LSAP HW3 Report

1 LDAP Directory Service

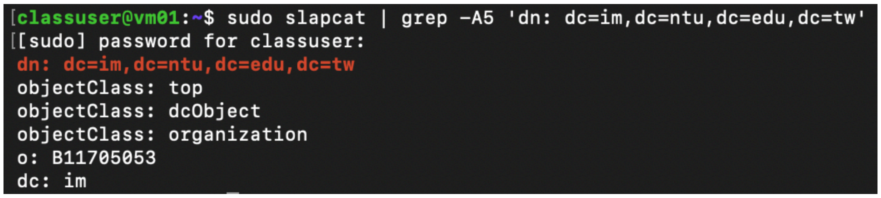
(1) 安裝與初始設定

安裝 LDAP 伺服器與工具，並把 Base DN/組織與管理者密碼寫入初始資料庫。

```
sudo apt update
sudo apt install -y slapd ldap-utils
sudo dpkg-reconfigure slapd
```

Domain: im.ntu.edu.tw → Base DN: dc=im,dc=ntu,dc=edu,dc=tw
Organization: B11705053
設定 admin 密碼（用於日後 ldapadd/modify）

```
sudo slapcat | grep -A5 'dn: dc=im,dc=ntu,dc=edu,dc=tw'
```



```
classuser@vm01:~$ sudo slapcat | grep -A5 'dn: dc=im,dc=ntu,dc=edu,dc=tw'
[sudo] password for classuser:
dn: dc=im,dc=ntu,dc=edu,dc=tw
objectClass: top
objectClass: dcObject
objectClass: organization
o: B11705053
dc: im
```

Figure 1: 初始化設定

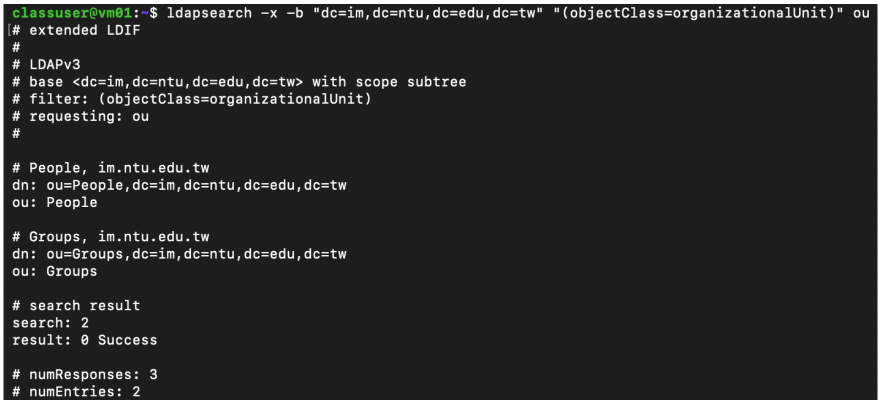
(2) 建立 OU（People / Groups）

建立常見的兩個組織單位，讓人與群組有清楚的容器。

```
cat > ou_base.ldif <<'LDIF'
dn: ou=People,dc=im,dc=ntu,dc=edu,dc=tw
objectClass: organizationalUnit
ou: People

dn: ou=Groups,dc=im,dc=ntu,dc=edu,dc=tw
objectClass: organizationalUnit
ou: Groups
LDIF

sudo ldapadd -x -D "cn=admin,dc=im,dc=ntu,dc=edu,dc=tw" -W -f ou_base.ldif
```

用 LDIF 描述兩個條目，再用 ldapadd 以 admin 身分匯入。

```
ldapsearch -x -b "dc=im,dc=ntu,dc=edu,dc=tw" "(objectClass=organizationalUnit)" ou
```



```
classuser@vm01:~$ ldapsearch -x -b "dc=im,dc=ntu,dc=edu,dc=tw" "(objectClass=organizationalUnit)" ou
# extended LDIF
#
# LDAPv3
# base <dc=im,dc=ntu,dc=edu,dc=tw> with scope subtree
# filter: (objectClass=organizationalUnit)
# requesting: ou
#

# People, im.ntu.edu.tw
dn: ou=People,dc=im,dc=ntu,dc=edu,dc=tw
ou: People

# Groups, im.ntu.edu.tw
dn: ou=Groups,dc=im,dc=ntu,dc=edu,dc=tw
ou: Groups

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

Figure 2: 建立 OU

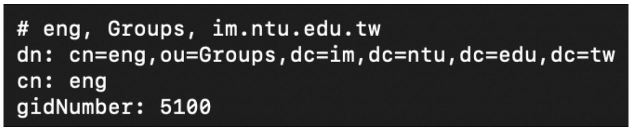
(3) 建立群組（eng / intern）

建立 UNIX 風格的 POSIX 群組，後面使用者可用 gidNumber 對應。

