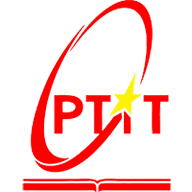
**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH**

****

**BÁO CÁO MÔN HỌC**

**IOT VÀ ỨNG DỤNG**

**ĐỀ TÀI 7: TRIỂN KHAI SKYNET IOT MESSAGING**

**Giảng viên hướng dẫn : ThS. ĐÀM MINH LỊNH**

**Sinh viên thực hiện 1 : VŨ HOÀNG MẠNH – N20DCCN001**

**Sinh viên thực hiện 2 : PHẠM KHÁNH BĂNG – N20DCCN007**

**Lớp : D20CQCNPM01-N**

**Khóa : 2020 - 2025**

**TP. Hồ Chí Minh ngày 5 tháng 1 năm 2024**

**LỜI CẢM ƠN**

Lời đầu tiên cho phép em xin được gửi lời cảm ơn chân thành đến Trường Học viện Công nghệ Bưu chính Viễn thông đã đưa môn học IoT và ứng dụng vào chương trình giảng dạy. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn – ThS. Đàm Minh Lịnh đã dạy dỗ, truyền đạt những kiến thức quý báu cũng như những kinh nghiệm của bản thân, đồng thời đưa ra những nhận xét đánh giá và góp ý cho những bài thuyết trình nhóm trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học của thầy, em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc trong việc làm thế nào để xây dựng cấu trúc của một bài nghiên cứu khoa học, cũng như các phương pháp vận dụng các cơ sở lý luận và luận cứ thực tế vào nghiên cứu. Đây chắc chắn sẽ là những kiến thức quý báu để em có thể cải thiện, nâng cao vốn kiến thức tự có của bản thân, tự tin trước những đề tài nghiên cứu khoa học trong tương lai.

Bộ môn IoT và ứng dụng là môn học thú vị, bổ ích và có tính thực tiễn cao khi giúp sinh viên tham gia môn học này có được sự tư duy logic nhanh nhẹn, vận dụng kết hợp các kiến thức đã học một cách hợp lý. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và chưa có nhiều trải nghiệm thực tế nên bài tiểu luận của em còn gặp nhiều sai sót khó tránh khỏi. Em rất mong được thầy xem xét và đánh giá bài tiểu luận của em, đồng thời đưa ra những góp ý giúp em có thể tránh những lỗi sai cơ bản, không đáng có để em có thể thực hiện tốt hơn trong tương lai.

Em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 5 tháng 1 năm 2023

Nhóm sinh viên thực hiện

Vũ Hoàng Mạnh – Phạm Khánh Băng

**MỤC LỤC**

# CHƯƠNG 1: TỔNG QUAN VỀ SKYNET

## **1.1. Khái niệm**

Skynet IoT là một hệ thống IoT (Internet of Things) được phát triển để quản lý và kết nối các thiết bị thông minh. Nó cho phép các thiết bị này giao tiếp với nhau và với internet để thu thập dữ liệu, chia sẻ thông tin và thực hiện các chức năng tự động.

Skynet IoT thường được sử dụng để tạo ra các môi trường thông minh, từ nhà thông minh đến các hệ thống quản lý công nghiệp và thành phố thông minh. Nó giúp tối ưu hóa việc quản lý, giám sát và điều khiển các thiết bị từ xa thông qua internet, mang lại tiện ích và hiệu quả trong việc quản lý các mạng thiết bị kết nối.

Skynet được thiết kế để kết hợp các chức năng của trí tuệ nhân tạo, IoT và công nghệ sổ kế toán phân phối của công nghệ blockchain. Skynet nhằm mục tiêu mang những công nghệ của tương lai đến với cuộc sống của chúng ta, trong đó chúng ta sẽ sống trong một thế giới thông minh, nơi tất cả các thiết bị có thể tự học hỏi, giao tiếp với nhau mà không cần sự can thiệp của con người.

Trước đây, IOT và trí tuệ nhân tạo hoạt động độc lập với cơ chế tập trung, tuy nhiên với sự xuất hiện của blockchain,đó chìa khóa để phân cấp và kết hợp các ngành công nghiệp này.

## **1.2. Các tính năng của SkyNet**

Skynet IoT là một hệ thống quản lý IoT (Internet of Things) mạnh mẽ và linh hoạt. Nó được thiết kế để kết nối và điều khiển các thiết bị thông minh từ xa thông qua internet. Điều này cho phép bạn quản lý, giám sát và tương tác với các thiết bị thông minh từ bất kỳ đâu, chỉ cần có kết nối internet.

Các tính năng chính của Skynet IoT bao gồm:

* Kết nối đa dạng thiết bị: Skynet IoT hỗ trợ nhiều loại thiết bị khác nhau, từ cảm biến đến máy móc công nghiệp và thiết bị gia đình thông minh như đèn, máy lạnh, camera, v.v.
* Thu thập và phân tích dữ liệu: Hệ thống này có khả năng thu thập dữ liệu từ các thiết bị, sau đó phân tích và hiển thị thông tin một cách trực quan để người dùng có thể hiểu rõ hơn về hoạt động của các thiết bị.
* Điều khiển từ xa: Người dùng có thể điều khiển các thiết bị từ xa thông qua giao diện của Skynet IoT trên điện thoại di động, máy tính hoặc thiết bị kết nối internet khác.
* Tự động hóa: Skynet IoT cung cấp khả năng tự động hóa các hoạt động dựa trên dữ liệu thu thập được. Ví dụ, bạn có thể lập lịch bật/tắt đèn tự động hoặc điều chỉnh nhiệt độ trong nhà dựa trên các thông số cảm biến.
* Bảo mật: Hệ thống này cũng chú trọng đến vấn đề bảo mật thông tin, đảm bảo rằng dữ liệu và điều khiển các thiết bị chỉ được truy cập bởi những người được ủy quyền.

Skynet IoT mang lại tiện ích lớn trong việc quản lý thông tin và điều khiển các thiết bị từ xa, giúp tối ưu hóa hoạt động và tiết kiệm thời gian cho người sử dụng.

## **1.3. Ưu thế của SkyNet**

Skynet là giao thức mang lại sự đổi mới, Opensingularity sẽ tập hợp tất cả các đề xuất thiết kế vào Skynet để có thể kết nối tất cả các thiết bị trong mạng. Để khắc phục các hạn chế trong phần cứng thiết bị IoT và để giải quyết vấn đề với việc chấp nhận tiền điện tử, Skynet được bao bọc bởi lõi chip lõi xử lý thần kinh của Skynet, có thể sửa đổi việc sử dụng mã thông báo mật mã và loại bỏ các hạn chế của phần cứng IoT. Ngoài ra, Skynet còn bao gồm một mạng blockchain vô hạn gọi là SON cung cấp ứng dụng cần thiết cho các lõi để giao tiếp và tương tác chặt chẽ trong mạng.

# CHƯƠNG 2. CÔNG NGHỆ SỬ DỤNG

## **2.1. Ubuntu**

a. Khái niệm

Ubuntu là một hệ điều hành trên máy tính, và nó được phát triển dựa trên Linux/Debian GNU. Lần đầu được giới thiệu vào năm 2004, tính đến năm 2007, Ubuntu đây là phiên bản chiếm 30% số bản tùy biến của Linux được cài đặt trên máy tính, và cũng là bản tuỳ biến Linux phổ biến nhất. Công ty Canonical đã và đang chịu trách nhiệm trong việc tài trợ Ubuntu, giúp cho hệ điều hành này có thể phát triển trong tương lai.

