เรื่อง Chat

จัดทำโดย

1. นายธีระเดช จะเรดรัมย์ 670112418042

เสนอ

ผู้ช่วยศาสตราจารย์ดรัสวิน วงศ์ปรเมษฐ์ รายงานเล่มนี้เป็นส่วนหนึ่งของรายวิชา

เทคโนโลยีอินเทอร์เน็ต

(4132203) มหาวิทยาลัยราชภัฏบุรีรัมย์

คำนำ

รายงานฉบับนี้จัดทำขึ้นเพื่อศึกษาเรื่อง **Chat** ซึ่งเป็นเครื่องมือสำคัญในการโต้ตอบแบบทันทีในยุคดิจิทัล โดยมีวัตถุประสงค์เพื่อทำความเข้าใจในประเด็นสำคัญต่าง ๆ ได้แก่ ความหมาย ประวัติความเป็นมา โปรโตคอลที่ใช้ (เช่น TCP, UDP, WebSocket) รวมถึงภาษาโปรแกรมที่เกี่ยวข้องในการพัฒนา

นอกจากนี้ รายงานยังได้นำเสนอการประยุกต์ใช้งานระบบแชทในสภาพแวดล้อมจริง โดยแสดงตัวอย่าง **Docker Image** จาก Docker Hub และการกำหนดค่าด้วยไฟล์ **docker-compose.yml** เพื่อให้เห็นภาพรวมของการจัดการและติดตั้งระบบแชทในรูปแบบคอนเทนเนอร์

ผู้จัดทำหวังเป็นอย่างยิ่งว่าเนื้อหาในรายงานนี้จะเป็นประโยชน์ต่อการศึกษาและทำความเข้าใจหลักการทำงานและการประยุกต์ใช้เทคโนโลยีแชทได้อย่างสมบูรณ์

จัดทำโดย

ธีระเดช จะเรดรัมย์

12 พฤศจิกายน 2568

ความหมายของระบบแชท

ระบบแชท (Chat System) หรือ Instant Messaging (IM) คือบริการที่อนุญาตให้ผู้ใช้สื่อสารแลกเปลี่ยนข้อความ, เสียง, หรือวิดีโอ แบบเรียลไทม์ (Real-time) ผ่านอินเทอร์เน็ต ลักษณะสำคัญคือ ความฉับไว และการสื่อสารแบบสองทิศทาง ซึ่งแตกต่างจากการส่งอีเมล ปัจจุบันขยายไปสู่ Chatbots และ Live Web Chat

ประวัติความเป็นมา

ยุคบุกเบิก (1980s): เริ่มต้นที่ **Internet Relay Chat (IRC)** ในปี 1988 โดย Jarkko Oikarinen ถือเป็นรากฐานของการสนทนากลุ่มบนอินเทอร์เน็ต

ยุคทองของ IM (Late 1990s): โปรแกรม Instant Messaging เชิงพาณิชย์ได้รับความนิยม เช่น AOL Instant Messenger (AIM), ICQ, และ MSN Messenger ซึ่งเน้นการสื่อสารส่วนตัวเป็นหลัก

ยุค Mobile และ Social Media (2010s – ปัจจุบัน): การมาของสมาร์ทโฟนทำให้เกิดแอปพลิเคชันอย่าง **WhatsApp**, **LINE**, และ **Facebook Messenger** ที่เปลี่ยนการแชทให้เป็นแพลตฟอร์มแบบครบวงจรและนำไปใช้ในธุรกิจ (เช่น Slack และ Chatbots)

พอร์ทที่ใช้งาน

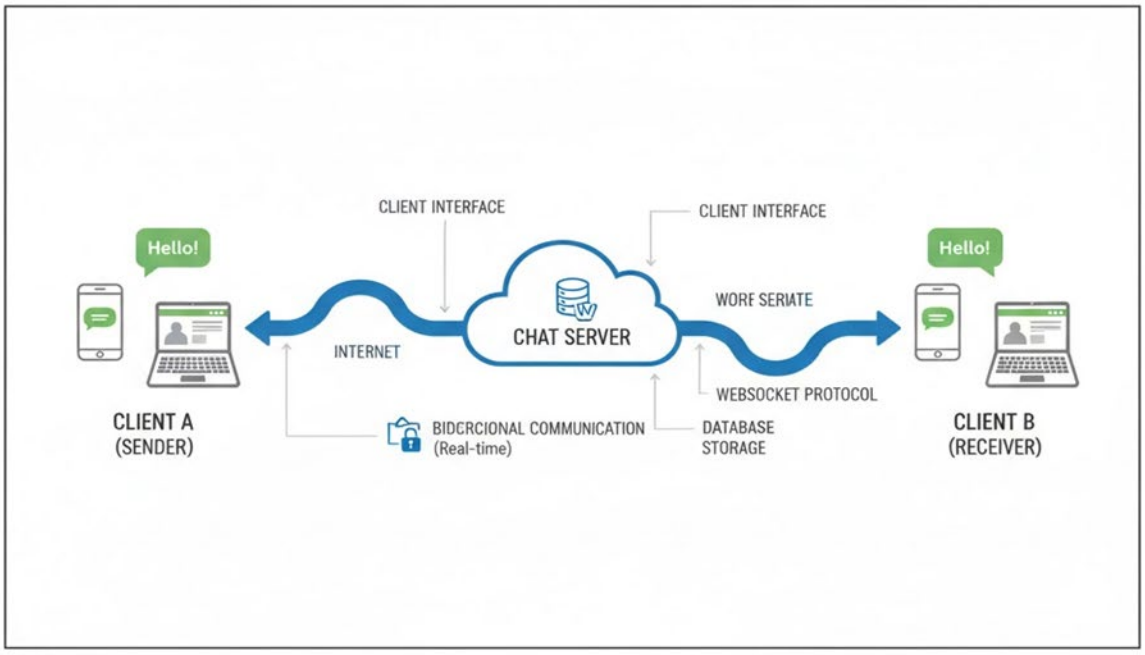
- **Port 80 / 443:** ใช้เริ่มต้นการเชื่อมต่อ **WebSocket** (HTTP/HTTPS)
- **พอร์ทที่กำหนดเอง (Custom Ports):** สำหรับ Chat Server ที่พัฒนาขึ้นเอง (เช่น Port 3000)