```
cat > groups.ldif <<'LDIF'
dn: cn=eng,ou=Groups,dc=im,dc=ntu,dc=edu,dc=tw
objectClass: posixGroup
cn: eng
gidNumber: 5100

dn: cn=intern,ou=Groups,dc=im,dc=ntu,dc=edu,dc=tw
objectClass: posixGroup
cn: intern
gidNumber: 5101
LDIF

sudo ldapadd -x -D "cn=admin,dc=im,dc=ntu,dc=edu,dc=tw" -W -f groups.ldif
```

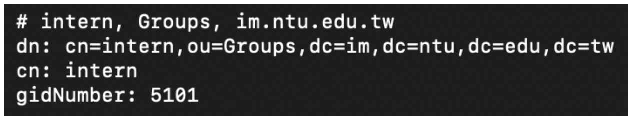
建兩個群組條目，分別給工程/實習用。

```
ldapsearch -x -b "dc=im,dc=ntu,dc=edu,dc=tw" "(cn=eng)" cn gidNumber
ldapsearch -x -b "dc=im,dc=ntu,dc=edu,dc=tw" "(cn=intern)" cn gidNumber
```



```
# eng, Groups, im.ntu.edu.tw
dn: cn=eng,ou=Groups,dc=im,dc=ntu,dc=edu,dc=tw
cn: eng
gidNumber: 5100
```

Figure 3: 建立工程群組



```
# intern, Groups, im.ntu.edu.tw
dn: cn=intern,ou=Groups,dc=im,dc=ntu,dc=edu,dc=tw
cn: intern
gidNumber: 5101
```

Figure 4: 建立實習群組

(4) 建立三個使用者並加入群組

1. 依題目建立 3 個帳號（first/last/student-id），並綁定到對應群組。
2. 產生密碼雜湊（SSHA）：slappasswd
3. 建立使用者條目（uidNumber/gidNumber/家目錄/登入殼等）：
4. 把使用者加入群組（memberUid）：

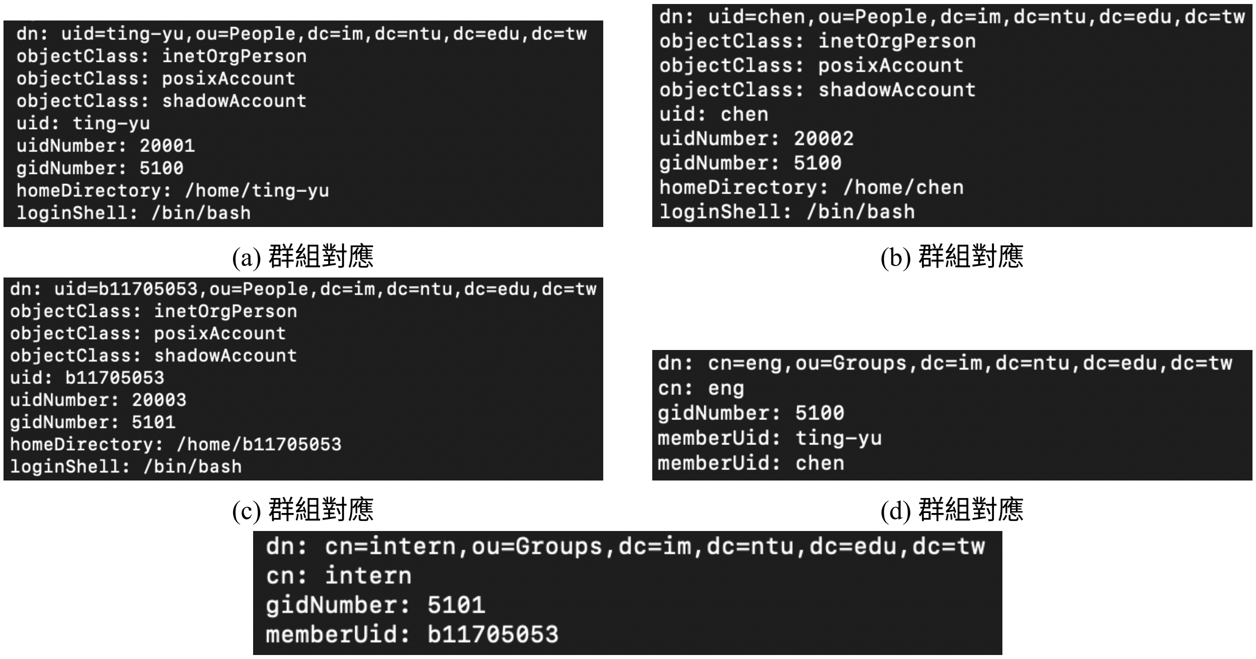
```
cat > group_members.ldif <<'LDIF'
dn: cn=eng,ou=Groups,dc=im,dc=ntu,dc=edu,dc=tw
changetype: modify
add: memberUid
memberUid: ting-yu
-
add: memberUid
memberUid: chen