Có ba loại phiên bản Ubuntu mà người dùng cần biết đến, bao gồm Ubuntu phiên bản thông thường, phiên bản hỗ trợ lâu dài (LTS) và các dự án khác. Thông thường, bạn nên sử dụng phiên bản hỗ trợ lâu dài để nhận được hỗ trợ trong vòng 3 năm đối với máy tính để bàn và 5 năm đối với máy chủ.

b. Một số tính năng chính của Ubuntu

* Thừa hưởng tính năng nổi bật của Linux:

Vì đây là phiên bản hệ điều hành phát triển dựa trên Linux, vậy nên sẽ thừa hưởng được các tính năng đặc biệt của Linux. Chẳng hạn như khả năng tùy biến hiệu suất làm việc, tốc độ, cũng như khả năng bảo mật trước sự tấn công của [malware](https://www.thegioididong.com/hoi-dap/malware-la-gi-tan-cong-phat-tan-malware-la-hinh-thuc-tan-1314306#hmenuid1) hay virus.

* Hỗ trợ người dùng trong việc cài đặt:

Để kiểm tra sự tương thích của các phiên bản với máy tính, bạn có thể sử dụng đĩa chạy trực tiếp phiên bản đó của hệ điều hành, từ đó người dùng sẽ quyết định xem có thể cài đặt trên thiết bị của họ hay không thông qua phần mềm Ubiquity.

Ngoài ra, người dùng sử dụng Windows có thể phân vùng lại đĩa cứng để cài đặt Ubuntu, và bạn cũng có thể gỡ bỏ nó khá dễ dàng.ngay cả khi đang sử dụng Windows.

* Giao diện:

Ubuntu hiện đang sử dụng giao diện đồ hoạ Unity, và giao diện này được thiết kế ra với mục đích tối ưu diện tích màn hình sử dụng, đồng thời cũng mang lại trải nghiệm dùng thân thiện.

* Ứng dụng:

Hệ điều hành Ubuntu đã được cài sẵn các phần mềm [mã nguồn mở](https://www.thegioididong.com/hoi-dap/ma-nguon-mo-la-gi-nhung-dieu-ban-chua-biet-ve-ma-nguon-mo-1319225#hmenuid1), chẳng hạn như trình duyệt Internet Firefox, bộ ứng dụng văn phòng đến từ LibreOffice (phiên bản Ubuntu 10.04 trở đi) hay trình tải file [torrent](https://www.thegioididong.com/hoi-dap/torrent-la-gi-nhung-dieu-can-biet-ve-torrent-truoc-khi-su-1307154#hmenuid1) Tranmission. Ngoài ra, hệ điều hành này cũng có một kho các ứng dụng miễn phí có tên là Ubuntu Software Center.

c. Ưu điểm của hệ điều hành Ubuntu

* Không yêu cầu cấu hình mạnh

Thông thường, quá trình cài đặt Ubuntu trên máy tính sẽ giao động từ 20 - 30 phút, tuỳ theo cấu hình của máy, và bạn chỉ cần có thiết bị có dung lượng RAM từ 256MB trở lên. Và cấu hình khuyên dùng thì chỉ cần RAM trên 384MB, và CPU là 700 MHz (x86).

* Có nhiều ứng dụng miễn phí

Bạn không cần tốn quá nhiều chi phí cho các ứng dụng, và chúng đều có mặt trên Ubuntu Software Center. Chẳng hạn như bộ ứng dụng văn phòng LibreOffice miễn phí cũng đã được cài sẵn trên hệ điều hành Ubuntu.

* Khả năng tương tác sâu

Mặc dù đôi khi phải sử dụng các câu lệnh để tương tác với thiết bị, nhưng nhờ vậy mà người dùng có thể điều khiển phần mềm một cách sâu hơn, từ đó hiệu quả công việc sẽ tốt hơn, nhất là khi so sánh với các giao diện đồ họa. Đồng thời, đây là hệ điều hành mã nguồn mở, thế nên Ubuntu có độ tùy biến cao.

* Có tính bảo mật cao

Bạn không cần lo lắng quá nhiều về việc có cần cài đặt các phần mềm diệt virus như trên Windows bởi vì số lượng malware hay virus khá ít trên hệ điều hành Ubuntu.

## **2.2. Redis**

a. Khái niệm

Ngày nay, khái niệm NoSQL trở nên không còn xa lạ trong giới Công Nghệ Thông Tin (CNTT). Đi kèm với đó là sự ra đời của hàng loạt hệ quản trị cơ sở dữ liệu (DBMS) phát triển dựa trên đặc thù của NoSQL: Non-relational (không quan hệ), Distributed (phân tán), Open-source (mã nguồn mở), Horizontally scalable (dễ dàng mở rộng theo chiều ngang).

Redis là cơ sở dữ liệu mang phong cách NoSQL, lưu trữ dữ liệu với dạng KEY-VALUE với nhiều tính năng được sử dụng rộng rãi. Nó có thể hỗ trợ nhiều kiểu dữ liệu như: strings, hashes, lists, sets, sorted. Đồng thời có thể cho phép scripting bằng ngôn ngữ Lua.

NoSQL là một khái niệm chỉ về một lớp các hệ cơ sở dữ liệu không sử dụng mô hình quan hệ (RDBMS). RDBMS vốn tồn tại khá nhiều nhược điểm như có hiệu năng không tốt nếu kết nối dữ liệu nhiều bảng lại hay khi dữ liệu trong một bảng là rất lớn.

Thuật ngữ NoSQL đánh dấu bước phát triển của thế hệ CSDL mới: phân tán (distributed) + không ràng buộc (non-relational).

b. Đặc chưng cơ bản của Redis

* Data model

Khác với RDMS như MySQL, hay PostgreSQL, Redis không có bảng. Redis lưu trữ data dưới dạng key-value. Thực tế thì memcache cũng làm vậy, nhưng kiểu dữ liệu của memcache bị hạn chế, không đa dạng được như Redis, do đó không hỗ trợ được nhiều thao tác từ phía người dùng. Dưới đây là sơ lược về các kiểu dữ liệu Redis dùng để lưu value.

* STRING: Có thể là string, integer hoặc float. Redis có thể làm việc với cả string, từng phần của string, cũng như tăng/giảm giá trị của integer, float.
* LIST: Danh sách liên kết của các strings. Redis hỗ trợ các thao tác push, pop từ cả 2 phía của list, trim dựa theo offset, đọc 1 hoặc nhiều items của list, tìm kiếm và xóa giá trị.
* SET Tập hợp các string (không được sắp xếp). Redis hỗ trợ các thao tác thêm, đọc, xóa từng phần tử, kiểm tra sự xuất hiện của phần tử trong tập hợp. Ngoài ra Redis còn hỗ trợ các phép toán tập hợp, gồm intersect/union/difference.
* HASH: Lưu trữ hash table của các cặp key-value, trong đó key được sắp xếp ngẫu nhiên, không theo thứ tự nào cả. Redis hỗ trợ các thao tác thêm, đọc, xóa từng phần tử, cũng như đọc tất cả giá trị.
* ZSET (sorted set): Là 1 danh sách, trong đó mỗi phần tử là map của 1 string (member) và 1 floating-point number (score), danh sách được sắp xếp theo score này. Redis hỗ trợ thao tác thêm, đọc, xóa từng phần tử, lấy ra các phần tử dựa theo range của score hoặc của string.
* Master/Slave Replication

Đây không phải là đặc trưng quá nổi bật, các DBMS khác đều có tính năng này, tuy nhiên chúng ta nêu ra ở đây để nhắc nhở rằng, Redis không kém cạnh các DBMS về tình năng Replication.

