

## Task 1: สร้าง User Accounts สำหรับ Team (30 นาที)

### 1.1 สร้าง Users และ Groups:

#### 1. สร้างกลุ่ม (Groups)

- `sudo groupadd developers`
- `sudo groupadd testers`
- `sudo groupadd dbadmin`

`groupadd` ใช้สำหรับสร้างกลุ่มใหม่ในระบบที่นี้คุณสร้าง 3 กลุ่มหลัก:

`developers` → สำหรับนักพัฒนา

`testers` → สำหรับผู้ทดสอบ

`dbadmin` → สำหรับผู้ดูแลฐานข้อมูล

#### 2. สร้างผู้ใช้ (Users) และกำหนดกลุ่ม

- `sudo useradd -m -s /bin/bash -G developers alice`
- `sudo useradd -m -s /bin/bash -G developers bob`
- `sudo useradd -m -s /bin/bash -G testers charlie`
- `sudo useradd -m -s /bin/bash -G dbadmin david`

`useradd` ใช้สร้างบัญชีผู้ใช้ใหม่คำสั่งที่ใช้ประกอบไปด้วย:

`-m` → สร้าง home directory ให้ผู้ใช้อัตโนมัติ (/home/username)

`-s /bin/bash` → ตั้งค่า shell เริ่มต้นเป็น bash

`-G <group>` → กำหนดกลุ่มที่ผู้ใช้สังกัด

ดังนั้น:

- alice และ bob อยู่ในกลุ่ม `developers`
- charlie อยู่ในกลุ่ม `testers`
- david อยู่ในกลุ่ม `dbadmin`

#### 3. ตั้งรหัสผ่านให้แต่ละผู้ใช้

- `sudo passwd alice`
- `sudo passwd bob`
- `sudo passwd charlie`
- `sudo passwd david`

- `passwd` ใช้เปลี่ยนหรือกำหนดรหัสผ่าน

- ระบบจะให้กรอกรหัสใหม่ 2 ครั้ง (เพื่อยืนยัน)

หลังจากตั้งรหัสผ่านเสร็จ ผู้ใช้ก็สามารถเข้าสู่ระบบได้

ผลลัพธ์

- ได้ Users และ Groups ตามที่กำหนดครบถ้วน
- แต่ละ User มี home directory ของตัวเอง และมี shell พร้อมใช้งาน
- บัญชีทั้งหมดถูกแบ่ง role ตามหน้าที่ (`developers/testers/dbadmin`) → ทำให้จัดสิทธิ์การเข้าถึงไฟล์/โฟลเดอร์ได้ง่ายขึ้น

```
[deeno@arch ~]$ sudo groupadd developers
[sudo] password for deeno:
[deeno@arch ~]$ sudo groupadd testers
[deeno@arch ~]$ sudo groupadd dbadmin
[deeno@arch ~]$ sudo useradd -s /bin/bash -G developers alice
[deeno@arch ~]$ sudo useradd -s /bin/bash -G developers bob
[deeno@arch ~]$ sudo useradd -s /bin/bash -G testers charlie
[deeno@arch ~]$ sudo useradd -s /bin/bash -G dbadmin david
[deeno@arch ~]$ sudo passwd alice
New password:
Retype new password:
passwd: password updated successfully
[deeno@arch ~]$ sudo passwd bob
New password:
Retype new password:
passwd: password updated successfully
[deeno@arch ~]$ sudo passwd charlie
New password:
Retype new password:
passwd: password updated successfully
[deeno@arch ~]$ sudo passwd david
New password:
Retype new password:
passwd: password updated successfully
[deeno@arch ~]$
```

## 1.2 ตั้งค่า Password Policy

เพื่อกำหนดนโยบายรหัสผ่าน (Password Policy) ให้รัดกุม ปลอดภัย และเป็นมาตรฐานเดียวกันสำหรับผู้ใช้งานทุกคนในระบบ

### 1. แก้ไขไฟล์ `/etc/login.defs`

`sudo nano /etc/login.defs`

```
[deeno@arch ~]$ sudo nano /etc/login.defs
```

ปรับค่าที่เกี่ยวข้องกับอายุการใช้งานของรหัสผ่าน:

- `PASS_MAX_DAYS 90` → รหัสผ่านมีอายุใช้งานสูงสุด 90 วัน
- `PASS_MIN_DAYS 7` → ผู้ใช้ต้องรออย่างน้อย 7 วันก่อนจะเปลี่ยนรหัสผ่านใหม่ได้
- `PASS_WARN_AGE 14` → เตือนผู้ใช้ 14 วันก่อนที่รหัสผ่านจะหมดอายุ
- `PASS_MIN_LEN 12` → กำหนดความยาวขั้นต่ำของรหัสผ่าน 12 ตัวอักษร

```
GNU nano 2.8.6 /etc/login.defs
#
# Password aging controls:
#
# PASS_MAX_DAYS Maximum number of days a password may be used.
# PASS_MIN_DAYS Minimum number of days allowed between password changes.
# PASS_WARN_AGE Number of days warning given before a password expires.
#
PASS_MAX_DAYS 30
PASS_MIN_DAYS 7
PASS_WARN_AGE 14
PASS_MIN_LEN 12
#
# Min/max values for automatic uid selection in useradd(8)
#
UID_MIN 1000
UID_MAX 60000
# System accounts
SYS_UID_MIN 500
SYS_UID_MAX 999
# Extra per user uids
SUB_UID_MIN 100000
SUB_UID_MAX 600100000
SUB_UID_COUNT 65536
#
# Min/max values for automatic gid selection in groupadd(8)
#
GID_MIN 1000
GID_MAX 60000
# System accounts
SYS_GID_MIN 500
SYS_GID_MAX 999
# Extra per user group ids
SUB_GID_MIN 100000
SUB_GID_MAX 600100000
SUB_GID_COUNT 65536
#
# Max number of login(1) retries if password is bad
#
LOGIN_RETRIES 5
#
```

## 2. ติดตั้ง libpam-pwquality

(สำหรับ Arch Linux ใช้ `pacman -S`, ใน Debian/Ubuntu ใช้ `apt install`)