dn: cn=intern,ou=Groups,dc=im,dc=ntu,dc=edu,dc=tw
changetype: modify
add: memberUid
memberUid: b11705053
LDIF

sudo ldapmodify -x -D "cn=admin,dc=im,dc=ntu,dc=edu,dc=tw" -W -f group_members.ldif
```

```
# 使用者屬性
ldapsearch -x -b "dc=im,dc=ntu,dc=edu,dc=tw" "(uid=ting-yu)" uid uidNumber gidNumber
ldapsearch -x -b "dc=im,dc=ntu,dc=edu,dc=tw" "(uid=chen)" uid uidNumber gidNumber
ldapsearch -x -b "dc=im,dc=ntu,dc=edu,dc=tw" "(uid=b11705053)" uid uidNumber gidNumber

# 群組成員
ldapsearch -x -b "dc=im,dc=ntu,dc=edu,dc=tw" "(cn=eng)" cn gidNumber memberUid
ldapsearch -x -b "dc=im,dc=ntu,dc=edu,dc=tw" "(cn=intern)" cn gidNumber memberUid
```

(5) 建 CA、產生 CSR、簽發 server.crt、啟用 LDAPS

建立自簽 CA（ca.key/ca.crt），之後用來簽伺服器憑證

```
openssl genrsa -out ~/ca.key 4096

openssl req -x509 -new -nodes -key ~/ca.key -sha256 -days 825 \
  -subj "/C=TW/ST=Taiwan/L=Taipei/O=IM Corp/OU=LSAP/CN=IM-CA" \
  -out ~/ca.crt

sudo cp ~/ca.crt /usr/local/share/ca-certificates/ldap_ca.crt
sudo update-ca-certificates
```



```
dn: uid=ting-yu,ou=People,dc=im,dc=ntu,dc=edu,dc=tw
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: ting-yu
uidNumber: 20001
gidNumber: 5100
homeDirectory: /home/ting-yu
loginShell: /bin/bash
```

(a) 群組對應

```
dn: uid=chen,ou=People,dc=im,dc=ntu,dc=edu,dc=tw
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: chen
uidNumber: 20002
gidNumber: 5100
homeDirectory: /home/chen
loginShell: /bin/bash
```

(b) 群組對應

```
dn: uid=b11705053,ou=People,dc=im,dc=ntu,dc=edu,dc=tw
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: b11705053
uidNumber: 20003
gidNumber: 5101
homeDirectory: /home/b11705053
loginShell: /bin/bash
```

(c) 群組對應

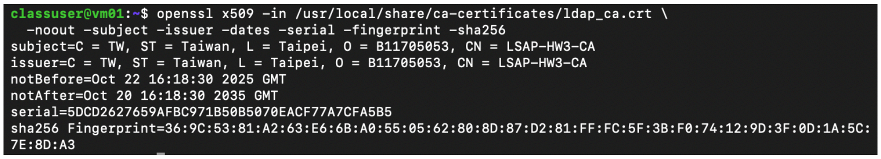
```
dn: cn=eng,ou=Groups,dc=im,dc=ntu,dc=edu,dc=tw
cn: eng
gidNumber: 5100
memberUid: ting-yu
memberUid: chen
```

(d) 群組對應

