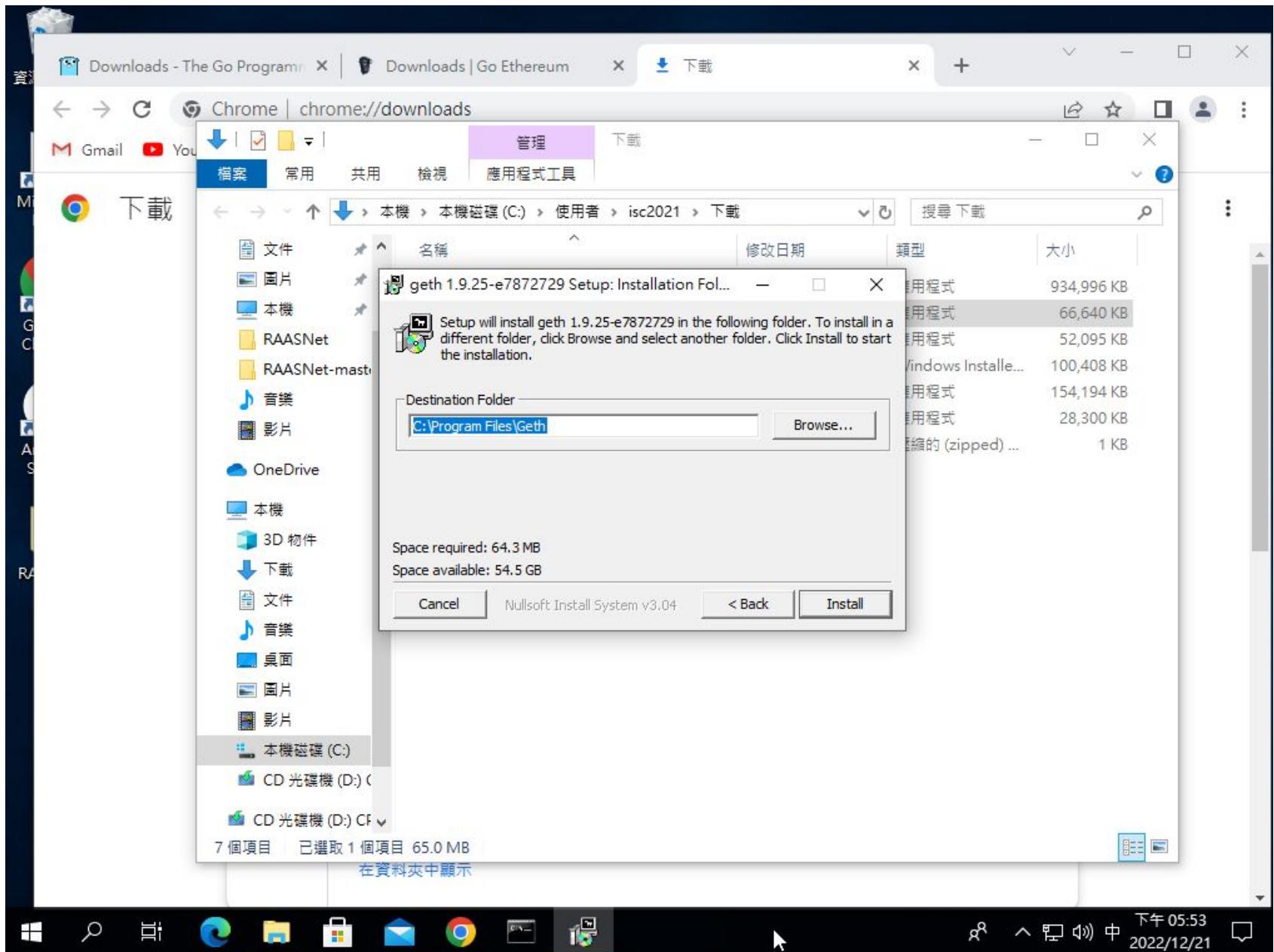
5. 安裝 Geth <https://geth.ethereum.org/downloads/>

6. 下滑安裝 Geth 1.9.25 Installer



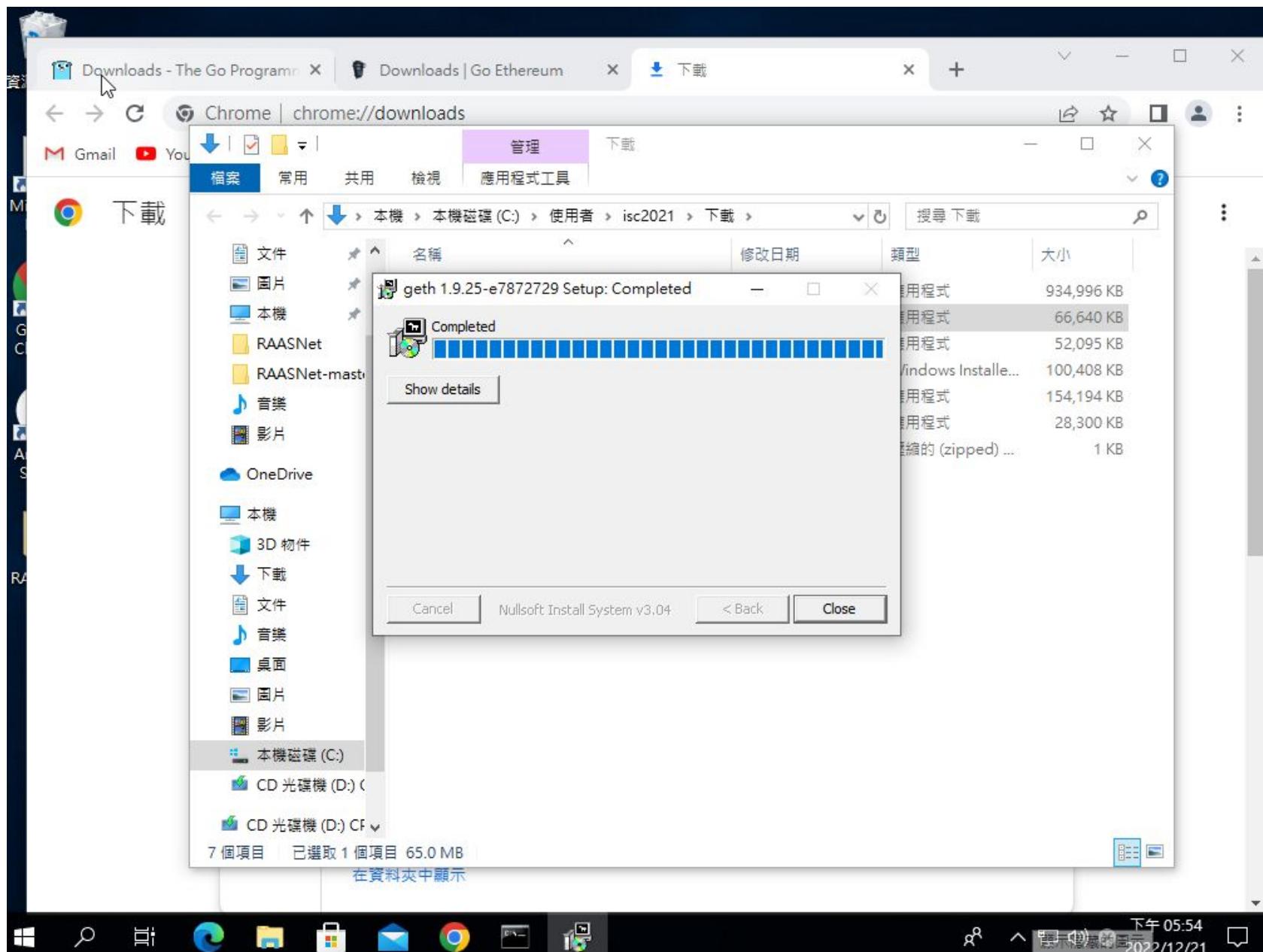
5. 安裝 Geth <https://geth.ethereum.org/downloads/>

