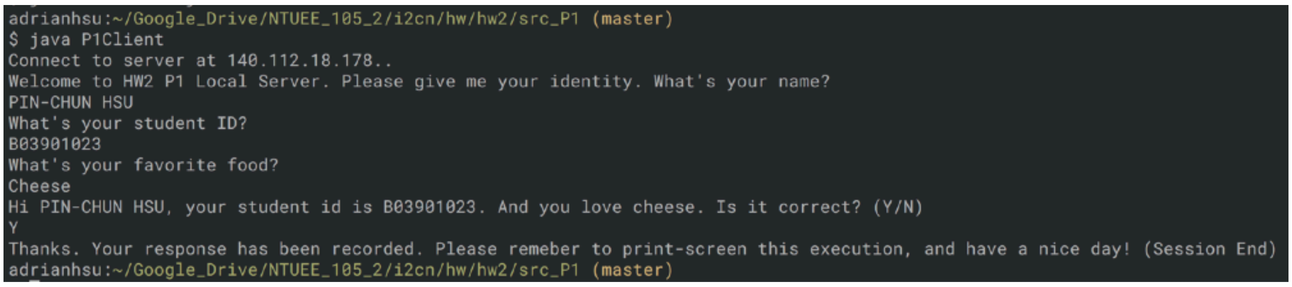
B03901023 電機三 許秉鈞

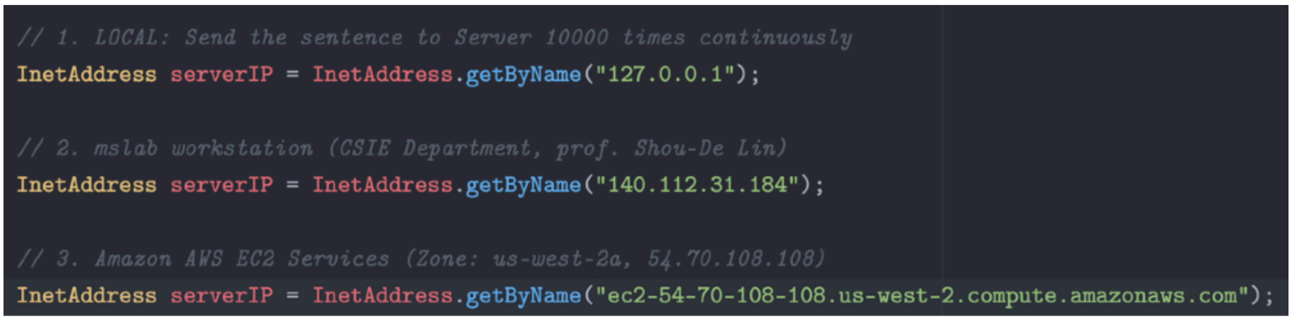
P1



```
adrianhsu:~/Google_Drive/NTUEE_105_2/i2cn/hw/hw2/src_P1 (master)
$ java P1Client
Connect to server at 140.112.18.178..
Welcome to HW2 P1 Local Server. Please give me your identity. What's your name?
PIN-CHUN HSU
What's your student ID?
B03901023
What's your favorite food?
Cheese
Hi PIN-CHUN HSU, your student id is B03901023. And you love cheese. Is it correct? (Y/N)
Y
Thanks. Your response has been recorded. Please remeber to print-screen this execution, and have a nice day! (Session End)
adrianhsu:~/Google_Drive/NTUEE_105_2/i2cn/hw/hw2/src_P1 (master)
```

