



รายงาน Mini Project เรื่อง ระบบ SSH (Secure Shell)

จัดทำโดย

นายธีรเทพ ชัยวิฑูรณกุล ปี 2 หมู่ 1 รหัส 670112418011

เสนอ

อาจารย์ผู้สอน ผู้ช่วยศาสตราจารย์ดร.สวิน วงศ์ประเมษฐ์

โครงการนี้เป็นส่วนหนึ่งของรายวิชา เทคโนโลยีอินเทอร์เน็ต (4132203)

ภาคเรียนที่ 2 ปีการศึกษา 2568

สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาลัยราชภัฏบุรีรัมย์

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของการจัดทำ Mini Project โดยมีวัตถุประสงค์เพื่อศึกษาและทำความเข้าใจระบบ SSH (Secure Shell) ซึ่งเป็นโปรโตคอลสำหรับการเชื่อมต่อและควบคุมเครื่องคอมพิวเตอร์จากระยะไกลอย่างปลอดภัย โดยมีการเข้ารหัสข้อมูลที่ส่งผ่านเครือข่าย เพื่อป้องกันการดักฟังและการโจมตีจากบุคคลที่สาม

เนื้อหาในรายงานครอบคลุมความหมาย หลักการทำงาน โครงสร้าง พอร์ตที่ใช้งาน รูปแบบคำสั่ง การยืนยันตัวตน และการใช้งาน SSH บน Docker รวมถึงตัวอย่างไฟล์ docker-compose พร้อมคำอธิบายและภาพประกอบ

ผู้จัดทำหวังว่ารายงานฉบับนี้จะเป็นประโยชน์ต่อผู้อ่านในการเรียนรู้ การติดตั้ง และการประยุกต์ใช้ SSH ในระบบเครือข่ายหรือเครื่องเซิร์ฟเวอร์จริง

ลงชื่อผู้จัดทำ

นายธีรเทพ ชัยวิฑูรณกุล

เนื้อหา

ระบบ SSH คืออะไร

SSH ย่อมาจาก **Secure Shell** เป็นโปรโตคอลที่ใช้ในการสื่อสารระหว่างเครื่องลูกข่าย (Client) และเครื่องเซิร์ฟเวอร์ (Server) ผ่านเครือข่าย โดยมุ่งเน้นเรื่อง “ความปลอดภัย” ของข้อมูลที่รับส่ง ผู้ใช้สามารถเชื่อมต่อ ควบคุม และจัดการไฟล์บนเครื่องปลายทางได้เหมือนอยู่หน้าเครื่องจริง

ประวัติความเป็นมา

- **ปี 1995:** Tatu Ylönen นักวิจัยชาวฟินแลนด์ พัฒนา SSH รุ่นแรก (SSH-1) เพื่อทดแทน Telnet ที่ไม่มีการเข้ารหัส
- **ปี 1997:** ออกเวอร์ชัน SSH-2 ที่เพิ่มความปลอดภัยและรองรับการยืนยันตัวตนหลายแบบ
- **ปัจจุบัน:** SSH กลายเป็นมาตรฐานสำคัญในการเข้าถึงเซิร์ฟเวอร์ Linux/Unix และอุปกรณ์ Network ต่าง ๆ ทั่วโลก

หลักการทำงานของ SSH

1. ผู้ใช้พิมพ์คำสั่ง SSH เพื่อเชื่อมต่อกับเครื่องปลายทาง
2. เซิร์ฟเวอร์ส่ง Public Key ให้ลูกข่ายเพื่อยืนยันตัวตน
3. ผู้ใช้ยืนยันตัวตนด้วย Password หรือ Private Key
4. เมื่อการยืนยันสำเร็จ จะมีการเข้ารหัสการสื่อสารทั้งหมดด้วย AES, RSA หรือ อื่น ๆ
5. สามารถส่งคำสั่ง คัดลอกไฟล์ หรือเปิด Tunnel เพื่อเข้าถึงบริการอื่น ๆ ได้อย่างปลอดภัย