* In-memory

Không như các DBMS khác lưu trữ dữ liệu trên đĩa cứng, Redis lưu trữ dữ liệu trên RAM, và đương nhiên là thao tác đọc/ghi trên RAM. Với người làm CNTT bình thường, ai cũng hiểu thao tác trên RAM nhanh hơn nhiều so với trên ổ cứng, nhưng chắc chắn chúng ta sẽ có cùng câu hỏi: Điều gì xảy ra với data của chúng ta khi server bị tắt?

Rõ ràng là toàn bộ dữ liệu trên RAM sẽ bị mất khi tắt server, vậy làm thế nào để Redis bảo toàn data và vẫn duy trì được ưu thế xử lý dữ liệu trên RAM. Chúng ta sẽ cùng tìm hiểu về cơ chế lưu dữ liệu trên ổ cứng của Redis trong phần tiếp theo của bài viết.

* Persistent redis

Dù làm việc với data dạng key-value lưu trữ trên RAM, Redis vẫn cần lưu trữ dữ liệu trên ổ cứng. 1 là để đảm bảo toàn vẹn dữ liệu khi có sự cố xảy ra (server bị tắt nguồn) cũng như tái tạo lại dataset khi restart server, 2 là để gửi data đến các slave server, phục vụ cho tính năng replication. Redis cung cấp 2 phương thức chính cho việc sao lưu dữ liệu ra ổ cứng, đó là RDB và AOF. RDB (Redis DataBase file)

c. Cách thức làm việc

RDB thực hiện tạo và sao lưu snapshot của DB vào ổ cứng sau mỗi khoảng thời gian nhất định.

Ưu điểm

* RDB cho phép người dùng lưu các version khác nhau của DB, rất thuận tiện khi có sự cố xảy ra.
* Bằng việc lưu trữ data vào 1 file cố định, người dùng có thể dễ dàng chuyển data đến các data centers khác nhau, hoặc chuyển đến lưu trữ trên Amazon S3.
* RDB giúp tối ưu hóa hiệu năng của Redis. Tiến trình Redis chính sẽ chỉ làm các công việc trên RAM, bao gồm các thao tác cơ bản được yêu cầu từ phía client như thêm/đọc/xóa, trong khi đó 1 tiến trình con sẽ đảm nhiệm các thao tác disk I/O. Cách tổ chức này giúp tối đa hiệu năng của Redis. Khi restart server, dùng RDB làm việc với lượng data lớn sẽ có tốc độ cao hơn là dùng AOF.

Nhược điểm

* RDB không phải là lựa chọn tốt nếu bạn muốn giảm thiểu tối đa nguy cơ mất mát dữ liệu. Thông thường người dùng sẽ set up để tạo RDB snapshot 5 phút 1 lần (hoặc nhiều hơn). Do vậy, trong trường hợp có sự cố, Redis không thể hoạt động, dữ liệu trong những phút cuối sẽ bị mất.
* RDB cần dùng fork() để tạo tiến trình con phục vụ cho thao tác disk I/O. Trong trường hợp dữ liệu quá lớn, quá trình fork() có thể tốn thời gian và server sẽ không thể đáp ứng được request từ client trong vài milisecond hoặc thậm chí là 1 second tùy thuộc vào lượng data và hiệu năng CPU.

## **2.3. MongoDB**

a. Khái niệm

MongoDB là một database hướng tài liệu (document), một dạng NoSQL database. Vì thế, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON. [MongoDB](https://topdev.vn/viec-lam-it/mongodb-kt90) sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau. Các dữ liệu được lưu trữ trong document kiểu JSON nên truy vấn sẽ rất nhanh.

MongoDB lần đầu ra đời bởi MongoDB Inc., tại thời điểm đó là thế hệ 10, vào tháng Mười năm 2007, nó là một phần của sản phẩm PaaS (Platform as a Service) tương tự như Windows Azure và Google App Engine. Sau đó nó đã được chuyển thành nguồn mở từ năm 2009.

MongoDB đã trở thành một trong những NoSQL database nổi trội nhất bấy giờ, được dùng làm backend cho rất nhiều website như eBay, SourceForge và The New York Times.

Các feature của MongoDB gồm có:

* Các ad hoc query: hỗ trợ search bằng field, các phép search thông thường, regular expression searches, và range queries.
* Indexing: bất kì field nào trong BSON document cũng có thể được index.
* Replication: có ý nghĩa là “nhân bản”, là có một phiên bản giống hệt phiên bản đang tồn tại, đang sử dụng. Với cơ sở dữ liệu, nhu cầu lưu trữ lớn, đòi hỏi cơ sở dữ liệu toàn vẹn, không bị mất mát trước những sự cố ngoài dự đoán là rất cao. Vì vậy, người ta nghĩ ra khái niệm “nhân bản”, tạo một phiên bản cơ sở dữ liệu giống hệt cơ sở dữ liệu đang tồn tại, và lưu trữ ở một nơi khác, đề phòng có sự cố.
* Aggregation: Các Aggregation operation xử lý các bản ghi dữ liệu và trả về kết quả đã được tính toán. Các phép toán tập hợp nhóm các giá trị từ nhiều Document lại với nhau, và có thể thực hiện nhiều phép toán đa dạng trên dữ liệu đã được nhóm đó để trả về một kết quả duy nhất. Trong SQL, count(\*) và GROUP BY là tương đương với Aggregation trong MongoDB.
* Lưu trữ file: MongoDB được dùng như một hệ thống file tận dụng những function trên và hoạt động như một cách phân phối qua sharding.

b. Khi nào sử dụng MongoDB?

* Quản lý và truyền tải content – Quản lý đa dạng nhiều product của content chỉ trong một kho lưu trữ data cho phép thay đổi và phản hồi nhanh chóng mà không chịu thêm phức tạp thêm từ hệ thống content.
* Cấu trúc Mobile và Social – MongoDB cung cấp một platform có sẵn, phản xạ nhanh, và dễ mở rộng cho phép rất nhiều khả năng đột phá, phân tích real-time, và hỗ trợ toàn cầu.
* Quản lý data khách hàng – Tận dụng khả năng query nhanh chóng cho phân tích real-time trên cơ sở dữ liệu người dùng cực lớn vớ các mô hình data phức tạp bằng các schema linh hoạt và tự động sharding cho mở rộng chiều ngang.

c. Ưu điểm của MongoDB

* Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.
* Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

d. Nhược điểm của MongoDB

* Không ứng dụng được cho các mô hình giao dịch nào có yêu cầu độ chính xác cao do không có ràng buộc.
* Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng.
* Dữ liệu lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.
* Mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

## **2.4. NodeJS**

a. Khái niệm

Node.js là một mã nguồn mở, một môi trường cho các máy chủ và ứng dụng mạng. Node.js sử dụng Google V8 JavaScript engine để thực thi mã, và một tỷ lệ lớn các mô-đun cơ bản được viết bằng JavaScript. Các ứng dụng node.js thì được viết bằn JavaScript.