```
dn: cn=intern,ou=Groups,dc=im,dc=ntu,dc=edu,dc=tw
cn: intern
gidNumber: 5101
memberUid: b11705053
```

(e) 群組對應

Figure 5: 群組對應（整合排版）

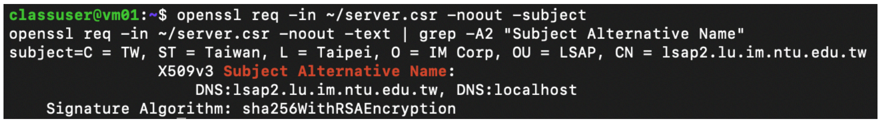


```
classuser@vm01:~$ openssl x509 -in /usr/local/share/ca-certificates/ldap_ca.crt \
  -noout -subject -issuer -dates -serial -fingerprint -sha256
subject=C = TW, ST = Taiwan, L = Taipei, O = B11705053, CN = LSAP-HW3-CA
issuer=C = TW, ST = Taiwan, L = Taipei, O = B11705053, CN = LSAP-HW3-CA
notBefore=Oct 22 16:18:30 2025 GMT
notAfter=Oct 20 16:18:30 2035 GMT
serial=5DCD2627659AFBC971B50B5070EACF77A7CFA5B5
sha256 Fingerprint=36:9C:53:81:A2:63:E6:6B:A0:55:05:62:80:8D:87:D2:81:FF:FC:5F:3B:F0:74:12:9D:3F:0D:1A:5C:
7E:8D:A3
```

Figure 6: ca.crt

LDAP 主機產生 server.key 與 server.csr

```
FQDN="lsap2.lu.im.ntu.edu.tw"
openssl genrsa -out ~/server.key 2048
openssl req -new -key ~/server.key \
  -subj "/C=TW/ST=Taiwan/L=Taipei/O=IM Corp/OU=LSAP/CN=${FQDN}" \
  -addext "subjectAltName=DNS:${FQDN},DNS:localhost" \
  -out ~/server.csr
```



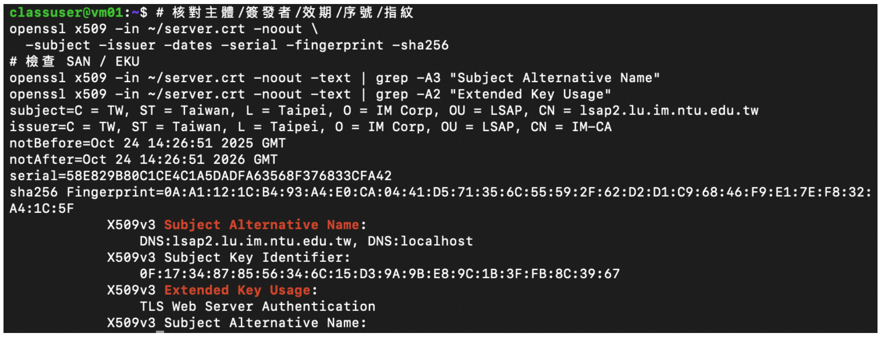
```
classuser@vm01:~$ openssl req -in ~/server.csr -noout -subject
openssl req -in ~/server.csr -noout -text | grep -A2 "Subject Alternative Name"
subject=C = TW, ST = Taiwan, L = Taipei, O = IM Corp, OU = LSAP, CN = lsap2.lu.im.ntu.edu.tw
                X509v3 Subject Alternative Name:
                    DNS:lsap2.lu.im.ntu.edu.tw, DNS:localhost
    Signature Algorithm: sha256WithRSAEncryption
```

Figure 7: server.csr

用 CA 簽 server.csr 產生 server.crt，並確保含 SAN 與 EKU: serverAuth。

```
cat > ~/openssl-ext.cnf <<'EXT'
basicConstraints=CA:FALSE
keyUsage=digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
```

