

Link นำเสนอ :

<https://youtu.be/GzeERihE8-8?si=NihoROLMKm3teFmv>

(วิดีโอ Demo อยู่ท้ายคลิปนะครับ)

Link Source Code (config-it.zip) :

[https://drive.google.com/drive/folders/1UCSbDJG\\_FGo609YwUpSPTb5yoDpzGfVZ?usp=sharing](https://drive.google.com/drive/folders/1UCSbDJG_FGo609YwUpSPTb5yoDpzGfVZ?usp=sharing)

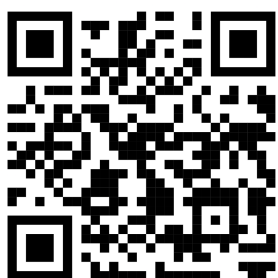
**\*\* ให้ run ไฟล์ main.py เพื่อใช้งาน \*\***

Link Github Repository :

<https://github.com/Pikachu051/config-it>

**\*\* ไฟล์ใน Github ยังขาด .env ที่เก็บ token ของ bot ซึ่งไม่สามารถอัปขึ้นไปได้ตามนโยบายของ Discord โดยทางคณะผู้จัดทำได้แนบไฟล์ .env ไว้ให้แล้วใน Link Source Code \*\***

QR สำหรับการเชิญ bot เข้า Discord Server :





## รายงาน

**หัวข้อ** การพัฒนา Discord BOT เพื่อการ Configuration

จัดทำโดย

นางสาวชุตติกาญจน์ ใจคง	65070053
นางสาวญาดาวิ ฤกษ์ประสูต	65070055
นายภูริพัทธ์ ชนะภัย	65070181
นายศุภชัย มณีรัตน์	65070224
นายสหชินเดช เกตุดี	65070232
นายสุทธิเกียรติ หักยาสมบุรณ์	65070240

เสนอ

รศ.ดร. โชติพัชร ภรณ์วลัย

ผศ.ดร. สุเมธ ประภาวัต

ผศ.ดร. ลภัส ประดิษฐ์ทัศนีย์

รายงานวิชา COMMUNICATION NETWORK INFRASTRUCTURE

สาขาวิชาเทคโนโลยีสารสนเทศ ภาคเรียนที่ 2 ปี การศึกษา 2566

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

## คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา 06016419 COMMUNICATION NETWORK INFRASTRUCTURE โดยมีวัตถุประสงค์เพื่อพัฒนา BOT Discord ให้สามารถใช้การ Configuration และเพื่อให้คนที่ไม่มีความรู้สามารถใช้งานได้ ในการจัดทำผู้จัดทำได้นำองค์ความรู้ที่ได้เรียน มาประยุกต์ใช้ในการพัฒนาครั้งนี้ด้วย หากมีข้อผิดพลาดประการใด คณะผู้จัดทำต้องขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

## สารบัญ

เรื่อง	หน้า
ที่มาและความสำคัญ	5
วัตถุประสงค์ของรายงาน	5
การสร้างบอท Discord	6
เครื่องมือและเทคนิคที่ใช้	7
Command ที่ใช้ในการพัฒนา	8
ผลการทดลองใช้งาน	17
สรุปผลการทดลอง	40
ประโยชน์	40
บรรณานุกรม	41

## ที่มาและความสำคัญ

เนื่องจากปัจจุบันแอปพลิเคชัน Discord นั้นเป็นที่นิยมมากในหมู่ผู้คน ไม่ว่าจะเป็นใช้เป็นที่สื่อสาร เข้าร่วมกิจกรรมในวงเล็กและวงกว้าง การประชุม หรือการเล่นเกม สตรีมมิงเกม การแชร์ไฟล์ได้แบบสะดวกรวดเร็ว และที่สำคัญ Discord มีบอท (Bot) ซึ่งบอทสามารถช่วยในการฟังเพลง หรือเล่นเกมเป็นต้น ทางผู้จัดทำจึงเห็นว่าเราสามารถสร้างบอทให้เกิดประโยชน์ในด้านของเน็ตเวิร์ค ร่วมกับรายวิชา COMMUNICATION NETWORK INFRASTRUCTURE นำมาประยุกต์เป็นบอทที่สามารถโต้ตอบกับผู้ใช้ Config Router และ Switch ได้ โดยการใช้ Secure Shell (SSH) ไปยังเซิร์ฟเวอร์ ผู้ใช้จะทำการพิมพ์คำสั่งในช่องแชท และบอทก็จะตอบกลับผู้ใช้ ซึ่งวิธีนี้จะทำให้เกิดความรวดเร็วและสะดวกต่อผู้ใช้งานมากขึ้น

จากที่กล่าวมาข้างต้น สรุปแล้วทางผู้จัดทำได้สร้างบอทขึ้นมาในแอปพลิเคชัน Discord และใช้คำสั่งทำ Config ได้โดยใช้ภาษา Python และ Library Netmiko ทางผู้จัดทำหวังว่าโครงงานนี้จะเป็นประโยชน์กับทุกคนไม่มากก็น้อย

## วัตถุประสงค์ของรายงาน

1. เพื่อการศึกษาและพัฒนาการใช้คำสั่ง Config ทางเน็ตเวิร์ค
2. เพื่อการศึกษาขั้นตอนการสร้าง BOT Discord โดยใช้ภาษา Python
3. เพื่อการศึกษาการประยุกต์ใช้งานร่วมกับ Netmiko
4. เพื่อการศึกษาและพัฒนาการทำงานของ BOT Discord ให้มีประสิทธิภาพมากขึ้น

## การสร้าง BOT Discord

การสร้างบอท Discord นั้นสามารถทำได้โดยใช้ Discord API ร่วมกับภาษาโปรแกรมมิ่งที่คุณชื่นชอบ เช่น Python, JavaScript, หรือ Java ในที่นี้ผู้จัดทำจะสร้างบอท Discord โดยใช้ภาษา Python ด้วย Discord.py library เป็นตัวช่วย

ขั้นตอนการสร้างบอท Discord แบ่งออกเป็นขั้นตอนหลักๆ ดังนี้:

### 1. สร้างแอปพลิเคชัน Discord และรับ Token:

- เข้าสู่หน้าเว็บของ Discord Developer Portal:  
<https://discord.com/developers/applications>
- คลิกที่ "New Application" เพื่อสร้างแอปพลิเคชันใหม่
- ตั้งชื่อให้แอปพลิเคชันของคุณและกด "Create"
- เมื่อสร้างแอปพลิเคชันแล้ว ไปที่แท็บ "Bot" และกด "Add Bot"
- คุณจะได้อ Token ของบอทที่สร้างขึ้นมา จำ Token นี้ไว้เพื่อใช้ในการเชื่อมต่อบอทกับ Discord

### 2. สร้างโปรเจกต์ Python และติดตั้ง Discord.py:

- สร้างโปรเจกต์ใหม่ใน Python
- ใช้คำสั่ง `pip install discord.py` เพื่อติดตั้ง Discord.py library ในโปรเจกต์ของคุณ

### 3. เขียนโค้ดสร้างบอท Discord:

- สร้างไฟล์ Python และเขียนโค้ดสร้างบอทด้วย Discord.py
- เชื่อมต่อบอทกับ Discord โดยใช้ Token ที่ได้รับจากขั้นตอนที่ 1
- เพิ่มฟังก์ชันและคำสั่งที่ต้องการให้บอทดำเนินการ เช่น ส่งข้อความ, ตอบกลับข้อความจากผู้ใช้, หรือปฏิสัมพันธ์กับเซิร์ฟเวอร์ Discord อื่น ๆ