`sudo pacman -S libpwquality`

## 3. แก้ไขไฟล์ `/etc/pam.d/common-password`

`sudo nano /etc/pam.d/common-password`

```
[deeno@arch ~]$ sudo nano /etc/login.defs
[deeno@arch ~]$ sudo pacman -S libpwquality
resolving dependencies...
looking for conflicting packages...

Package (2)      New Version  Net Change  Download Size
core/cracklib    2.10.3-1     0.91 MiB   0.27 MiB
extra/libpwquality 1.4.5-6     0.43 MiB   0.09 MiB

Total Download Size: 0.36 MiB
Total Installed Size: 1.34 MiB

:: Proceed with installation? [Y/n] y
:: Retrieving packages...
libpwquality-1.4.5-6...  93.8 KiB   291 KiB/s 00:00 [=====] 100%
cracklib-2.10.3-1-x86_64 277.2 KiB   726 KiB/s 00:00 [=====] 100%
Total (2/2)              371.0 KiB   879 KiB/s 00:00 [=====] 100%
(2/2) checking keys in keyring [=====] 100%
(2/2) checking package integrity [=====] 100%
(2/2) loading package files [=====] 100%
(2/2) checking for file conflicts [=====] 100%
(2/2) checking available disk space [=====] 100%
:: Processing package changes... [=====] 100%
(4/2) installing cracklib [=====] 100%
(2/2) installing libpwquality [=====] 100%
Optional dependencies for libpwquality
python: Python bindings [installed]
Running post-transaction hooks...
(1/1) Arming ConditionNeedsUpdate...
[deeno@arch ~]$ sudo nano /etc/pam.d/common-password
[deeno@arch ~]$
```

password requisite pam\_pwquality.so retry=3 minlen=12 difok=3 ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1

ความหมายของแต่ละพารามิเตอร์:

- `retry=3` → ให้ลองใส่รหัสผ่านใหม่ได้ 3 ครั้ง
- `minlen=12` → รหัสผ่านต้องยาวอย่างน้อย 12 ตัวอักษร

- difok=3 → รหัสผ่านใหม่ต้องแตกต่างจากรหัสผ่านเก่าอย่างน้อย 3 ตัวอักษร
- ucredit=-1 → ต้องมีอักษร **ตัวพิมพ์ใหญ่** อย่างน้อย 1 ตัว
- lcredit=-1 → ต้องมีอักษร **ตัวพิมพ์เล็ก** อย่างน้อย 1 ตัว
- dcredit=-1 → ต้องมี **ตัวเลข** อย่างน้อย 1 ตัว
- ocredit=-1 → ต้องมี **อักขระพิเศษ** อย่างน้อย 1 ตัว

```
GNU nano 8.6 /etc/pam.d/common-password
password requisite pam_pwquality.so retry=3 minlen=12 difok=3 ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1
Modified
```

```
[deeno@arch ~]$ cat /etc/passwd | tail -4
david:x:1004:1008::/home/david:/bin/bash
elasticsearch:x:966:966::/var/lib/elasticsearch:/sbin/nologin
kibana:x:965:965::/var/lib/kibana:/sbin/nologin
logstash:x:964:964::/var/lib/logstash:/sbin/nologin
[deeno@arch ~]$
```

## 2.1 สร้าง Sudo Groups:

# สร้าง custom sudo groups

sudo groupadd sudo-developers

sudo groupadd sudo-limited

# เพิ่ม users เข้า groups

sudo usermod -aG sudo-developers alice

sudo usermod -aG sudo-developers bob

sudo usermod -aG sudo-limited charlie

กลุ่ม sudo-developers → alice, bob (สิทธิ์เต็ม)

กลุ่ม sudo-limited → charlie (สิทธิ์เฉพาะบางคำสั่ง)

```
deeno@arch ~$ sudo groupadd sudo-developers
[deeno@arch ~]$ sudo groupadd sudo-limited
[deeno@arch ~]$ sudo usermod -s6 sudo-developers alice
[deeno@arch ~]$ sudo usermod -s6 sudo-developers bob
[deeno@arch ~]$ sudo usermod -s6 sudo-limited charlie
[deeno@arch ~]$
```

## 2.2 Configure Sudoers:

sudo EDITOR=nano visudo

```
deeno@arch ~$ sudo EDITOR=nano visudo
[deeno@arch ~]$
```

เพิ่มกฎ:

```
# Developers - full sudo access
%sudo-developers ALL=(ALL:ALL) ALL

# Limited sudo - specific commands only
%sudo-limited ALL=(ALL) /usr/bin/systemctl status *, /usr/bin/tail /var/log/*,
/bin/ps

# Database admin - database commands only
david ALL=(ALL) /usr/bin/mysql, /usr/bin/mysqldump, /bin/systemctl restart
mysql

# Sudo session timeout (15 minutes)
Defaults timestamp_timeout=15

# Log sudo commands
Defaults logfile="/var/log/sudo.log"

Defaults log_input, log_output
```

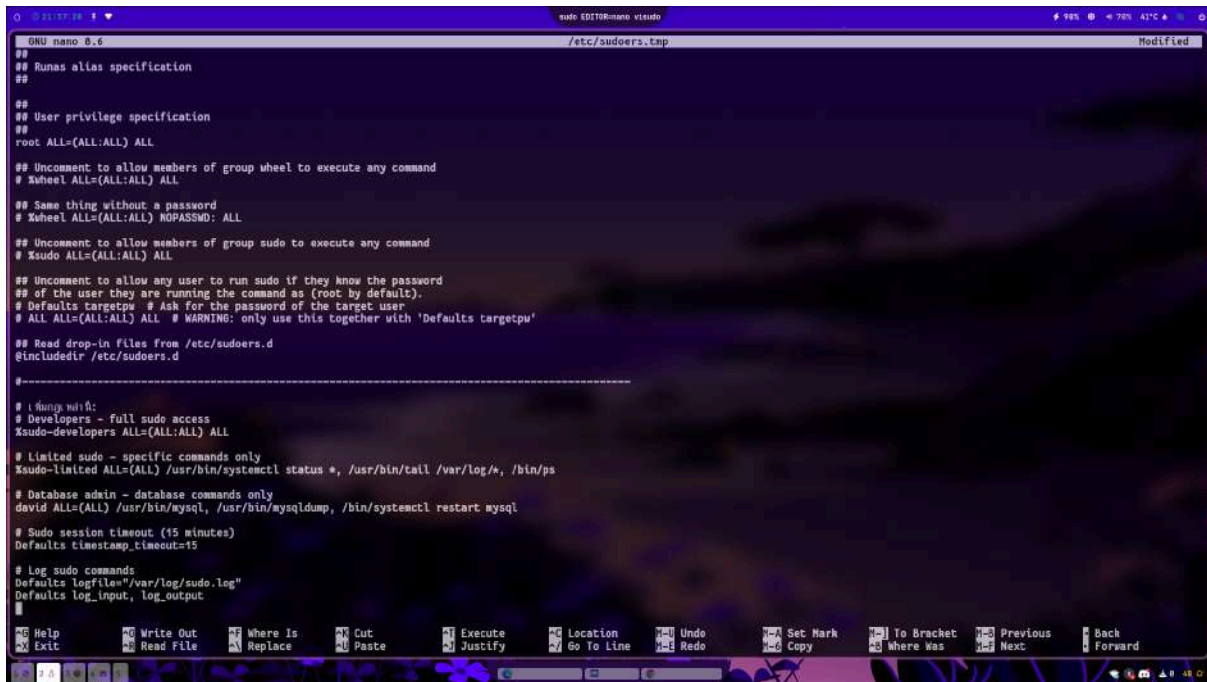
sudo-developers → ใช้ sudo ได้เต็มทุกคำสั่ง