Node.js chứa một thư viện built-in cho phép các ứng dụng hoạt động như một Webserver mà không cần phần mềm như Nginx, Apache HTTP Server hoặc IIS. Node.js cung cấp kiến trúc hướng sự kiện (event-driven) và non-blocking I/O API, tối ưu hóa thông lượng của ứng dụng và có khả năng mở rộng cao. Mọi hàm trong Node.js là không đồng bộ (asynchronous). Do đó, các tác vụ đều được xử lý và thực thi ở chế độ nền (background processing)

b. Ứng dụng của NodeJS

* Xây dựng websocket server (Chat server)
* Hệ thống Notification (Giống như facebook hayTwitter)
* Ứng dụng upload file trên client
* Các máy chủ quảng cáo
* Các ứng dụng dữ liệu thời gian thực khác.

c. Ưu điểm NodeJS

* Đặc điểm nổi bật của Node.js là nó nhận và xử lý nhiều kết nối chỉ với một single-thread. Điều này giúp hệ thống tốn ít RAM nhất và chạy nhanh nhất khi không phải tạo thread mới cho mỗi truy vấn giống PHP. Ngoài ra, tận dụng ưu điểm non-blocking I/O của Javascript mà Node.js tận dụng tối đa tài nguyên của server mà không tạo ra độ trễ như PHP
* JSON APIs Với cơ chế event-driven, non-blocking I/O(Input/Output) và mô hình kết hợp với Javascript là sự lựa chọn tuyệt vời cho các dịch vụ Webs làm bằng JSON.
* Ứng dụng trên 1 trang( Single page Application) Nếu bạn định viết 1 ứng dụng thể hiện trên 1 trang (Gmail?) NodeJS rất phù hợp để làm. Với khả năng xử lý nhiều Request/s đồng thời thời gian phản hồi nhanh. Các ứng dụng bạn định viết không muốn nó tải lại trang, gồm rất nhiều request từ người dùng cần sự hoạt động nhanh để thể hiện sự chuyên nghiệp thì NodeJS sẽ là sự lựa chọn của bạn.
* Shelling tools unix NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra 1 luồng khiến cho hiệu xuất hoạt động đạt mức tối đa nhất và tuyệt vời nhất.
* Streamming Data (Luồng dữ liệu) Các web thông thường gửi HTTP request và nhận phản hồi lại (Luồng dữ liệu). Giả xử sẽ cần xử lý 1 luồng giữ liệu cực lớn, NodeJS sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.
* Ứng dụng Web thời gian thực Với sự ra đời của các ứng dụng di động & HTML 5 nên Node.js rất hiệu quả khi xây dựng những ứng dụng thời gian thực (real-time applications) như ứng dụng chat, các dịch vụ mạng xã hội như Facebook, Twitter,…

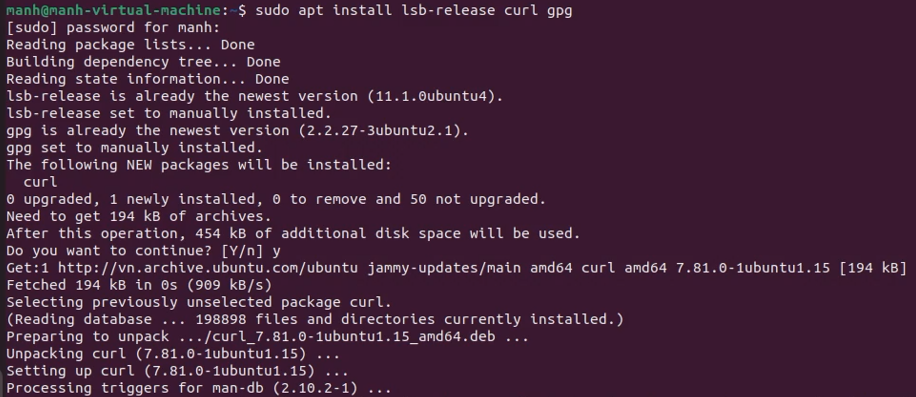
d. Nhược điểm NodeJS

* Ứng dụng nặng tốn tài nguyên Nếu bạn cần xử lý các ứng dụng tốn tài nguyên CPU như encoding video, convert file, decoding encryption… hoặc các ứng dụng tương tự như vậy thì không nên dùng NodeJS (Lý do: NodeJS được viết bằng C++ & Javascript, nên phải thông qua thêm 1 trình biên dịch của NodeJS sẽ lâu hơn 1 chút ). Trường hợp này bạn hãy viết 1 Addon C++ để tích hợp với NodeJS để tăng hiệu suất tối đa !
* NodeJS và ngôn ngữ khác NodeJS, PHP, Ruby, Python .NET …thì việc cuối cùng là phát triển các App Web. NodeJS mới sơ khai như các ngôn ngữ lập trình khác. Vậy nên bạn đừng hi vọng NodeJS sẽ không hơn PHP, Ruby, Python… ở thời điểm này. Nhưng với NodeJS bạn có thể có 1 ứng dụng như mong đợi, điều đó là chắc chắn!

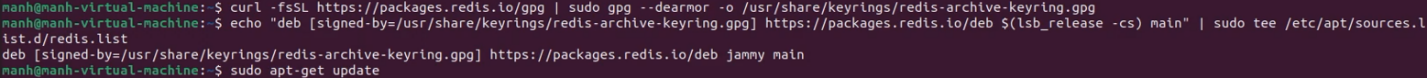
# CHƯƠNG 3. CÀI ĐẶT TRÊN UBUNTU

## **3.1. Cài đặt Redis**

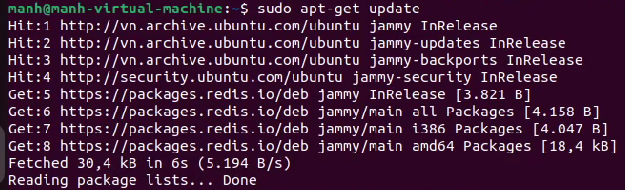
Thực hiện cài đặt lsb-release, curl và gpg trước:



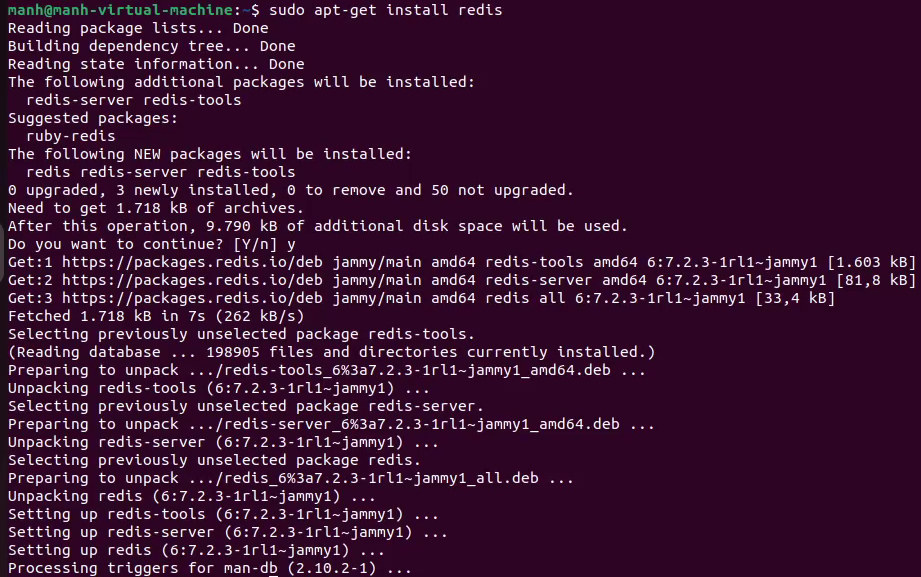
Thêm repository vào apt index:



Update apt:

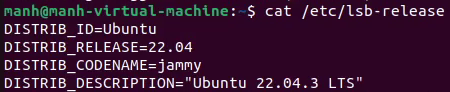


Cài đặt apt:

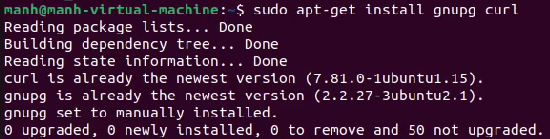


## **3.2. Cài đặt MongoDB**

Kiểm tra phiên bản Ubuntu (ở đây là 22.04):



Cài đặt gnupg và curl:



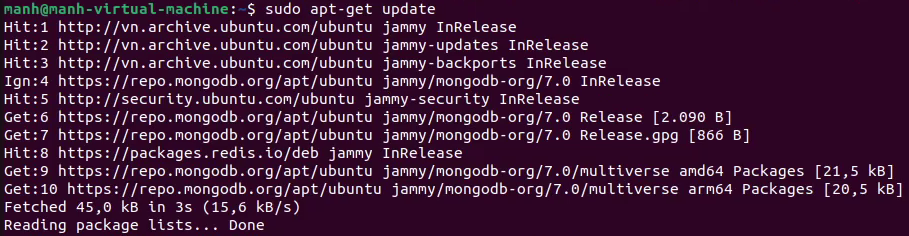
Import MongoDB public GPG key:



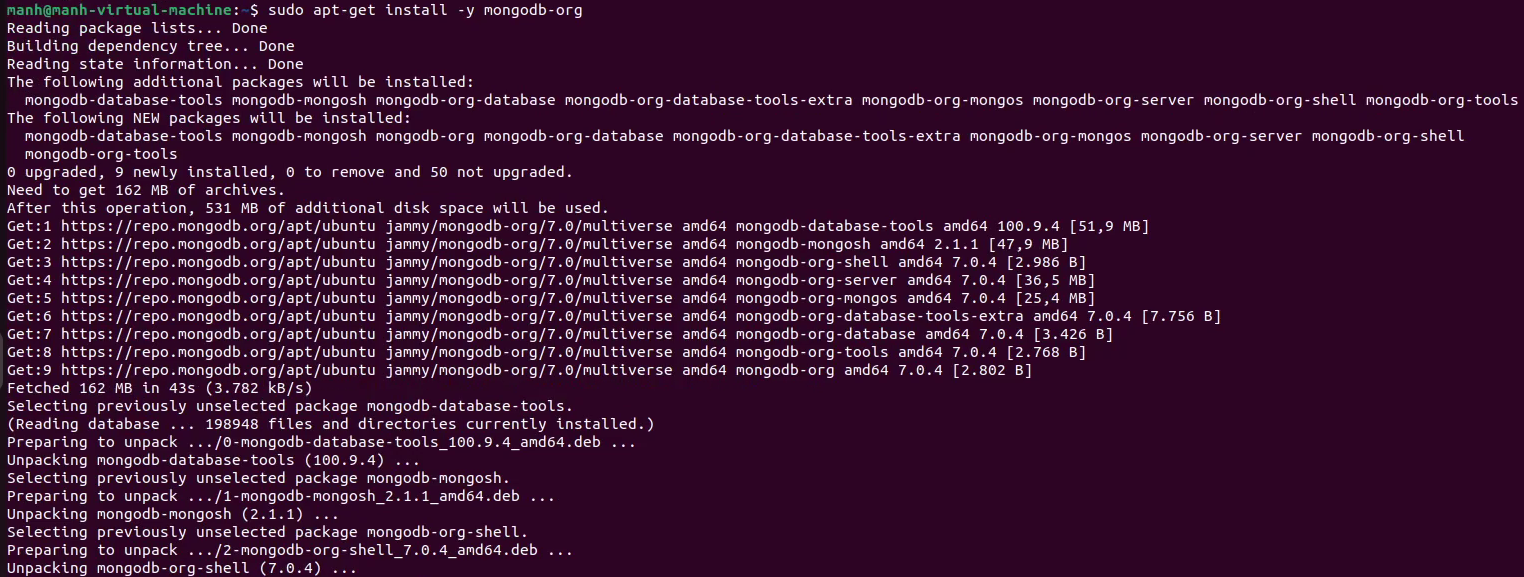
Tạo file /etc/apt/sources.list.d/mongodb-org-7.0.list cho Ubuntu 22.04:



Reload lại database local package:



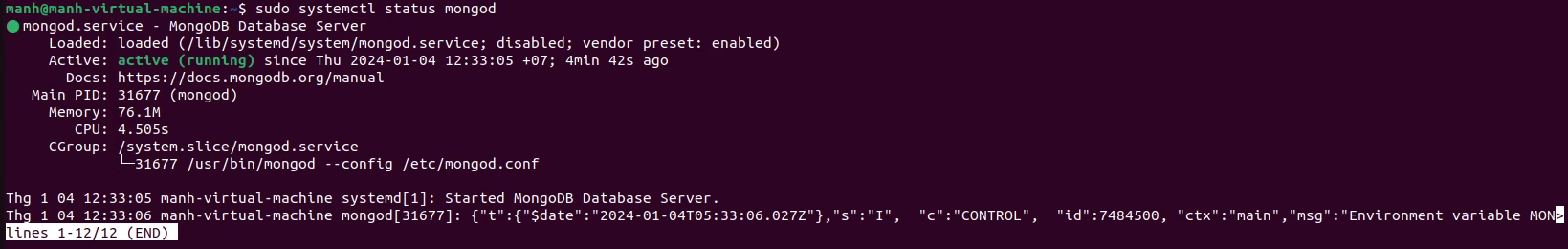
Cài đặt MongoDB packages:



Khởi động MongoDB:



Xác nhận MongoDB đã khởi động thành công:

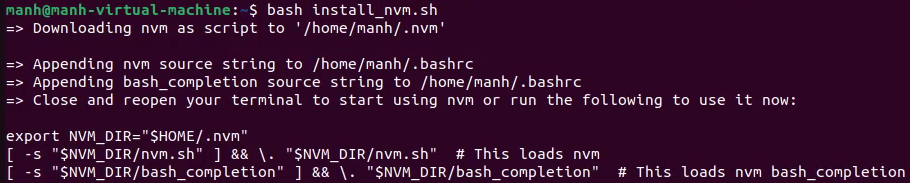


## **3.3. Cài đặt nvm**

Download install script:



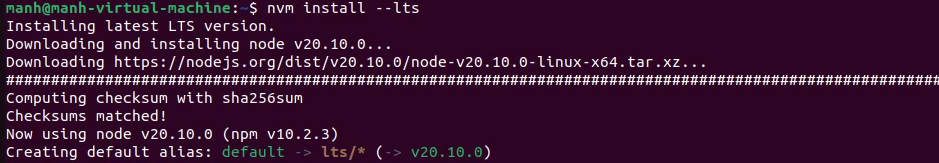
Chạy install script:



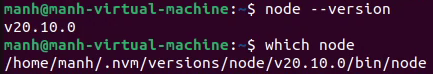
Xác nhận nvm hoạt động:



Cài đặt phiên bản LTS hiện tại của Node.js:

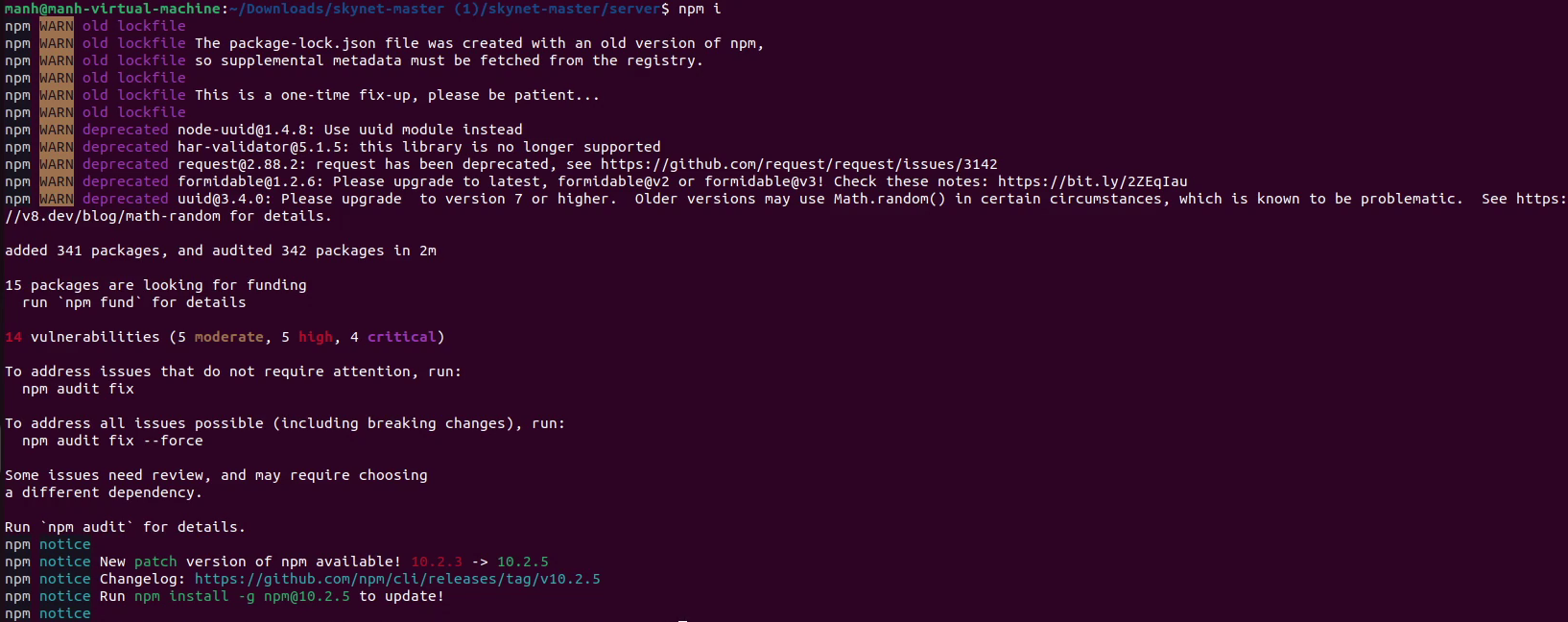


Xác nhận phiên bản của node:

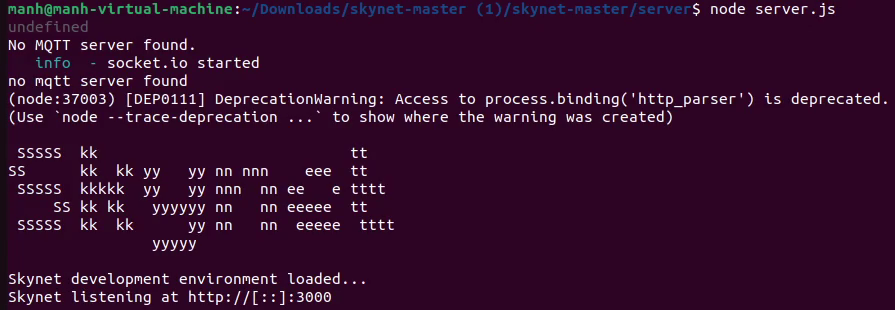


# CHƯƠNG 4. THỰC HIỆN NỘI DUNG

\Cài đặt node\_modules trong thư mục server:



Khởi động SkyNet server:



## **4.1. Thực hiện lấy status của SkyNet platform:**

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



Sau khi thực hiện câu lệnh, chương trình sẽ hiển thị các thông tin về status của SkyNet platform: skynet, timestamp và eventCode.

## **4.2. Thực hiện lấy danh sách các device dựa trên key/value**

curl -X GET <http://localhost:3000/devices?type=drone>



Sau khi thực hiện câu lệnh, chương trình sẽ hiển thị các device có key/value tương ứng (ở đây là type=drone):

{"devices":["74521970-aac9-11ee-8c4e-39f35f991d31","b4102751-aac9-11ee-8c4e-39f35f991d31","3e8e8770-ab22-11ee-bc12-d5fcbafcd168"]}

## **4.3. Thực hiện lấy tất cả thông tin (trừ token) của 1 device cụ thể**

curl -X GET <http://localhost:3000/devices>/74521970-aac9-11ee-8c4e-39f35f991d31



Chương trình sẽ trả về danh sách thông tin của device đang muốn tìm kiếm. Các thông tin được hiển thị bao gồm: \_id, name, type, color, ipAddress, uuid và timestamp.

## **4.4. Thực hiện đăng ký device trên mạng SkyNet**

curl -X POST -d "name=arduino&type=drone&color=black" <http://localhost:3000/devices>



Khi đăng ký device, chúng ta có thể thêm bao nhiêu thuộc tính vào đối tượng đều được. Như trên câu lệnh thì đang đăng ký 1 device mới với 3 thuộc tính là name=arduino, type=drone và color=black.

## **4.5. Thực hiện update thông tin của device**

curl -X PUT -d "token=123&color=blue&online=true" <http://localhost:3000/devices/74521970-aac9-11ee-8c4e-39f35f991d31>



Khi thực hiện lệnh update thì chúng ta cần nhập thêm cả token cho vấn đề bảo mật. Sau khi thực hiện thành công, chương trình sẽ trả về thông tin của thiết bị với các thông tin được update.

## **4.6. Thực hiện xóa 1 device**

curl -X DELETE -d "token=123 " http://localhost:3000/devices/74521970-aac9-11ee-8c4e-39f35f991d31



## **4.7. MyDevices**

curl -X GET http://localhost:3000/mydevices/022c4331-ab2c-11ee-869f-61687366b788?token=lfsifg5qn593nfr1r58gn1

Uuid: 022c4331-ab2c-11ee-869f-61687366b788

Token:lfsifg5qn593nfr1r58gn1

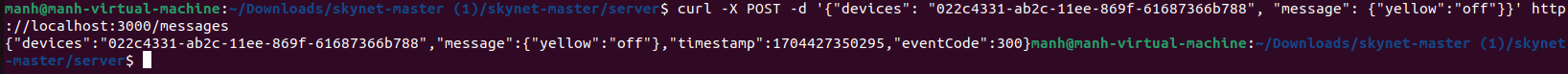
## **4.8. Thực hiện chức năng gửi tin nhắn**

a. Gửi tin nhắn cho tất cả các device

curl -X POST -d '{"devices": "\*", "message": {"yellow":"off"}}' [http://localhost:3000/messages](http://localhost:3000/messages)

b. Gửi tin nhắn cho 1 device cụ thể

curl -X POST -d '{"devices": "022c4331-ab2c-11ee-869f-61687366b788", "message": {"yellow":"off"}}' http://localhost:3000/messages