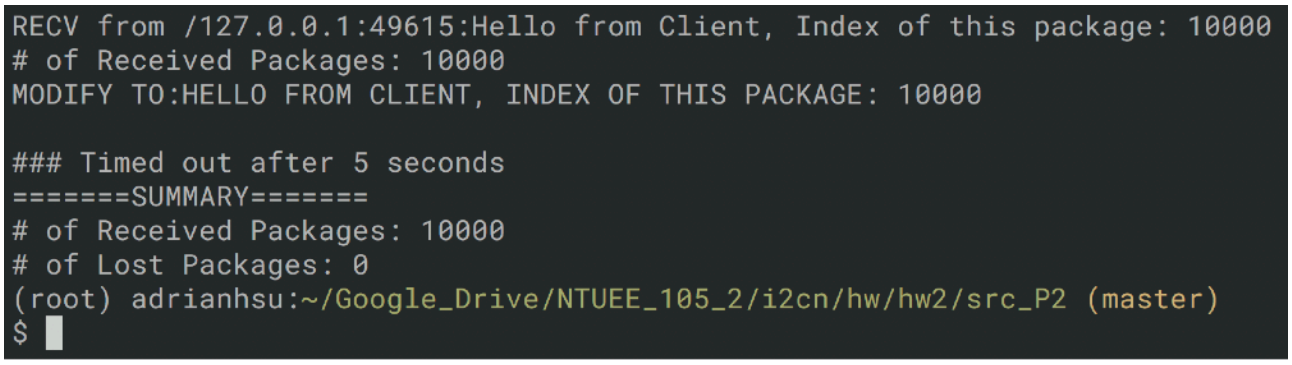
P2

我做了 3 種 cases (bonus 做了兩種) ，請見以下程式碼：



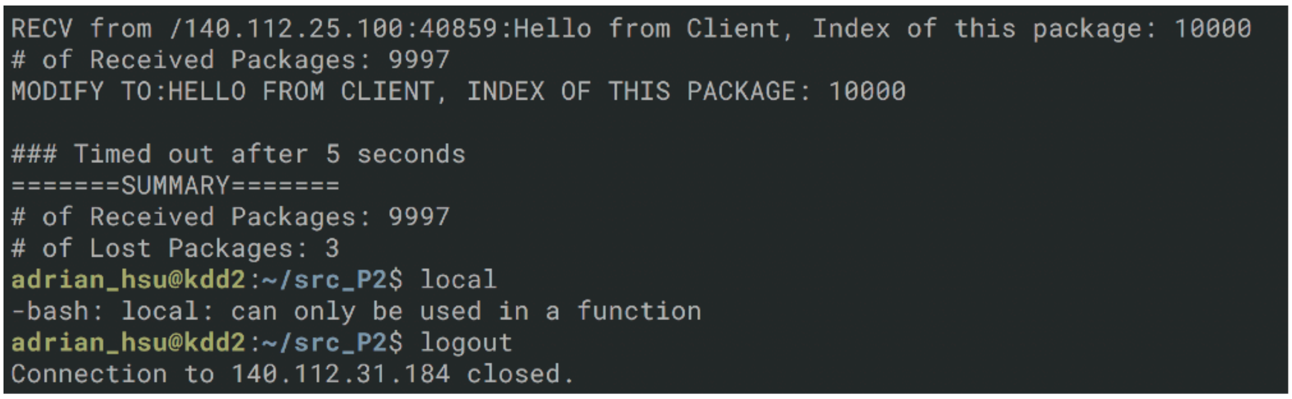
```java
// 1. LOCAL: Send the sentence to Server 10000 times continuously
InetAddress serverIP = InetAddress.getByName("127.0.0.1");

// 2. mslab workstation (CSIE Department, prof. Shou-De Lin)
InetAddress serverIP = InetAddress.getByName("140.112.31.184");

// 3. Amazon AWS EC2 Services (Zone: us-west-2a, 54.70.108.108)
InetAddress serverIP = InetAddress.getByName("ec2-54-70-108-108.us-west-2.compute.amazonaws.com");
```

LOCAL



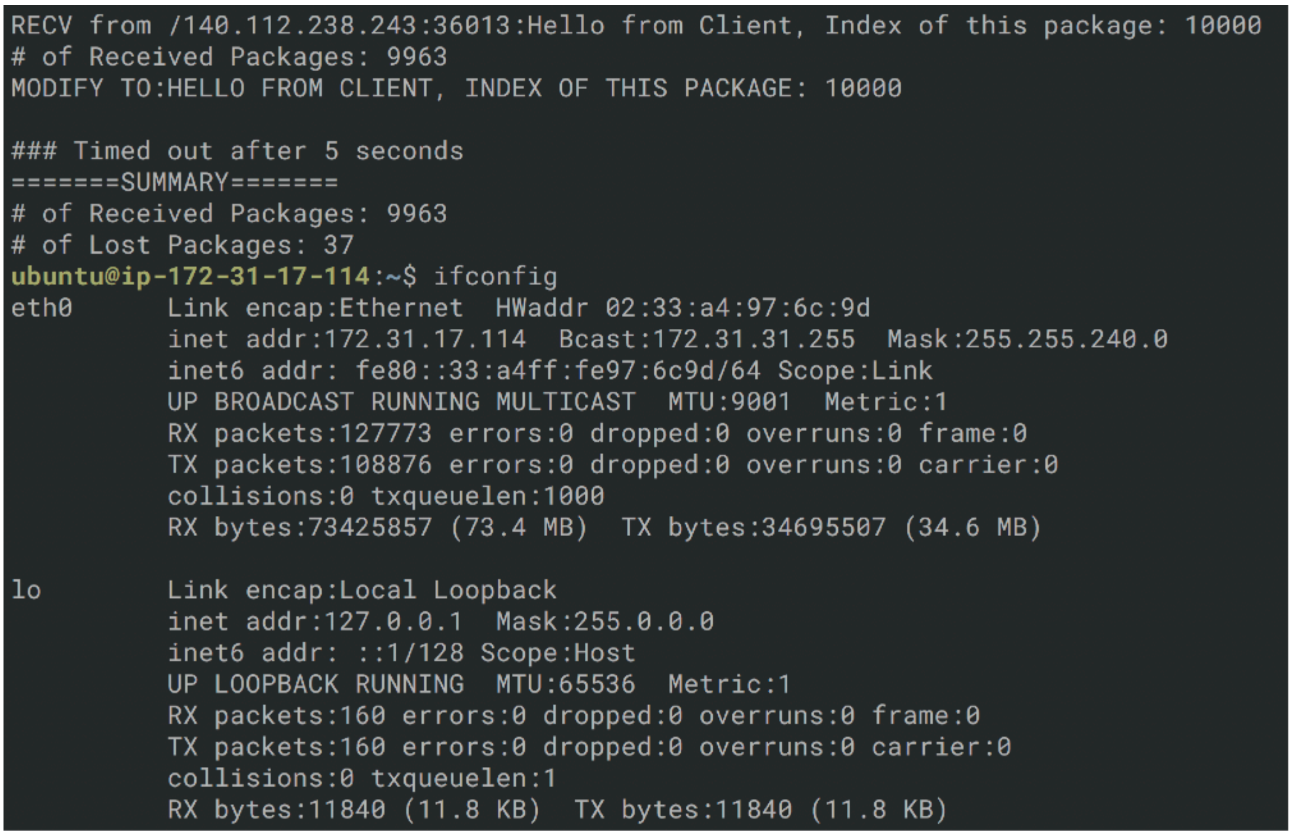
```
RECV from /127.0.0.1:49615:Hello from Client, Index of this package: 10000
# of Received Packages: 10000
MODIFY TO:HELLO FROM CLIENT, INDEX OF THIS PACKAGE: 10000

### Timed out after 5 seconds
=======SUMMARY=======
# of Received Packages: 10000
# of Lost Packages: 0
(root) adrianhsu:~/Google_Drive/NTUEE_105_2/i2cn/hw/hw2/src_P2 (master)
$ 
```