```
extendedKeyUsage=serverAuth
subjectAltName=@alt_names
[alt_names]
DNS.1=lsap2.lu.im.ntu.edu.tw
DNS.2=localhost
EXT

openssl x509 -req -in ~/server.csr \
  -CA ~/ca.crt -CAkey ~/ca.key -CAcreateserial \
  -out ~/server.crt -days 365 -sha256 -extfile ~/openssl-ext.cnf
```



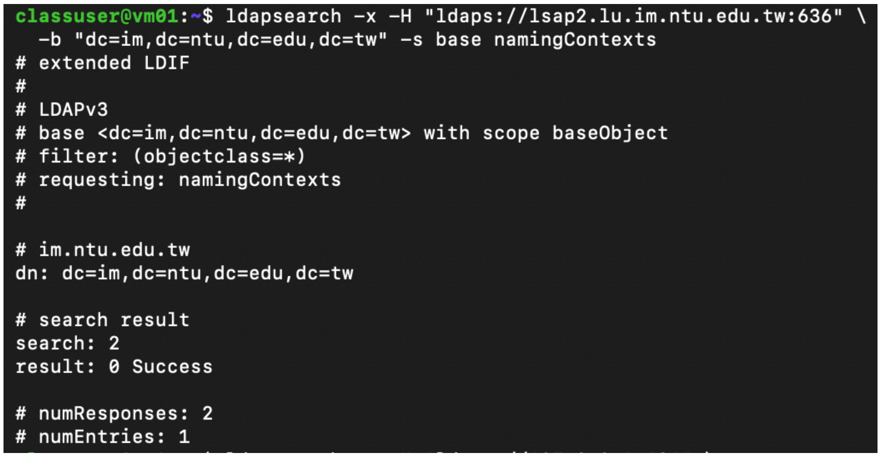
```
classuser@vm01:~$ # 核對主體/簽發者/效期/序號/指紋
openssl x509 -in ~/server.crt -noout \
  -subject -issuer -dates -serial -fingerprint -sha256
# 檢查 SAN / EKU
openssl x509 -in ~/server.crt -noout -text | grep -A3 "Subject Alternative Name"
openssl x509 -in ~/server.crt -noout -text | grep -A2 "Extended Key Usage"
subject=C = TW, ST = Taiwan, L = Taipei, O = IM Corp, OU = LSAP, CN = lsap2.lu.im.ntu.edu.tw
issuer=C = TW, ST = Taiwan, L = Taipei, O = IM Corp, OU = LSAP, CN = IM-CA
notBefore=Oct 24 14:26:51 2025 GMT
notAfter=Oct 24 14:26:51 2026 GMT
serial=58E829B80C1CE4C1A5DADFA63568F376833CFA42
sha256 Fingerprint=0A:A1:12:1C:B4:93:A4:E0:CA:04:41:D5:71:35:6C:55:59:2F:62:D2:D1:C9:68:46:F9:E1:7E:F8:32:
A4:1C:5F
            X509v3 Subject Alternative Name:
                DNS:lsap2.lu.im.ntu.edu.tw, DNS:localhost
            X509v3 Subject Key Identifier:
                0F:17:34:87:85:56:34:6C:15:D3:9A:9B:E8:9C:1B:3F:FB:8C:39:67
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Subject Alternative Name:
```

Figure 8: server.crt

(6) Enable LDAPS and Trust Your CA Locally

讓 slapd 使用自簽 CA 與伺服器憑證，開啟 ldaps:///（TCP 636），並在客戶端信任 CA，使 LDAP 工具能以 TLS 成功查詢。

```
ldapsearch -x -H ldaps://<domain-of-the-VM>:<assigned-port-of-the-VM> \
-b "dc=im,dc=ntu,dc=edu,dc=tw" -s base namingContexts
```

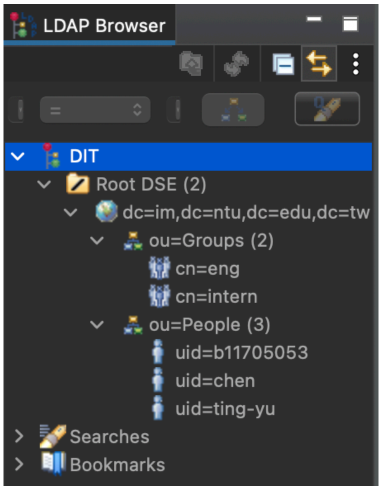


```
classuser@vm01:~$ ldapsearch -x -H "ldaps://lsap2.lu.im.ntu.edu.tw:636" \
  -b "dc=im,dc=ntu,dc=edu,dc=tw" -s base namingContexts
# extended LDIF
#
# LDAPv3
# base <dc=im,dc=ntu,dc=edu,dc=tw> with scope baseObject
# filter: (objectclass=*)
# requesting: namingContexts
#