6. 下滑安裝 Geth 1.9.25 Installer



5. 安裝 Geth <https://geth.ethereum.org/downloads/>

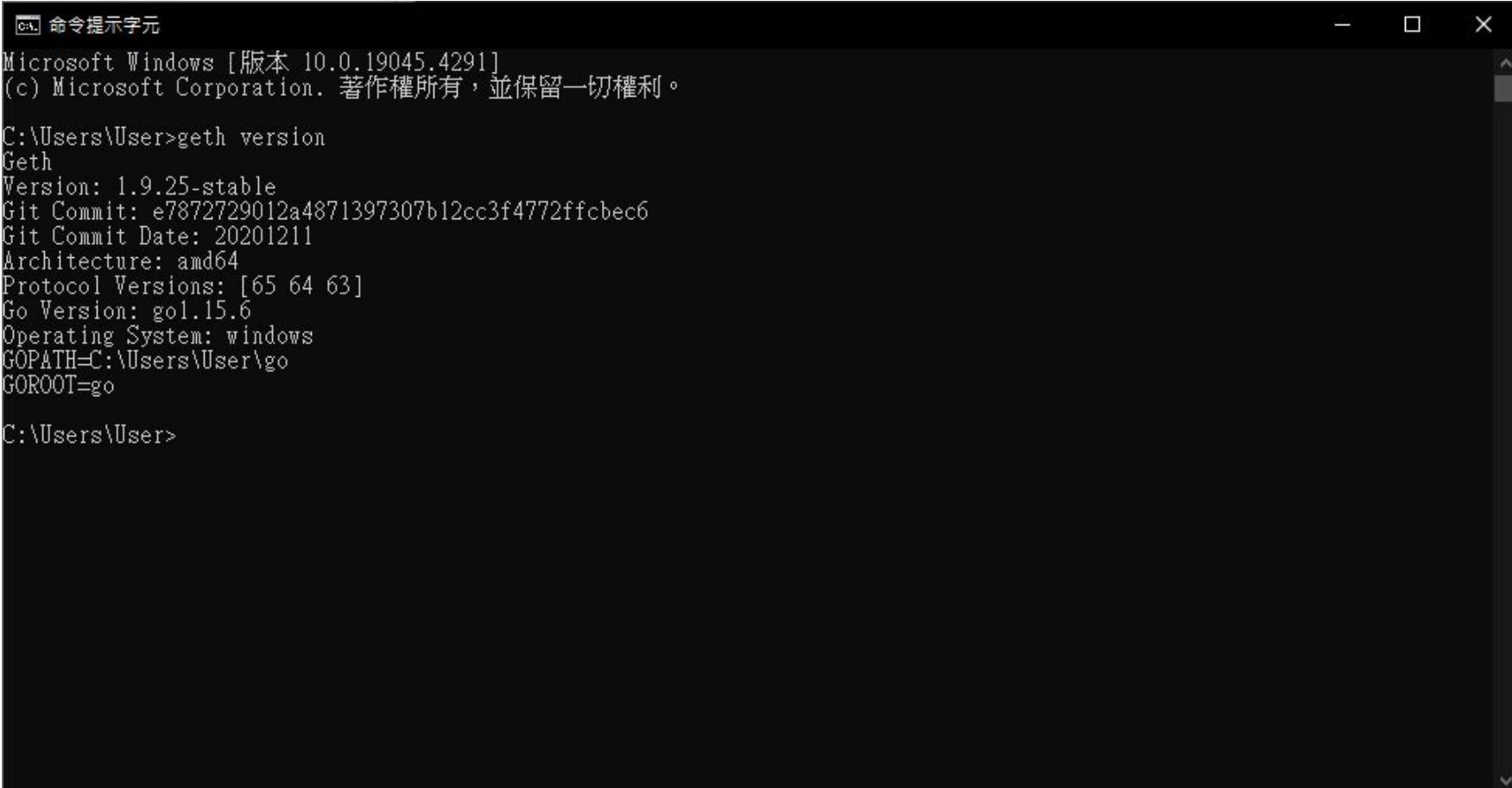
6. 下滑安裝 Geth 1.9.25 Installer



7. 測試安裝是否成功

7.1 打開CMD

7.2 輸入 geth version



```
命令提示字元
Microsoft Windows [版本 10.0.19045.4291]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\User>geth version
Geth
Version: 1.9.25-stable
Git Commit: e7872729012a4871397307b12cc3f4772ffcbec6
Git Commit Date: 20201211
Architecture: amd64
Protocol Versions: [65 64 63]
Go Version: go1.15.6
Operating System: windows
GOPATH=C:\Users\User\go
GOROOT=go

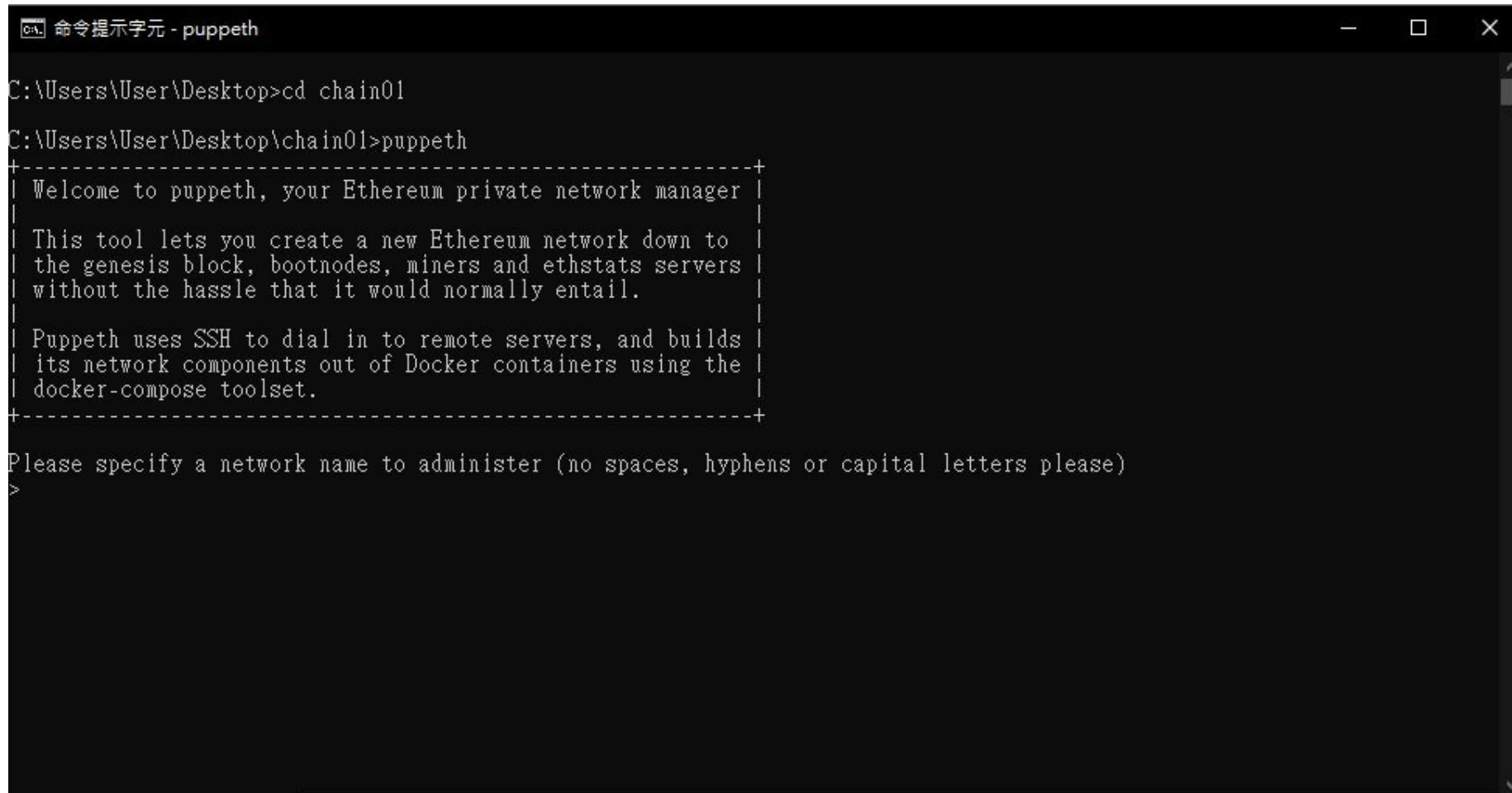
C:\Users\User>
```

8. 利用puppet 架設私鏈

8.1 打開CMD

8.2 CD到指定資料夾

8.3 輸入puppeth進入介面



```
C:\Users\User\Desktop>cd chain01
C:\Users\User\Desktop\chain01>puppeth
+-----+
| Welcome to puppeth, your Ethereum private network manager |
| This tool lets you create a new Ethereum network down to |
| the genesis block, bootnodes, miners and ethstats servers |
| without the hassle that it would normally entail.          |
| Puppet uses SSH to dial in to remote servers, and builds |
| its network components out of Docker containers using the |
| docker-compose toolset.                                     |
+-----+
Please specify a network name to administer (no spaces, hyphens or capital letters please)
>
```

8. 利用puppet 架設私鏈

8.4 輸入區塊鏈網路名稱

```
命令提示字元 - puppeth

C:\Users\User\Desktop\chain01>puppeth
+-----+
| Welcome to puppeth, your Ethereum private network manager |
|
| This tool lets you create a new Ethereum network down to |
| the genesis block, bootnodes, miners and ethstats servers |
| without the hassle that it would normally entail. |
|
| Puppeth uses SSH to dial in to remote servers, and builds |
| its network components out of Docker containers using the |
| docker-compose toolset. |
+-----+
Please specify a network name to administer (no spaces, hyphens or capital letters please)
> yujen_net

Sweet, you can set this via --network=yujen_net next time!

[32mINFO [0m[04-18|01:01:00.482] Administering Ethereum network
[33mWARN [0m[04-18|01:01:00.501] No previous configurations found
[32mname [0m=yujen_net
[33mpath [0m=.puppeth\yujen_net

What would you like to do? (default = stats)
1. Show network stats
2. Configure new genesis
3. Track new remote server
4. Deploy network components
>
```

