**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG** **CƠ SỞ TP.HCM**

- - - 🙞 🕮 🙜 - - -

Icon

Description automatically generated

**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**MÔN IOT VÀ ỨNG DỤNG**

**Đề tài**

**SKYNET IOT MESSAGING PLATFORM**

**Giảng viên hướng dẫn:** Th.S Đàm Minh Lịnh

**Nhóm sinh viên thực hiện:** Huỳnh Ngọc Ánh – N20DCCN003

Phạm Thụy Dạ Thảo – N20DCCN145

*TP. HCM, năm 2024*

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin gửi lời tri ân sâu sắc đến các thầy cô trường Học Viện Công Nghệ Bưu Chính Viễn thông cơ sở tại TP.HCM đã tận tình dẫn dắt và truyền đạt cho chúng em rất nhiều kiến thức quý báu trong các học kỳ vừa qua.

Đặc biệt, chúng em xin gửi lời cảm ơn chân thành đến ThS Đàm Minh Lịnh. Thầy đã hướng dẫn tận tình, truyền đạt kiến thức, chỉ bảo cho nhóm em trong suốt thời gian học tập và thực hiện đề tài.

Tuy nhiên, vì thời gian và kiến thức chuyên môn còn hạn chế cùng nhiều yếu tố khách quan khác nên trong quá trình thực hiện đề tài không tránh khỏi những sai sót. Kính mong được lời nhận xét và góp ý của thầy, để chúng em có thể hoàn thiện tốt hơn nữa.

*TP.HCM, năm 2024*

**MỤC LỤC**

[PHẦN 1: GIỚI THIỆU CHUNG 1](#_Toc155306871)

[1. Giới thiệu Skynet 1](#_Toc155306872)

[2. Giới thiệu Ubuntu 1](#_Toc155306873)

[2.1. Ubuntu là gì? 1](#_Toc155306874)

[2.2. WSL là gì? 1](#_Toc155306875)

[3. Giới thiệu Git 2](#_Toc155306876)

[4. Giới thiệu NodeJs 3](#_Toc155306877)

[5. Giới thiệu NPM 3](#_Toc155306878)

[PHẦN 2: TIẾN HÀNH CÀI ĐẶT 6](#_Toc155306879)

[1. Cài đặt WSL 6](#_Toc155306880)

[2. Cài đặt Ubuntu 6](#_Toc155306881)

[3. Cài đặt Redis trên Ubuntu 7](#_Toc155306882)

[4. Cài đặt MongoDB trên Ubuntu 7](#_Toc155306883)

[5. Cài đặt git và software-properties-common 9](#_Toc155306884)

[6. Cài đặt skynet 11](#_Toc155306885)

[7. Chạy Project 12](#_Toc155306886)

[PHẦN 3: DEMO 13](#_Toc155306887)

[PHẦN 4: TỔNG KẾT 14](#_Toc155306888)

[TÀI LIỆU THAM KHẢO 15](#_Toc155306889)

**PHẦN 1: GIỚI THIỆU CHUNG**

**1. Giới thiệu Skynet**

* Skynet là một nền tảng mã nguồn mở, hỗ trợ một nền tảng tin nhắn tức thời được thiết kế để chạy trên mạng lưới các mạng IoT có chung API hoặc giao thức truyền thông.
* Skynet API hỗ trợ các giao thức sau: HTTP, REST, WebSockets, MQTT (Message Queue Telemetry Transport) để đảm bảo việc truyền tải thông điệp
* Skynet cho phép các thiết bị như thiết bị nhà thông minh, cảm biến, drones, Arduinos, Raspberry Pis, ... đăng kí vào network của nó
* Khi các thiết bị đăng ký với Skynet, chúng được gán một định danh duy nhất là UUID dài 36 kí tự cùng với một secret token
* Những thiết bị khi đăng kí vào mạng skynet có thể đăng kí đến thiết bị khác và gửi thông điệp đến nó

**2. Giới thiệu Ubuntu**

**2.1. Ubuntu là gì?**

Ubuntu là một hệ điều hành trên máy tính, và nó được phát triển dựa trên Linux/Debian GNU, có rất nhiều tính năng hữu ích cho các máy tính xách tay, desktop và cả máy chủ.

Ưu điểm:

* Ubuntu hoàn toàn miễn phí.
* Đáp ứng nhu cầu của đa số người dùng.
* Hệ điều hành hoàn toàn không có virus.
* Mọi phần mềm được cập nhật tự động.
* Ubuntu được thiết kế để bảo mật.
* Đa ngôn ngữ.

Hạn chế:

* Khó dùng, khó làm quen đối với những người chuyển từ Windows sang.
* Thiếu nhiều phần mềm chuyên dụng cao.
* Bộ gõ tiếng việt trên Ubuntu khó khăn hơn so với Windows.

**2.2. WSL là gì?**

WSL viết tắt cho Windows Subsystem for Linux cho phép bạn cài đặt Ubuntu Terminal Environment trên hệ điều hành Windows giúp nhanh chóng tiết kiệm thời gian mà bạn không cần chuyển qua chuyển lại giữa 2 hệ điều hành.

Với WSL ta có thể chạy các lệnh, các ứng dụng trực tiếp từ dòng lệnh Windows mà không phải bận tâm về việc tạo/quản lý máy ảo như trước đây. Cụ thể, một số lưu ý mà Microsoft liệt kê có thể làm với WSL:

* Chọn sử dụng distro Linux từ Microsoft Store: Hiện giờ đang có các Distro Linux rất gọn nhẹ trên Store sử dụng được ngày như Ubuntu, Debian...
* Chạy được từ dòng lệnh các lệnh linux như ls, grep, sed ... hoặc bất kỳ chương trình nhị phân 64 bit (ELF-64) nào của Linux.
* Chạy được các công cụ như: vim, emacs…, các ngôn ngữ lập trình như NodeJS, JavaScript, C/C++, C#..., các dịch vụ như MySQL, Apache, lighthttpd...
* Có thể thực hiện cài đặt các gói từ trình quản lý gói của Distro đó (như lệnh apt trên Ubuntu).
* Từ Windows có thể chạy các ứng dụng Linux (dòng lệnh).
* Từ Linux có thể gọi ứng dụng của Windows.