# im.ntu.edu.tw
dn: dc=im,dc=ntu,dc=edu,dc=tw

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Figure 9: TLS 成功查詢

(7) Apache Directory Studio（ADS）連線與截圖

以圖形化工具驗證 LDAPS 與樹狀內容，提供作業需要的截圖。



```
DIT
  Root DSE (2)
    dc=im,dc=ntu,dc=edu,dc=tw
      ou=Groups (2)
        cn=eng
        cn=intern
      ou=People (3)
        uid=b11705053
        uid=chen
        uid=ting-yu
Searches
Bookmarks
```

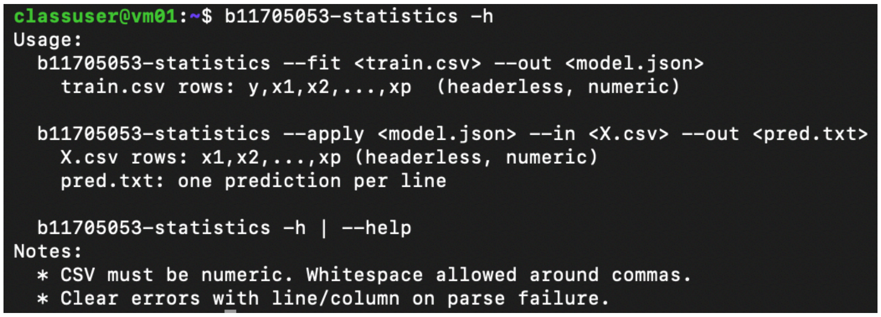
Figure 10: LDAPS 與樹狀內容

2 Custom APT Repository

(1) 打包與本機驗證（.deb）

目的：先確定套件可安裝與可執行。

```
ls -l ~/b11705053-statistics_1.0.0_amd64.deb
sudo dpkg -i ~/b11705053-statistics_1.0.0_amd64.deb || sudo apt -f install -y
dpkg -s b11705053-statistics | grep -E 'Status|Version'
which b11705053-statistics
b11705053-statistics -h
```



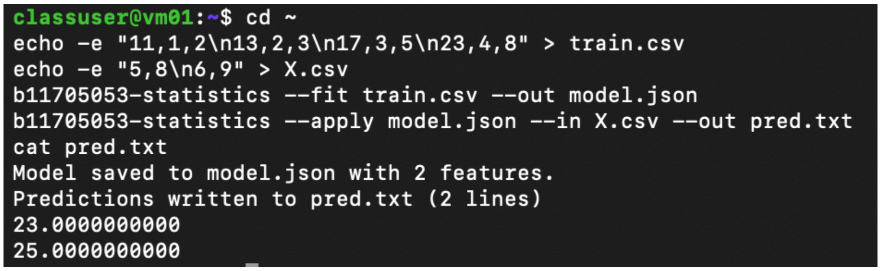
```
classuser@vm01:~$ b11705053-statistics -h
Usage:
  b11705053-statistics --fit <train.csv> --out <model.json>
    train.csv rows: y,x1,x2,...,xp  (headerless, numeric)

  b11705053-statistics --apply <model.json> --in <X.csv> --out <pred.txt>
    X.csv rows: x1,x2,...,xp (headerless, numeric)
    pred.txt: one prediction per line

  b11705053-statistics -h | --help
Notes:
  * CSV must be numeric. Whitespace allowed around commas.
  * Clear errors with line/column on parse failure.
```

Figure 11: 幫助訊息

(2) 建立倉庫目錄與放入套件

目的：建出 APT 標準結構，將 .deb 置於 pool/。

```
sudo install -d /var/www/html/apt/{pool/main,dists/stable/main/binary-amd64}
sudo cp ~/b11705053-statistics_1.0.0_amd64.deb /var/www/html/apt/pool/main/
```



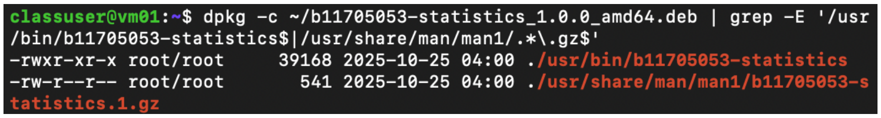
```
classuser@vm01:~$ cd ~
echo -e "11,1,2\n13,2,3\n17,3,5\n23,4,8" > train.csv
echo -e "5,8\n6,9" > X.csv
b11705053-statistics --fit train.csv --out model.json
b11705053-statistics --apply model.json --in X.csv --out pred.txt
cat pred.txt
Model saved to model.json with 2 features.
Predictions written to pred.txt (2 lines)
23.0000000000
25.0000000000
```