8. 利用puppet 架設私鏈

8.5 選擇Configure new genesis (2) (它會幫助我們建立初始區塊)

```
命令提示字元 - puppeth
| Welcome to puppeth, your Ethereum private network manager
| This tool lets you create a new Ethereum network down to
| the genesis block, bootnodes, miners and ethstats servers
| without the hassle that it would normally entail.
|
| Puppeth uses SSH to dial in to remote servers, and builds
| its network components out of Docker containers using the
| docker-compose toolset.
+-----+
Please specify a network name to administer (no spaces, hyphens or capital letters please)
> yujen_net

Sweet, you can set this via --network=yujen_net next time!

[32mINFO [0m[04-18|01:01:00.482] Administering Ethereum network
[33mWARN [0m[04-18|01:01:00.501] No previous configurations found
[32mname [0m=yujen_net
[33mpath [0m=.puppeth\yujen_net

What would you like to do? (default = stats)
1. Show network stats
2. Configure new genesis
3. Track new remote server
4. Deploy network components
> 2

What would you like to do? (default = create)
1. Create new genesis from scratch
2. Import already existing genesis
>
```

8. 利用puppet 架設私鏈

8.6 選擇Create new genesis from scratch (1)

```
命令提示字元 - puppeth
|
| Puppet uses SSH to dial in to remote servers, and builds
| its network components out of Docker containers using the
| docker-compose toolset.
+-----+
Please specify a network name to administer (no spaces, hyphens or capital letters please)
> yujen_net
Sweet, you can set this via --network=yujen_net next time!
[32mINFO [0m[04-18|01:01:00.482] Administering Ethereum network
[33mWARN [0m[04-18|01:01:00.501] No previous configurations found
[32mname [0m=yujen_net
[33mpath [0m=.puppeth\yujen_net
What would you like to do? (default = stats)
1. Show network stats
2. Configure new genesis
3. Track new remote server
4. Deploy network components
> 2
What would you like to do? (default = create)
1. Create new genesis from scratch
2. Import already existing genesis
> 1
```

8. 利用puppet 架設私鏈

8.7 選擇共識演算法 (二擇一)

```
命令提示字元 - puppeth
| docker-compose toolset.
+-----+
Please specify a network name to administer (no spaces, hyphens or capital letters please)
> yujen_net

Sweet, you can set this via --network=yujen_net next time!

[32mINFO [0m[04-18|01:01:00.482] Administering Ethereum network
[33mWARN [0m[04-18|01:01:00.501] No previous configurations found
[32mname [0m=yujen_net
[33mpath [0m=.puppeth\yujen_net

What would you like to do? (default = stats)
1. Show network stats
2. Configure new genesis
3. Track new remote server
4. Deploy network components
> 2

What would you like to do? (default = create)
1. Create new genesis from scratch
2. Import already existing genesis
> 1

Which consensus engine to use? (default = clique)
1. Ethash - proof-of-work
2. Clique - proof-of-authority
> 1

Which accounts should be pre-funded? (advisable at least one)
> 0x
```

8. 利用puppet 架設私鏈

8.8 先不設定pre-funded 和 precompile-addresses

8.9 連續點擊兩下Enter略過兩個環節

8.10 設定 chain/network ID

```
命令提示字元 - puppeth
3. Track new remote server
4. Deploy network components
> 2

What would you like to do? (default = create)
1. Create new genesis from scratch
2. Import already existing genesis
> 1

Which consensus engine to use? (default = clique)
1. Ethash - proof-of-work
2. Clique - proof-of-authority
> 1

Which accounts should be pre-funded? (advisable at least one)
> 0x

Should the precompile-addresses (0x1 .. 0xff) be pre-funded with 1 wei? (advisable yes)
>

Specify your chain/network ID if you want an explicit one (default = random)
> 15
[32mINFO [0m[04-18|01:08:20.385] Configured new genesis block

What would you like to do? (default = stats)
1. Show network stats
2. Manage existing genesis
3. Track new remote server
4. Deploy network components
>
```

8. 利用puppet 架設私鏈

8.11 選擇 Manage existing genesis以利我們輸出設定好的初始區塊 (2)

```
命令提示字元 - puppeth

What would you like to do? (default = create)
 1. Create new genesis from scratch
 2. Import already existing genesis
> 1

Which consensus engine to use? (default = clique)
 1. Ethash - proof-of-work
 2. Clique - proof-of-authority
> 1

Which accounts should be pre-funded? (advisable at least one)
> 0x

Should the precompile-addresses (0x1 .. 0xff) be pre-funded with 1 wei? (advisable yes)
>

Specify your chain/network ID if you want an explicit one (default = random)
> 15
[32mINFO [0m[04-18|01:08:20.385] Configured new genesis block

What would you like to do? (default = stats)
 1. Show network stats
 2. Manage existing genesis
 3. Track new remote server
 4. Deploy network components
> 2
```

8. 利用puppet 架設私鏈

8.12 選擇Export genesis configurations 輸出初始區塊 (2)

```
命令提示字元 - puppet
1. Create new genesis from scratch
2. Import already existing genesis
> 1

Which consensus engine to use? (default = clique)
1. Ethash - proof-of-work
2. Clique - proof-of-authority
> 1

Which accounts should be pre-funded? (advisable at least one)
> 0x

Should the precompile-addresses (0x1 .. 0xff) be pre-funded with 1 wei? (advisable yes)
>

Specify your chain/network ID if you want an explicit one (default = random)
> 15
[32mINFO [0m[04-18|01:08:20.385] Configured new genesis block

What would you like to do? (default = stats)
1. Show network stats
2. Manage existing genesis
3. Track new remote server
4. Deploy network components
> 2

1. Modify existing configurations
2. Export genesis configurations
3. Remove genesis configuration
> 2
```

8. 利用puppet 架設私鍊

8.13 直接輸入Enter 讓所有設定檔案放入所在路徑中

```
命令提示字元 - puppeth
Which consensus engine to use? (default = clique)
1. Ethash - proof-of-work
2. Clique - proof-of-authority
> 1

Which accounts should be pre-funded? (advisable at least one)
> 0x

Should the precompile-addresses (0x1 .. 0xff) be pre-funded with 1 wei? (advisable yes)
>

Specify your chain/network ID if you want an explicit one (default = random)
> 15
[32mINFO [0m[04-18|01:08:20.385] Configured new genesis block

What would you like to do? (default = stats)
1. Show network stats
2. Manage existing genesis
3. Track new remote server
4. Deploy network components
> 2

1. Modify existing configurations
2. Export genesis configurations
3. Remove genesis configuration
> 2

Which folder to save the genesis specs into? (default = current)
Will create yujen_net.json, yujen_net-aleth.json, yujen_net-harmony.json, yujen_net-parity.json
>
```

8. 利用puppet 架設私鍊

8.14 輸入 Ctrl+C 離開 Puppeth

```
命令提示字元
4. Deploy network components
> 2

1. Modify existing configurations
2. Export genesis configurations
3. Remove genesis configuration
> 2

Which folder to save the genesis specs into? (default = current)
Will create yujen_net.json, yujen_net-aleth.json, yujen_net-harmony.json, yujen_net-parity.json
>
[32mINFO [0m[04-18|01:11:21.493] Saved native genesis chain spec
[32mINFO [0m[04-18|01:11:21.497] Saved genesis chain spec
t-aleth.json
[32mINFO [0m[04-18|01:11:21.498] Saved genesis chain spec
et-parity.json
[32mINFO [0m[04-18|01:11:21.499] Saved genesis chain spec
net-harmony.json

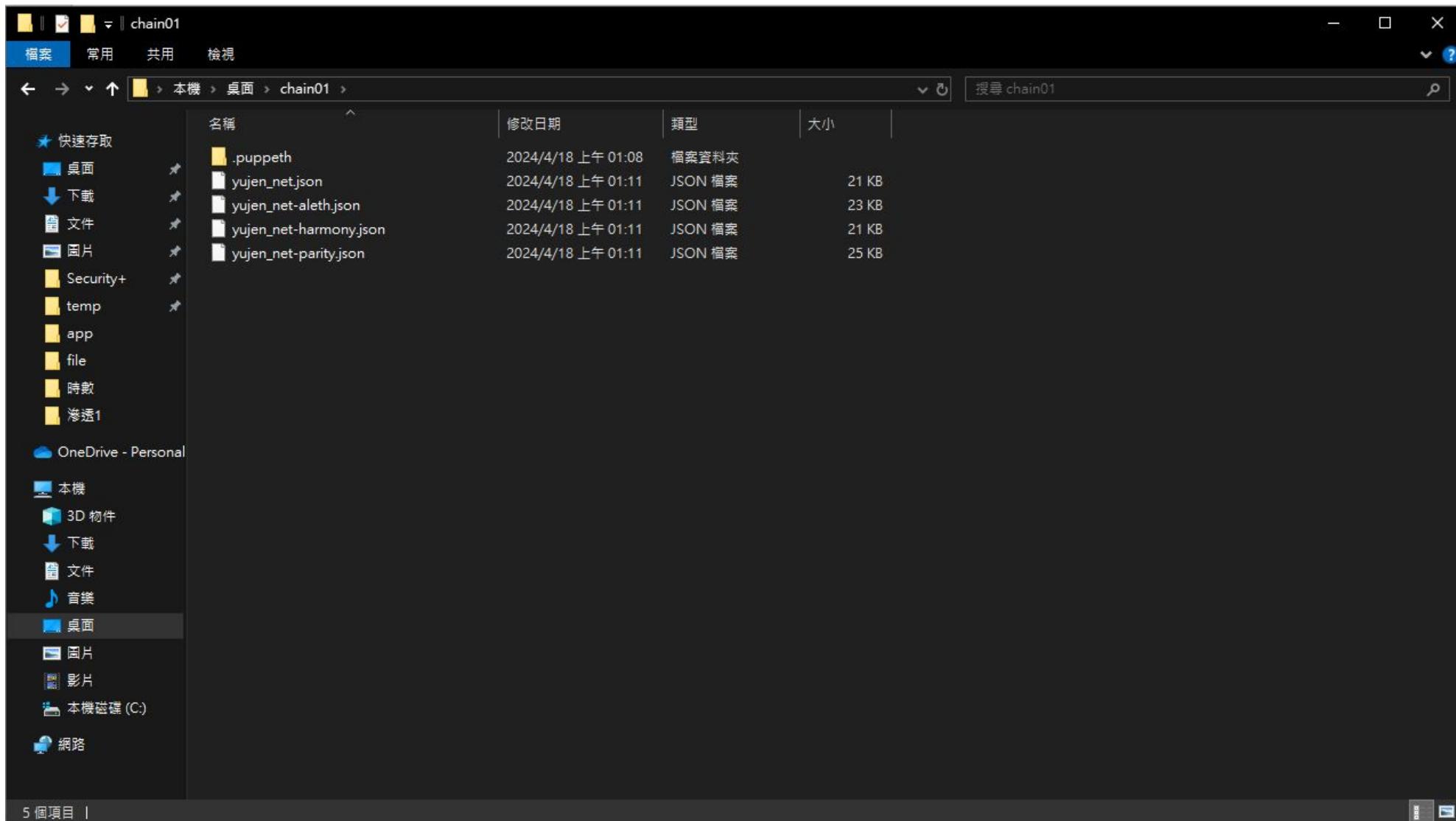
[32mpath [0m=yujen_net.json
[32mcclient [0m=aleth [32mpath [0m=yujen_ne
t-aleth.json
[32mcclient [0m=parity [32mpath [0m=yujen_n
et-parity.json
[32mcclient [0m=harmony [32mpath [0m=yujen_
net-harmony.json

What would you like to do? (default = stats)
1. Show network stats
2. Manage existing genesis
3. Track new remote server
4. Deploy network components
> [35mCRIT [0m[04-18|01:12:35.343] Failed to read user input
[35merr [0m=EOF

C:\Users\User\Desktop\chain01>
```