明達館 to 資工系館工作站 (林守德教授的lab主機 kdd2)



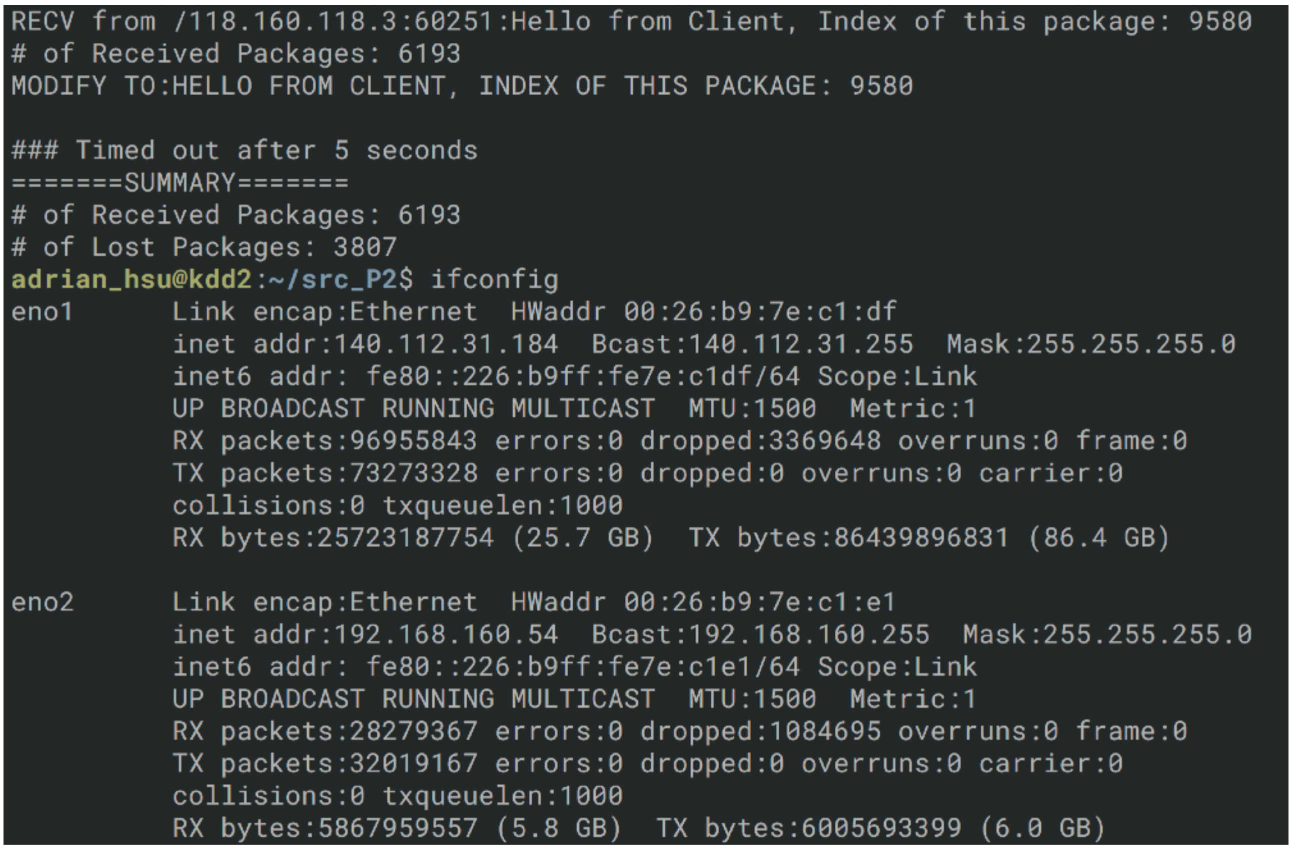
```
RECV from /140.112.25.100:40859:Hello from Client, Index of this package: 10000
# of Received Packages: 9997
MODIFY TO:HELLO FROM CLIENT, INDEX OF THIS PACKAGE: 10000

### Timed out after 5 seconds
=======SUMMARY=======
# of Received Packages: 9997
# of Lost Packages: 3
adrian_hsu@kdd2:~/src_P2$ local
-bash: local: can only be used in a function
adrian_hsu@kdd2:~/src_P2$ logout
Connection to 140.112.31.184 closed.
```