sudo-limited → ใช้ได้เฉพาะ: systemctl status, tail /var/log/\*, ps

david (dbadmin) → ใช้ได้เฉพาะ: mysql, mysqldump, systemctl restart mysql

ตั้งเวลา session timeout = 15 นาที

เก็บ log ทุกคำสั่งที่รันด้วย sudo ลงไฟล์ /var/log/sudo.log



```
GNU nano 2.9.6 /etc/sudoers.conf
##
## Runas alias specification
##
##
## User privilege specification
##
root ALL=(ALL:ALL) ALL

## Uncomment to allow members of group wheel to execute any command
## wheel ALL=(ALL:ALL) ALL

## Same thing without a password
## wheel ALL=(ALL:ALL) NOPASSWD: ALL

## Uncomment to allow members of group sudo to execute any command
## sudo ALL=(ALL:ALL) ALL

## Uncomment to allow any user to run sudo if they know the password
## of the user they are running the command as (root by default).
## Defaults targetpw. # Ask for the password of the target user
## ALL ALL=(ALL:ALL) ALL # WARNING: only use this together with 'Defaults targetpw'

## Read drop-in files from /etc/sudoers.d
@includedir /etc/sudoers.d

# Things which
# Developers - Full sudo access
Xsudo-developers ALL=(ALL:ALL) ALL

# Limited sudo - specific commands only
Xsudo-limited ALL=(ALL) /usr/bin/systemctl status *, /usr/bin/tail /var/log/*, /bin/ps

# Database admin - database commands only
david ALL=(ALL) /usr/bin/mysql, /usr/bin/mysqldump, /bin/systemctl restart mysql

# Sudo session timeout (15 minutes)
Defaults timestamp_timeout=15

# Log sudo commands
Defaults logfile="/var/log/sudo.log"
Defaults log_input, log_output
```

## 2.3 ทดสอบ Sudo Permissions:

sudo -u alice sudo ls /root

ใช้งานได้ → ยืนยันว่า **alice** มี full sudo access

sudo -u charlie sudo systemctl status ssh

ใช้ได้ (เพราะอนุญาต **systemctl status**)

sudo -u charlie sudo pacman -Syu

ไม่อนุญาต (ขึ้น error ว่าไม่มีสิทธิ์) → ยืนยันว่า **charlie** ถูกจำกัดตาม policy



```
[deeno@arch ~]$ sudo -u alice sudo ls /root
[sudo] password for alice:
[deeno@arch ~]$ sudo -u charlie sudo systemctl status ssh
[sudo] password for charlie:
Sorry, user charlie is not allowed to execute '/usr/bin/systemctl status.ssh' as root on arch.
[deeno@arch ~]$ sudo -u charlie sudo pacman -Syu
[sudo] password for charlie:
Sorry, user charlie is not allowed to execute '/usr/bin/pacman -Syu' as root on arch.
[deeno@arch ~]$
```

## 3.1 Backup และแก้ไข SSH Config:

sudo cp /etc/ssh/sshd\_config /etc/ssh/sshd\_config.backup

sudo nano /etc/ssh/sshd\_config

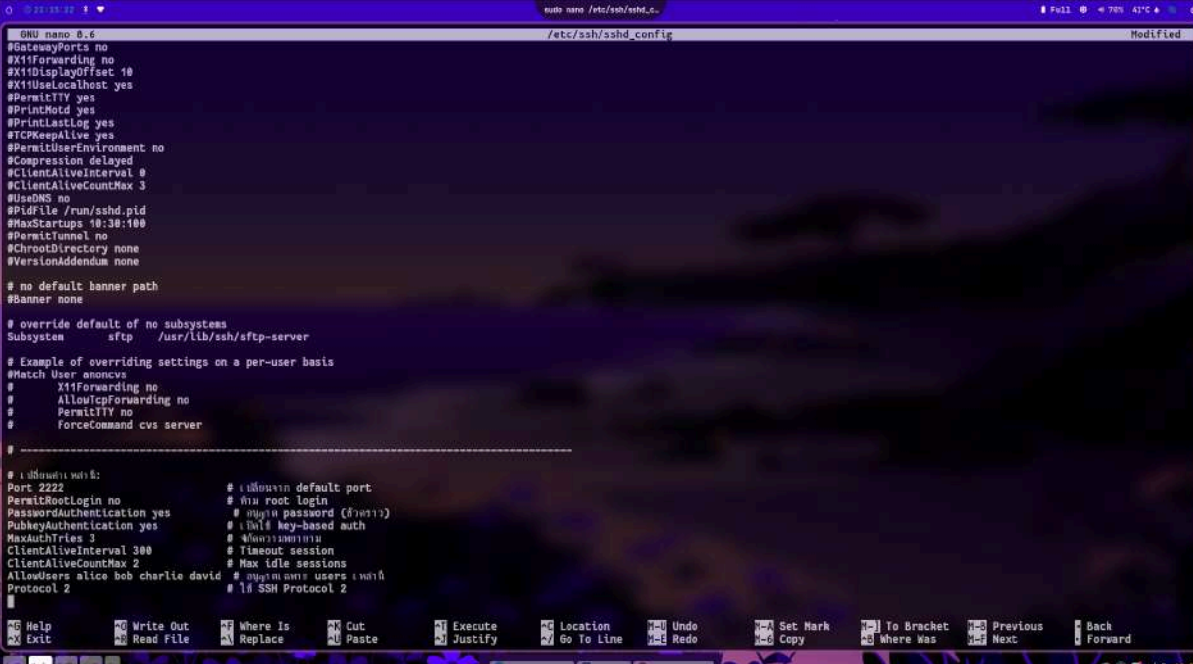


```
[deeno@arch ~]$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.backup
[deeno@arch ~]$ sudo nano /etc/ssh/sshd_config
[deeno@arch ~]$
```



# เปลี่ยนค่าเหล่านี้:

Port 2222	# เปลี่ยนจาก default port
PermitRootLogin no	# ห้าม root login
PasswordAuthentication yes	# อนุญาต password (ชั่วคราว)
PubkeyAuthentication yes	# เปิดใช้ key-based auth
MaxAuthTries 3	# จำกัดความพยายาม
ClientAliveInterval 300	# Timeout session
ClientAliveCountMax 2	# Max idle sessions
AllowUsers alice bob charlie david	# อนุญาตเฉพาะ users เหล่านี้
Protocol 2	# ใช้ SSH Protocol 2



```
GNU nano 2.9.6 /etc/ssh/sshd_config
# GatewayPorts no
#X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
#PrintMotd yes
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# override default of no subsystems
Subsystem sftp /usr/lib/ssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#    X11Forwarding no
#    AllowTcpForwarding no
#    PermitTTY no
#    ForceCommand cvs server

# เปลี่ยนค่าเหล่านี้:
Port 2222                # เปลี่ยนจาก default port
PermitRootLogin no       # ห้าม root login
PasswordAuthentication yes # อนุญาต password (ชั่วคราว)
PubkeyAuthentication yes # เปิดใช้ key-based auth
MaxAuthTries 3           # จำกัดความพยายาม
ClientAliveInterval 300  # Timeout session
ClientAliveCountMax 2    # Max idle sessions
AllowUsers alice bob charlie david # อนุญาตเฉพาะ users เหล่านี้
Protocol 2               # ใช้ SSH Protocol 2
```

### 3.2 สร้าง SSH Keys:

#### 1. สร้าง key pair ขนาด 4096-bit

`sudo -u alice ssh-keygen -t rsa -b 4096 -C "alice@company.com"`

- ได้ไฟล์ private key → `/home/alice/.ssh/id_rsa`

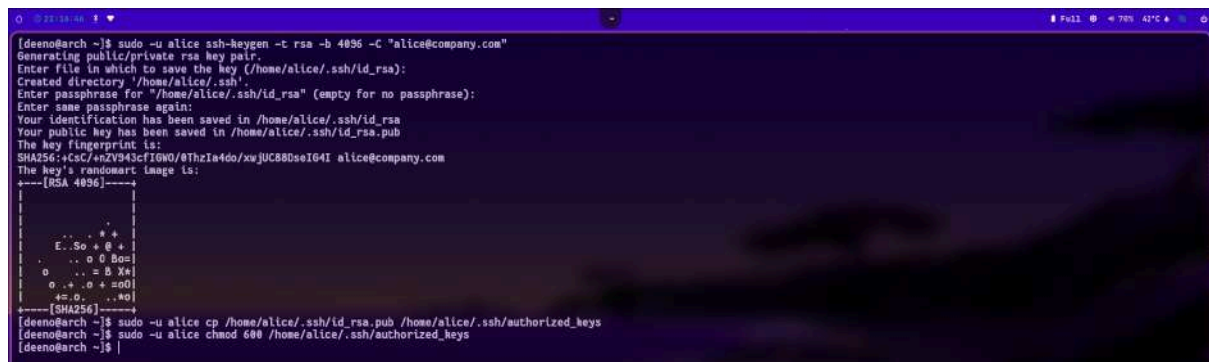
- ได้ไฟล์ public key → `/home/alice/.ssh/id_rsa.pub`

#### 2. ตั้งค่า authorized\_keys

`sudo -u alice cp /home/alice/.ssh/id_rsa.pub /home/alice/.ssh/authorized_keys`

`sudo -u alice chmod 600 /home/alice/.ssh/authorized_keys`

- อนุญาตให้ alice login ด้วย key-based authentication



```
[deeno@arch ~]$ sudo -u alice ssh-keygen -t rsa -b 4096 -C "alice@company.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alice/.ssh/id_rsa):
Created directory '/home/alice/.ssh'.
Enter passphrase for '/home/alice/.ssh/id_rsa' (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alice/.ssh/id_rsa
Your public key has been saved in /home/alice/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:4CtC/47V943cfIGN0/8ThzIe4do/xwJUC88DseIG4I alice@company.com
The key's randomart image is:
+-----[RSA 4096]-----+
|
| .. . + +
| . . . + +
| E . So + +
| . . . 0 0 Bos|
| . . . = B Xa|
| 0 + . 0 + s00|
| + . 0 . . +Q|
+-----[SHA256]-----+
[deeno@arch ~]$ sudo -u alice cp /home/alice/.ssh/id_rsa.pub /home/alice/.ssh/authorized_keys
[deeno@arch ~]$ sudo -u alice chmod 600 /home/alice/.ssh/authorized_keys
[deeno@arch ~]$
```

### 3.3 Configure SSH Banner:

#### สร้าง warning banner

`sudo nano /etc/ssh/ssh_banner.txt`

ใส่ข้อความเนื้อหา banner:

\*\*\*\*\*

**WARNING: Authorized access only!**

All connections are monitored and recorded.

Disconnect immediately if you are not an  
authorized user.

\*\*\*\*\*

**ตรงนี้คือข้อความ Warning Banner ที่จะโชว์ให้ผู้ใช้เห็นทุกครั้งก่อนเข้าสู่ระบบผ่าน SSH (ก่อน login prompt)**

เพิ่มใน sshd\_config

เปิดไฟล์การตั้งค่า SSH:

`sudo nano /etc/ssh/sshd_config`

เพิ่ม (หรือแก้ไข) บรรทัดนี้:

Banner `/etc/ssh/ssh_banner.txt`



```
[deeno@arch ~]$ sudo nano /etc/ssh/ssh_banner.txt
[deeno@arch ~]$ |

GNU nano 2.9.3 /etc/ssh/ssh_banner.txt
*****
WARNING: Authorized access only!
All connections are monitored and recorded.
Disconnect immediately if you are not an
authorized user.
*****

GNU nano 2.9.3 /etc/ssh/sshd_config
Modified
#X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
#PrintMotd yes
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# override default of no subsystems
Subsystem sftp /usr/lib/ssh/sftp-server

# Example of overriding settings on a per-user basis
Match User anoncvs
#
X11Forwarding no
#
AllowTcpForwarding no
#
PermitTTY no
#
ForceCommand cvs server

# (เพิ่มเพิ่มเพิ่ม)
Port 2222 # เปลี่ยนจาก default port
PermitRootLogin no # ห้าม root login
PasswordAuthentication yes # อนุญาต password (ถ้ามี)
PubkeyAuthentication yes # อนุญาต key-based auth
MaxAuthTries 3 # พยายามเชื่อมต่อ
ClientAliveInterval 300 # Timeout session
ClientAliveCountMax 2 # Max idle sessions
AllowUsers alice bob charlie david # อนุญาตเฉพาะ users เหล่านี้
Protocol 2 # ใช้ SSH Protocol 2

Banner /etc/ssh/ssh_banner.txt
```

### 3.4 Restart SSH และทดสอบ:

ทดสอบ config ก่อน restart

`sudo sshd -t`

คำสั่งนี้ใช้ตรวจสอบว่าไฟล์ `sshd_config` ที่เราแก้ไขมี syntax ถูกต้องหรือไม่

ถ้า ไม่มี output อะไรเลย แสดงว่า config ถูกต้อง

ถ้า config ผิด จะมี error ใ้มา เช่น path ไม่ถูก หรือ option พิมพ์ผิด

Restart SSH service

`sudo systemctl restart sshd`

ใช้คำสั่งสตาร์ท daemon ของ SSH server (sshd)