พอร์ตที่ใช้งาน

โปรโตคอล	พอร์ต	รายละเอียด
SSH	22	พอร์ตหลักในการเชื่อมต่อแบบเข้ารหัส
SFTP	22	ใช้โอนถ่ายไฟล์ผ่าน SSH
SCP	22	คัดลอกไฟล์ระหว่างเครื่องแบบปลอดภัย

รูปแบบการใช้งาน SSH

1. การเชื่อมต่อเครื่องปลายทาง `ssh username@hostname`

ตัวอย่าง: `ssh root@192.168.1.10`

2. การคัดลอกไฟล์ผ่าน SCP

`scp file.txt user@192.168.1.10:/home/user/`

3. การสร้าง SSH Key

`ssh-keygen -t rsa -b 4096`

4. การส่ง Public Key ไปยังเครื่องปลายทาง

`ssh-copy-id user@192.168.1.10`

5. การเปิด Port Forwarding

`ssh -L 8080:localhost:80 user@server`

การใช้งาน SSH ร่วมกับภาษาโปรแกรม

ตัวอย่าง Python (ใช้ Paramiko Library)

```
import paramiko
```

```
ssh = paramiko.SSHClient()
```

```
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
```

```
ssh.connect("192.168.1.10", username="root", password="1234")
```

```
stdin, stdout, stderr = ssh.exec_command("ls -la")
```

```
print(stdout.read().decode())
```

```
ssh.close()
```