### 4. เชื่อมต่อบอทกับ Discord:

- เมื่อเขียนโค้ดเสร็จสิ้น บันทึกและรันโปรแกรม Python ของคุณ

- ใช้ลิงก์ที่สร้างขึ้นจาก Discord Developer Portal เพื่อเชื่อมต่อบอกของคุณกับเซิร์ฟเวอร์ Discord ที่คุณต้องการ

## เครื่องมือและเทคนิคที่ใช้

### Python Libraries

#### Netmiko

Netmiko เป็นไลบรารี Python ที่รองรับหลายผู้ผลิตและทำให้การกำหนดค่าและการจัดการอุปกรณ์เครือข่ายเป็นอัตโนมัติง่ายขึ้น ซึ่งมีgrngunให้การติดต่อที่สม่ำเสมอและง่ายต่อการใช้งาน ในการจัดการอุปกรณ์เครือข่ายข้ามผู้ผลิตหลายราย เช่น เราเตอร์, สวิตช์, และไฟวอลล์

Netmiko ใช้โปรโตคอล Secure Shell (SSH) เพื่อเชื่อมต่อกับอุปกรณ์เครือข่ายและให้ฟังก์ชันและเมทอดในการดำเนินการคำสั่งบนอุปกรณ์ รองรับผู้ผลิตอุปกรณ์เครือข่ายหลายราย เช่น Cisco, Juniper, Arista, และ Huawei

ด้วย Netmiko, ผู้ดูแลระบบและวิศวกรเครือข่ายสามารถอัตโนมัติงานที่พบบ่อยในเครือข่าย เช่น กำหนดค่าอุปกรณ์, อัปเดตเฟิร์มแวร์, ดำเนินการสำรองข้อมูลและคืนค่า และการตรวจสอบเครือข่าย Netmiko ช่วยให้ผู้ใช้สามารถทำงานกับอุปกรณ์โดยโปรแกรมได้อย่างอัตโนมัติ ประหยัดเวลาและลดความน่าจะเป็นของข้อผิดพลาดหรือการกำหนดค่าผิด

Netmiko ถูกสร้างขึ้นบน Paramiko SSH library และมีเซ็ทของเมทอดและคุณสมบัติที่ขยายขึ้นโดยเฉพาะสำหรับการอัตโนมัติเครือข่าย นอกจากนี้มันยังรองรับการใช้กุญแจ SSH ที่กำหนดเองซึ่งช่วยเพิ่มความปลอดภัยและความเป็นไปตามข้อบังคับขององค์กร

เนื่องจาก Netmiko เป็นหนึ่งในไลบรารีที่เป็นเจ้าของของ Python ทำให้เราไม่จำเป็นต้องติดตั้งมันโดยแยกออกมา

## Command ที่ใช้ในการพัฒนา

### Connection

- **!create\_connection <ip> <username> <password>** คำสั่งนี้ใช้การเพิ่มอุปกรณ์ของผู้ใช้ให้บอกได้รู้จัก และแสดงผลลัพธ์เป็นการเชื่อมต่อกับชื่อผู้ใช้นั้นๆ และจำนวนเครื่องที่เชื่อมต่อ
- **!show\_connection <index>** คำสั่งนี้ใช้ในการแสดงการเชื่อมต่อของผู้ใช้ที่บอกทำความรู้จัก

### Help

- **!command\_list** คำสั่งที่ใช้ในการดู command ทั้งหมดที่สามารถใช้งานได้ โดยแสดงรายละเอียดของคำสั่งและลักษณะการใช้งานของแต่ละ command

### Basic Configuration command list

- **!show\_run <index>** คำสั่งนี้ใช้สำหรับแสดงการตั้งค่าปัจจุบันของอุปกรณ์ โดยส่งคำสั่ง "show run" ไปยังอุปกรณ์ และแสดงผลลัพธ์ที่ได้กลับไปยังช่องสนทนาใน Discord
- **!show\_run\_int <index> <interface>** คำสั่งนี้ใช้สำหรับแสดงการตั้งค่าของอินเตอร์เฟซที่กำหนด โดยส่งคำสั่ง "show run int {interface}" ไปยังอุปกรณ์ และแสดงผลลัพธ์ที่ได้กลับไปยังช่องสนทนาใน Discord
- **!save\_config <index>** คำสั่งนี้ใช้สำหรับบันทึกการตั้งค่าปัจจุบันของอุปกรณ์ โดยส่งคำสั่ง "wr" (write memory) ไปยังอุปกรณ์ เพื่อบันทึกการเปลี่ยนแปลงทั้งหมดที่ได้ทำไว้
- **!hostname <index> <hostname>** คำสั่งนี้ใช้สำหรับตั้งค่า Hostname ของอุปกรณ์ โดยรับพารามิเตอร์เป็นชื่อ Hostname ที่ต้องการตั้งค่า และส่งคำสั่ง "hostname {hostname}" ไปยังอุปกรณ์



- **!vlan\_ip\_add <index> <vlan> <ip\_address> <netmask>** คำสั่งนี้ใช้สำหรับเพิ่มที่อยู่ IP สำหรับ VLAN บนอุปกรณ์ โดยรับพารามิเตอร์เป็น ID ของ VLAN, ที่อยู่ IP, และมาสก์เน็ตเวิร์ก และส่งคำสั่ง "int {vlan}, ip add {ip\_address} {netmask}" ไปยังอุปกรณ์ เพื่อเพิ่มที่อยู่ IP
- **!vlan\_no\_shut <index> <id>** คำสั่งนี้ใช้สำหรับเปิดการใช้งาน VLAN บนอุปกรณ์ โดยรับพารามิเตอร์เป็น ID ของ VLAN และส่งคำสั่ง "vlan {id}, no shut" ไปยังอุปกรณ์ เพื่อเปิดการใช้งาน VLAN
- **!int\_ip\_add <index> <interface> <ip\_address> <netmask>** คำสั่งนี้ใช้สำหรับเพิ่มที่อยู่ IP สำหรับอินเทอร์เฟซบนอุปกรณ์ โดยรับพารามิเตอร์เป็นชื่อของอินเทอร์เฟซ, ที่อยู่ IP, และมาสก์เน็ตเวิร์ก และส่งคำสั่ง "int {interface}", "no ip add", "ip add {ip\_address} {netmask}" ไปยังอุปกรณ์
- **!add\_gateway <index> <ip\_gateway>** คำสั่งนี้ใช้สำหรับกำหนด IP Default Gateway บนอุปกรณ์ โดยรับพารามิเตอร์เป็นที่อยู่ IP ของเกตเวย์เริ่มต้น และส่งคำสั่ง "ip default-gateway {ip\_gateway}" ไปยังอุปกรณ์ เพื่อกำหนด IP Default Gateway
- **!int\_no\_shut <index> <interface>** คำสั่งนี้ใช้สำหรับเปิดการใช้งานอินเทอร์เฟซบนอุปกรณ์ โดยรับพารามิเตอร์เป็นชื่อของอินเทอร์เฟซ และส่งคำสั่ง "int {interface}", "no shut" ไปยังอุปกรณ์ เพื่อเปิดการใช้งานอินเทอร์เฟซ
- **!int\_shut <index> <interface>** คำสั่งนี้ใช้สำหรับปิดการใช้งานอินเทอร์เฟซบนอุปกรณ์ โดยรับพารามิเตอร์เป็นชื่อของอินเทอร์เฟซ และส่งคำสั่ง "int {interface}", "shut" ไปยังอุปกรณ์ เพื่อปิดการใช้งานอินเทอร์เฟซ
- **!banner <index> <string>** คำสั่งนี้ใช้สำหรับตั้งค่า Banner Message of the Day (MOTD) บนอุปกรณ์ โดยรับพารามิเตอร์เป็นข้อความที่ต้องการให้แสดง และส่งคำสั่ง "banner motd # {ข้อความ} #" ไปยังอุปกรณ์ เพื่อตั้งค่าข้อความ MOTD