ทดสอบการเชื่อมต่อ

`ssh -p 2222 alice@localhost`

- `ssh` = ใช้เชื่อมต่อไปยัง SSH server
- `-p 2222` = ระบุ port ที่ SSH server ฟังอยู่ (ในที่นี้คือ **2222** ไม่ใช่ค่า default 22)
- `alice@localhost` = พยายามเชื่อมต่อไปยัง host localhost ด้วย user alice

```
[deeno@arch ~]$ sudo sshd -t
[deeno@arch ~]$ sudo systemctl restart sshd
[deeno@arch ~]$ ssh -p 2222 alice@localhost
The authenticity of host '[localhost]:2222 ([::1]:2222)' can't be established.
ED25519 key fingerprint is SHA256:1W5zRepJrJ5nEv35no3yu9RIyKjYMcKwMUQ16g3t0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2222' (ED25519) to the list of known hosts.
# I am banner:
*****
WARNING: Authorized access only!
All connections are monitored and recorded.
Disconnect immediately if you are not an
authorized user.
*****
alice@localhost's password:
[alice@arch ~]$
```

#### 4.1 Configure UFW:

##### # Reset UFW to default

`sudo ufw --force reset`

- ลบกฎทั้งหมดที่เคยตั้งไว้
- คืนค่า UFW เป็นค่าเริ่มต้น
- `--force` ใช้เพื่อไม่ให้ถามยืนยัน

##### # Set default policies

`sudo ufw default deny incoming`

`sudo ufw default allow outgoing`

- `deny incoming` = บล็อกการเชื่อมต่อจากภายนอกทั้งหมด เว้นแต่จะอนุญาตเป็นพิเศษ
- `allow outgoing` = อนุญาตการเชื่อมต่อจากเครื่องเราออกไปภายนอกได้

##### # Allow SSH (new port)

`sudo ufw allow 2222/tcp`

- อนุญาตการเชื่อมต่อ SSH ผ่าน port 2222 (ที่คุณตั้งใน `sshd_config`)
- โปรโตคอล TCP เท่านั้น

##### # Allow web services

`sudo ufw allow 80/tcp`

`sudo ufw allow 443/tcp`

- เปิดให้เข้าถึง HTTP (80) และ HTTPS (443)
- ใช้สำหรับ web server เช่น Apache, Nginx

##### # Allow specific IPs only for SSH (optional)

`# sudo ufw allow from 192.168.1.0/24 to any port 2222`

- จำกัดให้เฉพาะ IP/Subnet ที่กำหนดสามารถเข้า SSH ได้
- ตัวอย่าง: `192.168.1.0/24` = เครือข่าย local เท่านั้น

##### # Enable UFW

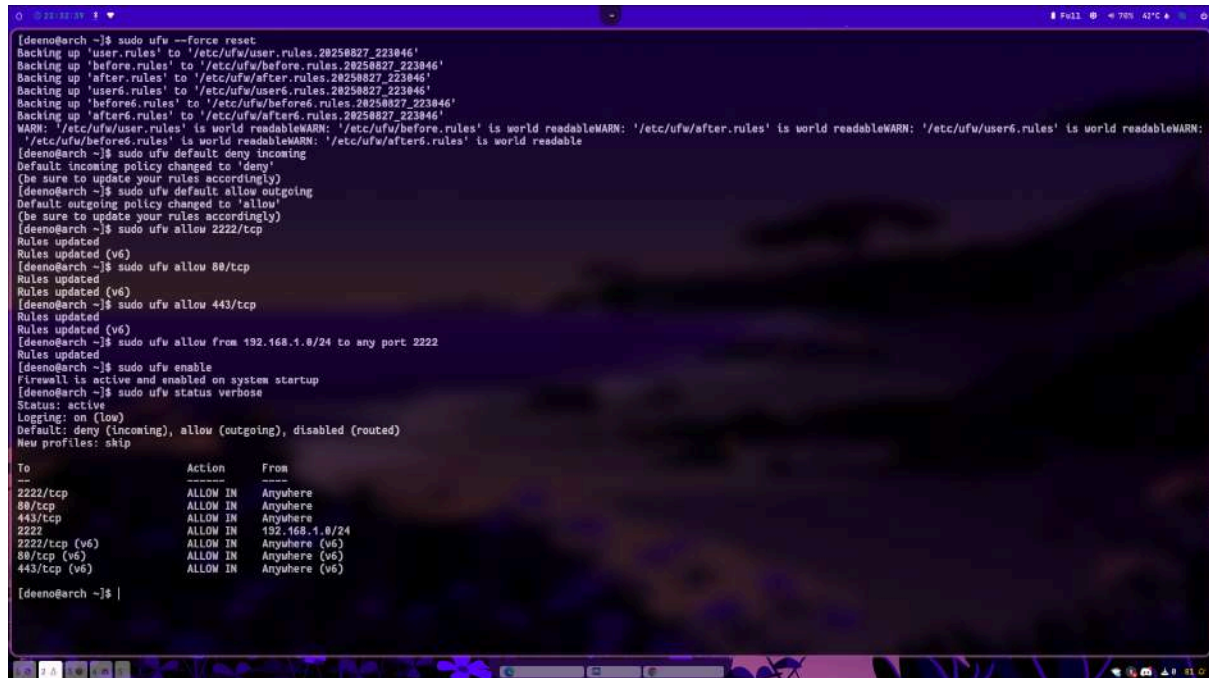
`sudo ufw enable`

- เปิดใช้งาน firewall อย่างเป็นทางการ
- หลังจากนี้ traffic จะถูกกรองตามกฎที่ตั้งไว้