明達館 to AWS（美國西岸、Amazon提供的server免費服務）



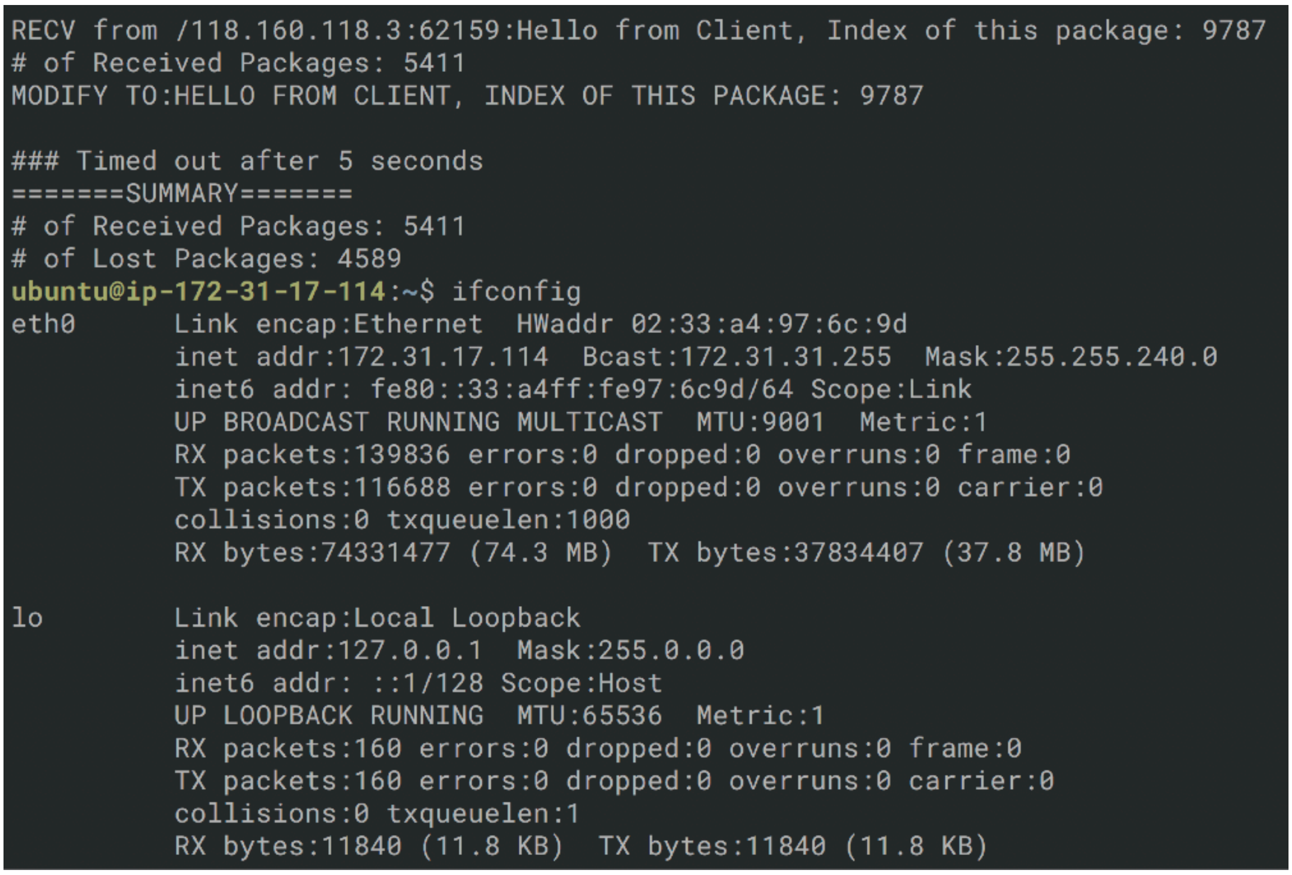
```
RECV from /140.112.238.243:36013:Hello from Client, Index of this package: 10000
# of Received Packages: 9963
MODIFY TO:HELLO FROM CLIENT, INDEX OF THIS PACKAGE: 10000

### Timed out after 5 seconds
=======SUMMARY=======
# of Received Packages: 9963
# of Lost Packages: 37
ubuntu@ip-172-31-17-114:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 02:33:a4:97:6c:9d
          inet addr:172.31.17.114  Bcast:172.31.31.255  Mask:255.255.240.0
          inet6 addr: fe80::33:a4ff:fe97:6c9d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9001  Metric:1
          RX packets:127773 errors:0 dropped:0 overruns:0 frame:0
          TX packets:108876 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:73425857 (73.4 MB)  TX bytes:34695507 (34.6 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)
```