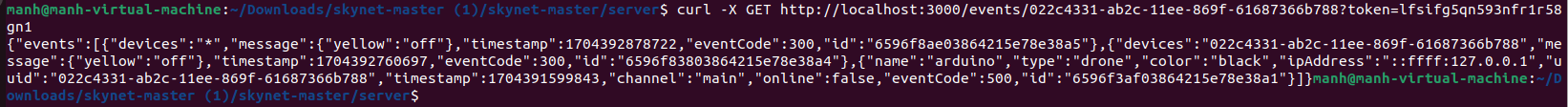
c. Gửi tin nhắn cho mảng các device:

curl -X POST -d '{"devices": ["022c4331-ab2c-11ee-869f-61687366b788","74521970-aac9-11ee-8c4e-39f35f991d31"], "message": {"yellow":"off"}}' <http://localhost:3000/messages>



## **4.9. Thực hiện lấy danh sách 10 event gần nhất của 1 device**

curl -X GET <http://localhost:3000/events/022c4331-ab2c-11ee-869f-61687366b788?token=lfsifg5qn593nfr1r58gn1>



Sau khi thực hiện thành công, chương trình sẽ trả về danh sách của 10 event liên quan tới device đã tìm kiếm.

## **4.10. Subscribe**

curl -X GET <http://localhost:3000/subscribe/e0201f40-ab7c-11ee-9be3-0f18db460b63?token=sqf4gnsuk8d11ryyht6jo>

## **4.11. Authenticate**

curl -X GET <http://localhost:3000/authenticate/022c4331-ab2c-11ee-869f-61687366b788?token=lfsifg5qn593nfr1r58gn1>



Chương trình sẽ trả về uuid và authenticaton: true hoặc false dựa trên tính hợp lệ của thông tin xác thực uuid/token.

curl -X GET <http://localhost:3000/mydevices/022c4331-ab2c-11ee-869f-61687366b788?token=lfsifg5qn593nfr1r58gn1>