# Show status

`sudo ufw status verbose`

- แสดงสถานะ UFW ปัจจุบัน
- verbose จะบอกละเอียดขึ้น เช่น default policy และ logging



```
[deeno@arch ~]$ sudo ufw --force reset
Backing up 'user.rules' to '/etc/ufw/user.rules.20250827_223046'
Backing up 'before.rules' to '/etc/ufw/before.rules.20250827_223046'
Backing up 'after.rules' to '/etc/ufw/after.rules.20250827_223046'
Backing up 'user6.rules' to '/etc/ufw/user6.rules.20250827_223046'
Backing up 'before6.rules' to '/etc/ufw/before6.rules.20250827_223046'
Backing up 'after6.rules' to '/etc/ufw/after6.rules.20250827_223046'
WARN: '/etc/ufw/user.rules' is world readableWARN: '/etc/ufw/before.rules' is world readableWARN: '/etc/ufw/after.rules' is world readableWARN: '/etc/ufw/user6.rules' is world readableWARN: '/etc/ufw/before6.rules' is world readableWARN: '/etc/ufw/after6.rules' is world readable
[deeno@arch ~]$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(Be sure to update your rules accordingly)
[deeno@arch ~]$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(Be sure to update your rules accordingly)
[deeno@arch ~]$ sudo ufw allow 2222/tcp
Rules updated
Rules updated (v6)
[deeno@arch ~]$ sudo ufw allow 88/tcp
Rules updated
Rules updated (v6)
[deeno@arch ~]$ sudo ufw allow 443/tcp
Rules updated
Rules updated (v6)
[deeno@arch ~]$ sudo ufw allow from 192.168.1.0/24 to any port 2222
Rules updated
Rules updated (v6)
[deeno@arch ~]$ sudo ufw enable
Firewall is active and enabled on system startup
[deeno@arch ~]$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To Action From
--
2222/tcp ALLOW IN Anywhere
88/tcp ALLOW IN Anywhere
443/tcp ALLOW IN Anywhere
2222 ALLOW IN 192.168.1.0/24
2222/tcp (v6) ALLOW IN Anywhere (v6)
88/tcp (v6) ALLOW IN Anywhere (v6)
443/tcp (v6) ALLOW IN Anywhere (v6)

[deeno@arch ~]$
```

## 4.2 Advanced UFW Rules:

# Rate limiting for SSH

`sudo ufw limit 2222/tcp`

- เหมือนกับ allow แต่มี rate limiting
- ถ้า IP เดียวกันพยายามเชื่อมต่อ SSH (port 2222) เกิน 6 ครั้งใน 30 วินาที → UFW จะ block ชั่วคราว
- ใช้ป้องกัน brute-force SSH login attacks ได้ดี

# Allow MySQL only from specific network

`sudo ufw allow from 192.168.1.0/24 to any port 3306`

- อนุญาตให้ เฉพาะเครื่องในเครือข่าย 192.168.1.0/24 เข้าถึง MySQL (port 3306)
- เครื่องอื่น ๆ นอก subnet จะถูกปฏิเสธ

# Log all denied connections

`sudo ufw logging on`

- เปิดการบันทึก log ของการเชื่อมต่อที่ถูก deny
- Logs จะถูกเขียนไว้ที่ /var/log/ufw.log หรือบางระบบรวมอยู่ใน /var/log/syslog
- ใช้สำหรับ ตรวจสอบความพยายามเจาะระบบ

# Show numbered rules

`sudo ufw status numbered`

- แสดงกฎ UFW ทั้งหมดพร้อมลำดับเลขกำกับ

```
[deeno@arch ~]$ sudo ufw limit 2222/tcp
Rule updated
Rule updated (v6)
[deeno@arch ~]$ sudo ufw allow from 192.168.1.0/24 to any port 3386
Rule added
[deeno@arch ~]$ sudo ufw logging on
Logging enabled
[deeno@arch ~]$ sudo ufw status numbered
Status: active

To Action From
--
[ 1] 2222/tcp LIMIT IN Anywhere
[ 2] 88/tcp ALLOW IN Anywhere
[ 3] 443/tcp ALLOW IN Anywhere
[ 4] 2222 ALLOW IN 192.168.1.0/24
[ 5] 3386 ALLOW IN 192.168.1.0/24
[ 6] 2222/tcp (v6) LIMIT IN Anywhere (v6)
[ 7] 88/tcp (v6) ALLOW IN Anywhere (v6)
[ 8] 443/tcp (v6) ALLOW IN Anywhere (v6)
[deeno@arch ~]$
```

## 5.1 Install Monitoring Tools:

# Install required packages

`sudo apt update`

`sudo apt install fail2ban logwatch sysstat htop iotop`

- fail2ban → ตรวจสอบ log เช่น /var/log/auth.log และบล็อก IP อัตโนมัติถ้าเจอการ brute-force login
- logwatch → สรุปรายงาน log ระบบ (daily summary) แล้วส่งไปทาง email หรือ ไฟล์รายงาน
- sysstat → มีเครื่องมืออย่าง iostat, mpstat, sar สำหรับเก็บสถิติ CPU, I/O, memory → ใช้ monitor performance แบบระยะยาว
- htop → real-time process monitoring (คล้าย top แต่ใช้ง่ายกว่า)
- iotop → ใช้ดู process ไหนที่ใช้ disk I/O มาก