8. 利用puppet 架設私鍊

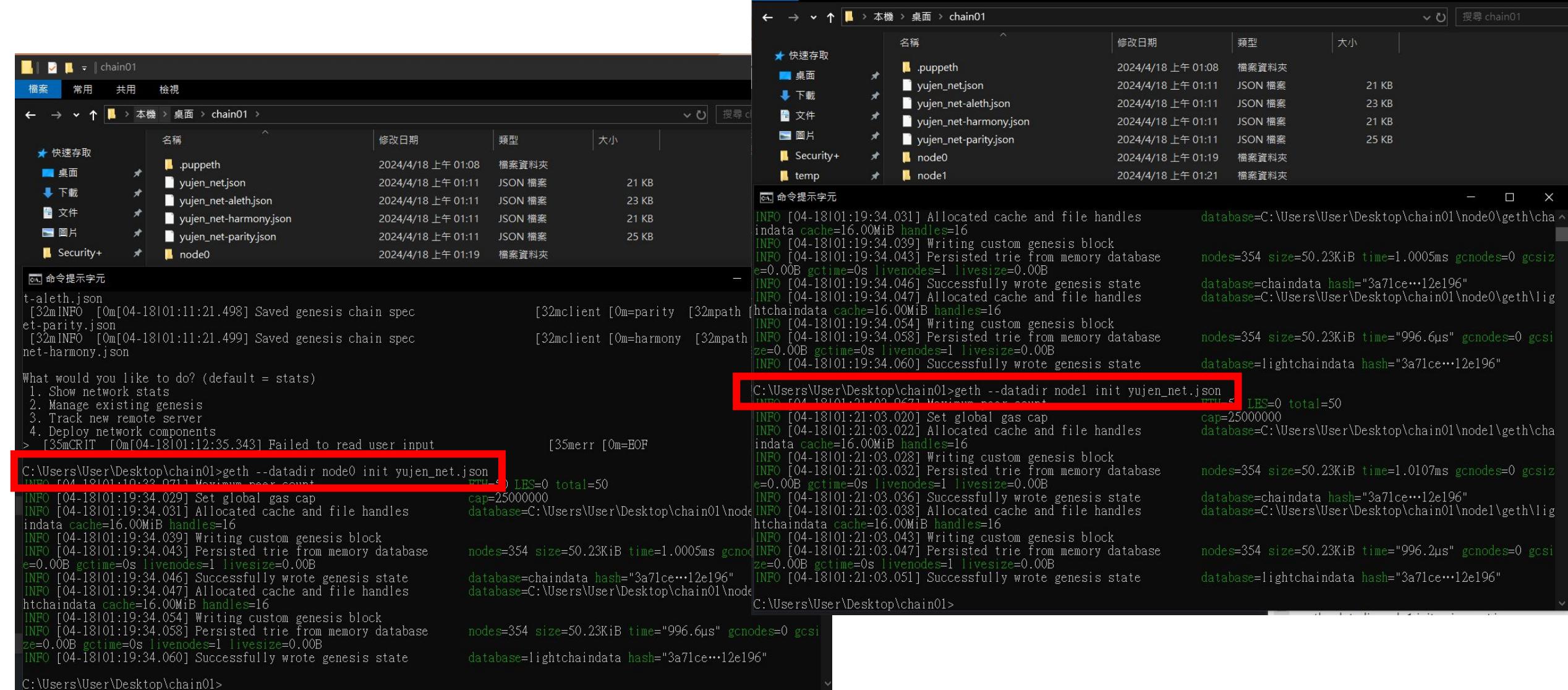
8.15 確認路徑中的文件，初始區塊設定完成。



9. 初始化節點

9.1 於同樣路徑下的CMD中輸入指令 `geth --datadir (node_name) init (net_name).json`

9.2 此處建立兩個節點，即執行上述指令二次。



10. 節點

10.1 CD至區塊鏈所在路徑，開啟節點

node 0:

```
geth --datadir node0 --nodiscover --networkid 15 --port 30015 --rpc --rpcport 8549 --rpcapi "web3,net,eth,admin,personal"  
--rpccorsdomain "*" --ipcdisable --allow-insecure-unlock console
```

node 1:

```
geth --datadir node1 --nodiscover --networkid 15 --port 30017 --rpc --rpcport 8550 --rpcapi "web3,net,eth,admin,personal"  
--rpccorsdomain "*" --ipcdisable --allow-insecure-unlock console
```

10. 節點

10.1 CD至區塊鏈所在路徑，開啟節點 node0

```
命令提示字元 - geth --datadir node0 --nodiscover --networkid 15 --port 30016 --rpc --rpcport 8549 -rpcapi "web3,net,eth,admin,personal" --rp...  
C:\Users\User\Desktop\chain01>geth --datadir node0 --nodiscover --networkid 15 --port 30016 --rpc --rpcport 8549 -rpcapi "web3,net,eth,admin,personal" --rpccorsdomain "*" --ipcdisable --allow-insecure-unlock console  
INFO [04-18|01:23:51.904] Maximum peer count: 50  
WARN [04-18|01:23:51.926] The flag --rpc is deprecated and will be removed in the future, please use --http  
WARN [04-18|01:23:51.927] The flag --rpcport is deprecated and will be removed in the future, please use --http.port  
WARN [04-18|01:23:51.928] The flag --rpccorsdomain is deprecated and will be removed in the future, please use --http.co  
rsdomain  
WARN [04-18|01:23:51.930] The flag --rpcapi is deprecated and will be removed in the future, please use --http.api  
INFO [04-18|01:23:51.961] Set global gas cap: 25000000  
INFO [04-18|01:23:51.963] Allocated trie memory caches: clean=256.00MiB dirty=256.00MiB  
INFO [04-18|01:23:51.964] Allocated cache and file handles: database=C:\Users\User\Desktop\chain01\node0\geth\cha  
in\data\cache=512.00MiB handles=8192  
INFO [04-18|01:23:51.982] Opened ancient database: database=C:\Users\User\Desktop\chain01\node0\geth\cha  
in\data\ancient  
INFO [04-18|01:23:51.985] Initialised chain configuration: config="{ChainID: 15 Homestead: 0 DAO: <nil> DAOSupp  
rt: false EIP150: 0 EIP155: 0 EIP158: 0 Byzantium: 0 Constantinople: 0 Petersburg: 0 Istanbul: 0, Muir Glacier: <nil>, Y  
ODO v2: <nil>, Engine: ethash}"  
INFO [04-18|01:23:51.988] Disk storage enabled for ethash caches: dir=C:\Users\User\Desktop\chain01\node0\geth\ethash c  
ount=3  
INFO [04-18|01:23:51.989] Disk storage enabled for ethash DAGs: dir=C:\Users\User\AppData\Local\Ethash count=2  
INFO [04-18|01:23:51.990] Initialising Ethereum protocol: versions="[65 64 63]" network=15 dbversion=<nil>  
WARN [04-18|01:23:51.992] Upgrade blockchain database version: from=<nil> to=8  
INFO [04-18|01:23:51.994] Loaded most recent local header: number=0 hash="3a71ce...12e196" td=524288 age=19m37s  
INFO [04-18|01:23:51.995] Loaded most recent local full block: number=0 hash="3a71ce...12e196" td=524288 age=19m37s  
INFO [04-18|01:23:51.997] Loaded most recent local fast block: number=0 hash="3a71ce...12e196" td=524288 age=19m37s  
INFO [04-18|01:23:51.999] Regenerated local transaction journal: transactions=0 accounts=0  
INFO [04-18|01:23:52.005] Allocated fast sync bloom: size=512.00MiB  
INFO [04-18|01:23:52.008] Starting peer-to-peer node: instance=Geth/v1.9.25-stable-e7872729/windows-amd64/g  
o1.15.6
```

10. 節點

10.2 建立節點帳戶 personal.newAccount() 並輸入新帳戶的 passphrase

```
命令提示字元 - geth --datadir node0 --nodiscover --networkid 15 --port 30016 --rpc --rpcport 8549 -rpcapi "web3,net,eth,admin,personal" --rp... -  X

o1.15.6
INFO [04-18|01:23:52.009] Initialized fast sync bloom items=354 errorrate=0.000 elapsed=1.993ms
INFO [04-18|01:23:52.015] New local node record seq=1 id=71a770f9708aaafb ip=127.0.0.1 udp=0 tcp=3001
6
INFO [04-18|01:23:52.016] HTTP server started endpoint=127.0.0.1:8549 cors=* vhosts=localhost
INFO [04-18|01:23:52.017] Started P2P networking self="enode://1f50e1d3089009f82a61eb2f0ca71175eb9e933
b2c2c455cc084a8ab09d9c6e31ddfbefffa01ab338af4019af4e9c7332e09306ad213a176fc0913d74fc66326@127.0.0.1:30016?discport=0"
WARN [04-18|01:23:52.050] Served eth_coinbase reqid=3 t=0s err="etherbase must be explicitly specified"
Welcome to the Geth JavaScript console!

instance: Geth/v1.9.25-stable-e7872729/windows-amd64/gol.15.6
at block: 0 (Thu Apr 18 2024 01:04:14 GMT+0800 (CST))
datadir: C:\Users\User\Desktop\chain01\node0
modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d
> INFO [04-18|01:23:54.793] Mapped network port proto=tcp extport=30016 intport=30016 interface="UP
NP IGDv1-IP1"

>
> personal.newAccount()
Passphrase:
Repeat passphrase:
INFO [04-18|01:27:47.888] Your new key was generated address=0xB1B1DA1E9857e922fb38E3Fb44A57Ac4f44778FA
WARN [04-18|01:27:47.891] Please backup your key file! path=C:\Users\User\Desktop\chain01\node0\keystore\UTC
--2024-04-17T17-27-46.846422800Z--b1b1dale9857e922fb38e3fb44a57ac4f44778fa
WARN [04-18|01:27:47.893] Please remember your password!
"0xb1b1dale9857e922fb38e3fb44a57ac4f44778fa"
```

10. 節點

10.3 查詢 node0 的節點資訊 admin.nodeInfo.enode, 並將其複製下來保存 把當中的 IP 改為 127.0.0.1

```
命令提示字元 - geth --datadir node0 --nodiscover --networkid 15 --port 30016 --rpc --rpcport 8549 -rpcapi "web3,net,eth,admin,personal" --rp... ━ ━ ×

[INFO] [04-18|01:23:52.017] Started P2P networking self="enode://1f50e1d3089009f82a61eb2f0ca71175eb9e933...b2c2c455cc084a8ab09d9c6e31ddfbefffa01ab338af4019af4e9c7332e09306ad213a176fc0913d74fc66326@127.0.0.1:30016?discport=0"
[WARN] [04-18|01:23:52.050] Served eth_coinbase reqid=3 t=0s err="etherbase must be explicitly specified"
Welcome to the Geth JavaScript console!

instance: Geth/v1.9.25-stable-e7872729/windows-amd64/g01.15.6
at block: 0 (Thu Apr 18 2024 01:04:14 GMT+0800 (CST))
datadir: C:\Users\User\Desktop\chain01\node0
modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d
> INFO [04-18|01:23:54.793] Mapped network port proto=tcp extport=30016 intport=30016 interface="UPNP IGDv1-IP1"

>
> personal.newAccount()
Passphrase:
Repeat passphrase:
[INFO] [04-18|01:27:47.888] Your new key was generated address=0xB1B1DA1E9857e922fb38E3Fb44A57Ac4f44778FA
[WARN] [04-18|01:27:47.891] Please backup your key file! path=C:\Users\User\Desktop\chain01\node0\keystore\UTC--2024-04-17T17-27-46.846422800Z--b1b1dale9857e922fb38e3fb44a57ac4f44778fa
[WARN] [04-18|01:27:47.893] Please remember your password!
"0xb1b1dale9857e922fb38e3fb44a57ac4f44778fa"

> admin.nodeInfo.enode
[INFO] [04-18|01:29:28.524] New local node record seq=2 id=71a770f9708aaafb ip=10.1.1.2 udp=0 tcp=3001
6
"enode://1f50e1d3089009f82a61eb2f0ca71175eb9e933b2c2c455cc084a8ab09d9c6e31ddfbefffa01ab338af4019af4e9c7332e09306ad213a176fc0913d74fc66326@10.1.1.2:30016?discport=0"
>
```

10. 節點

10.4 在 node1 連結節點 node0 admin.addPeer(admin.nodeInfo.enode)

```
命令提示字元 - geth --datadir node1 --nodiscover --networkid 15 --port 30017 --rpc --rpcport 8550 --rpcapi "web3,net,eth,admin,personal" --r... - □ ×

INFO [04-18|01:59:49.785] Allocated fast sync bloom size=512.00MiB
INFO [04-18|01:59:49.786] Starting peer-to-peer node instance=Geth/v1.9.25-stable-e7872729/windows-amd64/gol.15.6
INFO [04-18|01:59:49.788] Initialized fast sync bloom items=354 errorrate=0.000 elapsed=1.993ms
INFO [04-18|01:59:49.796] New local node record seq=4 id=662b9b0b2d9d129e ip=127.0.0.1 udp=0 tcp=30017
INFO [04-18|01:59:49.796] HTTP server started endpoint=127.0.0.1:8550 cors=* vhosts=localhost
INFO [04-18|01:59:49.798] Started P2P networking self="enode://7ac8fdb6bfe76ad2cab9241f2fd90a0fc5736ab33a8f8ab80e4050de65db92fdd25dc30b0674ac89fdde5595f6162bf4d264ddee82d81f5191ca7654ca2fc407@127.0.0.1:30017?discport=0"
WARN [04-18|01:59:49.830] Served eth_coinbase reqid=3 t=0s err="etherbase must be explicitly specified"
Welcome to the Geth JavaScript console!

instance: Geth/v1.9.25-stable-e7872729/windows-amd64/gol.15.6
at block: 0 (Thu Apr 18 2024 01:04:14 GMT+0800 (CST))
  datadir: C:\Users\User\Desktop\chain01\node1
  modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d
> INFO [04-18|01:59:52.573] Mapped network port proto=tcp extport=30017 intport=30017 interface="UPNP, IGDv1, IP1"
> admin.addPeer("enode://1f50e1d3089009f82a61eb2f0ca71175eb9e933b2c2c455cc084a8ab09d9c6e31ddfbeffa01ab338af4019af4e9c7332e09306ad213a176fc0913d74fc66326@127.0.0.1:30016?discport=0")
INFO [04-18|02:00:> 39.483] Looking for peers peercount=0 tried=0 static=1
```

10. 節點

10.5.1 輸入 net.peerCount 檢查 連結節點

```
> net.peerCount
1
> INFO [04-18|02:01:14.484] Looking for peers
>                                         peercount=1 tried=1 static=1
```

10.5.2 輸入 admin.peers 檢查 連結節點

```
> admin.peers
[{
  caps: ["eth/63", "eth/64", "eth/65"],
  enode: "enode://7ac8fdb6bfe76ad2cab9241f2fd90a0fc5736ab33a8f8ab80e4050de65db92fdd25dc30b0674ac89fdde5595f6162bf4d264
ddee82d81f5191ca7654ca2fc407@127.0.0.1:52828",
  id: "662b9b0b2d9d129e53026aad3c20d83fc714940b39181731f896127fe1c40f72",
  name: "Geth/v1.9.25-stable-e7872729/windows-amd64/go1.15.6",
  network: {
    inbound: true,
    localAddress: "127.0.0.1:30016",
    remoteAddress: "127.0.0.1:52828",
    static: false,
    trusted: false
  },
  protocols: {
    eth: {
      difficulty: 524288,
      head: "0x3a71ce1fe6deeddbbec6f28cfe4737cda8f63e2a5c4d828e6681480b7012e196",
      version: 65
    }
  }
}]
```

