

**ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**



PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

**Đề tài 507: Xây dựng ứng dụng giám sát các thiết bị mạng
dựa trên SNMP**

SINH VIÊN THỰC HIỆN:

Tên sinh viên: Trần Đình Văn. LỚP: 21TCLC_DT4

Tên sinh viên: Trần Thị Kim Tiến. LỚP: 21TCLC_DT4

GIẢNG VIÊN HƯỚNG DẪN: Nguyễn Thế Xuân Ly

Đà Nẵng Ngày 4 , tháng 1 năm 2024

MỤC LỤC

.....	1
MỤC LỤC	2
DANH SÁCH HÌNH VẼ.....	4
MỞ ĐẦU (GIỚI THIỆU ĐỀ TÀI)	5
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	6
1.1. Tổng quan về giao thức SNMP.....	6
1.1.1. Giao thức SNMP là gì?	6
1.1.2. Các phiên bản của giao thức SNMP	6
1.1.3. Các thành phần trong SNMP	7
1.1.4. Object ID	8
1.1.5. Object access	9
1.1.6. Management Information Base – MIB.....	10
1.1.7. Các phương thức của SNMP	11
1.1.8. Các cơ chế bảo mật cho SNMP.....	14
1.1.9. Cấu trúc bản tin SNMP	16
CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	17
2.1. Giới thiệu về phần mềm Zabbix	17
2.1.1. Khái niệm	17
2.1.2. Ưu điểm.....	17
2.1.3. Kiến trúc của hệ thống giám sát Zabbix.....	17
2.1.4. Cơ chế hoạt động.....	Error! Bookmark not defined.
2.1.5. Tính năng của Zabbix.....	19
2.1.6. Cấu trúc thư mục	20
2.1.7. Các phần tử cơ bản trong Zabbix	Error! Bookmark not defined.
2.2. Phân tích yêu cầu:	20
2.3. Phân tích các thành phần của hệ thống:.....	21
2.4. Phân tích cơ chế hoạt động của hệ thống:	22
2.5. Các phần tử cơ bản của hệ thống giám sát:	23
2.6. Các chức năng cơ bản của hệ thống giám sát:	25

CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ	26
3.1. Môi trường triển khai	26
3.2. Triển khai cấu hình	26
3.2.1. Cấu hình Zabbix Server trên Zabbixsrv	26
3.2.2. Cấu hình deploy Web Interface tương tác với Zabbix Server trên Zabbixsrv	28
3.2.3. Cấu hình Web Interface thiết lập các tham số để giám sát PC1 và Router1	29
3.3. Triển khai giám sát	33
3.3.1. Giám sát PC1 thông qua Web Interface	33
3.3.2. Giám sát Router1 thông qua Web Interface	36
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	37
TÀI LIỆU THAM KHẢO	38
[1] https://vinahost.vn/snmp-la-gi/	38
[2] Tên chủ sở hữu, Tên bài viết, url, ngày truy cập	38
[3] https://mdungblog.wordpress.com/2020/01/06/ly-thuyet-giao-thuc-snmp-toan-tap/	38
[4] https://www.zabbix.com/documentation/current/en/manual/api	38
[5] https://thegioifirewall.com/huong-dan-cau-hinh-giam-sat-switch-cisco-voi-giao-thuc-snmp-tren-zabbix/	38
Link source code : https://github.com/dinhvan2310/pbl4_snmp_ui	38

DANH SÁCH HÌNH VẼ

<i>Hình 1 Giao thức SNMP</i>	6
<i>Hình 2 Kiến trúc của SNMP</i>	7
<i>Hình 3 Quan hệ giữa Network management station và Network Element</i>	8
<i>Hình 4 Hình minh hoạ quá trình lấy sysName.0</i>	9
<i>Hình 5 : Minh hoạ MIB tree</i>	10
<i>Hình 6 Bảng các phương thức cơ bản của SNMP</i>	11
<i>Hình 7 Minh hoạ các phương thức của SNMPv1</i>	14
<i>Hình 8 Zabbix Server</i>	18
<i>Hình 9 Zabbix Proxy</i>	18
<i>Hình 10 Zabbix agent</i>	18
<i>Hình 11 Giao diện Web</i>	19
<i>Hình 12 Cơ chế hoạt động Zabbix</i>	Error! Bookmark not defined.
<i>Hình 13 Sơ đồ hoạt động của Zabbix</i>	23
<i>Hình 14 : Tạo host group Window SNMP</i>	30
<i>Hình 15 Tạo host PC1</i>	30
<i>Hình 16 Tạo mới một Trigger</i>	31
<i>Hình 17 : Tạo host groups Router SNMP</i>	32
<i>Hình 18 Bảng thống kê các Items của PC1</i>	33
<i>Hình 19 Đồ thị dung lượng CPU đã sử dụng</i>	34
<i>Hình 20 Đồ thị dung lượng Ram</i>	34
<i>Hình 21 Bảng giám sát dung lượng ổ cứng</i>	35
<i>Hình 22 Màn hình hiển thị cảnh báo của Host PC1</i>	35
<i>Hình 23 Bảng thống kê các Items của Router1</i>	36
<i>Hình 24 Đồ thị dung lượng CPU đã sử dụng</i>	36
<i>Hình 25 Đồ thị dung lượng CPU đã sử dụng</i>	37