การใช้งานร่วมกับภาษา (Programming Languages)

- ยกตัวอย่างภาษาที่นิยมใช้ในการพัฒนา Chat Application:

- **Node.js/JavaScript:** นิยมใช้คู่กับ **WebSocket** (เช่น library ชื่อ **Socket.IO**) สำหรับการสื่อสารแบบ Real-time
- **Python:** ใช้สำหรับ Chatbot (โดยเฉพาะด้าน AI/NLP) หรือ Backend
- **Go, Java:** ใช้สำหรับระบบ Chat ขนาดใหญ่ที่มีประสิทธิภาพสูง

BASIC CHAT SYSTEM ARCHITECTURE



REAL-TIME MESSAGING FLOW

•

การใช้งาน chat ด้วย Docker

docker-compose

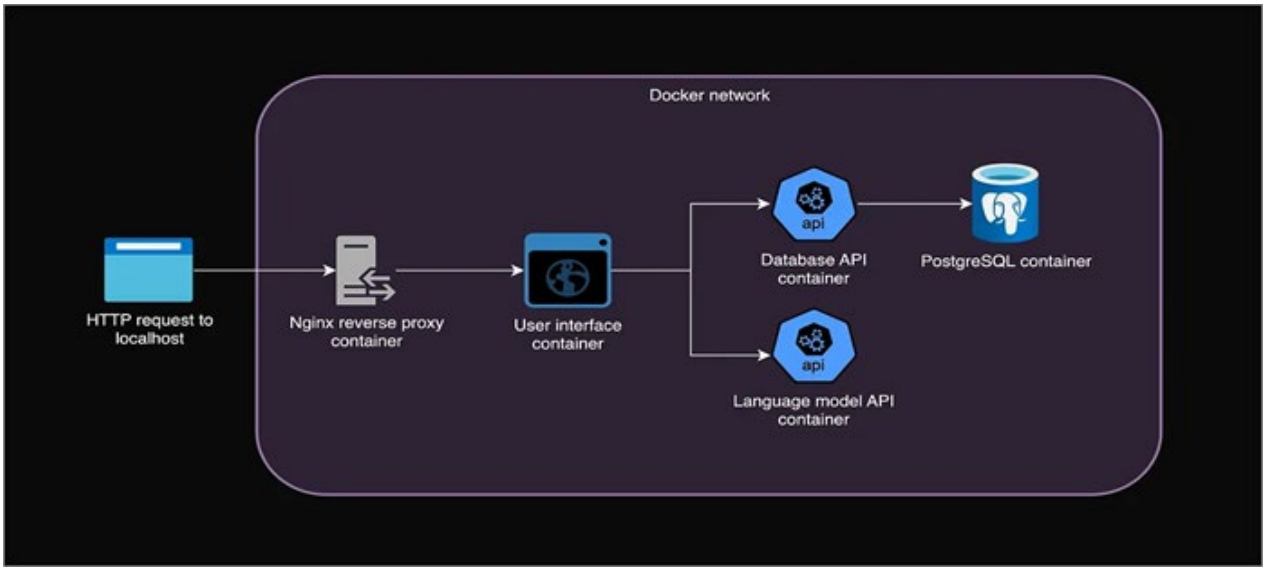
```
version: '3.7'


services:
  # 1. Chat Application Server (Rocket.Chat)
  rocketchat:
    image: rocketchat/rocket.chat:latest
    container_name: rocketchat
    restart: unless-stopped
    environment:
      - MONGO_URL=mongodb://mongo:27017/rocketchat  # เชื่อมต่อกับคอนเทน
เนอร์ mongo
    ports:
      - 3000:3000 # เปิดพอร์ท 3000 สำหรับการเข้าถึงจากภายนอก
    depends_on:
      - mongo
  # 2. Database Service (MongoDB)
  mongo:
    image: mongo:4.0
    container_name: mongo
```

restart: unless-stopped

volumes:

- ./data/db:/data/db # จัดเก็บข้อมูลอย่างถาวร (Persistent Storage)



อ้างอิง

1. https://aws.amazon.com/th/what-is/chatbot/
2. https://deadsimplechat.com/blog/websockets-and-nodejs-real-time-chat-app/
3. https://docs.rocket.chat/docs/deploy-with-docker-docker-compose
4. https://www.byteplus.com/en/topic/99863
5. https://th.wikipedia.org/wiki/%E0%B8%88%E0%B8%B1%E0%B8%81%E0%B8%A3%E0%B8%81%E0%B8%A5%E0%B8%AA%E0%B8%99%E0%B8%97%E0%B8%99%E0%B8%B2