內湖 to 資工系館（林守德教授實驗室）#這筆掉了3807個packages，其實有些狀況只掉1000個左右、網路順的話甚至只掉10個以內，蠻不一定的。



```
RECV from /118.160.118.3:60251:Hello from Client, Index of this package: 9580
# of Received Packages: 6193
MODIFY TO:HELLO FROM CLIENT, INDEX OF THIS PACKAGE: 9580

### Timed out after 5 seconds
=======SUMMARY=======
# of Received Packages: 6193
# of Lost Packages: 3807
adrian_hsu@kdd2:~/src_P2$ ifconfig
eno1      Link encap:Ethernet  HWaddr 00:26:b9:7e:c1:df
          inet addr:140.112.31.184  Bcast:140.112.31.255  Mask:255.255.255.0
          inet6 addr: fe80::226:b9ff:fe7e:c1df/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:96955843 errors:0 dropped:3369648 overruns:0 frame:0
          TX packets:73273328 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:25723187754 (25.7 GB)  TX bytes:86439896831 (86.4 GB)

eno2      Link encap:Ethernet  HWaddr 00:26:b9:7e:c1:e1
          inet addr:192.168.160.54  Bcast:192.168.160.255  Mask:255.255.255.0
          inet6 addr: fe80::226:b9ff:fe7e:c1e1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:28279367 errors:0 dropped:1084695 overruns:0 frame:0
          TX packets:32019167 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5867959557 (5.8 GB)  TX bytes:6005693399 (6.0 GB)
```

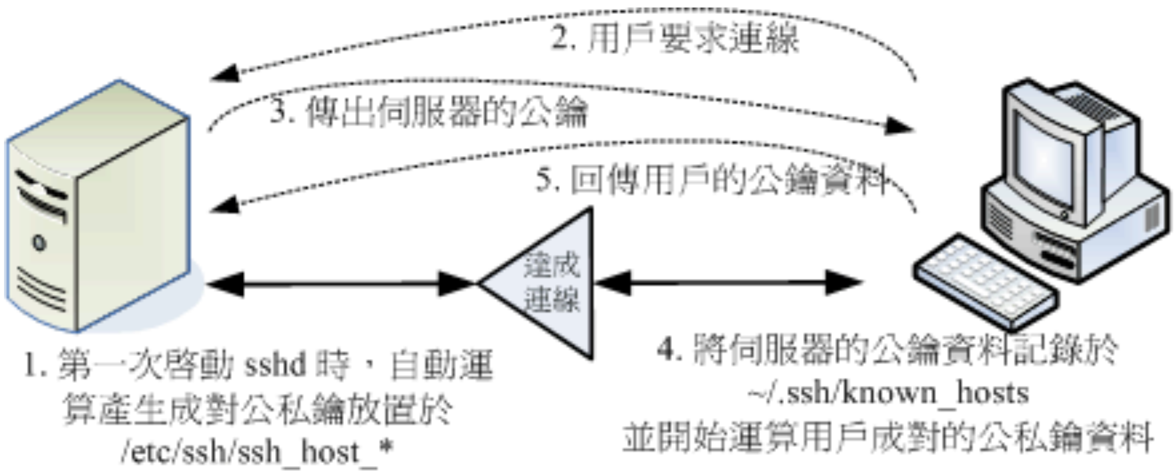
內湖 to AWS（美國西岸）#這筆掉了4589個packages，距離太遠了！