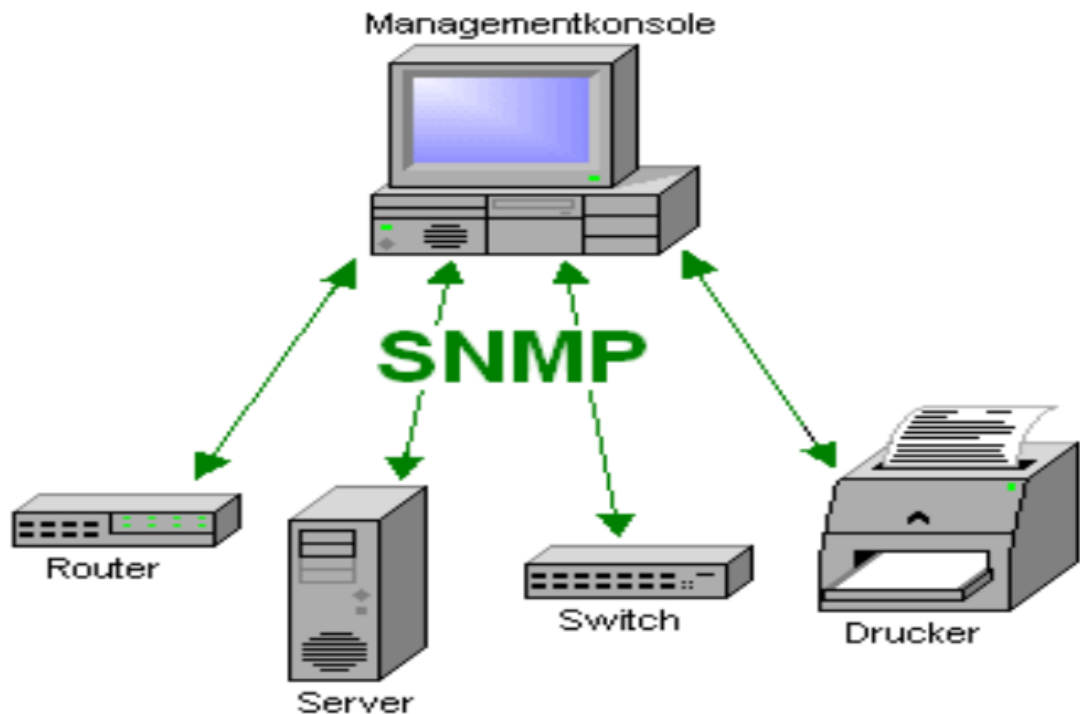
MỞ ĐẦU (GIỚI THIỆU ĐỀ TÀI)

Cùng với sự phát triển của công nghệ thông tin, sự đầu tư cho hạ tầng mạng trong mỗi doanh nghiệp ngày càng tăng cao, dẫn đến việc quản trị sự cố một hệ thống mạng gặp rất nhiều khó khăn. Đi cùng với những lợi ích khi phát triển hạ tầng mạng như băng thông cao, khối lượng dữ liệu trong mạng lớn, đáp ứng được nhu cầu của người dùng, hệ thống mạng phải đối đầu với rất nhiều thách thức như các cuộc tấn công bên ngoài, tính sẵn sàng của thiết bị, tài nguyên của hệ thống,... Một trong những giải pháp hữu hiệu nhất để giải quyết vấn đề này là thực hiện việc giải pháp giám sát mạng, dựa trên những thông tin thu thập được thông qua quá trình giám sát, các nhân viên quản trị mạng có thể phân tích, đưa ra những đánh giá, dự báo, giải pháp nhằm giải quyết những vấn đề trên. Để thực hiện giám sát mạng có hiệu quả, một chương trình giám sát phải đáp ứng được các yêu cầu sau: phải đảm bảo chương trình luôn hoạt động, tính linh hoạt, chức năng hiệu quả, đơn giản trong triển khai, chi phí thấp. Hiện nay có khá nhiều phần mềm hỗ trợ việc giám sát mạng có hiệu quả như Nagios, Zabbix, PRTG,...

Vì vậy, Em đã chọn đề tài “ Xây dựng hệ thống giám sát mạng dựa trên SNMP thông qua phần mềm mã nguồn mở Zabbix”, một phần mềm mã nguồn mở với nhiều chức năng mạnh mẽ cho phép quản lý các thiết bị, dịch vụ trong hệ thống mạng. Với mục tiêu nghiên cứu, tìm hiểu về giải pháp giúp mọi người có cái nhìn tổng quan về một hệ thống giám sát mạng hoàn chỉnh, đồng thời đưa ra một giải pháp cụ thể đối với một hệ thống mạng dành cho doanh nghiệp.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT**1.1. Tổng quan về giao thức SNMP****1.1.1. Giao thức SNMP là gì?**

SNMP (Simple Network Management Protocol) là giao thức tầng ứng dụng trong mô hình TCP/IP được dùng để quản lý và giám sát tất cả thiết bị mạng và các chức năng có liên quan.