Figure 12: 確認輸出

(3) 產生索引與 Release

目的：用 dpkg-scanpackages 生成 Packages，再以 apt-ftparchive 生成 Release。

```
cd /var/www/html/apt
sudo bash -c 'dpkg-scanpackages --multiversion pool > dists/stable/main/binary-amd64/
    Packages'
sudo gzip -kf dists/stable/main/binary-amd64/Packages

sudo tee apt-ftparchive.conf >/dev/null <<'CONF'
APT::FTPArchive::Release {
  Origin "b11705053";
  Label "b11705053 APT";
  Suite "stable";
  Codename "stable";
  Architectures "amd64";
  Components "main";
  Description "Custom repo for b11705053-statistics";
};
CONF

sudo bash -c 'apt-ftparchive -c apt-ftparchive.conf release dists/stable > dists/stable/
    Release'
```



```
classuser@vm01:~$ dpkg -c ~/b11705053-statistics_1.0.0_amd64.deb | grep -E '/usr
/bin/b11705053-statistics$|/usr/share/man/man1/.*\.gz$'
-rwxr-xr-x root/root     39168 2025-10-25 04:00 ./usr/bin/b11705053-statistics
-rw-r--r-- root/root       541 2025-10-25 04:00 ./usr/share/man/man1/b11705053-s
tatistics.1.gz
```

Figure 13: deb 內含正確檔案

(4) GPG 金鑰與簽署

目的：以私鑰簽出 InRelease/Release.gpg，並匯出公鑰供客戶端信任。

```
gpg --quick-generate-key "b11705053 APT Repo <you@example.com>" default default never
gpg --list-keys "b11705053 APT Repo <you@example.com>"

gpg --default-key "b11705053 APT Repo <you@example.com>" --clearsign \
  -o /tmp/InRelease /var/www/html/apt/dists/stable/Release
gpg --default-key "b11705053 APT Repo <you@example.com>" -abs \
  -o /tmp/Release.gpg /var/www/html/apt/dists/stable/Release
sudo mv /tmp/InRelease /var/www/html/apt/dists/stable/InRelease
sudo mv /tmp/Release.gpg /var/www/html/apt/dists/stable/Release.gpg

gpg --export -a "b11705053 APT Repo <you@example.com>" \
 | sudo tee /var/www/html/apt/b11705053-archive-keyring.gpg.asc >/dev/null
```

(5) 以 Apache 提供 HTTP

目的：讓倉庫經由 HTTP 對外可取用（你使用 Apache，而非臨時 127.0.0.1:8000）。

```
sudo apt install -y apache2
sudo systemctl enable --now apache2

curl -I http://<IP>/apt/dists/stable/InRelease
curl -I http://<IP>/apt/dists/stable/main/binary-amd64/Packages.gz
```

(6) 客戶端加入來源並安裝

目的：客戶端 dearmor 匯入公鑰、加入 source、更新索引並安裝。

```
sudo mkdir -p /usr/share/keyrings
curl -fsSL http://<IP>/apt/b11705053-archive-keyring.gpg.asc \
 | sudo gpg --dearmor -o /usr/share/keyrings/b11705053-archive-keyring.gpg
sudo chmod 644 /usr/share/keyrings/b11705053-archive-keyring.gpg

echo "deb [arch=amd64 signed-by=/usr/share/keyrings/b11705053-archive-keyring.gpg] \
http://<IP>/apt stable main" | sudo tee /etc/apt/sources.list.d/b11705053.list

sudo apt update
apt-cache policy b11705053-statistics | sed -n '1,12p'
sudo apt install -y b11705053-statistics
```

(7) 功能驗證

目的：用小型 CSV 跑訓練與推論，確認輸出正確。

```
echo -e "11,1,2\n13,2,3\n17,3,5\n23,4,8" > train.csv
echo -e "5,8\n6,9" > X.csv
b11705053-statistics --fit train.csv --out model.json
b11705053-statistics --apply model.json --in X.csv --out pred.txt
cat pred.txt # 23.0000000000 / 25.0000000000
```