# Install ELK stack components (optional)

`sudo apt install elasticsearch logstash kibana`

- ELK Stack = Elasticsearch + Logstash + Kibana
- Elasticsearch → เก็บและจัดทำดัชนี log (database สำหรับ log)
- Logstash → ตัวรวบรวม log จากหลาย ๆ แหล่ง มาประมวลผล/parse ก่อนเก็บเข้า Elasticsearch
- Kibana → Dashboard สำหรับดู log แบบ real-time, มี graph & visualization



```
[deeno@arch ~]$ sudo pacman -Syu
:: Synchronizing package databases...
core is up to date
extra                                     7.9 MiB 3.01 MiB/s 00:03 [-----] 100%
:: Starting full system upgrade...
there is nothing to do
[deeno@arch ~]$ sudo pacman -S fail2ban logwatch sysstat htop iotop
resolving dependencies...
Looking for conflicting packages...

Package (17)                New Version  Net Change  Download Size
extra/perl-clone            0.47-2       0.02 MiB   0.01 MiB
extra/perl-encode-locale   1.05-14      0.02 MiB   0.01 MiB
extra/perl-html-parser      3.83-2       0.33 MiB   0.15 MiB
extra/perl-html-tagset      3.24-3       0.02 MiB   0.01 MiB
extra/perl-http-date        6.06-4       0.01 MiB   0.01 MiB
extra/perl-http-message     7.00-2       0.16 MiB   0.07 MiB
extra/perl-io-html          1.004-7      0.02 MiB   0.01 MiB
extra/perl-lwp-mediatypes   6.04-7       0.06 MiB   0.02 MiB
extra/perl-uri              5.32-2       0.20 MiB   0.03 MiB
extra/python-pyinotify      0.9.6-15    0.29 MiB   0.05 MiB
extra/python-systemd        235-4        0.35 MiB   0.08 MiB
extra/whois                 5.6.4-1      0.24 MiB   0.06 MiB
extra/fail2ban              1.1.0-7      4.93 MiB   0.06 MiB
extra/htop                  3.4.1-1      0.38 MiB   0.16 MiB
extra/iotop                 0.6-12       0.24 MiB   0.06 MiB
extra/logwatch              7.11-1       2.13 MiB   0.38 MiB
extra/sysstat               12.7.7-1     1.61 MiB   0.44 MiB

Total Download Size: 2.47 MiB
Total Installed Size: 11.00 MiB

:: Proceed with installation? [Y/n] y
:: Retrieving packages...
perl-html-parser-3.83-2-x86_64 158.1 KiB 472 KiB/s 00:00 [-----] 100%
htop-3.4.1-1-x86_64            168.0 KiB 434 KiB/s 00:00 [-----] 100%
perl-uri-5.32-2-any            94.4 KiB 474 KiB/s 00:00 [-----] 100%
python-systemd-235-4-x86_64    76.9 KiB 427 KiB/s 00:00 [-----] 100%
sysstat-12.7.7-1-x86_64       451.8 KiB 729 KiB/s 00:01 [-----] 100%
logwatch-7.11-1-any           392.0 KiB 537 KiB/s 00:01 [-----] 100%
perl-http-message-7.00-2-any   78.3 KiB 611 KiB/s 00:00 [-----] 100%
whois-5.6.4-1-x86_64          66.1 KiB 574 KiB/s 00:00 [-----] 100%
iotop-0.6-12-any              58.4 KiB 483 KiB/s 00:00 [-----] 100%
fail2ban-1.1.0-7-any          883.5 KiB 1895 KiB/s 00:01 [-----] 100%
python-pyinotify-0.9.6-15-any  54.5 KiB 556 KiB/s 00:00 [-----] 100%
perl-io-html-1.004-7-any       14.4 KiB 277 KiB/s 00:00 [-----] 100%
perl-lwp-mediatypes-6.04-7-any 19.3 KiB 363 KiB/s 00:00 [-----] 100%
perl-encode-locale-1.05-14-any 18.0 KiB 149 KiB/s 00:00 [-----] 100%
```

```
perl-clone-0.47-2-x86_64       18.4 KiB 386 KiB/s 00:00 [-----] 100%
perl-http-date-6.06-4-any      9.6 KiB 150 KiB/s 00:00 [-----] 100%
Total (17/17)                 2.5 MiB 2.29 MiB/s 00:01 [-----] 100%
(17/17) checking keys in keyring [-----] 100%
(17/17) checking package integrity [-----] 100%
(17/17) loading package files [-----] 100%
(17/17) checking for file conflicts [-----] 100%
(17/17) checking available disk space [-----] 100%
:: Processing package changes...
( 1/17) installing python-pyinotify [-----] 100%
( 2/17) installing python-systemd [-----] 100%
( 3/17) installing whois [-----] 100%
( 4/17) installing fail2ban [-----] 100%
Optional dependencies for fail2ban
firewalld: for a firewall backend
ipset: for a firewall backend
iptables: for a firewall backend [installed]
nftables: for a firewall backend
( 5/17) installing perl-html-tagset [-----] 100%
( 6/17) installing perl-clone [-----] 100%
( 7/17) installing perl-encode-locale [-----] 100%
( 8/17) installing perl-http-date [-----] 100%
( 9/17) installing perl-io-html [-----] 100%
(10/17) installing perl-lwp-mediatypes [-----] 100%
(11/17) installing perl-uri [-----] 100%
(12/17) installing perl-http-message [-----] 100%
(13/17) installing perl-html-parser [-----] 100%
(14/17) installing logwatch [-----] 100%
Optional dependencies for logwatch
cron
perl-date-manip: human readable dates
(15/17) installing sysstat [-----] 100%
Optional dependencies for sysstat
tk: to use lsag
gnuplot: to use lsag
(16/17) installing htop [-----] 100%
Optional dependencies for htop
lm_sensors: show cpu temperatures [installed]
lsof: show files opened by a process [installed]
strace: attach to a running process
(17/17) installing iotop [-----] 100%
:: Running post-transaction hooks...
(1/6) Reloading system manager configuration...
(2/6) Creating temporary files...
(3/6) Arming ConditionNeedsUpdate...
(4/6) Checking for old perl modules...
(5/6) Updating icon theme caches...
(6/6) Updating the desktop file MIME type cache...
[deeno@arch ~]$
```

```
[deeno@arch ~]$ pacman -Q elasticsearch-bin kibana-bin logstash-bin
elasticsearch-bin 8.12.1
kibana-bin 8.12.1
logstash-bin 8.12.1
[deeno@arch ~]$
```

## 5.2 Configure Fail2Ban:

# Backup original config

sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.conf.backup

- เก็บไฟล์ jail.conf ตั้งเดิมไว้ เพื่อเวลา config มีปัญหาจะได้ย้อนกลับได้

#### # สร้าง local config

sudo nano /etc/fail2ban/jail.local

- การใช้ jail.local ดีกว่าแก้ jail.conf ตรง ๆ เพราะ update ของแพ็คเกจจะไม่ overwrite
- 
- Fail2Ban จะอ่าน jail.local กับค่าของ jail.conf

#### # เนื้อหาไฟล์:

##### [DEFAULT]

bantime = 3600 # บล็อก IP เป็นเวลา 1 ชั่วโมง  
findtime = 600 # จับช่วงเวลา 10 นาทีล่าสุด  
maxretry = 3 # ถ้าเจอผิดพลาดเกิน 3 ครั้งภายใน 10 นาที → ban  
backend = systemd # ใช้ systemd journal ในการอ่าน log

##### [sshd]

enabled = true  
port = 2222  
logpath = /var/log/auth.log  
maxretry = 3  
bantime = 3600

- ป้องกันการ brute-force SSH
- ใช้ port 2222 (ตามที่คุณเปลี่ยนใน sshd\_config)
- ถ้า login fail เกิน 3 ครั้งใน 10 นาที → ban IP 1 ชั่วโมง

##### [apache-auth]

enabled = true  
port = http,https  
logpath = /var/log/apache2/error.log

- ตรวจสอบการ login ผิดพลาดจาก Apache (เช่น basic auth, web login)
- ใช้ log จาก Apache error log

##### [apache-badbots]

enabled = true  
port = http,https  
logpath = /var/log/apache2/access.log  
bantime = 86400 # ban 24 ชั่วโมง  
maxretry = 1 # ถ้าเจอพฤติกรรมแบบ bad bot ครั้งเดียว → ban เถย



- ใช้ rules ที่ Fail2Ban มีมาเพื่อจับ **web crawler / bots** ที่ผิดปกติ
- เหมาะกับการป้องกัน DoS จาก bot หรือ crawler ที่ไม่เป็นมิตร

```

[deeno@arch ~]$ sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.conf.backup
[deeno@arch ~]$ sudo nano /etc/fail2ban/jail.local
[deeno@arch ~]$

GNU nano 2.8.6 /etc/fail2ban/jail.local
[DEFAULT]
bantime = 3600
findtime = 600
maxretry = 3
backend = systemd

[sshd]
enabled = true
port = 2222
logpath = /var/log/auth.log
maxretry = 3
bantime = 3600

[apache-auth]
enabled = true
port = http,https
logpath = /var/log/apache2/error.log

[apache-badbots]
enabled = true
port = http,https
logpath = /var/log/apache2/access.log
bantime = 86400
maxretry = 1