- **!vlan\_ip\_delete <index> <vlan>** คำสั่งที่ใช้ในการลบการกำหนดค่า IP address ออกจาก VLAN บนอุปกรณ์เครือข่าย โดยส่งคำสั่ง "int {vlan}", "no ip add" ไปยังอุปกรณ์
- **!vlan\_shut <index> <vlan>** คำสั่งที่ใช้สำหรับปิดการทำงานของ VLAN บนอุปกรณ์เครือข่าย โดยมีการรับพารามิเตอร์เป็น vlan และส่งคำสั่ง "int vlan {vlan}", "shut" ไปยังอุปกรณ์
- **!delete\_gateway <index> <ip\_gateway>** คำสั่งที่ใช้ในการลบค่า Default Gateway จากอุปกรณ์เครือข่าย โดยมีการรับพารามิเตอร์เป็น ip\_gateway และส่งคำสั่ง "no ip default-gateway {ip\_gateway}" ไปยังอุปกรณ์
- **!int\_ip\_delete <index> <interface>** คำสั่งที่ใช้ในการลบค่า IP address ของ Interface ของอุปกรณ์บนเครือข่าย โดยมีการรับพารามิเตอร์เป็น interface และส่งคำสั่ง "int {interface}", "no ip address" ไปยังอุปกรณ์
- **!traceroute <index> <ip\_address> <source\_ip=None>** คำสั่งที่ใช้ทดสอบการสื่อสารจากต้นทางไปยังปลายทางคล้ายกับการ Ping แต่จะเก็บรายละเอียดระหว่างการส่งข้อมูลไปในแต่ละ Hop ตลอดเส้นทางเพิ่มมา ทำให้เห็นเส้นทางจากต้นทางไปยังปลายทางทั้งหมด และเก็บค่า Round Trip Time (RTT) ของแต่ละ Hop

### Troubleshooting Cisco Commands List

- **!ping <index> <ip\_address>** คำสั่งนี้ใช้สำหรับทดสอบการเชื่อมต่อโดยส่งแพคเกจ PING ไปยังอุปกรณ์ที่กำหนด โดยรับพารามิเตอร์เป็น IP Address และส่งคำสั่ง "ping {ip\_address}" ไปยังอุปกรณ์ เมื่อเสร็จสิ้นจะแสดงผลลัพธ์ในรูปแบบ Packets sent, Packets received, Packet loss

- **!show\_vlan <index>** คำสั่งนี้ใช้สำหรับแสดงรายการของ VLAN บนอุปกรณ์ โดยส่งคำสั่ง "show vlan brief" ไปยังอุปกรณ์ และแสดงผลลัพธ์ที่ได้กลับไปยังช่องสนทนาใน Discord
- **!show\_int <index>** คำสั่งนี้ใช้สำหรับแสดงรายการของ interface บนอุปกรณ์ โดยส่งคำสั่ง "show ip int brief" ไปยังอุปกรณ์ และแสดงผลลัพธ์ที่ได้กลับไปยังช่องสนทนาใน Discord
- **!show\_hostname <index>** คำสั่งนี้ใช้สำหรับแสดง Hostname ของอุปกรณ์ โดยส่งคำสั่ง "show run | include hostname" ไปยังอุปกรณ์ และแสดงผลลัพธ์ที่ได้กลับไปยังช่องสนทนาใน Discord
- **!show\_mac\_table** คำสั่งที่ใช้สำหรับแสดงตาราง MAC address จากอุปกรณ์เครือข่าย โดยส่งคำสั่ง "show mac address-table" ไปยังอุปกรณ์ และแสดงผลลัพธ์ที่ได้กลับไปยังช่องสนทนาใน Discord

## Routing and VLAN Commands

- **!create\_route <index> <dest\_ip> <dest\_mask> <next\_hop>** คำสั่งนี้ใช้สำหรับเพิ่มเส้นทาง IP บนอุปกรณ์ โดยรับพารามิเตอร์เป็น destination IP address, subnet mask, และ next hop address และส่งคำสั่ง "ip route {des\_ip} {des\_mask} {next\_hop}" ไปยังอุปกรณ์ เพื่อเพิ่มเส้นทาง IP
- **!show\_route <index>** คำสั่งนี้ใช้สำหรับแสดงตารางเส้นทาง IP (IP routing table) บนอุปกรณ์ โดยส่งคำสั่ง "show ip route" ไปยังอุปกรณ์ และแสดงผลลัพธ์กลับไปยังช่องสนทนาใน Discord
- **!delete\_route <index> <dest\_ip> <dest\_mask> <next\_hop>** คำสั่งที่ใช้ในการลบเส้นทางการทำ routing ออกจาก routing table โดยรับพารามิเตอร์ "dest\_ip" network address ปลายทาง "dest\_mask" subnet mask ของปลายทาง "next\_hop" IP address ของ next hop จากนั้นใช้คำสั่ง "no ip route {dest\_ip} {des\_mask} {next\_hop}"

- **!rip <index> <networks>** คำสั่งนี้ใช้สำหรับกำหนดค่า RIP (Routing Information Protocol) บนอุปกรณ์ โดยรับพารามิเตอร์เป็น network และส่งคำสั่ง "router rip", "version 2", "no auto-summary", "network {networks}" ไปยังอุปกรณ์  
\*networks คือ IP address, IP address,...
- **!remove\_rip\_nw <index> <networks>** คำสั่งนี้ใช้ในการลบการกำหนดค่าเครือข่าย RIP (Routing Information Protocol) บนอุปกรณ์ โดยรับพารามิเตอร์เป็น network และส่งคำสั่ง "router rip", "no network {networks}" ไปยังอุปกรณ์  
\*networks คือ IP address, IP address,...
- **!disable\_rip <index>** คำสั่งนี้ใช้ในการปิดการใช้งาน RIP (Routing Information Protocol) บนอุปกรณ์ โดยส่งคำสั่ง "no router rip" ไปยังอุปกรณ์
- **!show\_rip <index>** คำสั่งนี้ใช้ในการแสดงข้อมูลเกี่ยวกับการใช้งานโปรโตคอล RIP (Routing Information Protocol) บนอุปกรณ์ โดยส่งคำสั่ง "show ip rip database" ไปยังอุปกรณ์ และแสดงผลลัพธ์กลับไปยังช่องสนทนาใน Discord
- **!eigrp <index> <networks> <asn>** คำสั่งนี้ใช้ในการกำหนดค่า EIGRP (Enhanced Interior Gateway Routing Protocol) บนอุปกรณ์ โดยรับพารามิเตอร์เป็น networks, AS number และส่งคำสั่ง "router eigrp {asn}", "network {network[0]} {network[1]}" ไปยังอุปกรณ์  
\*networks คือ IP address/subnet mask, IP address/subnet mask, ...  
\*network[0] คือ IP address  
\*network[1] คือ wildcard mask ที่ถูกแปลงมาจาก subnet mask
- **!remove\_eigrp\_nw <index> <networks> <asn>** คำสั่งนี้ใช้ในการลบการกำหนดค่าเครือข่าย EIGRP (Enhanced Interior Gateway Routing

Protocol) บนอุปกรณ์ โดยรับพารามิเตอร์เป็น networks, AS number และ  
ส่งคำสั่ง "router eigrp {asn}", "no network {network[0]}"

{network[1]} " ไปยังอุปกรณ์

\*networks คือ IP address/subnet mask, IP address/subnet mask,...