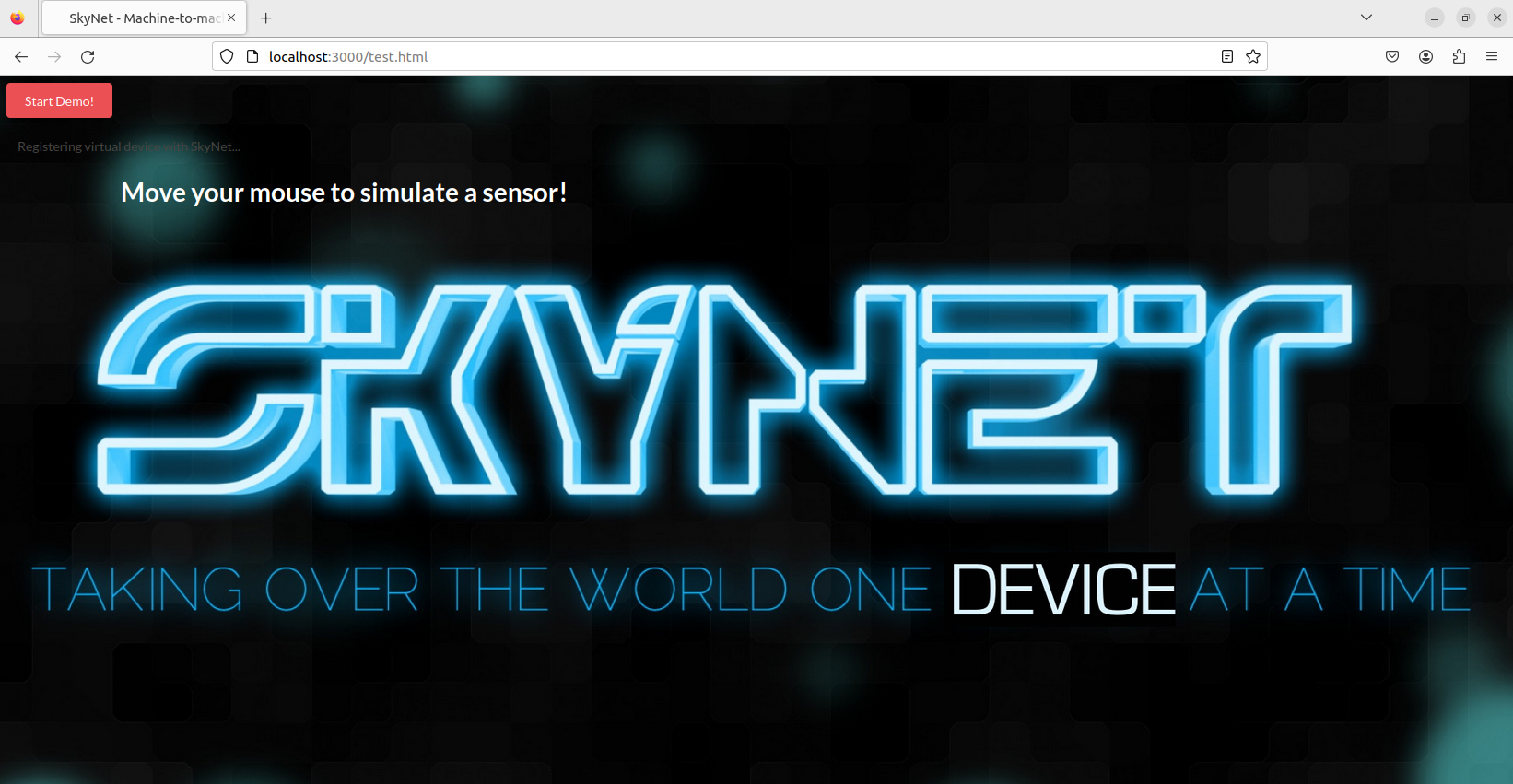
## **4.12. IP Address**

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

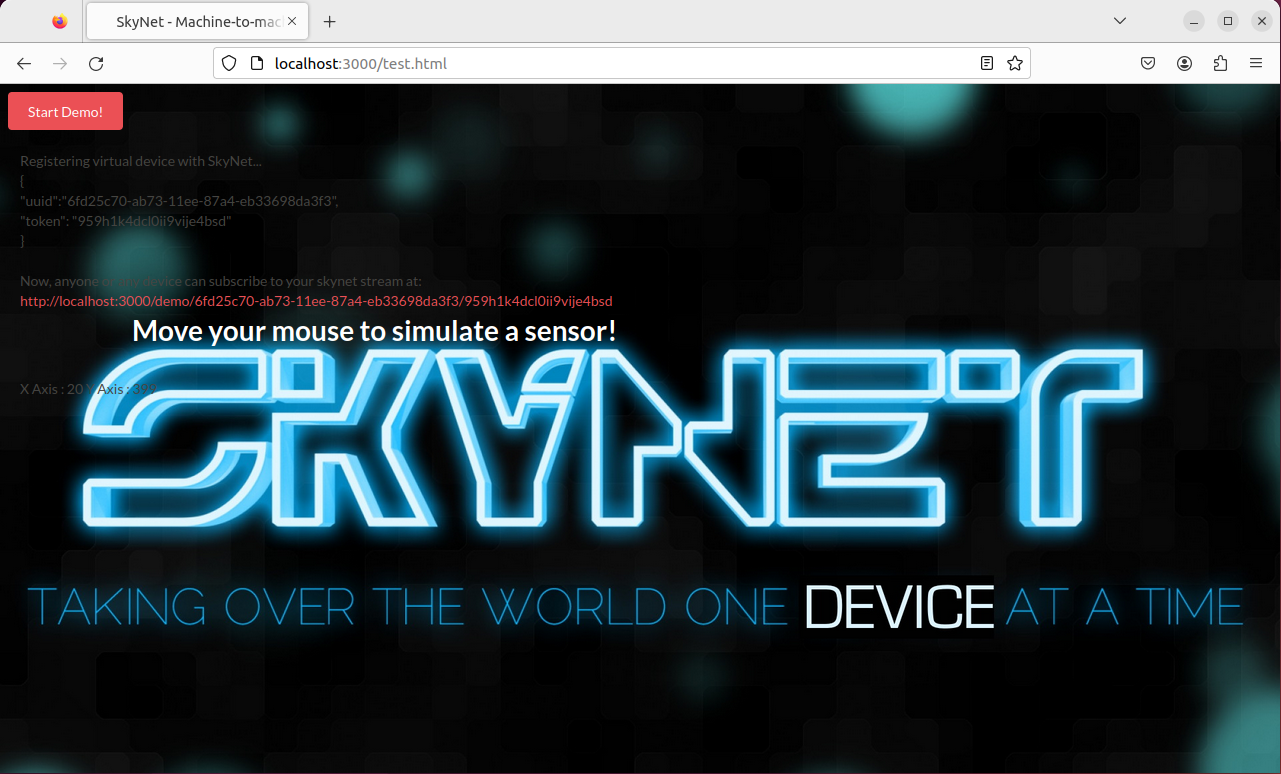


Chương trình sẽ trả về ipAddress của SkyNet.

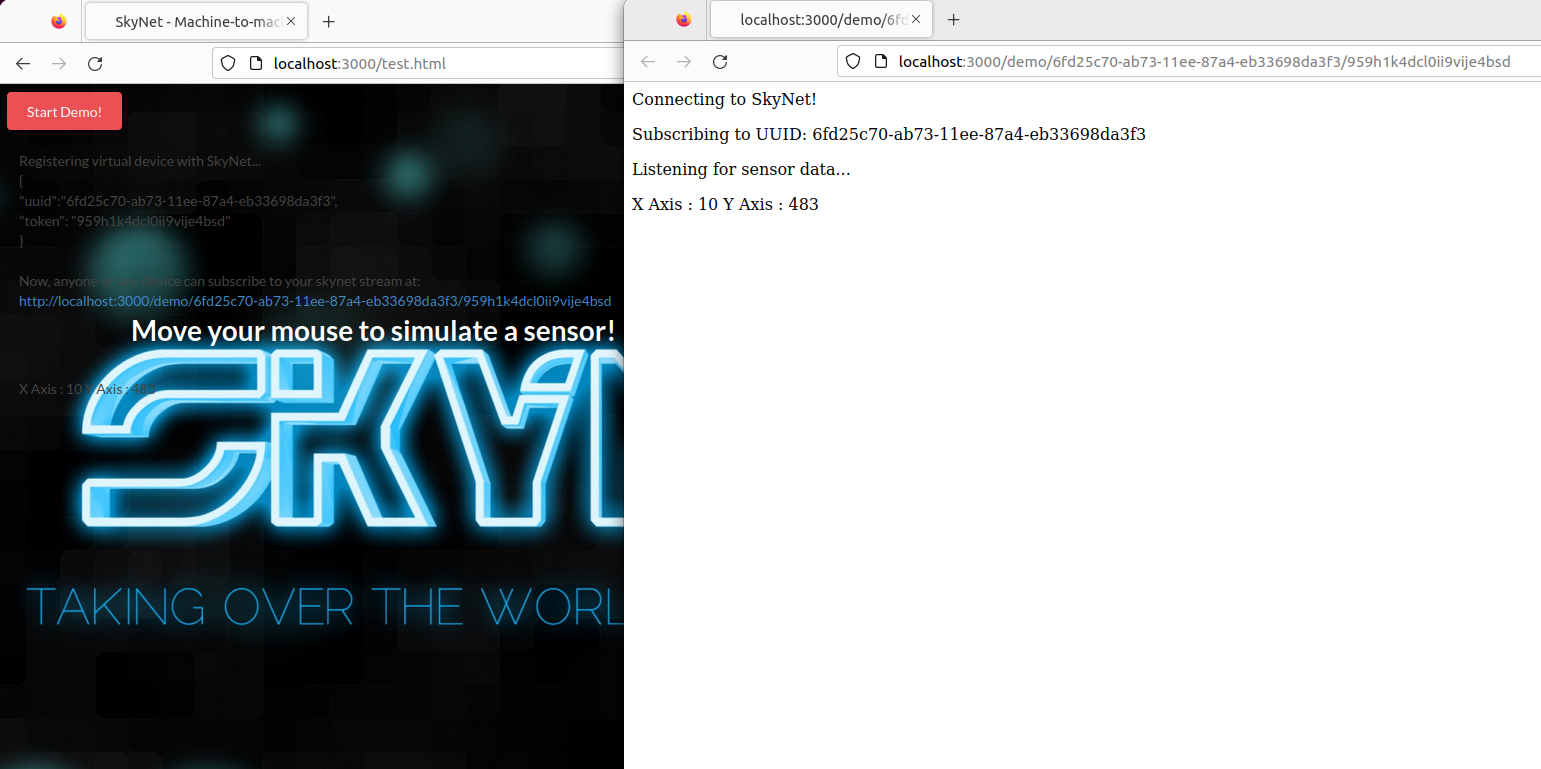
# CHƯƠNG 5. DEMO



Đây là giao diện Demo SkyNet khi bắt đầu khởi động. Để bắt đầu thực hiện Demo, chúng ta sẽ nhấn button “Start Demo!”.



Sau khi nhấn button “Start Demo!”, chương trình sẽ đăng ký thiết bị ảo tới SkyNet, đồng thời hiển thị uuid và token của device. Sau đó chương trình sẽ hiển thị 1 đường link demo cho device.



Đây là giao diện sau khi người dùng nhấn vào đường link ở trên. Khi bật chế độ cửa sổ song song, ta có thể thấy được khi di chuyển con chuột thì tọa độ X và Y của con chuột trên giao diện demo thay đổi đồng thời với tọa độ X và Y của con chuột trên giao diện SkyNet.

# CHƯƠNG 6. TỔNG KẾT

Nhóm chúng em đã thực hiện được các yêu cầu về chức năng của SkyNet, bao gồm:

* Hiển thị thông tin status của SkyNet
* Hiển thị danh sách device phù hợp theo key/value tìm kiếm
* Hiển thị chi tiết thông tin của 1 device cụ thể
* Đăng ký 1 device mới
* Cập nhật thông tin của 1 device
* Hủy đăng ký 1 device
* Thực hiện gửi tin nhắn tới tất cả device, mảng device và 1 device cụ thể
* Thực hiện lấy danh sách 10 event gần nhất của 1 device
* Thực hiện lấy authenticate
* Thực hiện lấy ip address
* Giao diện demo con trỏ chuột thay đổi tọa độ (x,y) cùng lúc với SkyNet

Tuy nhiên vẫn còn vài chức năng nhóm em chưa thực hiện được do còn nhiều thiếu sót trong quá trình hoàn thiện. Nhóm em rất mong được thầy góp ý để bài báo cáo của chúng em được hoàn thiện hơn. Chúng em xin chân thành cảm ơn.