```

### 5.3 Configure System Monitoring:

#### # Enable sysstat

`sudo systemctl enable sysstat`

`sudo systemctl start sysstat`

- sysstat = ชุดเครื่องมือเก็บสถิติระบบ (CPU, disk I/O, memory ฯลฯ) เช่น iostat, mpstat, sar
- enable + start = ให้ service เก็บสถิติ background ตลอดเวลา

#### # Create monitoring script

สร้างไฟล์

`sudo nano /usr/local/bin/system_monitor.sh`

`#!/bin/bash`

# System monitoring script

`DATE=$(date)`

`echo "=== System Monitor Report - $DATE ===" >> /var/log/system_monitor.log`

- พิมพ์ header ใต้ log พร้อม timestamp

# CPU Usage

`echo "CPU Usage:" >> /var/log/system_monitor.log`

`top -bn1 | grep "Cpu(s)" >> /var/log/system_monitor.log`

- ใช้ top (batch mode -bn1) ดึงค่า CPU usage ปัจจุบัน

- เก็บเฉพาะบรรทัด "Cpu(s)"

#### # Memory Usage

```
echo "Memory Usage:" >> /var/log/system_monitor.log
```

```
free -h >> /var/log/system_monitor.log
```

- รายงาน RAM ที่ใช้/เหลือ แบบ human readable (-h)

#### # Disk Usage

```
echo "Disk Usage:" >> /var/log/system_monitor.log
```

```
df -h >> /var/log/system_monitor.log
```

- เช็คพื้นที่ดิสก์ทั้งหมด (per partition)

#### # Active Users

```
echo "Active Users:" >> /var/log/system_monitor.log
```

```
who >> /var/log/system_monitor.log
```

- แสดง user ที่ login อยู่ในระบบ

#### # Failed Login Attempts

```
echo "Recent Failed Logins:" >> /var/log/system_monitor.log
```

```
tail -10 /var/log/auth.log | grep "Failed password" >> /var/log/system_monitor.log
```

- ดึง 10 บรรทัดล่าสุดจาก /var/log/auth.log ที่มีคำว่า "Failed password"
- ใช้ตรวจสอบการ login SSH ที่ล้มเหลว

```
echo "=====" >> /var/log/system_monitor.log
```

- แยก sectionให้อ่านง่าย

#### # Make executable

```
sudo chmod +x /usr/local/bin/system_monitor.sh
```

- ให้สิทธิ์ run script

#### # Add to crontab (run every hour)

Run with Crontab

```
sudo crontab -e
```

#### # เพิ่มบรรทัด:

```
0 * * * * /usr/local/bin/system_monitor.sh
```

- รันสคริปต์ทุกต้นชั่วโมง (Hourly monitoring)
- เก็บ log ลง /var/log/system\_monitor.log

```
[deeno@arch ~]$ sudo systemctl enable sysstat
Created symlink /etc/systemd/system/multi-user.target.wants/sysstat.service' → '/usr/lib/systemd/system/sysstat.service'.
Created symlink /etc/systemd/system/sysstat.service.wants/sysstat-collect.timer' → '/usr/lib/systemd/system/sysstat-collect.timer'.
Created symlink /etc/systemd/system/sysstat.service.wants/sysstat-summary.timer' → '/usr/lib/systemd/system/sysstat-summary.timer'.
Created symlink /etc/systemd/system/sysstat.service.wants/sysstat-rotate.timer' → '/usr/lib/systemd/system/sysstat-rotate.timer'.
[deeno@arch ~]$ sudo systemctl start sysstat
[deeno@arch ~]$ sudo nano /usr/local/bin/system_monitor.sh
[deeno@arch ~]$
[deeno@arch ~]$ sudo chmod +x /usr/local/bin/system_monitor.sh
[deeno@arch ~]$ sudo crontab -e
```

```
GNU nano 8.6 /usr/local/bin/system_monitor.sh
#!/bin/bash
# System monitoring script
DATE=$(date)
echo "==== System Monitor Report - $DATE ===" >> /var/log/system_monitor.log

# CPU Usage
echo "CPU Usage:" >> /var/log/system_monitor.log
top -bn1 | grep "Cpu(s)" >> /var/log/system_monitor.log

# Memory Usage
echo "Memory Usage:" >> /var/log/system_monitor.log
free -h >> /var/log/system_monitor.log

# Disk Usage
echo "Disk Usage:" >> /var/log/system_monitor.log
df -h >> /var/log/system_monitor.log

# Active Users
echo "Active Users:" >> /var/log/system_monitor.log
who >> /var/log/system_monitor.log

# Failed Login Attempts
echo "Recent Failed Logins:" >> /var/log/system_monitor.log
tail -10 /var/log/auth.log | grep "Failed password" >> /var/log/system_monitor.log

echo "===== " >> /var/log/system_monitor.log
```

## 5.4 Configure Log Rotation:

# Create logrotate config

sudo nano /etc/logrotate.d/system\_monitor

/var/log/system\_monitor.log {

- ระบุดว่าใช้กฎนี้กับไฟล์ log นี้เท่านั้น

daily

- หมุน log ทุกวัน (rotate ทุกวัน)

missingok

- ถ้าไฟล์ log หายไป ไม่ต้อง error ให้ข้ามไปเลย ๆ

rotate 30

- เก็บ log ย้อนหลังสูงสุด 30 ไฟล์ เก่ากว่านี้จะถูกลบอัตโนมัติ

compress

- บีบอัดไฟล์ log ที่ถูก rotate (gzip) เพื่อลดพื้นที่

delaycompress

- ไม่บีบไฟล์ที่เพิ่ง rotate ล่าสุด (ไฟล์ .1) กันที จะเริ่มบีบตอนรอบถัดไป → ป้องกันปัญหาถ้า log ยังถูกอ่านอยู่

notifempty

- ถ้า log ว่างเปล่า ไม่ต้อง rotate

copytruncate

- ทำสำเนาไฟล์แล้วตัดเนื้อหาออกจากต้นฉบับ → ใช้กับ service ที่ไม่รองรับ log reopen (เช่น script ของคุณ)
- ทำให้ log ยังถูกเขียนต่อได้ โดยไม่ต้อง restart service

}

The first screenshot shows a terminal window with the command `sudo nano /etc/logrotate.d/system_monitor` being executed. The second screenshot shows the contents of the file `/etc/logrotate.d/system_monitor` in nano editor. The configuration is as follows:

```

GNU nano 8.6 /etc/logrotate.d/system_monitor
/var/log/system_monitor.log {
    daily
    missingok
    rotate 30
    compress
    delaycompress
    notifempty
    copytruncate
}
  
```