\*network[0] คือ IP address

\*network[1] คือ wildcard mask ที่ถูกแปลงมาจาก subnet mask

- **!disable\_eigrp <index> <asn>** คำสั่งนี้ใช้ในการปิดการใช้งาน EIGRP (Enhanced Interior Gateway Routing Protocol) บนอุปกรณ์ โดยรับพารามิเตอร์เป็น AS number และส่งคำสั่ง "no router eigrp {asn}" ไปยังอุปกรณ์
- **!show\_eigrp <index>** คำสั่งนี้ใช้ในการแสดงข้อมูลเกี่ยวกับการใช้งานโปรโตคอล EIGRP (Enhanced Interior Gateway Routing Protocol) บนอุปกรณ์ โดยส่งคำสั่ง "show ip eigrp topology", "show ip eigrp neighbors" ไปยังอุปกรณ์ และแสดงผลลัพธ์กลับไปยังช่องสนทนาใน Discord
- **!ospf <index> <networks>** คำสั่งนี้ใช้ในการกำหนดค่า OSPF (Open Shortest Path First) บนอุปกรณ์ โดยรับพารามิเตอร์เป็น networks และส่งคำสั่ง "router ospf 1", "network {network[0]} {network[1]} area {network[2]}" ไปยังอุปกรณ์
  - \*networks คือ IP address/subnet mask/area ID, IP address/subnet mask/area ID,...
  - \*network[0] คือ IP address
  - \*network[1] คือ wildcard mask ที่ถูกแปลงมาจาก subnet mask
  - \*network[2] คือ area ID
- **!remove\_ospf\_nw <index> <networks>** คำสั่งนี้ใช้ในการลบเครือข่ายที่กำหนดไว้ในการกำหนดค่า OSPF (Open Shortest Path First) บนอุปกรณ์

โดยรับพารามิเตอร์เป็น networks และส่งคำสั่ง "no network {network[0]} {network[1]} area {network[2]}" ไปยังอุปกรณ์

\*network[0] คือ IP address

\*network[1] คือ wildcard mask ที่ถูกแปลงมาจาก subnet mask

\*network[2] คือ area ID

- **!disable\_ospf <index>** คำสั่งนี้ใช้ในการปิดการใช้งาน OSPF (Open Shortest Path First) บนอุปกรณ์ โดยส่งคำสั่ง "no router ospf 1" ไปยังอุปกรณ์และแสดงผลพร้อมกลับไปยังช่องสนทนาใน Discord
- **!show\_ospf <index>** คำสั่งนี้ใช้สำหรับแสดงข้อมูลเกี่ยวกับโปรโตคอล OSPF (Open Shortest Path First) โดยส่งคำสั่ง "show ip ospf database" ไปยังอุปกรณ์ และแสดงผลพร้อมกลับไปยังช่องสนทนาใน Discord
- **!bgp <index> <networks> <neighbors> <asn>** คำสั่งนี้ใช้สำหรับกำหนดค่า BGP (Border Gateway Protocol) บนอุปกรณ์ มีการกำหนด neighbors ซึ่งเป็นขั้นตอนที่สำคัญในการกำหนดการเชื่อมต่อระหว่างเรากับเครือข่ายอื่นๆ เพื่อแลกเปลี่ยนข้อมูล BGP กัน โดยรับพารามิเตอร์เป็น networks, neighbors, AS number และส่งคำสั่ง "router bgp {asn}", "network {network[0]} mask {network[1]}", "neighbor {neighbor[0]} remote-as {neighbor[1]}" ไปยังอุปกรณ์
  - \*networks คือ IP address/subnet mask, IP address/subnet mask,...
  - \*neighbors คือ IP address:asn, IP address:asn
- **!remove\_bgp\_nw <index> <networks> <asn>** คำสั่งนี้ใช้ในการลบเครือข่ายที่กำหนดไว้จาก configuration ของ BGP บนอุปกรณ์ โดยรับพารามิเตอร์เป็น networks, AS number และส่งคำสั่ง "router bgp {asn}", "no network {network[0]} mask {network[1]}" ไปยังอุปกรณ์
  - \*networks คือ IP address/subnet mask, IP address/subnet mask,...

- **!remove\_bgp\_neighbor <index> <neighbors> <asn>** คำสั่งนี้ใช้ลบเพื่อนบ้าน (neighbor) ออกจาก configuration ของ BGP บนอุปกรณ์ โดยรับพารามิเตอร์เป็น neighbors, AS number และส่งคำสั่ง "router bgp {asn}", "no {neighbor[0]} remote-as {neighbor[1]}"  
\*neighbors คือ IP address:asn,IP address:asn
- **!disable\_bgp <index> <asn>** คำสั่งนี้ใช้สำหรับปิดการใช้งาน BGP (Border Gateway Protocol) บนอุปกรณ์ สำหรับ Autonomous System ที่ระบุ โดยจะรับพารามิเตอร์เป็น AS Number และส่งคำสั่ง "no router bgp {asn}" ไปยังอุปกรณ์
- **!show\_bgp <index>** คำสั่งนี้ใช้สำหรับแสดงข้อมูลเกี่ยวกับโปรโตคอล BGP (Border Gateway Protocol) และส่งคำสั่ง "show ip bgp" ไปยังอุปกรณ์ และแสดงผลลัพท์กลับไปยังช่องสนทนาใน Discord
- **!show\_spanning\_tree <index>** คำสั่งนี้ใช้สำหรับแสดงข้อมูลการทำงานของ Spanning Tree Protocol (STP) บนอุปกรณ์ โดยส่งคำสั่ง "show spanning-tree" ไปยังอุปกรณ์ และแสดงผลลัพท์กลับไปยังช่องสนทนาใน Discord
- **!create\_vlan <index> <id>** คำสั่งนี้ใช้สำหรับสร้าง VLAN บนอุปกรณ์ โดยรับพารามิเตอร์เป็นหมายเลข ID ของ VLAN และส่งคำสั่ง "vlan {id}" ไปยังอุปกรณ์ เพื่อสร้าง VLAN
- **!delete\_vlan <index> <id>** คำสั่งที่ใช้สำหรับการลบ VLAN ออกจากอุปกรณ์บนเครือข่าย โดยรับพารามิเตอร์เป็นหมายเลข ID ของ VLAN และส่งคำสั่ง "no vlan {id}" ไปยังอุปกรณ์
- **!int\_switch\_mode <index> <interface> <mode>** คำสั่งนี้ใช้สำหรับการเปลี่ยนโหมดของอินเตอร์เฟซ เช่น access หรือ trunk โดยรับพารามิเตอร์เป็นชื่อของอินเตอร์เฟซ(interface) และโหมดที่ต้องการเปลี่ยน(mode) และส่งคำสั่ง "int {interface}", "switchport mode {mode}" ไปยังอุปกรณ์