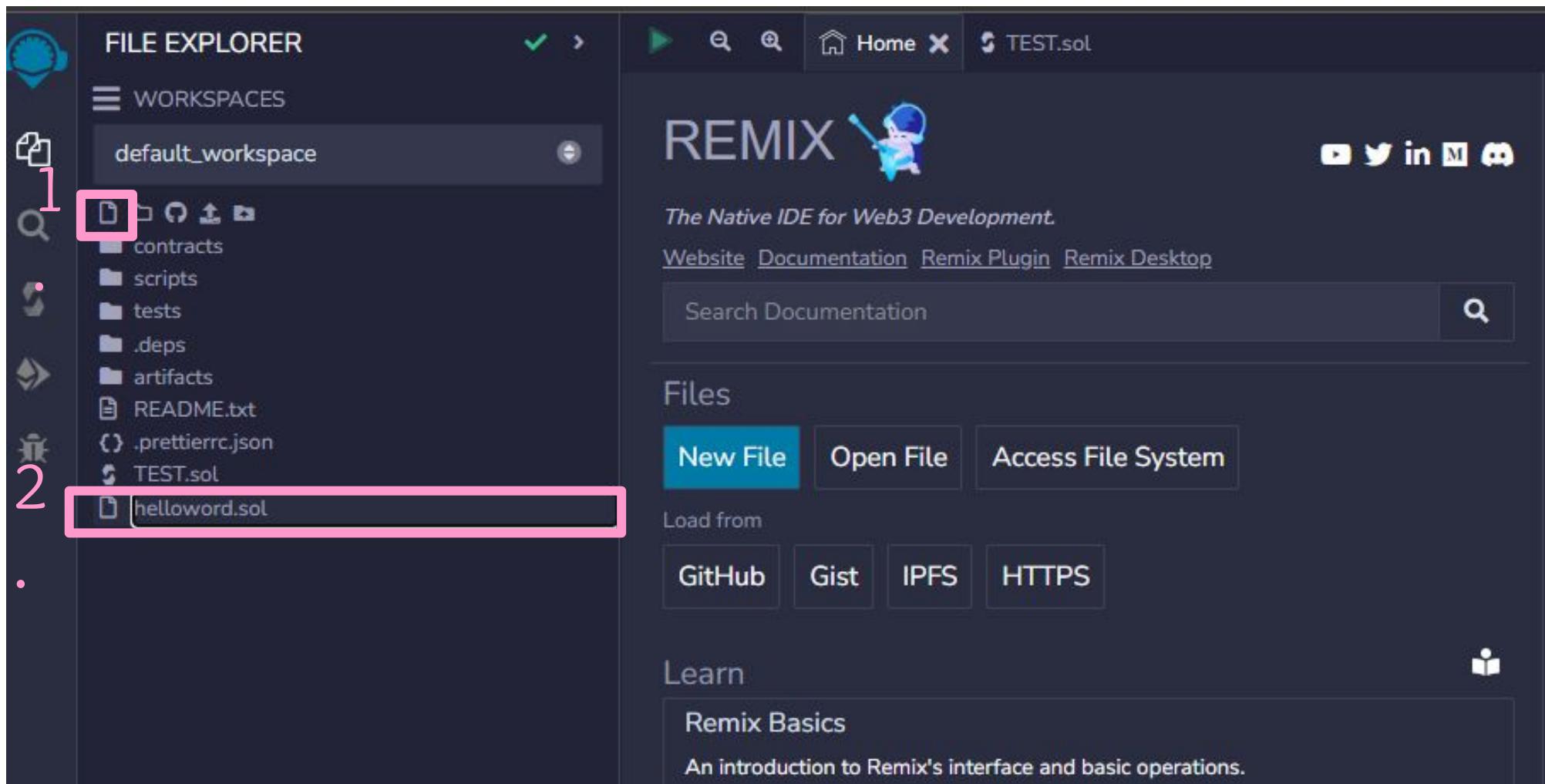
10. 常用指令

personal.newAccount()	創建帳戶
admin.nodeInfo.enode	查詢的節點資訊
admin.addPeer(admin.nodeInfo.enode)	連結節點
net.peerCount	查看已連接的節點個數
admin.peers	查看已連接的節點資訊
eth.coinbase	確認礦工
miner.start(1)	開始挖礦
miner.stop()	停止挖礦
eth.blockNumber	區塊鏈節點中區塊的總數
eth.getBalance(eth.accounts[0])	取得帳戶餘額
personal.unlockAccount(eth.coinbase)	解鎖帳戶

11. 撰寫智能合約

(1). 開啟remix : <https://remix.ethereum.org/>

(2). 點擊創建新文件，並命名為 helloworld.sol

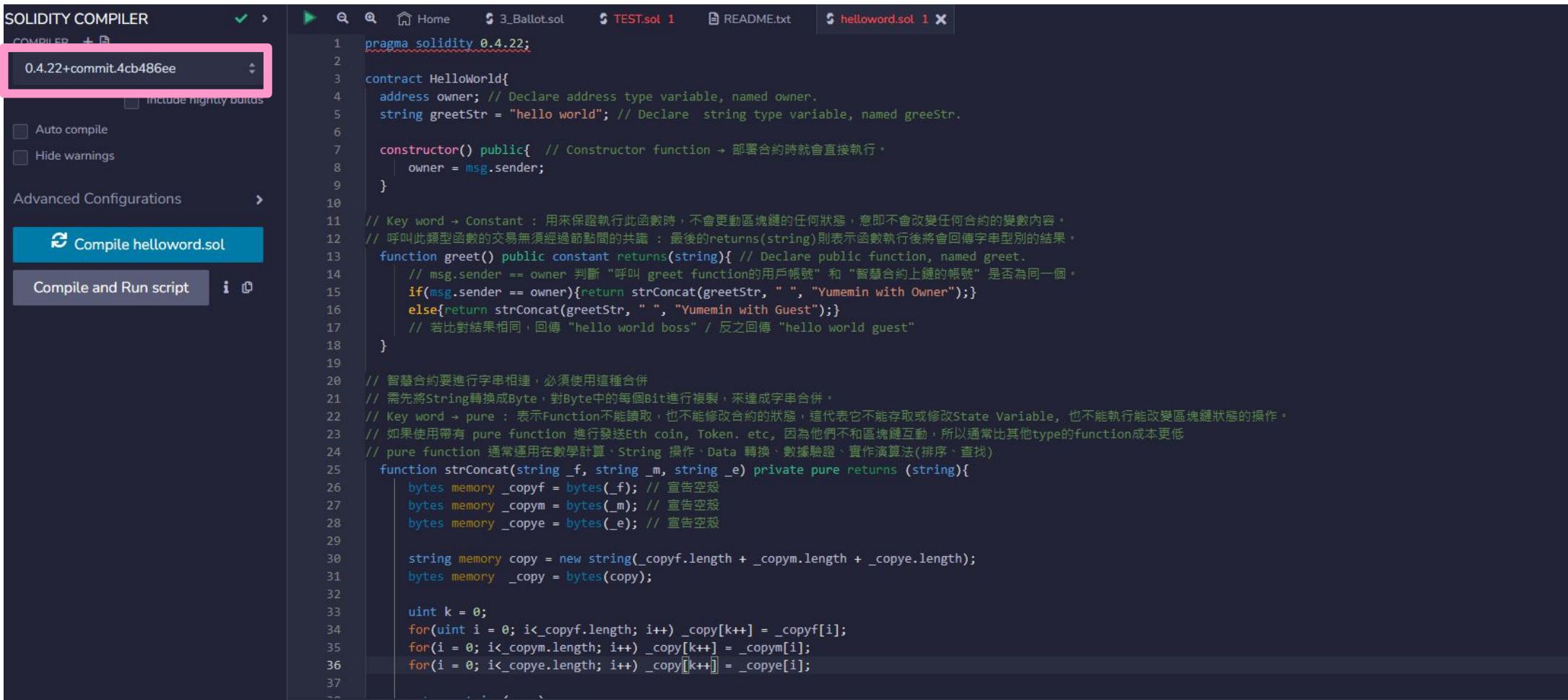


11. 撰寫智能合約

(3) 將智能合約複製到 helloworld.sol 中

<https://drive.google.com/file/d/1a3xf3GNu3xg4DidYFsGZ631pVQQg6AHq/view?usp=sharing>

(4) 將 Compiler 版本調成 0.4.22



The screenshot shows the Solidity Compiler interface with the following details:

- Compiler Version:** 0.4.22+commit.4cb486ee (highlighted with a red box)
- File:** helloworld.sol
- Code Content:** The code defines a contract named HelloWorld with a constructor that sets the owner to the msg.sender. It includes a greet() function that returns a string based on whether the caller is the owner or a guest. It also includes a strConcat(string _f, string _m, string _e) private pure returns (string) function that concatenates three strings.

```
pragma solidity 0.4.22;

contract HelloWorld{
    address owner; // Declare address type variable, named owner.
    string greetStr = "hello world"; // Declare string type variable, named greetStr.

    constructor() public{ // Constructor function → 部署合約時就會直接執行。
        owner = msg.sender;
    }

    // Key word → Constant : 用來保證執行此函數時，不會更動區塊鏈的任何狀態，意即不會改變任何合約的變數內容。
    // 呼叫此類型函數的交易無須經過節點間的共識：最後的returns(string)則表示函數執行後將會回傳字符串別的結果。
    function greet() public constant returns(string){ // Declare public function, named greet.
        // msg.sender == owner 判斷 "呼叫 greet function 的用戶帳號" 和 "智慧合約上鏈的帳號" 是否為同一個。
        if(msg.sender == owner){return strConcat(greetStr, " ", "Yumemin with Owner");}
        else{return strConcat(greetStr, " ", "Yumemin with Guest");}
        // 若比對結果相同，回傳 "hello world boss" / 反之回傳 "hello world guest"
    }

    // 智慧合約要進行字串相連，必須使用這種合併
    // 需先將String轉換成Byte，對Byte中的每個Bit進行複製，來達成字串合併。
    // Key word → pure : 表示Function不能讀取，也不能修改合約的狀態，這代表它不能存取或修改State Variable，也不能執行能改變區塊鏈狀態的操作。
    // 如果使用帶有 pure function 進行發送Eth coin, Token, etc，因為他們不和區塊鏈互動，所以通常比其他type的function成本更低
    // pure function 通常適用在數學計算、String 操作、Data 轉換、數據驗證、實作演算法(排序、查找)
    function strConcat(string _f, string _m, string _e) private pure returns (string){
        bytes memory _copyf = bytes(_f); // 廣告空殼
        bytes memory _copym = bytes(_m); // 廣告空殼
        bytes memory _copye = bytes(_e); // 廣告空殼

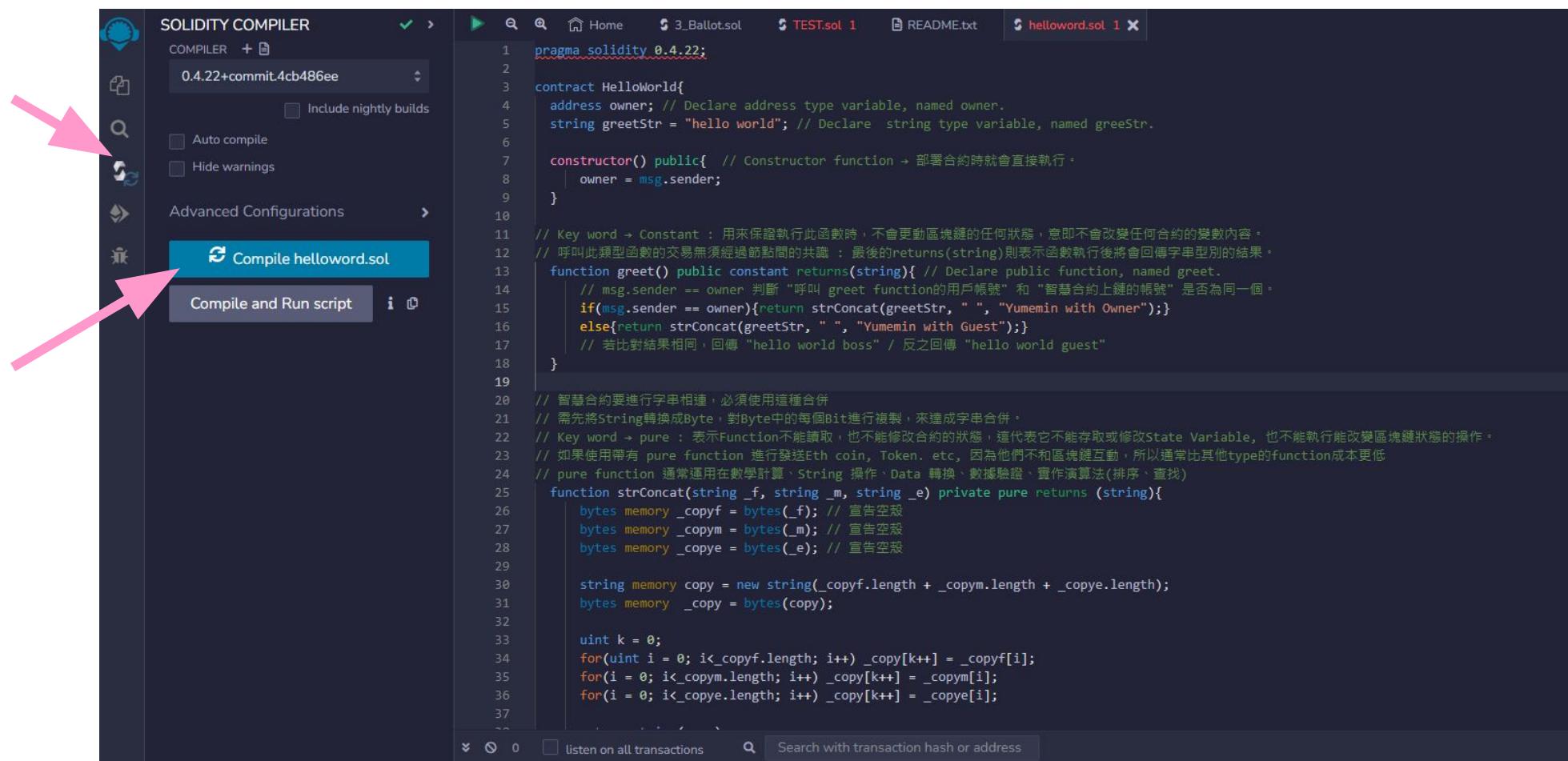
        string memory copy = new string(_copyf.length + _copym.length + _copye.length);
        bytes memory _copy = bytes(copy);

        uint k = 0;
        for(uint i = 0; i<_copyf.length; i++) _copy[k++] = _copyf[i];
        for(i = 0; i<_copym.length; i++) _copy[k++] = _copym[i];
        for(i = 0; i<_copye.length; i++) _copy[k++] = _copye[i];
    }
}
```

11. 撰寫智能合約

(5) 切換到 Solidity Compiler 頁面

(6) 點擊 Compile helloworld.sol



The screenshot shows the Solidity Compiler interface. On the left, there's a sidebar with icons for Home, TEST.sol, README.txt, and helloworld.sol (which is currently selected). Below the sidebar, there are buttons for 'Include nightly builds', 'Auto compile', 'Hide warnings', and 'Advanced Configurations'. A prominent blue button labeled 'Compile helloworld.sol' is highlighted with a pink arrow. Below it is another button labeled 'Compile and Run script'. The main area displays the Solidity code for 'helloworld.sol'. The code defines a contract named 'HelloWorld' with a constructor that sets the owner to the msg.sender. It includes a public function 'greet()' that returns a string. The code also contains a helper function 'strConcat' for concatenating strings. The code is annotated with comments explaining various Solidity features like constants, pure functions, and memory management.

```
pragma solidity 0.4.22;

contract HelloWorld{
    address owner; // Declare address type variable, named owner.
    string greetStr = "hello world"; // Declare string type variable, named greetStr.

    constructor() public{ // Constructor function → 部署合約時就會直接執行。
        owner = msg.sender;
    }

    // Key word → Constant : 用來保證執行此函數時，不會更動區塊鏈的任何狀態，意即不會改變任何合約的變數內容。
    // 呼叫此類型函數的交易無須經過節點間的共識：最後的returns(string)則表示函數執行後將會回傳字串型別的結果。
    function greet() public constant returns(string){ // Declare public function, named greet.
        // msg.sender == owner 判斷 "呼叫 greet function 的用戶帳號" 和 "智慧合約上鏈的帳號" 是否為同一個。
        if(msg.sender == owner){return strConcat(greetStr, " ", "Yumemin with Owner");}
        else{return strConcat(greetStr, " ", "Yumemin with Guest");}
        // 若比對結果相同，回傳 "hello world boss" / 反之回傳 "hello world guest"
    }

    // 智慧合約要進行字串相連，必須使用這種合併
    // 需先將String轉換成Byte，對Byte中的每個Bit進行複製，來達成字串合併。
    // Key word → pure : 表示Function不能讀取，也不能修改合約的狀態，這代表它不能存取或修改State Variable，也不能執行能改變區塊鏈狀態的操作。
    // 如果使用帶有 pure function 進行發送Eth coin, Token, etc.，因為他們不和區塊鏈互動，所以通常比其他type的function成本更低
    // pure function 通常適用在數學計算、String 操作、Data 轉換、數據驗證、實作演算法(排序、查找)
    function strConcat(string _f, string _m, string _e) private pure returns (string){
        bytes memory _copyf = bytes(_f); // 壓告空殼
        bytes memory _copym = bytes(_m); // 壓告空殼
        bytes memory _copye = bytes(_e); // 壓告空殼

        string memory copy = new string(_copyf.length + _copym.length + _copye.length);
        bytes memory _copy = bytes(copy);

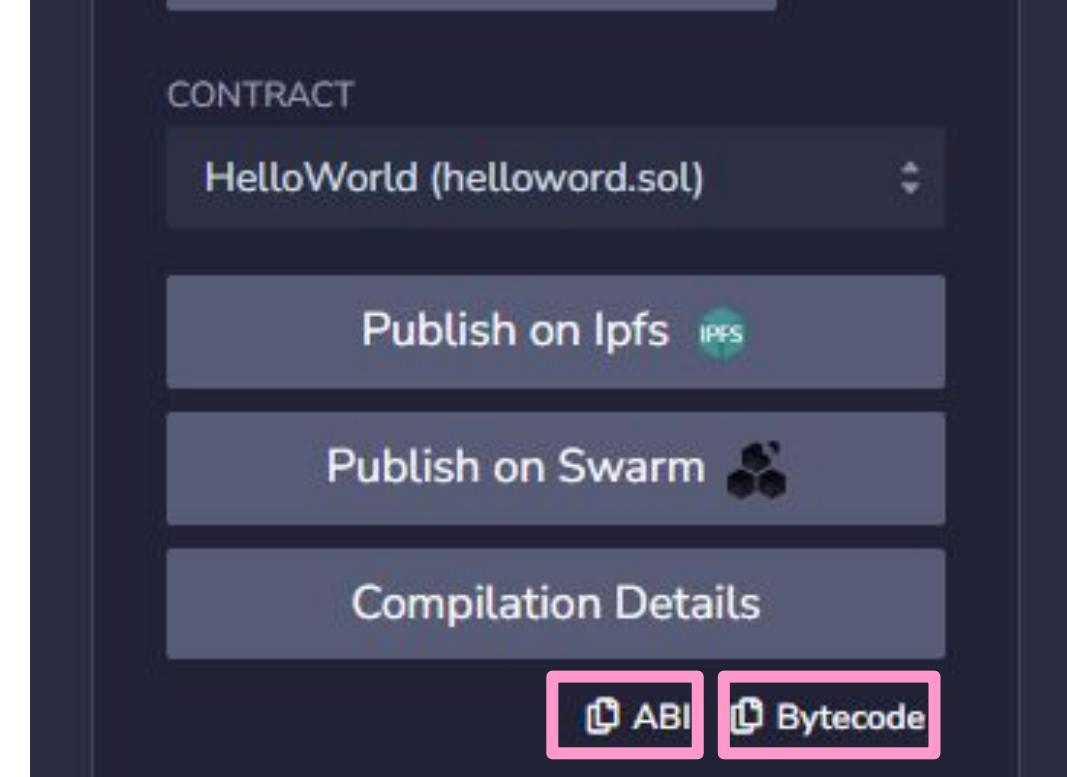
        uint k = 0;
        for(uint i = 0; i < _copyf.length; i++) _copy[k++] = _copyf[i];
        for(i = 0; i < _copym.length; i++) _copy[k++] = _copym[i];
        for(i = 0; i < _copye.length; i++) _copy[k++] = _copye[i];
    }
}
```