Hình 1 Giao thức SNMP

SNMP cung cấp ngôn ngữ chung cho các thiết bị mạng. Nhờ đó, SNMP tạo thuận lợi cho các nơi nhận khi chuyển tiếp thông tin quản lý trong môi trường single-vendor và multi-vendor ở mạng cục bộ (**LAN**). Đôi khi, quá trình này có thể diễn ra ở **mạng diện rộng (WAN)**. SNMP3 là phiên bản mới nhất của SNMP chứa đầy đủ tính năng bảo mật để xác thực và mã hóa tin nhắn. Thậm chí, nó đủ khả năng bảo vệ các gói trong khi truyền thông tin.

1.1.2. Các phiên bản của giao thức SNMP

Giao thức SNMP có 3 phiên bản khác nhau:

SNMP phiên bản 1 (thường được viết tắt là SNMPv1): đây là phiên bản triển khai đầu tiên và nó thường hoạt động trong đặc tả thông tin quản lý cấu trúc. Nó cũng đã được mô tả trong tài liệu RFC 1157.

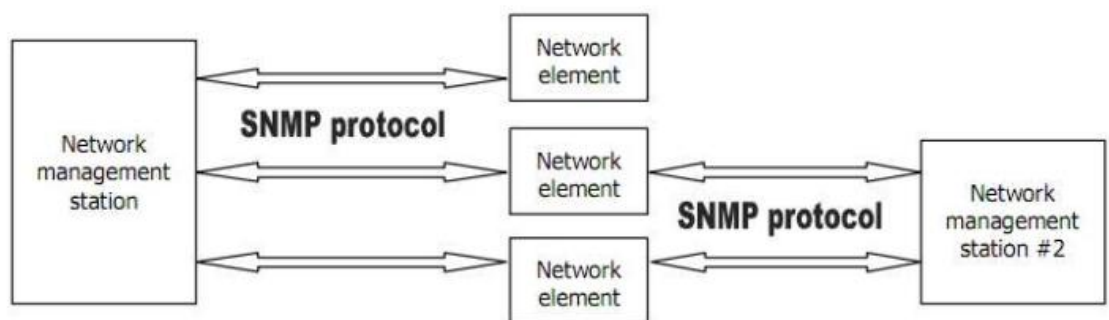
SNMP phiên bản 2 (SNMPv2): Đây là phiên bản đã được cải tiến nhằm hỗ trợ xử lý các lỗi khá hiệu quả. Nó thường được mô tả trong RFC 1901. Phiên bản này lần đầu tiên được giới thiệu trong RFC 1441. Ở thời điểm hiện tại, SNMPv2 là phiên bản thịnh hành nhất.

SNMP phiên bản 3 (SNMPv3): Đây là phiên bản có cải thiện tính năng bảo mật và quyền riêng tư. Nó thường được giới thiệu trong RFC 3410. Phiên bản này là mới nhất, có hỗ trợ xác thực, mã hóa các tập tin và các tin nhắn. SNMP có các gói tin bảo vệ trong quá trình truyền đi.

1.1.3. Các thành phần trong SNMP

Theo RFC1157, kiến trúc của SNMP bao gồm 2 thành phần: các trạm quản lý mạng (network management station) và các thành tố mạng (network element).

Network management station thường là một máy tính chạy phần mềm quản lý SNMP (SNMP management application), dùng để giám sát và điều khiển tập trung các network element.



Hình 2 Kiến trúc của SNMP

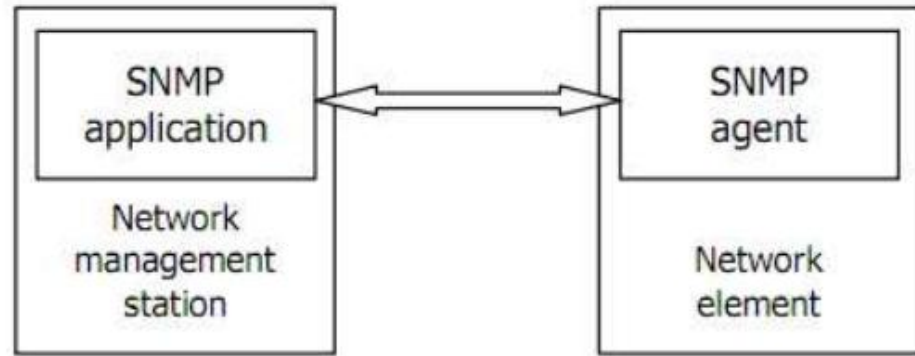
Network element là các thiết bị, máy tính, hoặc phần mềm tương thích SNMP và được quản lý bởi network management station. Như vậy element bao gồm device, host và application.

Một management station có thể quản lý nhiều element, một element cũng có thể được quản lý bởi nhiều management station. Vậy nếu một element được quản lý bởi 2 station thì điều gì sