รายงาน Mini Project เรื่อง: FTP (File Transfer Protocol)

เสนอ

อาจารผู้สอน ผู้ช่วยศาสตราจารย์ดรัสวิน วงศ์ปรเมษฐ์

จัดทำโดย

นาย ยุทธ์กรณ์ แจ่นประโคน 670112418050 ปี 2

โครงงานนี้เป็นส่วนหนึ่งของรายวิชา เทคโนโลยีอินเทอร์เน็ต (4132203)

ภาคเรียนที่ 2 ปีการศึกษา 2568

สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาลัยราชภัฏบุรีรัมย์

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของการจัดทำ Mini Project โดยมีวัตถุประสงค์เพื่อศึกษาและทำความเข้าใจระบบ FTP (File Transfer Protocol) ซึ่งเป็นโปรโตคอลพื้นฐานที่ใช้ในการถ่ายโอนไฟล์ระหว่างเครื่องลูกข่ายและเครื่องแม่ข่ายผ่านเครือข่ายอินเทอร์เน็ต โดยระบบ FTP มีบทบาทสำคัญในการจัดการไฟล์ในระบบเครือข่าย ทั้งในระดับผู้ใช้ทั่วไปและในระดับองค์กร

เนื้อหาในรายงานฉบับนี้ครอบคลุมถึงความหมายและประวัติความเป็นมาของ FTP หลักการทำงาน พอร์ตที่เกี่ยวข้อง การใช้งานร่วมกับภาษาโปรแกรม การติดตั้งและใช้งาน FTP Server ผ่าน Docker รวมถึงตัวอย่างไฟล์ docker-compose พร้อมคำอธิบายและภาพประกอบ เพื่อให้ผู้อ่านสามารถเข้าใจและนำไปประยุกต์ใช้งานได้จริง

ผู้จัดทำหวังว่ารายงานฉบับนี้จะเป็นประโยชน์ต่อผู้อ่านในการเรียนรู้และประยุกต์ใช้ระบบ FTP ในการพัฒนาและจัดการระบบเครือข่าย รวมถึงเป็นแนวทางในการศึกษาต่อยอดในหัวข้อที่เกี่ยวข้องในอนาคต

ลงชื่อผู้จัดทำ

นายยุทธ์กรณ์ แจ่นประโคน

เนื้อหา

ระบบ FTP คืออะไร

FTP (File Transfer Protocol) คือโปรโตคอลที่ใช้ในการถ่ายโอนไฟล์ระหว่างเครื่องลูกข่าย (Client) และเครื่องแม่ข่าย (Server) ผ่านเครือข่าย TCP/IP โดยทำงานในรูปแบบ Client-Server ผู้ใช้สามารถอัปโหลด ดาวน์โหลด หรือจัดการไฟล์บนเซิร์ฟเวอร์ได้จากระยะไกล

โครงสร้างของระบบ FTP แบ่งได้เป็น 3 ส่วนหลัก ได้แก่

1. FTP Client – โปรแกรมหรือคำสั่งที่ใช้เชื่อมต่อไปยัง FTP Server เช่น FileZilla, WinSCP หรือ ftp บน terminal

2. FTP Server – เครื่องแม่ข่ายที่ให้บริการจัดการไฟล์ เช่น อัปโหลด ดาวน์โหลด ลบ หรือเปลี่ยนชื่อไฟล์

3. ช่องทางการเชื่อมต่อ – ใช้พอร์ต 21 สำหรับควบคุมการเชื่อมต่อ และพอร์ต 20 หรือพอร์ตแบบสุ่มสำหรับส่งข้อมูล (Data Connection)

ระบบ FTP ใช้วิธีการส่งข้อมูลแบบ Client–Server โดยเครื่องลูกข่ายจะส่งคำสั่งไปยังเซิร์ฟเวอร์ผ่านช่องทางควบคุม และเซิร์ฟเวอร์จะตอบกลับหรือส่งข้อมูลผ่านช่องทางข้อมูลตามคำสั่งที่ได้รับ

ประวัติความเป็นมา

- ปี 1971: เริ่มมีการใช้งาน FTP บนเครือข่าย ARPANET
- ปี 1985: FTP ได้รับการรับรองใน RFC 959
- ปัจจุบัน: มีการพัฒนาเวอร์ชันที่ปลอดภัยมากขึ้น เช่น FTPS และ SFTP เพื่อรองรับการเข้ารหัสข้อมูล

หลักการทำงานของระบบ FTP

1. ผู้ใช้เปิดโปรแกรม FTP Client เช่น FileZilla หรือใช้คำสั่ง ftp ผ่าน terminal เพื่อเชื่อมต่อกับ FTP Server
2. FTP Client เชื่อมต่อกับ FTP Server ผ่านพอร์ต 21 ซึ่งเป็นช่องทางควบคุม (Control Connection)
3. ผู้ใช้เข้าสู่ระบบด้วยชื่อผู้ใช้และรหัสผ่าน หรือในบางกรณีอาจใช้ Anonymous Login
4. เมื่อเข้าสู่ระบบสำเร็จ ผู้ใช้สามารถส่งคำสั่งต่าง ๆ เช่น LIST, GET, PUT เพื่อจัดการไฟล์
5. เมื่อมีการส่งหรือรับไฟล์ จะมีการเปิดช่องทางข้อมูล (Data Connection) แยกต่างหากผ่านพอร์ต 20 (Active Mode) หรือพอร์ตแบบสุ่ม (Passive Mode)
6. หลังจากใช้งานเสร็จ ผู้ใช้สามารถพิมพ์คำสั่ง bye หรือ quit เพื่อยกเลิกการเชื่อมต่อ

FTP แบบดั้งเดิมไม่มีการเข้ารหัสข้อมูล ทำให้ข้อมูลที่ส่งผ่านเครือข่ายสามารถถูกดักฟังได้ จึงมีการพัฒนา FTPS และ SFTP เพื่อเพิ่มความปลอดภัย

พอร์ตที่ใช้งาน

โปรโตคอล	พอร์ต	รายละเอียด
FTP	21	พอร์ตควบคุมการเชื่อมต่อ
FTP	20	พอร์ตส่งข้อมูล (Active Mode)
SFTP	22	ใช้ SSH ในการถ่ายโอนไฟล์
FTPS	990	ใช้ SSL/TLS ในการเข้ารหัส

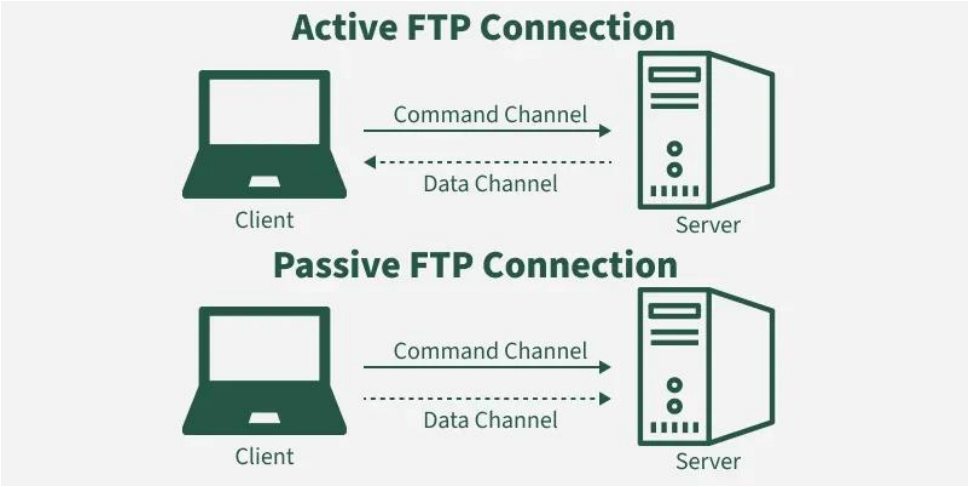
รูปแบบการใช้งาน FTP

การเชื่อมต่อ FTP Server	ftp 192.168.1.10
การเข้าสู่ระบบด้วยชื่อผู้ใช้และรหัสผ่าน	Name: user Password: *****
การอัปโหลดไฟล์	put file.txt
การดาวน์โหลดไฟล์	get file.txt
การแสดงรายการไฟล์	ls

การใช้งาน FTP ร่วมกับภาษาโปรแกรม

ตัวอย่าง Python (ใช้ ftplib)	from ftplib import FTP ftp = FTP("192.168.1.10") ftp.login(user="user", passwd="pass") ftp.retrlines("LIST") ftp.quit()

ตัวอย่างนี้ใช้เชื่อมต่อ FTP Server และแสดงรายการไฟล์



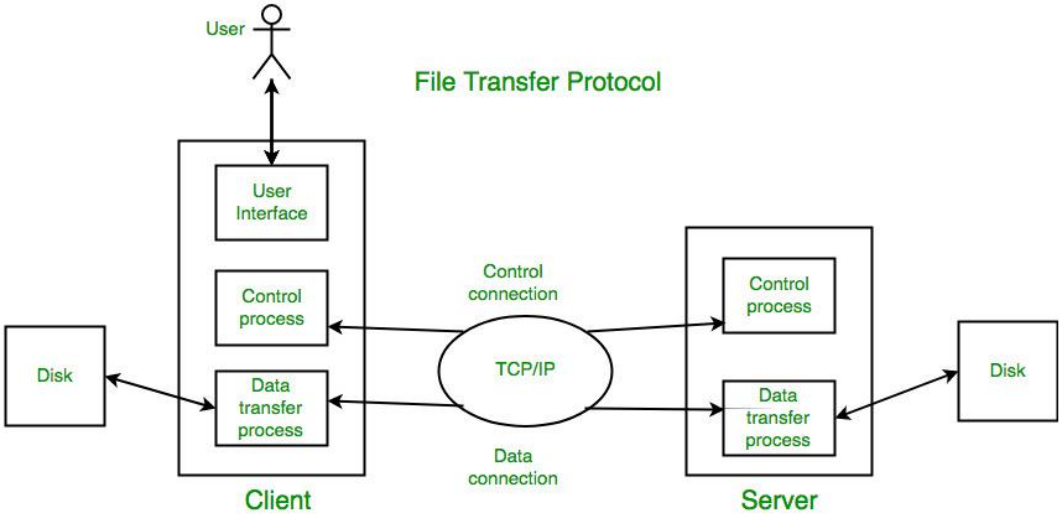


Image บน Docker Hub

Image	รายละเอียด
fauria/vsftpd	ใช้งานง่าย, รองรับ anonymous และ user login, ตั้งค่าผ่าน environment
stilliard/pure-ftpd	รองรับ passive mode, TLS, user mapping, ใช้ใน production ได้
delfer/alpine-ftp-server	เบา, รวดเร็ว, เหมาะกับงานที่ต้องการ footprint ต่ำ
panubo/vsftpd	ใช้ base image แบบ secure, เหมาะกับงานที่ต้องการความปลอดภัยสูง

คำสั่ง pull image

fauria/vsftpd	docker pull fauria/vsftpd
stilliard/pure-ftpd	docker pull stilliard/pure-ftpd
delfer/alpine-ftp-server	docker pull delfer/alpine-ftp-server
panubo/vsftpd	docker pull panubo/vsftpd

คำอธิบาย

- เปิดพอร์ต 21 และ 20 ของโฮสต์ เพื่อเชื่อมต่อกับ FTP Server ภายในคอนเทนเนอร์
- กำหนดตัวแปรสภาพแวดล้อม (Environment Variables) เช่น FTP_USER และ FTP_PASS เพื่อสร้างบัญชีผู้ใช้
- ใช้ Volume เพื่อเก็บไฟล์ข้อมูลแบบถาวร แม้คอนเทนเนอร์จะถูกลบหรือรีสตาร์ตก็ยังไม่หาย
- สามารถเชื่อมต่อผ่านโปรแกรม FTP Client เช่น FileZilla หรือใช้คำสั่ง ftp ผ่าน terminal ได้ทันที
- ตั้งค่า restart ให้คอนเทนเนอร์เปิดขึ้นอัตโนมัติเมื่อระบบรีบูต หรือมีการหยุดทำงานโดยไม่ได้สั่งปิด

อ้างอิง

1. https://piromweb.com/blog/ftp
2. https://th.wikipedia.org/wiki/FTP
3. https://hub.docker.com/r/fauria/vsftpd
4. https://docs.docker.com/compose/
5. https://www.geeksforgeeks.org/file-transfer-protocol-ftp/
6. https://www.scaler.com/topics/computer-network/file-transfer-protocol/
7. https://www.pynetlabs.com/ftp-file-transfer-protocol/