**ตัวอย่างนี้ใช้การเชื่อมต่อผ่าน SSH เพื่อรันคำสั่งบนเซิร์ฟเวอร์จากระยะไกล*

ภาพประกอบ: ภาพการทำงานของระบบ SSH

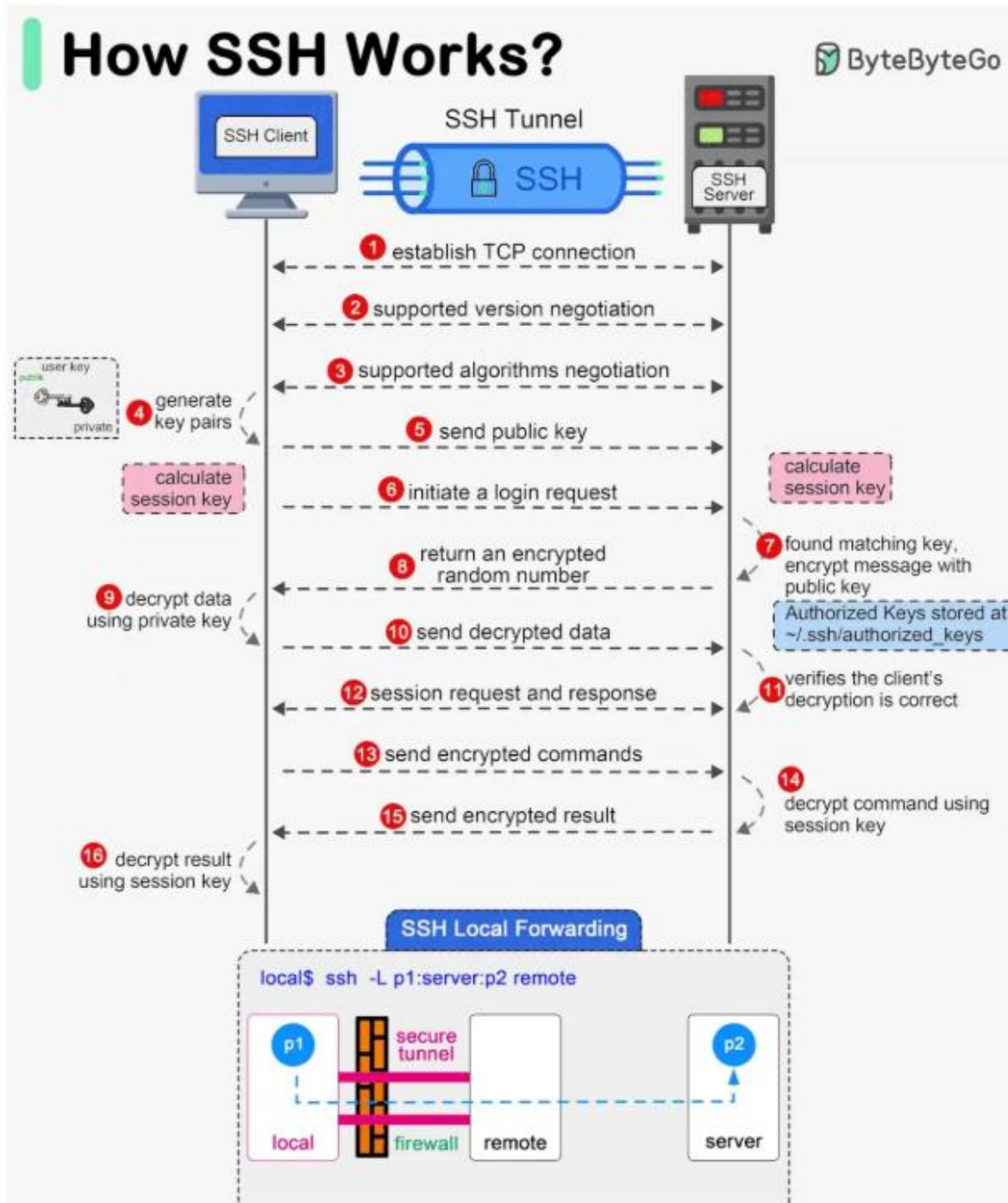


Image บน Docker Hub

ตัวอย่าง: Linux Server Image

Image	รายละเอียด
linuxserver/openssh-server	<ul style="list-style-type: none">- รองรับการเชื่อมต่อด้วย <i>Public Key</i>- รองรับการใช้งานแบบ <i>Password access</i>- รองรับ sudo- สามารถแม็บ Volume สำหรับเก็บคอนฟิก หรือแม็บโฟลเดอร์ที่ผู้ใช้ควรเข้าถึงได้

ตัวอย่างการ pull image

```
docker pull linuxserver/openssh-server
```

การ pull image ระบุเวอร์ชันเฉพาะ

```
docker pull linuxserver/openssh-server:latest
```

ตัวอย่างการใช้งาน (docker-compose):

services:

openssh-server:

image: linuxserver/openssh-server:latest

container_name: openssh-server

environment:

- PUID=1000
- PGID=1000
- TZ=Etc/UTC
- PUBLIC_KEY=yourpublickey # ถ้าใช้ public key
- PASSWORD_ACCESS=false # ตั้งให้ false ถ้าไม่อยากจะใช้ password
- USER_NAME=customuser # ถ้าต้องการ

volumes:

- /path/on/host/openssh-server/config:/config

ports:

- 2222:2222

restart: unless-stopped

คำอธิบาย

- เปิดพอร์ต 2222 ของโฮสต์ → 2222 ภายในคอนเทนเนอร์
- กำหนดตัวแปรสภาพแวดล้อม (Environment Variables)
- ใช้ Volume เพื่อเก็บไฟล์คอนฟิกและข้อมูลผู้ใช้แบบถาวร แม้คอนเทนเนอร์จะถูกลบหรือรีสตาร์ทก็ยังไม่หาย
- ตั้งค่า restart ให้คอนเทนเนอร์เปิดขึ้นอัตโนมัติเมื่อระบบรีบูต หรือถ้ามีการหยุดทำงานโดยไม่ได้สั่งปิด

อ้างอิง

- 1 <https://www.ssh.com/academy>
- 2 <https://www.openssh.com/>
- 3 <https://docs.docker.com/>
- 4 <https://datatracker.ietf.org/doc/html/rfc4251>
- 5 <https://blog.bytebytego.com/p/ep124-how-does-ssh-work>