11. 撰寫智能合約

(7) 將Bytecode複製下來，打開文字編輯器，貼上後再前方加上0x，並保存下來。



```
1 60806040526040805190810160405280600b81526020017f68656c6c6f2077
2
3
4
5 0x60806040526040805190810160405280600b81526020017f68656c6c6f2077
6
7
8
9
```



(8) 將 abi 複製下來，前往 <http://www.bejson.com/jsonviewernew/>，去除掉空格，並保存下來



```
{"constant":true,"inputs":[],"name":"greet","outputs":[]}
[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function"}, {""inputs":[]],"payable":false,"stateMutability":"nonpayable","type":"constructor"}]
```

12. 部署智能合約

(1) 到指定資料夾後，開啟節點

```
geth --datadir node1 --nodiscover --networkid 15 --port 30016 --rpc --rpcport 8549  
--rpcapi "web3,net,eth,admin,personal" --rpccorsdomain "*" --ipcdisable  
--allow-insecure-unlock console
```

(2) personal.newAccount() // 新增帳戶

(3) eth.accounts //查詢所有帳戶

(4) eth.coinbase //查詢礦工

(5) eth.getBalance(eth.accounts[0]) //查詢餘額

12. 部署智能合約

(6) personal.unlockAccount(eth.accounts[0], "password") //帳戶解鎖

```
To exit, press ctrl-d
> personal.unlockAccount(eth.accounts[0], 'admin')
true
> eth.getBalance(eth.accounts[0])
20000000000000000000000000000000
```

(7) miner.start(1)

(8) miner.stop()

(9) eth.getBalance(eth.accounts[0]) //再次查詢餘額，確定有錢錢

```
> personal.unlockAccount(eth.accounts[0], 'admin')
true
> eth.getBalance(eth.accounts[0])
20000000000000000000000000000000
> eth.accounts
["0x0a1d225a80eca3c6c11a8228a0b9355d2c72fc45"]
> abi=[{"inputs":[],"payable":false,"stateMutability":"pure","name":"greet","outputs":[{"name":"","type":"string"}]]
[{"inputs":[]}]
```

12. 部署智能合約

```
(10) abi =[{"inputs": [], "payable": false, "stateMutability": "nonpayable", "type": "function"}] // 宣告abi
```

```
> abi=[{"inputs":[],"payable":false,"stateMutability":"nonpayable","type":"constructor"}, {"constant":true,"inputs":[],"name":"greet","outputs":[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function"}]
[{
  inputs: [],
  payable: false,
  stateMutability: "nonpayable",
  type: "constructor"
}, {
  constant: true,
  inputs: [],
  name: "greet",
  outputs: [
    {
      name: "",
      type: "string"
    }
  ],
  payable: false,
  stateMutability: "view",
  type: "function"
}]
```

(11) bytecode=“0x6080604052604080 …… c3320cbf43085d6d61ea0ad8c190029” //宣告bytecode

```
(12) eth.estimateGas({data: bytecode}) //預估手續費
```

12. 部署智能合約

(13) MyContract = eth.contract(abi)

```
5627a7a72305820d3b2a8e095fa2232c884b8b5d2e60ea71Fd68c3320cbf430
> MyContract = eth.contract(abi)
{
  abi: [
    {
      inputs: [],
      payable: false,
      stateMutability: "nonpayable",
      type: "constructor"
    },
    {
      constant: true,
      inputs: [],
      name: "greet",
      outputs: [{...}],
      payable: false,
      stateMutability: "view",
      type: "function"
    }
  ],
  eth: {
    accounts: ["0x0a1d225a80eca3c6c11a8228a0b9355d2c72fc45"],
    blockNumber: 10,
    coinbase: "0x0a1d225a80eca3c6c11a8228a0b9355d2c72fc45",
    ...
  }
}
```

(14) contract1 = MyContract.new({from:"0x0a1d225a80eca3c6c11a8228a0b9355d2c72fc45",data:bytecode,gas:1200000})

創建合約實例, from後的地址為帳戶的address, gas後填寫願意支付的gas

```
> contract1 = MyContract.new({from:"0x0a1d225a80eca3c6c11a8228a0b9355d2c72fc45",data:bytecode,gas:1200000})
INFO [09-21|14:30:10.759] Submitted contract creation fullhash=0xc4d5dae23963f3775b2d77462c1b7e70a4c5ef8facd280be62cc120f7f17b41c contract=0x6800bbecc3B7f8F381B4A8E4cfb84A9C02C03c84
{
  abi: [
    {
      inputs: [],
      payable: false,
      stateMutability: "nonpayable",
      type: "constructor"
    },
    {
      constant: true,
      inputs: [],
      name: "greet",
      outputs: [{...}],
      payable: false,
      stateMutability: "view",
      type: "function"
    }
  ],
  address: undefined,
  transactionHash: "0xc4d5dae23963f3775b2d77462c1b7e70a4c5ef8facd280be62cc120f7f17b41c"
}
```

12. 部署智能合約

(15) contract1 //查看合約資訊

```
address: undefined,
transactionHash: "0xc4d5dae23963f3775b2d77462c1b7e70a4c5ef8facd280be62cc120f7f17b41c"
}
> contract1
{
  abi: [{  
    inputs: [],  
    payable: false,  
    stateMutability: "nonpayable",  
    type: "constructor"  
  }, {  
    constant: true,  
    inputs: [],  
    name: "greet",  
    outputs: [{}...],  
    payable: false,  
    stateMutability: "view",  
    type: "function"  
  }],  
  address: undefined,  
  transactionHash: "0xc4d5dae23963f3775b2d77462c1b7e70a4c5ef8facd280be62cc120f7f17b41c"
}
> txpool.status
{
```

(16) txpool.status // 查看是否有等待挖礦的交易

```
> txpool.status
{
  pending: 9,  
  queued: 0
}
> miner.start()
INFO [09-21|14:30:32.927] Updated mining threads
```

12. 部署智能合約

(17) miner.start(1) // 挖礦以將合約交易寫入鏈上。

(18) miner.stop() // 挖一下再停止挖礦。

(19) txpool.status // 查看是否有等待挖礦的交易，確認待挖礦交易降為0。

```
> txpool.status
{
  pending: 0,
  queued: 0
}
> contract1.address
```

(20) contract1.address // 查看已經部署到私有鏈上的智能合約address，並保存下來。

複製保
存



```
queued: 0
}
> contract1.address
"0x6800bbecc3b7f8f381b4a8e4cfb84a9c02c03c84"
> exit
INFO [00:21|14:21:55.694] HTTP server stopped
```

12. 部署智能合約

(21) contract = MyContract.at(**address**) //重新創建一個合約實例，並指名合約地址

```
> contract = myContract.at("0x6800bbecc3b7f8f381b4a8e4cfb84a9c02c03c84")
{
  abi: [
    {
      inputs: [],
      payable: false,
      stateMutability: "nonpayable",
      type: "constructor"
    },
    {
      constant: true,
      inputs: [],
      name: "greet",
      outputs: [{...}],
      payable: false,
      stateMutability: "view",
      type: "function"
    }
  ],
  address: "0x6800bbecc3b7f8f381b4a8e4cfb84a9c02c03c84",
  transactionHash: null,
  allEvents: function(),
  greet: function()
}
```

(22) contract.greet.call() //呼叫合約中的greet() 函式

(23) contract.greet.call({from: "0x0a1d225a80eca3c6c11a8228a0b9355d2c72fc45"}); //以特定用戶呼叫 greet() 函式

```
}
```

```
> contract.greet.call()
"hello world Yumemin with Guest"
> eth.accounts
["0x0a1d225a80eca3c6c11a8228a0b9355d2c72fc45"]
> contract.greet.call({from: "0x0a1d225a80eca3c6c11a8228a0b9355d2c72fc45"});
"hello world Yumemin with Owner"
```

<https://drive.google.com/file/d/1SwTBLzBYxOQtm6dGvOTc1Rx6ErNH1t9n/view?usp=sharing>