- **!int\_access\_vlan <index> <interface> <vlan\_id>** คำสั่งนี้ใช้สำหรับกำหนดการใช้งานพอร์ตในโหมด Switchport Access VLAN บนอุปกรณ์เครือข่าย โดยกำหนดให้พอร์ต(interface) ที่ระบุใช้งานใน VLAN ที่ระบุ ซึ่งมีการรับพารามิเตอร์เป็น interface, vlan\_id และส่งคำสั่ง "int {interface} switchport access vlan {vlan\_id}" ไปยังอุปกรณ์
- **!router\_on\_a\_stick <index> <interface> <vlan\_id> <ip\_address> <subnet\_mask>** คำสั่งนี้ใช้สำหรับการสร้าง sub-interface router โดยมีการรับพารามิเตอร์เป็น interface, vlan\_id, ip address, subnet mask และส่งคำสั่ง "int {interface}", "no shut", "shut", "exit", "int {interface}.{vlan\_id}", "encapsulation dot1Q {vlan\_id}", "ip address {ip\_address} {subnet\_mask}" ไปยังอุปกรณ์

## วิธีใช้งาน BOT

1. อุปกรณ์เครือข่ายที่จะทำการเชื่อมต่อกับบอทต้อง config ให้พร้อมสำหรับการ SSH ก่อน
2. ผู้ใช้ต้องทำการเชิญบอทเข้าไปในเซิร์ฟเวอร์ของตัวเอง โดยการสแกน QR ที่ให้ไว้
3. ต้องโหลดโค้ดของผู้พัฒนาลงในเครื่องจาก Link Github :  
<https://github.com/Pikachu051/config-it.git>
4. เมื่อจะเริ่มใช้งานผู้ใช้ต้องทำการรันโค้ดก่อนเริ่มเสมอ
5. จากนั้นให้ผู้ใช้ทำการสร้างการเชื่อมต่อกับอุปกรณ์ โดยใช้คำสั่ง  
!create\_connection
6. เมื่อต้องการทราบคำสั่งเพิ่มเติมสามารถใช้คำสั่ง !command\_list





## สำหรับการเชิญบอกเข้าเซิร์ฟ

### ผลการทดลอง

```

19:36 Mad !create_connection 192.168.21.131 cisco cisco
19:36 BOT CONFIG_IT
Connection created for waitwhat.he.is.mad with device #1.
19:36 Mad !create_connection 192.168.21.132
19:36 BOT CONFIG_IT
Error
Missing required arguments. Type !command_list to see the list of available
commands.

```

รูปที่ 1 !create\_connection 192.168.21.131 cisco cisco

```

19:40 Mad !show_connection
19:40 BOT CONFIG_IT @Mad
List of connected devices has been sent to your DM!

```

```

7:40 PM
Connected Devices
Device #1
IP: 192.168.21.131
Device #2
IP: 192.168.21.132

```

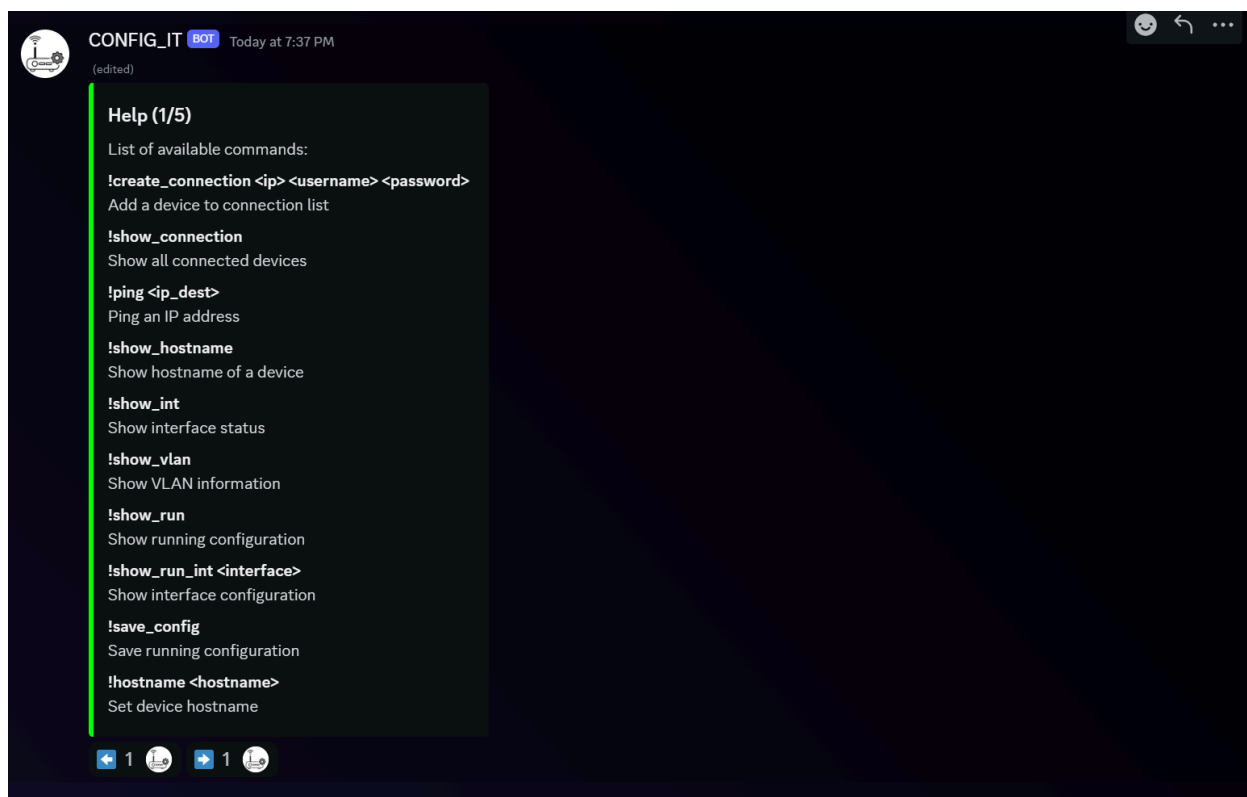
รูปที่ 2 !show\_connection

```

19:37 Mad !command_list
19:38 BOT CONFIG_IT @Mad
Command lists sent to your DM!

```

รูปที่ 3 !command\_list



รูปที่ 4 !command\_list

```

19:49 Mad !ping 1 192.168.21.132
19:49 BOT CONFIG_IT
Connecting to 192.168.21.131...
19:49 BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
Pinging to 192.168.21.132...
BOT CONFIG_IT
Ping Result
Packets sent
5
Packets received
4
-----
Packet received
4 (80% received)
Packet loss
1 (20% loss)

```

ຮູບ 5 !ping 1 192.168.21.132

```

19:50 Mad !show_hostname 1
19:50 BOT CONFIG_IT
Connecting to 192.168.21.131...
19:50 BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
Hostname: R1

```

ຮູບ 6 !show\_hostname 1

```

19:51 Mad !show_int 1
19:51 BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT

```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.21.131	YES	manual	up	up
FastEthernet1/0	192.168.22.1	YES	manual	up	up
FastEthernet1/1	unassigned	YES	unset	administratively down	down
GigabitEthernet2/0	unassigned	YES	unset	administratively down	down

ຮູບທີ່ 7 !show\_int 1

```

19:53 Mad !show_vlan 2
19:53 BOT CONFIG_IT
Connecting to 192.168.21.132...
BOT CONFIG_IT
Success
Connected to 192.168.21.132 successfully!
BOT CONFIG_IT

```

VLAN	Name	Status	Ports
1	default	active	Et0/0, Et0/1, Et0/2, Et0/3, Et1/0, Et1/1, Et1/2, Et1/3, Et2/0, Et2/1, Et2/2, Et2/3, Et3/0, Et3/1, Et3/2, Et3/3
20	VLAN0020	active	
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

ຮູບທີ່ 8 !show\_vlan 2

```

19:54 Mad !show_run 1
19:54 BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
19:54 BOT CONFIG_IT
Building configuration...

Current configuration : 1350 bytes
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
no ip icmp rate-limit unreachable
ip cef
!
!
!

```

ຮູບ 9 !show\_run 1

```

19:56 Mad !show_run_int 1 f0/0
19:56 BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
Building configuration...

Current configuration : 87 bytes
!
interface FastEthernet0/0
ip address 192.168.21.131 255.255.255.0
duplex full
end

```

ຮູບ 10 !show\_run\_int 1 f0/0

```
19:56 Mad !save_config 1
19:56 BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
Success
Configuration has been saved!
```

ຮູບທີ່ 11 !save\_config 1

```
19:57 Mad !hostname 1 R7
19:57 BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
Hostname has been set to R7
```

ຮູບທີ່ 12 !hostname 1 R7

```
20:05 Mad !create_route 1 0.0.0.0 0.0.0.0 192.168.21.1
20:05 BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
Route has been added!
```

ຮູບທີ່ 13 !create\_route 1

```

20:05 Mad !show_route 1
20:05 BOT CONFIG_IT
Connecting to 192.168.21.131...
20:05 BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
20:05 BOT CONFIG_IT
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.21.1 to network 0.0.0.0

C 192.168.21.0/24 is directly connected, FastEthernet0/0
C 192.168.22.0/24 is directly connected, FastEthernet1/0
S* 0.0.0.0/0 [1/0] via 192.168.21.1

```

ຮູບ 14 !show\_route 1

```

20:23 Mad !show_spanning_tree 2
20:23 BOT CONFIG_IT
Connecting to 192.168.21.132...
20:23 BOT CONFIG_IT
Success
Connected to 192.168.21.132 successfully!
20:23 BOT CONFIG_IT
VLAN0001
Spanning tree enabled protocol ieee
Root ID    Priority    32769
Address    aabb.cc00.0100
This bridge is the root
Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec

Bridge ID   Priority    32769 (priority 32768 sys-id-ext 1)
Address     aabb.cc00.0100
Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec
Aging Time  300 sec

Interface    Role Sts Cost      Prio.Nbr Type
-----
Et0/0        Desg FWD 100      128.1   Shr
Et0/1        Desg FWD 100      128.2   Shr
Et0/2        Desg FWD 100      128.3   Shr
Et0/3        Desg FWD 100      128.4   Shr
Et1/0        Desg FWD 100      128.5   Shr
Et1/1        Desg FWD 100      128.6   Shr

```

ຮູບ 15 !show\_spanning\_tree 2

```

20:29 Mad !banner 1 Authorized Users Only!
20:29 BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
Banner has been set!

```

ຮູບທີ່ 16 !banner 1 Authorized Users Only!

```

20:30 Mad !create_vlan 2 20
20:30 BOT CONFIG_IT
Connecting to 192.168.21.132...
BOT CONFIG_IT
Success
Connected to 192.168.21.132 successfully!
BOT CONFIG_IT
VLAN 20 created.

```

ຮູບທີ່ 17 !create 1 vlan 20

```

20:32 Mad !vlan_ip_add 2 20 192.168.23.1 255.255.255.0
20:32 BOT CONFIG_IT
Connecting to 192.168.21.132...
BOT CONFIG_IT
Success
Connected to 192.168.21.132 successfully!
BOT CONFIG_IT
IP Address 192.168.23.1 and Subnet Mask 255.255.255.0 has been added to VLAN 20.

```

ຮູບທີ່ 18 !vlan\_ip\_add 2 20 192.168.23.1 255.255.255.0



```

p32 Mad !vlan_no_shut 2 20
p32 BOT CONFIG_IT
    Connecting to 192.168.21.132...
BOT CONFIG_IT
    Success
    Connected to 192.168.21.132 successfully!
BOT CONFIG_IT
    No shutdown VLAN 20 succeed.

```

ຮູບທີ່ 19 !vlan\_no\_shut 2 20

```

Mad !delete_vlan 2 20
BOT CONFIG_IT
    Connecting to 192.168.21.132...
BOT CONFIG_IT
    Success
    Connected to 192.168.21.132 successfully!
BOT CONFIG_IT
    VLAN 20 has been deleted.

```

ຮູບທີ່ 20 !delete\_vlan 2 20

```

Mad !int_ip_add 1 f1/1 192.168.23.2 255.255.255.0
BOT CONFIG_IT
    Connecting to 192.168.21.131...
BOT CONFIG_IT
    Success
    Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
    IP Address 192.168.23.2 and Subnet Mask 255.255.255.0 has been added to Interface f1/1

```

ຮູບທີ່ 21 !int\_ip\_add 1 f1/1 192.168.23.2 255.255.255.0

```

Mad !add_gateway 1 192.168.21.1
BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
IP Default Gateway 192.168.21.1 has been set on the device.

```

ຮູບທີ່ 22 !add\_gateway 1 192.168.21.1

```

Mad !delete_gateway 1 192.168.21.1
BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
IP Default Gateway 192.168.21.1 has been deleted.

```

ຮູບທີ່ 23 !delete\_gateway 1 192.168.21.1

```

Mad !int_switch_mode 2 e1/1 access
BOT CONFIG_IT
Connecting to 192.168.21.132...
BOT CONFIG_IT
Success
Connected to 192.168.21.132 successfully!
BOT CONFIG_IT
Changed e1/1 to switchport mode access successfully!

```

ຮູບທີ່ 24 !int\_switch\_mode 2 e1/1 access

Mad !int\_no\_shut 1 f1/1

**BOT** CONFIG\_IT

Connecting to 192.168.21.131...

**BOT** CONFIG\_IT

**Success**

Connected to 192.168.21.131 successfully!

**BOT** CONFIG\_IT

Interface f1/1 is now no shutdown.

ຮູບທີ່ 25 !int\_no\_shut 1 f1/1

Mad !rip 1 192.168.21.0,192.168.22.0

**BOT** CONFIG\_IT

Connecting to 192.168.21.131...

**BOT** CONFIG\_IT

**Success**

Connected to 192.168.21.131 successfully!

**BOT** CONFIG\_IT

**Success**

RIP has been configured with the following configuration

**Network**

192.168.21.0

**Network**

192.168.22.0

ຮູບທີ່ 26 !rip 1 192.168.21.0, 192.168.22.0

Mad !remove\_rip\_nw 1 192.168.22.0

**BOT** CONFIG\_IT

Connecting to 192.168.21.131...

**BOT** CONFIG\_IT

**Success**

Connected to 192.168.21.131 successfully!

**BOT** CONFIG\_IT

**Success**

The following RIP networks has been removed.

**Network**

192.168.22.0

ပုံ ၂၇ !remove\_rip\_nw 1 192.168.22.0

Mad !show\_rip 1

**BOT** CONFIG\_IT

Connecting to 192.168.21.131...

**BOT** CONFIG\_IT

**Success**

Connected to 192.168.21.131 successfully!

**BOT** CONFIG\_IT

192.168.21.0/24 auto-summary  
 192.168.21.0/24 directly connected, FastEthernet0/0  
 192.168.22.0/24 is possibly down  
 192.168.22.0/24 is possibly down

ပုံ ၂၈ !show\_rip 1

Mad !disable\_rip 1

**BOT** CONFIG\_IT

Connecting to 192.168.21.131...

**BOT** CONFIG\_IT

**Success**

Connected to 192.168.21.131 successfully!

**BOT** CONFIG\_IT

**Success**

RIP has been disabled.

ပုံ ၂၉ !disable\_rip 1

Mad !ospf 1 192.168.21.0/24/0,192.168.22.0/24/0

**BOT** CONFIG\_IT

Connecting to 192.168.21.131...

**BOT** CONFIG\_IT

**Success**

Connected to 192.168.21.131 successfully!

**BOT** CONFIG\_IT

**Success**

OSPF has been configured with the following configuration

**Network**

192.168.21.0

**Wildcard**

0.0.0.255

**Area**

0

**Network**

192.168.22.0

**Wildcard**

0.0.0.255

**Area**

0

Existing networks currently within OSPF

**Network**

192.168.21.0

**Network**

192.168.22.0

ຮູບທີ່ 30 !ospf 1 192.168.21.0/24/0,192.168.22.0/24/0

Mad !remove\_ospf\_nw 1 192.168.22.0/24/0

**BOT** CONFIG\_IT

Connecting to 192.168.21.131...

**BOT** CONFIG\_IT

**Success**

Connected to 192.168.21.131 successfully!

**BOT** CONFIG\_IT

**Success**

The following OSPF networks has been removed.

**Network**

192.168.22.0 Area: 0

ຮູບທີ່ 31 !remove\_ospf\_nw 1 192.168.22.0/24/0

```

Mad !show_ospf 1
BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
OSPF Router with ID (192.168.23.2) (Process ID 1)
Router Link States (Area 0)
Link ID      ADV Router   Age         Seq#         Checksum Link count
192.168.23.2 192.168.23.2 29          0x80000002  0x00C3E9  1

```

រូបភាព 32 !show\_ospf 1

```

Mad !disable_ospf 1
BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
Success
OSPF has been disabled.

```

រូបភាព 33 !disable\_ospf 1

```
Mad !eigrp 1 192.168.21.0/24,192.168.22.0/24,192.168.23.0/24 1
```

**BOT CONFIG\_IT**

Connecting to 192.168.21.131...

**BOT CONFIG\_IT**

**Success**

Connected to 192.168.21.131 successfully!

**BOT CONFIG\_IT**

**Success**

EIGRP has been configured with the following configuration

**AS Number**

1

**Network**

192.168.21.0

**Subnet Mask**

0.0.0.255

-----

**Network**

192.168.22.0

**Subnet Mask**

0.0.0.255

-----

**Network**

192.168.23.0

**Subnet Mask**

0.0.0.255

-----

Existing networks currently within EIGRP

รูปที่ 34 !eigrp 1 192.168.21.0/24,192.168.22.0/24,192.168.23.0/24 1

Mad !remove\_eigrp\_nw 1 192.168.22.0/24 1

BOT CONFIG\_IT

Connecting to 192.168.21.131...

BOT CONFIG\_IT

**Success**

Connected to 192.168.21.131 successfully!

BOT CONFIG\_IT

**Success**

The following EIGRP networks has been removed.

**Network**

192.168.22.0

**Subnet Mask**

0.0.0.255

ຮູບ 35 !remove\_eigrp\_nw 1 192.168.22.0/24 1

Mad !show\_eigrp 1

BOT CONFIG\_IT

Connecting to 192.168.21.131...

BOT CONFIG\_IT

**Success**

Connected to 192.168.21.131 successfully!

BOT CONFIG\_IT

IP-EIGRP Topology Table for AS(1)/ID(192.168.23.2)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,  
r - reply Status, s - sia Status

P 192.168.21.0/24, 1 successors, FD is 28160  
via Connected, FastEthernet0/0

P 192.168.23.0/24, 1 successors, FD is 28160  
via Connected, FastEthernet1/1

IP-EIGRP neighbors for process 1

H	Address	Interface	Hold Uptime (sec)	SRTT (ms)	RTO	Q Cnt	Seq Num
0	192.168.23.1	Fa1/1	13 00:05:41	11	200	0	5

ຮູບ 36 !show\_eigrp 1



```
Mad !disable_eigrp 11
```

```
BOT CONFIG_IT
```

```
Connecting to 192.168.21.131...
```

```
BOT CONFIG_IT
```

```
Success
```

```
Connected to 192.168.21.131 successfully!
```

```
BOT CONFIG_IT
```

```
Success
```

```
EIGRP has been disabled.
```

ပုံ 37 !disable\_eigrp 11

```
Mad !bgp 1 192.168.21.0/24,192.168.22.0/24,192.168.23.0/24 192.168.23.1:25000 24000
```

```
BOT CONFIG_IT
```

```
Connecting to 192.168.21.131...
```

```
BOT CONFIG_IT
```

```
Success
```

```
Connected to 192.168.21.131 successfully!
```

```
BOT CONFIG_IT
```

```
Success
```

```
BGP has been configured with the following configuration
```

```
AS Number
```

```
24000
```

```
---Networks---
```

```
Network
```

```
192.168.21.0
```

```
Subnet Mask
```

```
255.255.255.0
```

```
Network
```

```
192.168.22.0
```

```
Subnet Mask
```

```
255.255.255.0
```

```
Network
```

```
192.168.23.0
```

```
Subnet Mask
```

```
255.255.255.0
```

```
---Neighbors---
```

```
Neighbor
```

```
192.168.23.1
```

```
Neighbor ASN
```

```
25000
```

ပုံ 38 !bgp 1 192.168.21.0/24,192.168.22.0/24,192.168.23.0/24 192.168.23.1:25000 24000

```

Mad !show_bgp 1
BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21:131 successfully!
BOT CONFIG_IT
BGP table version is 7, local router ID is 192.168.23.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop           Metric LocPrf Weight Path
*> 192.168.21.0    0.0.0.0             0         32768 i
*> 192.168.22.0    0.0.0.0             0         32768 i
* 192.168.23.0     192.168.23.1        0          0 25000 i
*>                 0.0.0.0             0         32768 i

```

ຮູບທີ່ 39 !show\_bgp 1

```

Mad !remove_bgp_nw 1 192.168.22.0/24 24000
BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21:131 successfully!
BOT CONFIG_IT
Success
The following BGP networks has been removed.
Network
192.168.22.0
Subnet Mask
255 255 255.0

```

ຮູບທີ່ 40 !remove\_bgp\_nw 1 192.168.22.0/24 24000

```

Mad !remove_bgp_neighbor 1 192.168.23.1:25000 24000
BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21:131 successfully!
BOT CONFIG_IT
Success
The following BGP neighbors has been removed.
Neighbor
192.168.23.1
Neighbor ASN
25000
-----

```

ຮູບທີ່ 41 !remove\_bgp\_neighbor 1 192.168.23.1:25000 24000

```

Mad !disable_bgp 1 24000
BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
Success
BGP has been disabled.

```

ຮູບ 42 !disable\_bgp 1 24000

```

Mad !delete_route 1 0.0.0.0 0.0.0.0 192.168.21.1
BOT CONFIG_IT
Connecting to 192.168.21.131...
BOT CONFIG_IT
Success
Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
Route has been deleted!

```

ຮູບ 43 !delete\_route 1 0.0.0.0 0.0.0.0 192.168.21.1

```

Mad !vlan_ip_delete 2 20
BOT CONFIG_IT
Connecting to 192.168.21.132...
BOT CONFIG_IT
Success
Connected to 192.168.21.132 successfully!
BOT CONFIG_IT
IP Address and Subnet Mask has been deleted from VLAN 20.