**3. Giới thiệu Git**

Git là hệ thống kiểm soát phiên bản phân tán mã nguồn mở. Các dự án thực tế thường có nhiều nhà phát triển làm việc song song. Vì vậy, một hệ thống kiểm soát phiên bản như Git là cần thiết để đảm bảo không có xung đột mã giữa các nhà phát triển. Ngoài ra, các yêu cầu trong dự án thay đổi thường xuyên. Vì vậy, cần một hệ thống cho phép nhà phát triển quay lại phiên bản cũ hơn của mã. Sử dụng Git sẽ đem tới cho các lập trình viên một kho lưu trữ, trong đó chứa đầy đủ các lịch sử thay đổi của hệ thống.

Git cũng có các lợi ích tuyệt vời đối với các dự án đang chạy song song. Đặc biệt là các dự án có cùng một cơ sở code. Việc sử dụng Git sẽ giúp phân nhánh tốt hơn. Qua đó giúp việc thực hiện các dự án dễ dàng và hiệu quả hơn

* Thao tác thực hiện nhanh, gọn, lẹ hơn, việc sử dụng an toàn hơn.
* Người dùng có thể dễ dàng kết hợp các nhánh với nhau. Qua đó giúp đơn giản hóa quy trình code theo nhóm.
* Khi sử dụng GIT, người dùng chỉ cần sử dụng clone mã nguồn từ khoa chứa/phiên bản nào đó có sẵn trong kho lưu trữ/ một nhánh trong kho lưu trữ. Hệ thống lưu trữ chính xác sẽ cho phép bạn có thể làm việc bất cứ lúc nào, ở bất cứ đâu.
* Việc lập trình các dự án song song sẽ cực kỳ dễ dàng, hiệu quả và chính xác.

**4. Giới thiệu NodeJs**

NodeJS là một môi trường runtime chạy JavaScript đa nền tảng và có mã nguồn mở, được sử dụng để chạy các ứng dụng web bên ngoài trình duyệt của client. Nền tảng này được phát triển bởi Ryan Dahl vào năm 2009, được xem là một giải pháp hoàn hảo cho các ứng dụng sử dụng nhiều dữ liệu nhờ vào mô hình hướng sự kiện (event-driven) không đồng bộ.

Ưu điểm:

* IO hướng sự kiện không đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời.
* Sử dụng JavaScript – một ngôn ngữ lập trình dễ học.
* Chia sẻ cùng code ở cả phía client và server.
* NPM (Node Package Manager) và module Node đang ngày càng phát triển mạnh mẽ.
* Cộng đồng hỗ trợ tích cực.
* Cho phép stream các file có kích thước lớn.

Nhược điểm:

* Không có khả năng mở rộng, vì vậy không thể tận dụng lợi thế mô hình đa lõi trong các phần cứng cấp server hiện nay.
* Khó thao tác với cơ sử dữ liệu quan hệ.
* Mỗi callback sẽ đi kèm với rất nhiều callback lồng nhau khác.
* Cần có kiến thức tốt về JavaScript.
* Không phù hợp với các tác vụ đòi hỏi nhiều CPU.

**5. Giới thiệu NPM**

NPM viết tắt của từ Node Package Manager là một công cụ tạo và quản lý các thư viện javascript cho Nodejs với mục đích hỗ trợ cho từng bản phát triển của Node. NPM ra đời với khả năng giải quyết các vấn đề trong việc quản lý các package và phiên bản của chúng.

Trong nhiều năm trở lại đây, NPM được số lượng lớn các lập trình viên Javascript sử dụng một cách rộng rãi để có thể chia sẻ hàng trăm nghìn các thư viện với những đoạn code đã được thực hiện một chức năng có sẵn đồng thời cài đặt nhiều hơn những module khác nhau phục vụ cho việc quản lý dependency. Điều này giúp các dự án mới tránh được việc phải viết lại các thư viện lập trình, các thành phần cơ bản thậm chí là những framework quan trọng khác.

NPM cung cấp 2 chức năng chính bao gồm:

* Người dùng có thể dễ dàng truy cập vào https://nodejs.org/en/ để tìm kiếm được các package bởi NPM được xem là kho lưu trữ trực tuyến cho các module hay package.
* NPM có khả năng quản lý các module và phiên bản của Javascript trong các dự án một cách dễ dàng hơn, nhanh chóng hơn và tiết kiệm nhiều thời gian hơn.

Cách thức hoạt động của NPM được phát huy một cách hiệu quả dựa trên hai vai trò chính như sau:

* NPM là một repository được sử dụng rộng rãi với mục đích publish project Node.js mã nguồn mở tương tự như một nền tảng trực tuyến các công cụ được viết bằng Javascript có thể được chia sẻ và publish bởi người dùng.
* NPM là công cụ dòng lệnh giúp người dùng có thể tương tác với các nền tảng trực tuyến như máy chủ hay trình duyệt. Tiện ích này sẽ hỗ trợ việc cài đặt hoặc gỡ cài đặt gói, quản lý do máy chủ thực hiện quản lý phiên bản, quản lý dependency cần thiết để chạy dự án.

**6. Giới thiệu Redis Server**

Redis (REmote DIctionary Server) là một mã nguồn mở được dùng để lưu trữ dữ liệu có cấu trúc, có thể sử dụng như một database, bộ nhớ cache hay một message broker.

Redis là một hệ thống cơ sở dữ liệu key-value in-memory, được sử dụng làm database, cache, message broker, và streaming engine.

Redis là một cơ sở dữ liệu trong bộ nhớ, có nghĩa là dữ liệu được lưu trữ trên bộ nhớ RAM của máy chủ lưu trữ. Điều này làm cho Redis nhanh bởi vì nó không phải chờ cho ổ cứng truyền thống tìm kiếm và đọc nhiều lần.

Là một cơ sở dữ liệu NoSQL, Redis cho phép bạn tạo cấu trúc dữ liệu ít lược đồ phù hợp với nhu cầu ứng dụng của bạn. Bạn có thể sử dụng Redis cho những cái đơn giản, như tạo một bộ đếm lượng khách truy cập, hoặc để biết thêm thông tin, như lưu trữ thông tin người dùng.

Redis hữu dụng khi bạn cần một cái gì đó thật sự rất nhanh. Redis có thể được sử dụng cho một loạt các trường hợp sử dụng khác nhau, như:

* Caching
* Queuing
* Pub/sub
* Machine learning

Lợi ích của Redis:

* Hiệu suất nhanh
* Cấu trúc dữ liệu linh hoạt
* Đơn giản và dễ sử dụng
* Khả năng sao chép, bền vững và nhất quán
* Độ khả dụng cao và quy mô linh hoạt
* Mã nguồn mở

**7. Giới thiệu MongoDB**

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL. MongoDB giúp các tổ chức lưu trữ lượng lớn dữ liệu trong khi vẫn hoạt động nhanh chóng.

MongoDB là một chương trình cơ sở dữ liệu mã nguồn mở được thiết kế theo kiểu hướng đối tượng trong đó các bảng được cấu trúc một cách linh hoạt cho phép các dữ liệu lưu trên bảng không cần phải tuân theo một dạng cấu trúc nhất định nào. Chính do cấu trúc linh hoạt này nên MongoDB có thể được dùng để lưu trữ các dữ liệu có cấu trúc phức tạp và đa dạng và không cố định (hay còn gọi là Big Data).

Lợi thế của MongoDB so với các cơ sở dữ liệu dạng quan hệ:

* Ít Schema hơn: Số trường, nội dung và kích cỡ của Document này có thể khác với Document khác.
* Cấu trúc của một đối tượng là rõ ràng.
* Không có các Join phức tạp.
* Khả năng truy vấn sâu hơn.
* MongoDB dễ dàng để mở rộng.
* Việc chuyển đổi/ánh xạ của các đối tượng ứng dụng đến các đối tượng cơ sở dữ liệu là không cần thiết.
* Sử dụng bộ nhớ nội tại để lưu giữ phần công việc, giúp truy cập dữ liệu nhanh hơn.

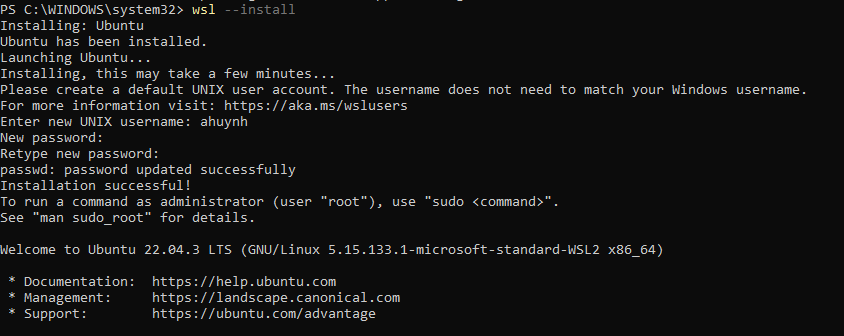
**PHẦN 2: TIẾN HÀNH CÀI ĐẶT**

1. **Cài đặt WSL**

Mở powershell prompt với quyền Administrator và run câu lệnh sau để cài đặt:

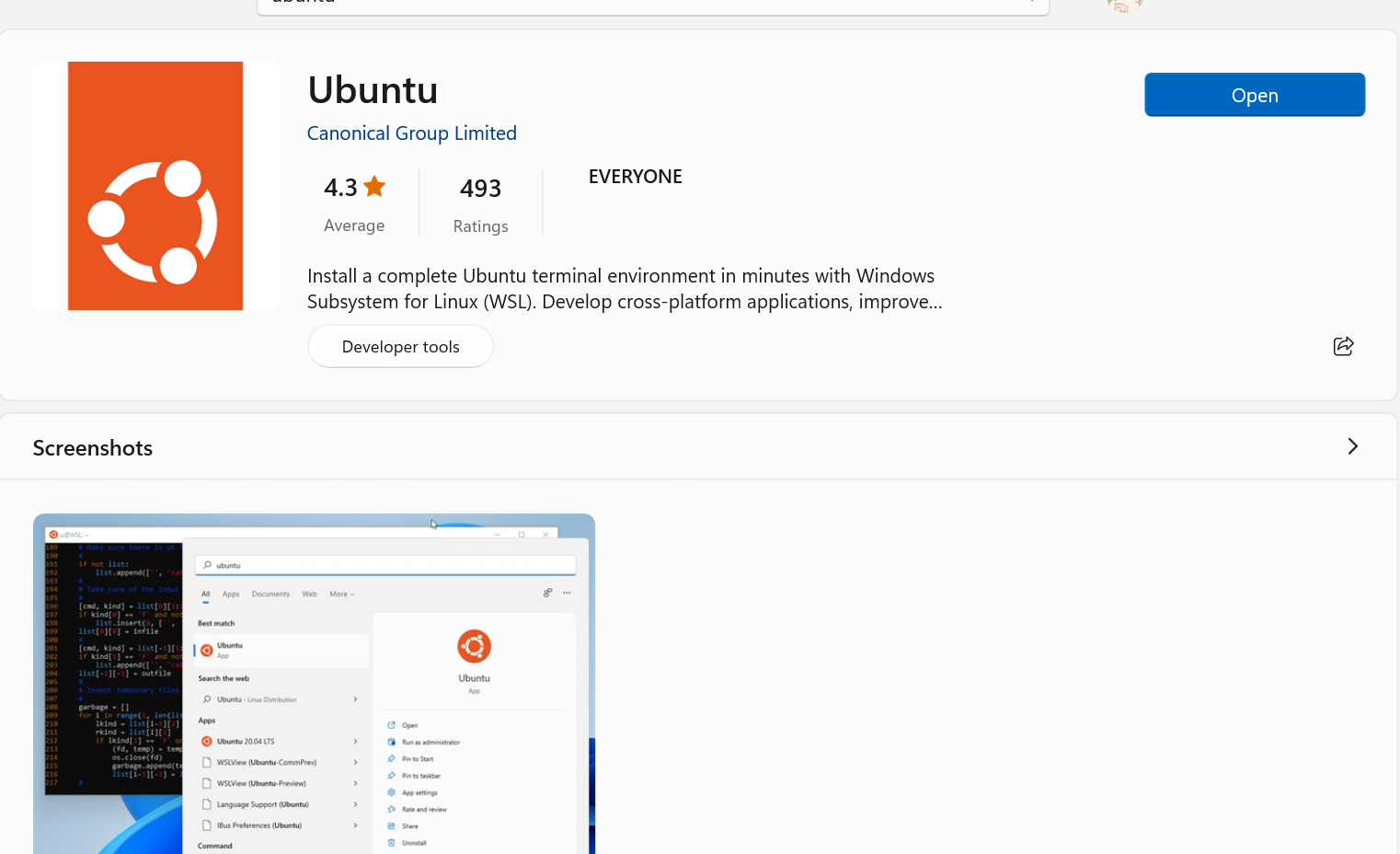
**wsl –install**

Dòng lệnh này sẽ enable tất cả các tính năng cần thiết để chạy WSL và cài đặt bản phân phối Ubuntu mặc định của Linux.



1. **Cài đặt Ubuntu**

Mở Microsoft Store để cài Ubuntu như hình dưới:

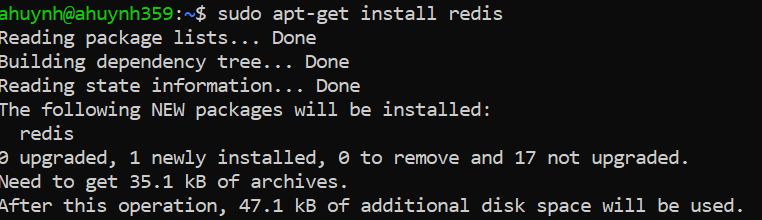


1. **Cài đặt Redis trên Ubuntu**

* Chạy lệnh sau:

**sudo apt-get update**

**sudo apt-get install redis**



* Chạy lệnh sau để start Redis server:

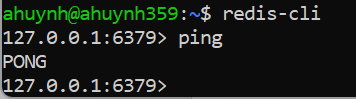
**sudo service redis-server start**

* Kiểm tra nếu redis server đang run:

**redis-cli**

**127.0.0.1:6379> ping**

**PONG**



1. **Cài đặt MongoDB trên Ubuntu**

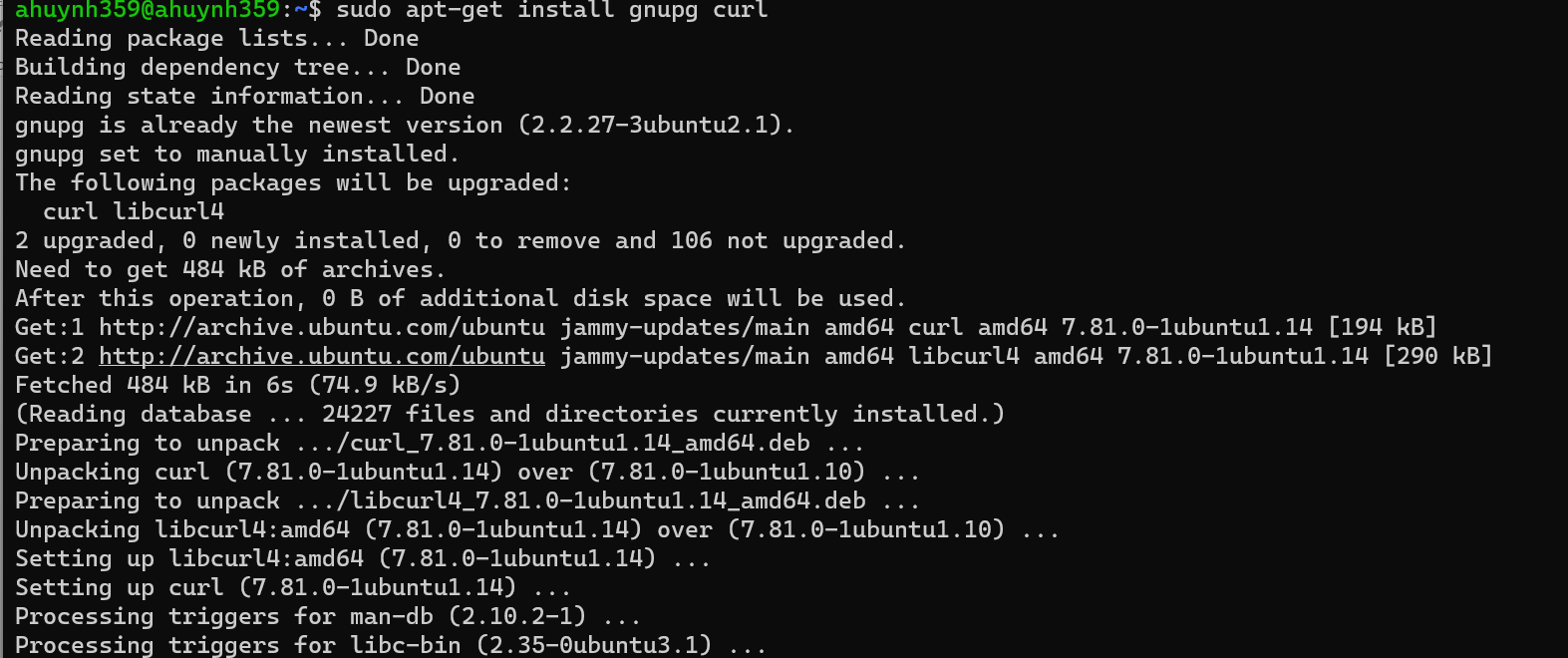
* Cài đặt gnupg và curl nếu nó chưa có:

+ GnuPG là viết tắt của "GNU Privacy Guard," là một chương trình mã nguồn mở dùng để mã hóa và ký số thông tin. Nó cung cấp một cơ sở hạ tầng an ninh cho việc gửi và nhận email, file, và các thông tin khác.

+ curl là một công cụ dòng lệnh và thư viện truyền tải dữ liệu với các URL. Nó hỗ trợ một loạt các giao thức truyền tải dữ liệu, bao gồm HTTP, HTTPS, FTP, FTPS, SCP, SFTP, LDAP và nhiều giao thức khác.

Sử dụng lệnh sau để cài đặt:

**sudo apt-get install gnupg curl**

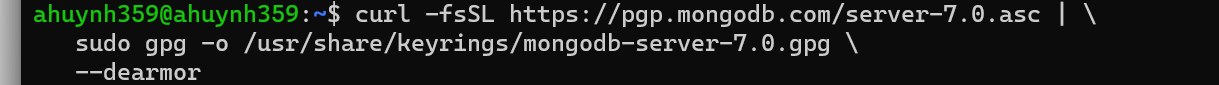


* Import MongoDB public GPG key từ trang web <https://pgp.mongodb.com/server-7.0.asc>

**curl -fsSL https://pgp.mongodb.com/server-7.0.asc | \**

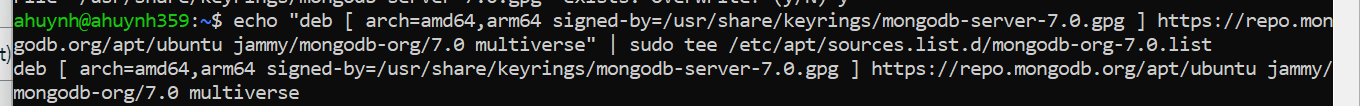
**sudo gpg -o /usr/share/keyrings/mongodb-server-7.0.gpg \**

**--dearmor**



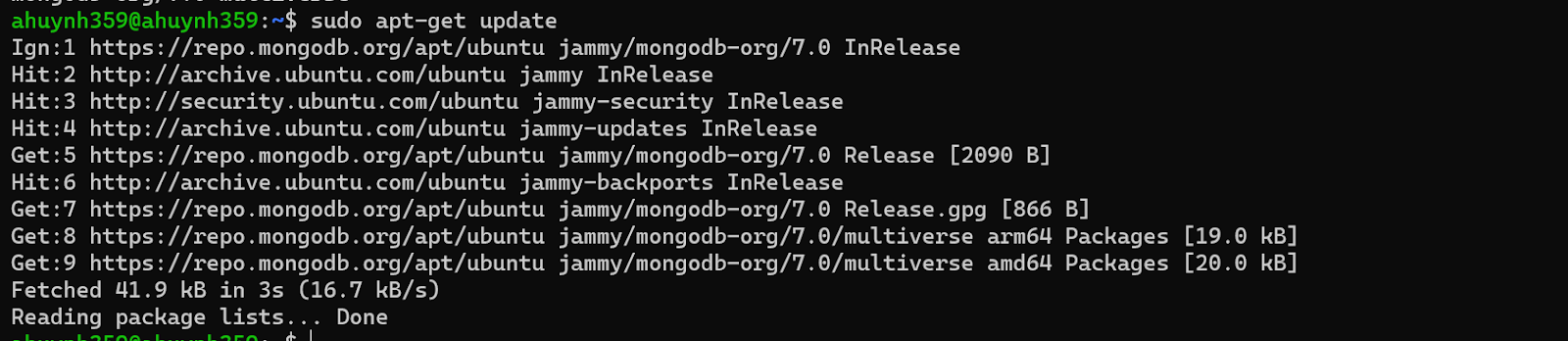
* Tạo list file cho MongoDB:

**echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg ] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-7.0.list**



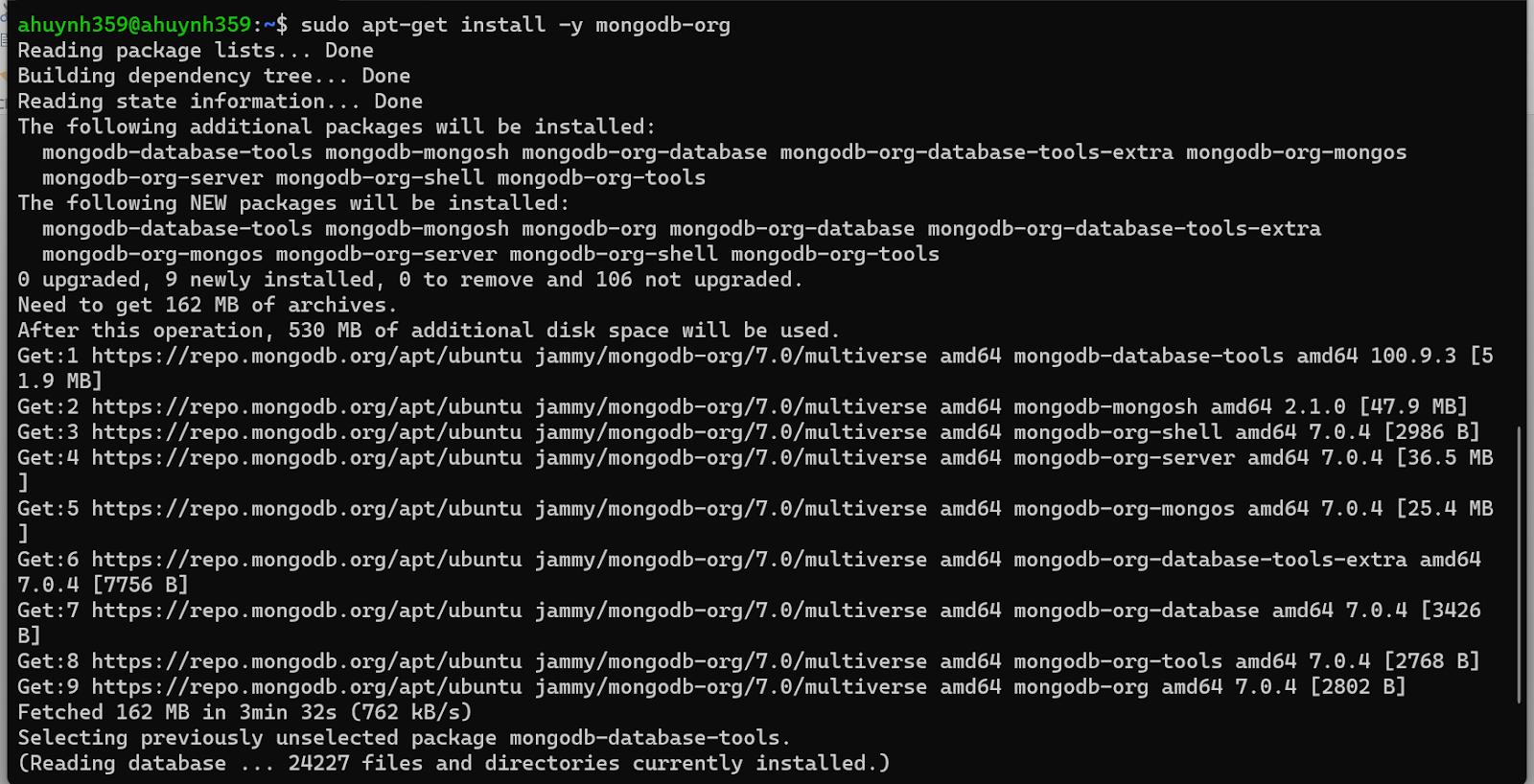
* Reload local package database:

**sudo apt-get update**



* Cài đặt MongoDB:

**sudo apt-get install -y mongodb-org**



* Start MongoDB:

**sudo systemctl start mongod**

* Xác nhận MongoDB start thành công:

**sudo systemctl status mongod**

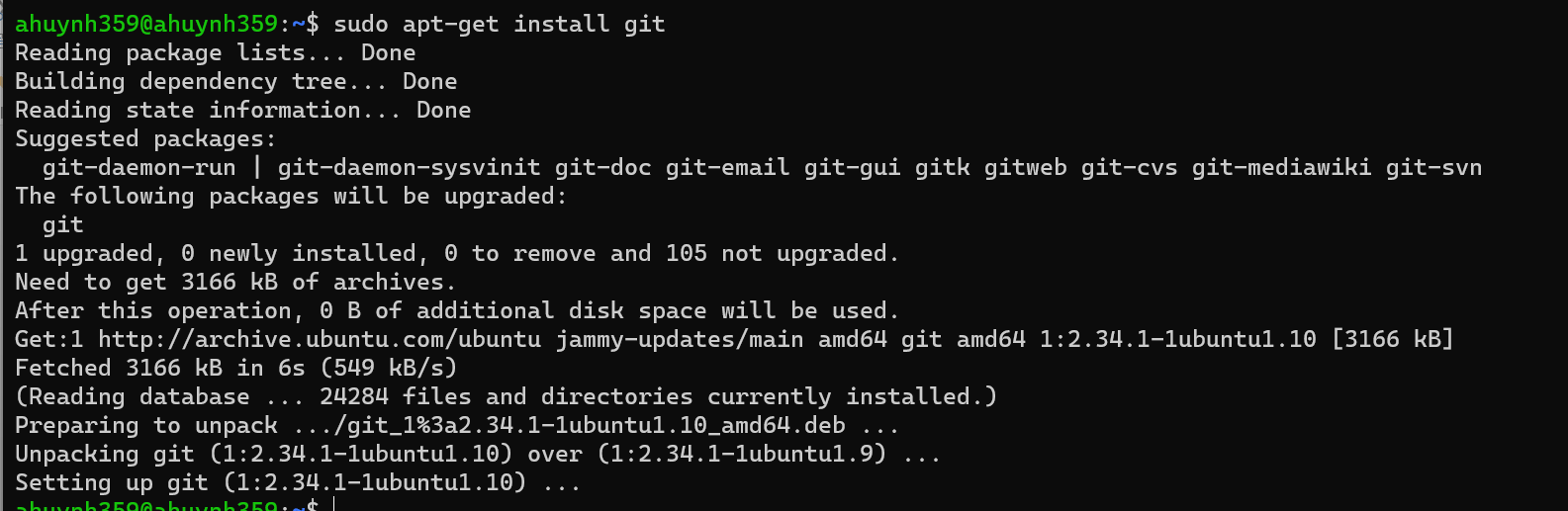


* Stop MongoDB:

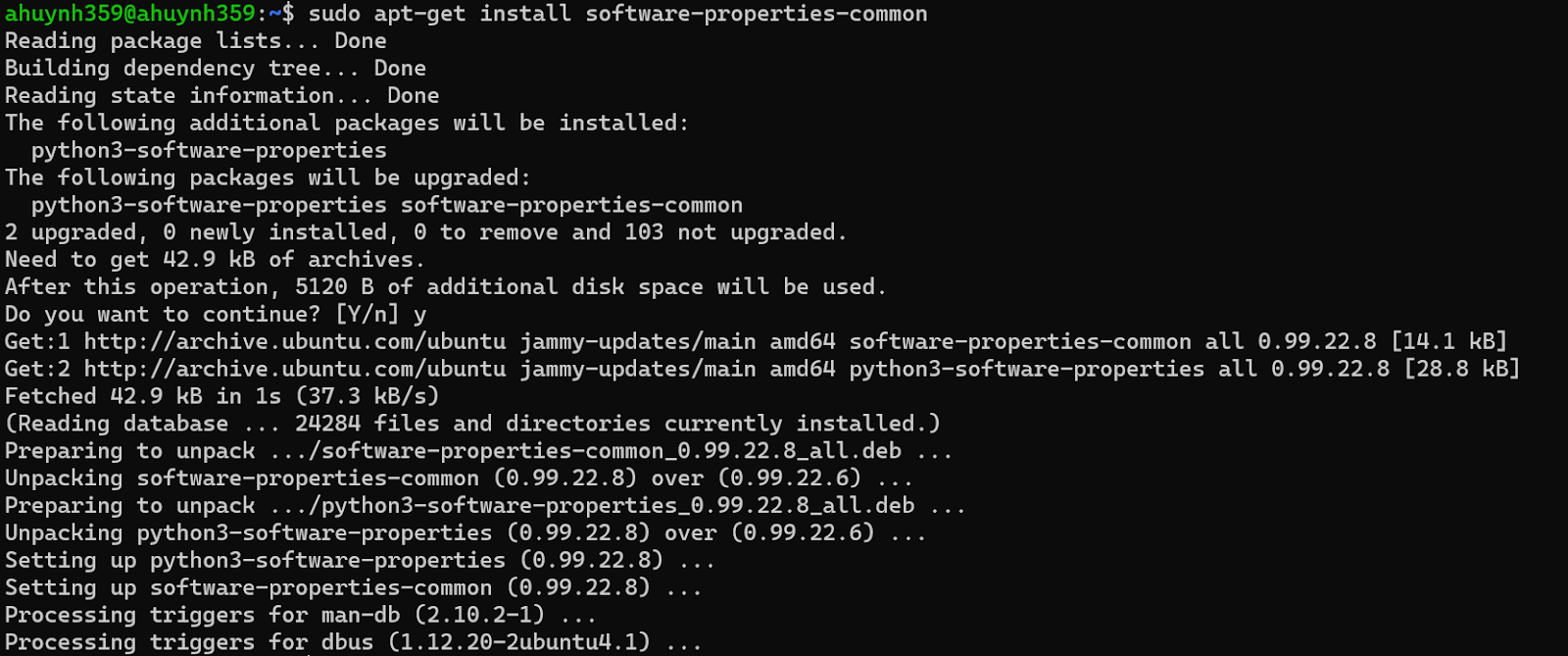
**sudo systemctl stop mongod**

1. **Cài đặt git và software-properties-common**

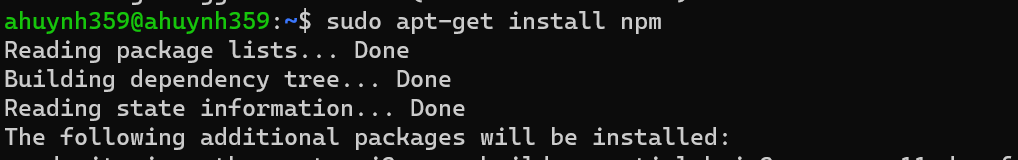
**sudo apt-get install git**



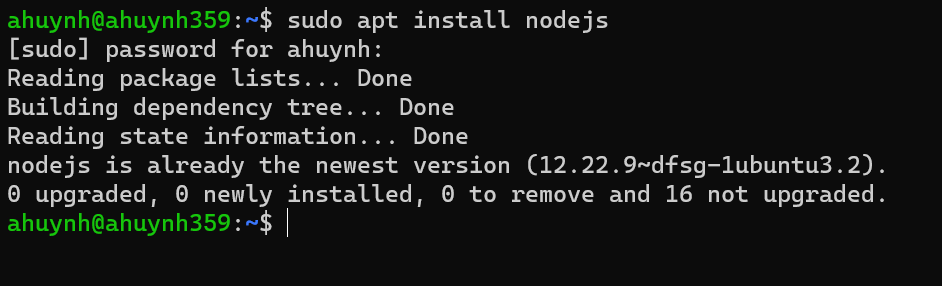
**sudo apt-get install software-properties-common**



**sudo apt-get install npm**



**sudo apt install nodejs**



Sau đó, update NodeJs sang version 12

Chúng ta sẽ sử dụng curl để cài đặt NVM qua command.

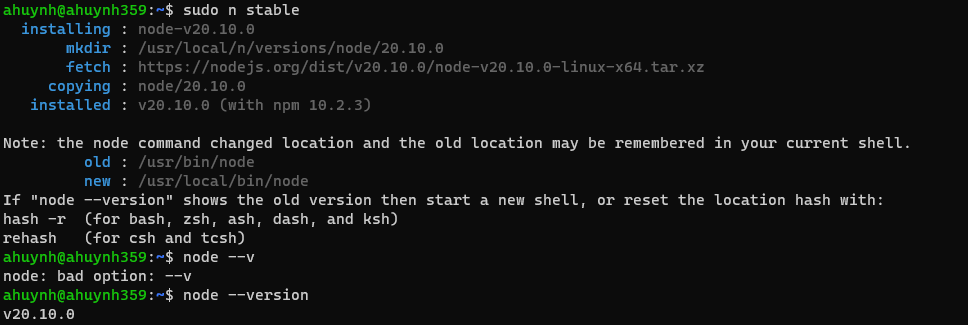
**curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash**

Trước khi bắt đầu với phiên bản mới của NodeJS, ta cần xóa các bộ nhớ đệm được liên kết với NPM.

**sudo npm cache clean -f**

https://lh7-us.googleusercontent.com/yidW6SSENYEFQj6XOlhEtCT1cHQqmdGsexX8c1FQtjxWbZqG-IFkYKOqzQdmksgVb1361RA3psKOzFHn3BYNJFgQm4GIJU7t93lNlrXrf3a1Jy7NDZ0cWwxZQskFLUvNNUofbnGJDCrk

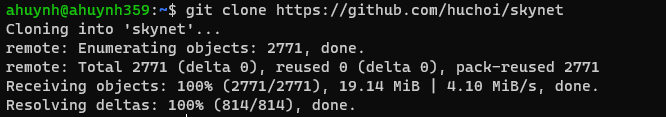
**sudo npm install -g n**



1. **Cài đặt skynet**

* Clone project skynet trên github:

**git clone https://github.com/huchoi/skynet**



* Thiết lập registry (kho lưu trữ) mặc định cho npm, trình quản lý gói của Node.js:

**npm config set registry** [**https://registry.npmjs.org/**](http://registry.npmjs.org/)

https://lh7-us.googleusercontent.com/_8ygEISluI7wy0r9fIUeKQbSBhsOT-Crjbex1NimQF2w3gHx7oV0fgpV7obnMOPnf18tSMq0hiEwlzx5w-xu6SFMGPYH9_fYi9OX1kRNtP28JEj5U8O9hOHgz4HMZLC4LbYyZEjLLU0z_uYtAemQdQ

* Di chuyển vào thư mục skynet cài đặt npm:

**cd skynet**

**npm install**

https://lh7-us.googleusercontent.com/9-rbjh10gSMuU9JNg5kv1YL6TIirGLBtKnYx4Vf7-NJCWzr0oaeeIGaJ-f-INM-0aW2pcKvmuAk1vcve9_wQuMl5pMLeePpgI4--6kP0LcVO3_bfAdQCsHJ7_cVZY9xtfJ0Re-eCJup1

1. **Chạy Project:**

* Update phiên bản NodeJS lên 14:

**cp config.js.sample config.js**

* Cấu hình lại file config.js như sau:

module.exports = {

  databaseUrl: "mongodb://localhost:27017/skynet",

  port: 3000,

  log: true,

  rateLimit: 10,

  redisHost: "127.0.0.1",

  redisPort: "6379"

}

* Start MongoDB và Redis:

**sudo systemctl start mongod**

**sudo service redis-server start**

* Sửa code trong file skynet.im thành **localhost:3000**

# **PHẦN 3: DEMO**

* **Gõ lệnh sau để return the current status of the SkyNet network:**

curl -X GET <http://localhost:3000/status>

* **Thêm một thiết bị:**

Đăng ký một thiết bị trên SkyNet network.

Ta có thể thêm 1 hoặc nhiều thuộc tính cho các đối tượng thiết bị.

SkyNet sẽ trả về UUID và token của mỗi thiết bị. Chúng ta sẽ có thể cần để sử dụng trong việc update các thiết bị sau này.

curl -X POST -d "name=drone&token=mytoken&color=yellow" <http://localhost:3000/devices>

https://lh7-us.googleusercontent.com/tUKqH1knqV0IwGN4niIwFQP77m7disScr22zxMnDcOmgjOGHZqMl6E96TrZAc9U7lJnfpx7VfIbzdyPfVdmfuLwpYTEAAk_vZBT0CqCwxImG_7yqjotufG7Jh2Wopnooecrr5yIlNnigmalYlfd83lE

* **Lấy thông tin thiết bị:**

Trả về tất cả các thông tin của thiết bị bằng UUID của nó.

curl <http://localhost:3000/devices/730cfcc0-ab6d-11ee-a4f3-d721ac356c20>

* **Thay đổi thông tin thiết bị:**

Cập nhật 1 đối tượng thiết bị, để bảo mật ta sẽ phải nhập token.

curl -X PUT -d "token=mytoken&color=red" <http://localhost:3000/devices/730cfcc0-ab6d-11ee-a4f3-d721ac356c20>

* **Lấy 10 sự kiện gần nhất cho thiết bị:**

Trả về 10 sự kiện gần nhất có liên quan tới thiết bị hoặc node.

curl -X GET [http://localhost:3000/events/](http://localhost:3000/events/b9b90cf0-8cd1-11ee-9e70-c13b613268f9/token=mytoken)730cfcc0-ab6d-11ee-a4f3-d721ac356c20?[token=mytoken](http://localhost:3000/events/b9b90cf0-8cd1-11ee-9e70-c13b613268f9/token=mytoken)

Event code

* 100 = Web socket connected
* 101 = Web socket identification
* 200 = System status API call
* 201 = Get events
* 202 =
* 203 =
* 300 = Incoming message
* 301 = Incoming SMS message
* 302 = Outgoung SMS message
* 400 = Register device
* 401 = Update device
* 402 = Delete device
* 403 = Query devices
* 500 = WhoAmI
* 600 = Gateway Config API call
* **Xóa thiết bị:**

Hủy đăng ký một thiết bị trên SkyNet network, để bảo mật ta phải nhập token.

curl -X DELETE -d "token=mytoken" <http://localhost:3000/devices/730cfcc0-ab6d-11ee-a4f3-d721ac356c20>

* **Trả về UUID**

Trả về tất cả thông tin (bao gồm token) của tất cả các thiết bị hoặc node phụ thuộc vào UUID của người dùng.

curl -X GET http://localhost:3000/authenticate/9ed17ff1-ab74-11ee-80f5-bd9e25d2b5b9?token=123

* **IP Adress:**

Trả về địa chỉ IP public của yêu cầu. Cái này có thể dùng khi làm việc với Skynet network sau tường lửa.

curl -X GET <http://localhost:3000/ipaddress>

* **Send message:**

Gửi một JSON message tới tất cả thiết bị hoặc một chuỗi thiết bị hoặc một thiết bị nào đó trên Skynet network.

curl -X POST -d '{"devices": "b86424a1-ab78-11ee-80f5-bd9e25d2b5b9", "message": {"red":"off"}}' http://localhost:3000/messages

# **PHẦN 4: TỔNG KẾT**

* + - Đáp ứng được các yêu cầu của đề tài.
    - Nguồn tài liệu để tìm hiểu và nghiên cứu không nhiều nên dẫn tới nhiều khó khăn và thách thức trong quá trình thực hiện đề tài.
    - Trong quá trình thực hiện đề tài, nhóm đã cố gắng tập trung tìm hiểu và tham khảo các tài liệu liên quan. Tuy nhiên, với thời gian và trình độ còn hạn chế, nên chúng em không thể tránh khỏi những thiếu sót. Nhóm rất mong nhận được các nhận xét và góp ý của thầy để chúng em có thể hoàn thiện hơn các kết quả nghiên cứu.

**TÀI LIỆU THAM KHẢO**

1. [Dr. Pankaj Dadhich](https://www.blogger.com/profile/06752400709546633350), [*‘Skynet in IoT’*](https://www.drpankajdadhich.com/2022/12/skynet-in-iot.html), Dr. [Pankaj Dadhich](https://www.blogger.com/profile/06752400709546633350), 09/12/2022.
2. Ubuntu, [*‘Install Ubuntu on WSL2 and get started with graphical applications’*](https://ubuntu.com/tutorials/install-ubuntu-on-wsl2-on-windows-11-with-gui-support#1-overview), Ubuntu.
3. Redis, [*‘Introduction to Redis’*](https://redis.io/docs/about/), Redis.
4. Redis, [*‘Install Redis on Windows’*](https://redis.io/docs/install/install-redis/install-redis-on-windows/), Redis.
5. MongoDB, [*‘Install MongoDB Community Edition on Ubuntu’*](https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/), MongoDB.
6. Brian Boucheron, [*‘How To Install Node.js on Ubuntu 20.04’*](https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-ubuntu-20-04), Digital Ocean, 24/01/2023.