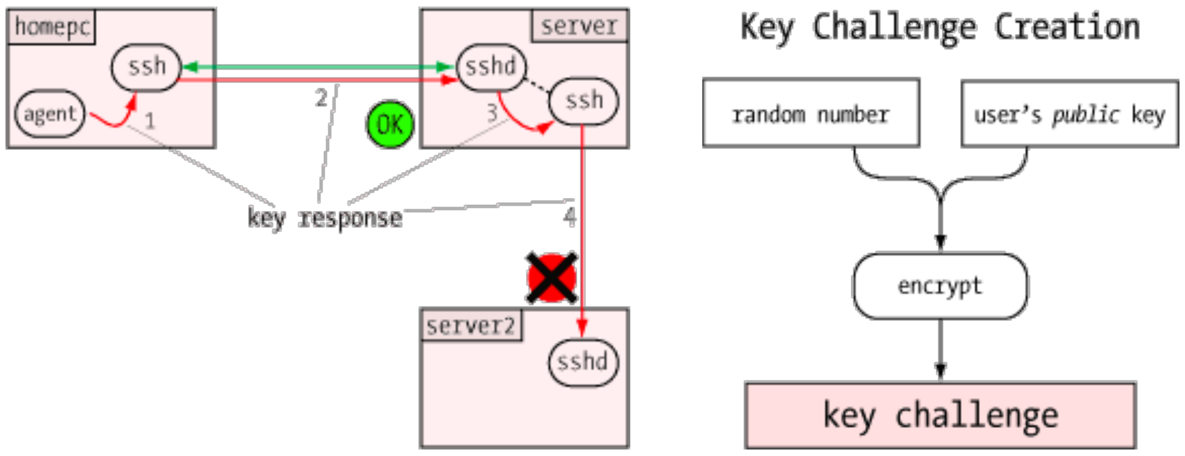
```
RECV from /118.160.118.3:62159:Hello from Client, Index of this package: 9787
# of Received Packages: 5411
MODIFY TO:HELLO FROM CLIENT, INDEX OF THIS PACKAGE: 9787

### Timed out after 5 seconds
=======SUMMARY=======
# of Received Packages: 5411
# of Lost Packages: 4589
ubuntu@ip-172-31-17-114:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 02:33:a4:97:6c:9d
          inet addr:172.31.17.114  Bcast:172.31.31.255  Mask:255.255.240.0
          inet6 addr: fe80::33:a4ff:fe97:6c9d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9001  Metric:1
          RX packets:139836 errors:0 dropped:0 overruns:0 frame:0
          TX packets:116688 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:74331477 (74.3 MB)  TX bytes:37834407 (37.8 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)
```

3. Secure Shell (SSH) [Application Layer Protocol]

SSH是一種跨越Application Layer跟Transport Layer的安全協定，在系上的工作站、網路上的amazon aws服務都很常用到，尤其是遠端的伺服器連線。SSH的基本架構如下：



圖片來源為鳥哥的Linux 私房菜。因為系上最常用到SSH，所以我選擇這個協定來研讀。我們可以從途上看到，SSH的基本架構最主要分為三個步驟：一、產生公私鑰；二、要求連線並傳送金鑰；三、將公鑰資料記錄，運算成對的公私鑰資料。比較特別的是，SSH為了安全考量（SSH非常重視安全驗證）有三個協定，分別為傳輸層協定（伺服器認證、資料機密保管）、用戶認證協定（伺服器用來身份鑑別用戶端）、連線協定（公私鑰通道），這些協定構成基本框架，讓許多高層協定都可以使用。前面提及SSH的安全驗證很重要，在wikipedia上面的資料來源寫道：在用戶端、SSH有兩種級別的驗證：
一、基於密碼的安全驗證，也是我們在工作站連線的方法，只要知道帳號密碼就可以登入、所有傳輸資料都被加密，但可能會被其他伺服器冒充讓我們走錯終點；二、基於金鑰的安全驗證，也是我在上一題作業的aws服務用的方法，他會給一份.pem檔、裡面是很長的字串，需要把公有金鑰放在目標存取的伺服器、用戶端在需要請求時、就把自己的公有金鑰傳過去進行比對，如果一致就會被用公有金鑰加密、又稱質詢（Challenge），確認通過才能傳到用戶端。

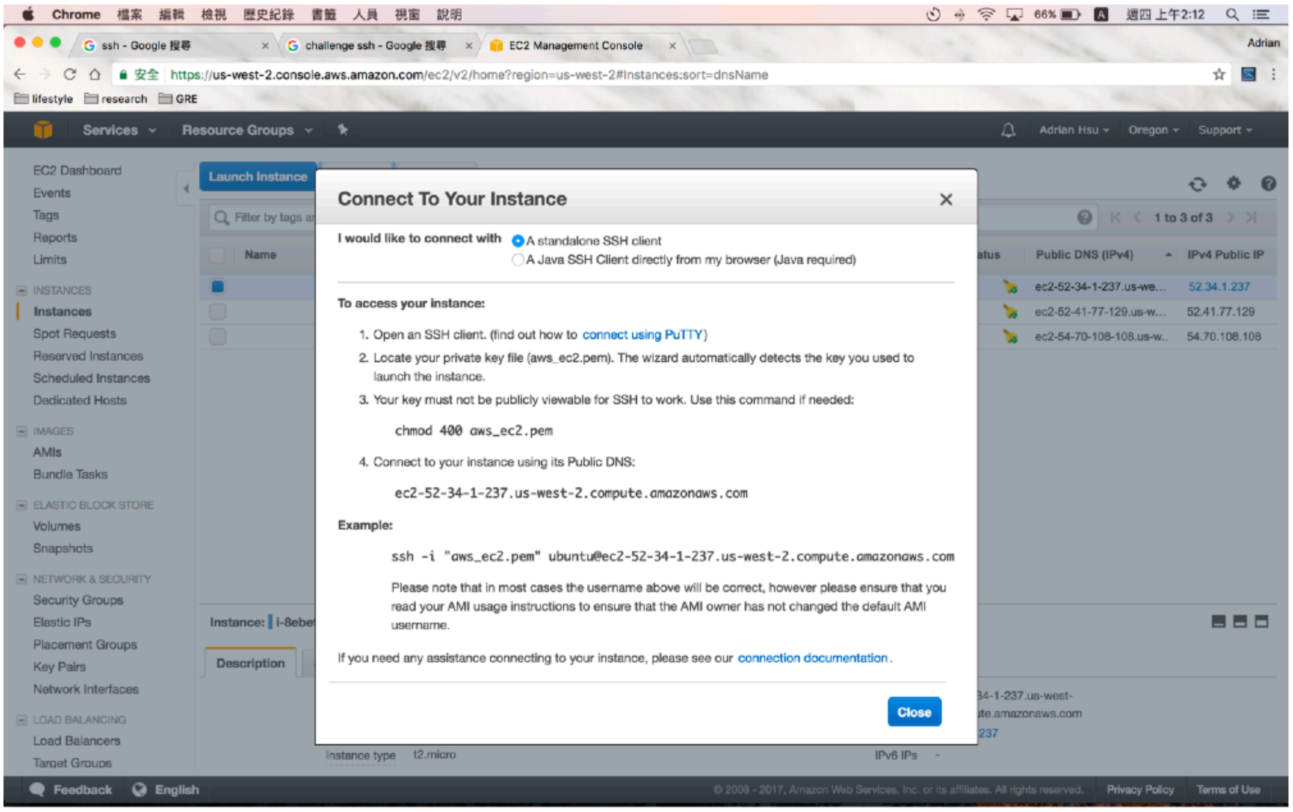


參考上面兩張圖，就是用基於金鑰的方式安全驗證，並在最後做key challenge。

研讀後請解釋該協定運作方式，並試舉一例做解釋。 - AWS Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) 是一種 Web 服務，提供可調整的運算容量 (也就是 Amazon 資料中心內的伺服器)，以用來建置和託管軟體系統，簡單來說，就是個置放在美國的工作站，而且很多大軟體公司都有用他們的運算空間。這個服務就是用SSH可以連上的、而且也是用剛剛提到的第二種（金鑰安全認證）。

要連線的話，就用 -i 加上 這.pem檔案的位置，其他SSH 的指令就照樣打就可以。下圖是我實際操作時、網站的連線教學，有standalone SSH client、Java SSH Client directly from my browser (Java required)這兩個可以選。



Connect To Your Instance

I would like to connect with [✓ A standalone SSH client] A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (aws_ec2.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 aws_ec2.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-52-34-1-237.us-west-2.compute.amazonaws.com
```

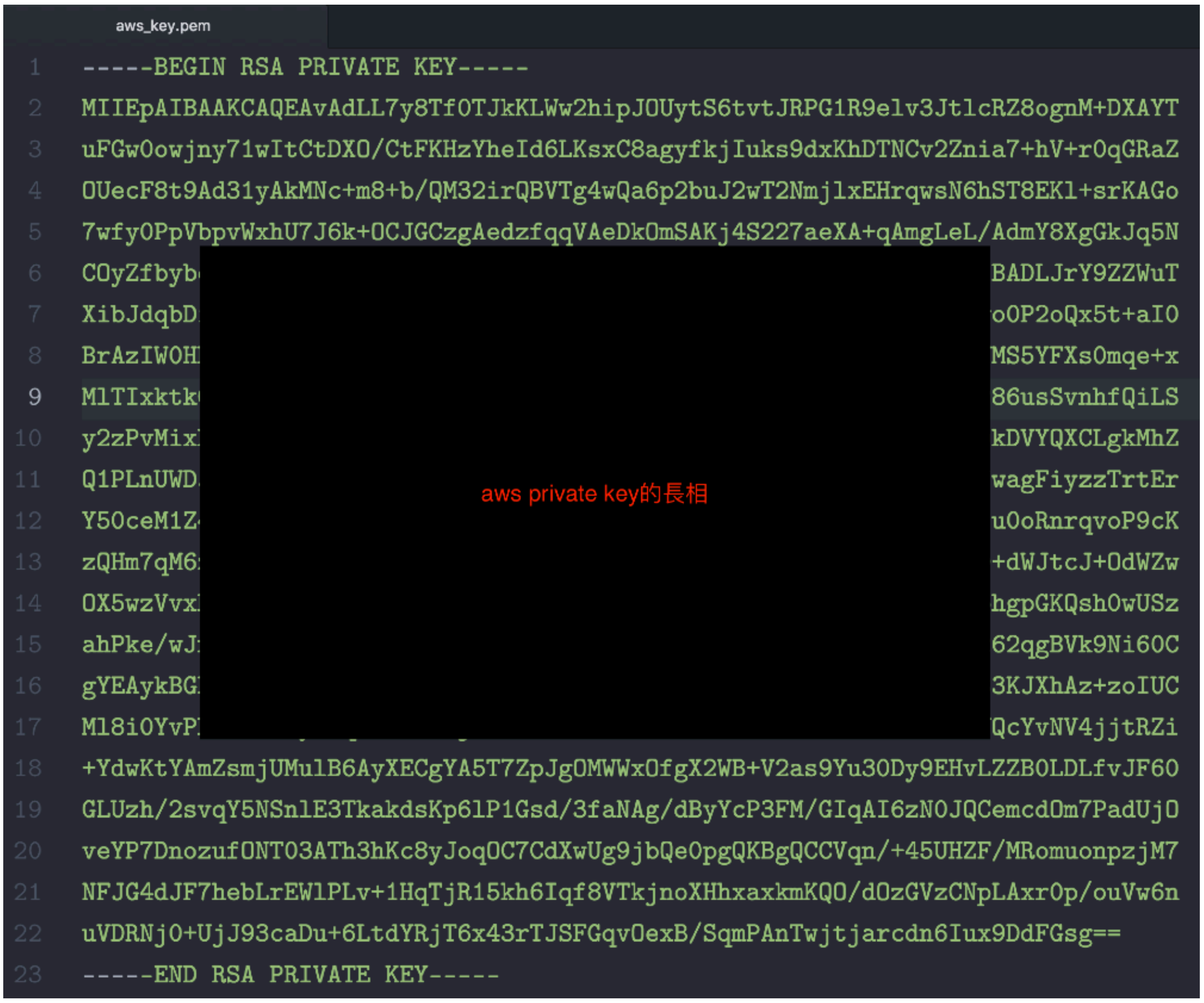
Example:

```
ssh -i "aws_ec2.pem" ubuntu@ec2-52-34-1-237.us-west-2.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation.

下圖是我建立好自己的EC2雲端後，從網站下載下來的私鑰檔案：aws_key.pem



aws_key.pem

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAvAdLL7y8Tf0TJkKLWw2hipJ0UytS6tvtJRPG1R9elv3JtlcRZ8ognM+DXAYT
uFGw0owjny71wItCtDX0/CtFKHzYheId6LKsxC8agyfkjIuks9dxKhDTNCv2Znia7+hV+r0qGRaZ
0UecF8t9Ad31yAkMNc+m8+b/QM32irQBVTg4wQa6p2buJ2wT2NmjlxEHrqwsN6hST8EKl+srKAGo
7wfy0PpVbpvWxhU7J6k+0CJGCzgAedzfqqVAeDk0mSAKj4S227aeXA+qAmgLeL/AdmY8XgGkJq5N
C0yZfbyb[redacted]BADLJrY9ZZWuT
XibJdqbD                                                        o0P2oQx5t+aI0
BrAzIW0H                                                        MS5YFXs0mqe+x
M1TIxktk                                                        86usSvnhfQiLS
y2zPvMix                                                        kDVYQXCLgkMhZ
Q1PLnUWD                                                        wagFiyzzTrtEr
Y50ceM1Z                                                        u0oRnrqvoP9cK
zQHm7qM6                                                        +dWJtcJ+0dWZw
0X5wzVvx                                                        hgpGKQsh0wUSz
ahPke/wJ                                                        62qgBVk9Ni60C
gYEAykBG                                                        3KJXhAz+zoIUC
M18i0YvP                                                        QcYvNV4jjtRZi
+YdwKtYAmZsmjUMulB6AyXECgYA5T7ZpJg0MWWx0fgX2WB+V2as9Yu30Dy9EHvLZZB0LDLfvJF60
GLUzh/2svqY5NSnlE3TkakdsKp6lP1Gsd/3faNAg/dByYcP3FM/GIqAI6zN0JQCemcd0m7PadUj0
veYP7Dnozuf0NT03ATh3hKc8yJoq0C7CdXwUg9jbQe0pgQKBgQCCVqn/+45UHZF/MRomuonpzjM7
NFJG4dJF7hebLrEWlPLv+1HqTjR15kh6Iqf8VTkjnoXHhxaxkmKQ0/d0zGVzCNpLAxr0p/ouVw6n
uVDRNj0+UjJ93caDu+6LtdYRjT6x43rTJSFGqv0exB/SqmPAnTwjtjarcdn6Iux9DdFGsg==
-----END RSA PRIVATE KEY-----
```

可以看到，他的連線方式跟前面講的SSH運作原理一樣：

- Open an SSH client. (find out how to connect using PuTTY)
- Locate your private key file (aws_ec2.pem). The wizard automatically detects the key you used to launch the instance.
- Your key must not be publicly viewable for SSH to work. Use this command if needed:
  - `chmod 400 aws_ec2.pem`
- Connect to your instance using its Public DNS:
  - `ec2-52-34-1-237.us-west-2.compute.amazonaws.com`

因為我沒有用PuTTY而是直接用bash連，所以client端就蠻容易上手的，我覺得這個例子讓我學到最多的是私鑰跟公鑰的實際使用方式，收穫良多。