```

ຮູບ 44 !vlan\_ip\_delete 2 20

```
Mad !int_ip_delete 1 f1/1
```

```
BOT CONFIG_IT
```

```
Connecting to 192.168.21.131...
```

```
BOT CONFIG_IT
```

```
Success
```

```
Connected to 192.168.21.131 successfully!
```

```
BOT CONFIG_IT
```

```
Success
```

```
The IP Address has been removed from the interface.
```

ຮູບທີ່ 45 !int\_ip\_delete 1 f1/1

```
Mad !int_access_vlan 2 e1/2 20
```

```
BOT CONFIG_IT
```

```
Connecting to 192.168.21.132...
```

```
BOT CONFIG_IT
```

```
Success
```

```
Connected to 192.168.21.132 successfully!
```

```
BOT CONFIG_IT
```

```
Interface e1/2 is now accessed in VLAN 20!
```

ຮູບທີ່ 46 !int\_access\_vlan 2 e1/2 20

```

Mad !show_mac_table 2
BOT CONFIG_IT
Connecting to 192.168.21.132...
BOT CONFIG_IT
Success
Connected to 192.168.21.132 successfully!
BOT CONFIG_IT
Mac Address Table
-----
Vlan    Mac Address      Type      Ports
----    -
1       000c.2987.ebfe    DYNAMIC   Et0/0
1       0050.56c0.0001    DYNAMIC   Et0/0
1       ca01.05de.0000    DYNAMIC   Et0/0
1       ca01.05de.001c    DYNAMIC   Et0/1
Total Mac Addresses for this criterion: 4

```

ຮູບທີ່ 47 !show\_mac\_table 2

```

Mad !vlan_shut 2 20
BOT CONFIG_IT
Connecting to 192.168.21.132...
BOT CONFIG_IT
Success
Connected to 192.168.21.132 successfully!
BOT CONFIG_IT
Success
The VLAN 20 has been shutdown.

```

ຮູບທີ່ 48 !vlan\_shut 2 20

```
Mad !router_on_a_stick 1 f1/1 20 192.168.23.1 255.255.255.0
```

```
BOT CONFIG_IT
```

```
Connecting to 192.168.21.131...
```

```
BOT CONFIG_IT
```

```
Success
```

```
Connected to 192.168.21.131 successfully!
```

```
BOT CONFIG_IT
```

```
Success
```

```
Router on a stick has been configured.
```

ຮູບ 49 !router\_on\_a\_stick 1 f1/1 20 192.168.23.1 255.255.255.0

```
Mad !show_run_int 1 f1/1.20
```

```
BOT CONFIG_IT
```

```
Connecting to 192.168.21.131...
```

```
BOT CONFIG_IT
```

```
Success
```

```
Connected to 192.168.21.131 successfully!
```

```
BOT CONFIG_IT
```

```
Building configuration...
```

```
Current configuration : 99 bytes
```

```
!
```

```
interface FastEthernet1/1.20
```

```
 encapsulation dot1Q 20
```

```
 ip address 192.168.23.1 255.255.255.0
```

```
end
```

ຮູບ 50 !show\_run\_int 1 f1/1.20

```
7 Mad !traceroute 1 192.168.21.132
7 BOT CONFIG_IT
  Connecting to 192.168.21.131...
BOT CONFIG_IT
  Success
  Connected to 192.168.21.131 successfully!
BOT CONFIG_IT
  Traceroute
  Tracing the route to 192.168.21.132
BOT CONFIG_IT
  Traceroute results:
  1 192.168.21.132 4 msec 4 msec 8 msec
```

ສູນທີ 51 !traceroute 1 192.168.21.132

## สรุปผลการทดลอง

ด้วยการสร้างบอท Discord ที่สามารถโต้ตอบกับผู้ใช้และดำเนินการ Config Router และ Switch ผ่าน SSH ด้วย Netmiko ผู้จัดได้ทำการสร้างช่องทางการสื่อสาร และการจัดการเครือข่ายที่มีประสิทธิภาพในแอปพลิเคชัน Discord

บอท Discord ที่เราสร้างขึ้นใช้ Discord.py library เพื่อเชื่อมต่อและโต้ตอบกับ เซิร์ฟเวอร์ Discord ผ่าน Discord API จากคำสั่งและการตอบกลับที่กำหนดไว้ล่วงหน้า ทำให้สามารถตอบคำถามหรือปฏิบัติตามคำสั่งของผู้ใช้งานในช่องแชทได้ ทำให้ไม่ต้องเข้า แอปพลิเคชันอื่น และช่วยลดการเสียเวลา Config Router และ Switch

คำสั่งต่างๆ ใช้งานผ่าน Netmiko ซึ่งสร้างการเชื่อมต่อ SSH กับ Server ทำให้สามารถส่งคำสั่งตาม Command ที่ผู้จัดทำเขียนขึ้นมาได้ commandที่ใช้จะเขียนตามหลักของ Cisco และมีการปรับให้คำสั่งกระชับมากขึ้น เพื่อให้ใช้ได้รวดเร็วมากขึ้น รวมถึงการรับข้อมูลตอบกลับเพื่อส่งกลับให้ผู้ใช้ผ่านแชทของ Discord และสามารถใช้บริการได้ ทุกเมื่อขอเพียงเชื่อมต่ออินเทอร์เน็ต ทำให้ผู้ใช้สามารถจัดการเครือข่ายของตนได้อย่างรวดเร็วและสะดวกสบาย

## ประโยชน์ที่ได้รับ

1. ได้นำความรู้จากในห้องเรียนมาต่อยอด
2. ช่วยต่อการใช้งานไม่ซับซ้อน
3. อำนวยความสะดวกแก่คนที่ต้องเดินทาง หรือ ทำงานนอกสถานที่



## บรรณานุกรม

1. การสร้าง Discord bot สืบค้นเมื่อ วันที่ 31 มีนาคม 2567 ,  
<https://appmaster.io/th/blog/discord-bot-withiisraangaelaephimaipyan-gechirfew-r>, <https://www.youtube.com/watch?v=2k9x0s3awss>
2. Netmiko: A Python Library สืบค้นวันที่ 31 มีนาคม 2567,  
<https://medium.com/@cyber.stack/netmiko-a-python-library-for-managing-network-devices-7927c935e2da>
3. Netmiko: A Python Library สืบค้นวันที่ 31 มีนาคม 2567,  
<https://ktbyers.github.io/netmiko/docs/netmiko/index.html>
4. Cisco command Cheat Sheet สืบค้นวันที่ 1 เมษายน 2567,  
[https://www.netwrix.com/cisco\\_commands\\_cheat\\_sheet.html](https://www.netwrix.com/cisco_commands_cheat_sheet.html)

## หน้าที่และภาระงานที่รับผิดชอบ

นางสาวชุตติกาญจน์ ใจคง 65070053	- Coding + Testing, Document
นางสาวญาตาวิ ฤกษ์ประสูต 65070055	- Coding, Document, Slide
นายภูริพัทธ์ ชนะภัย 65070181	- Coding + Debugging
นายศุภชัย มณีรัตน์ 65070224	- Coding, Slide
นายสหชินเดช เกตุดี 65070232	- Coding + Testing + Debugging
นายสุทธิเกียรติ หักยาสมบุรณ์ 65070240	- Coding, Document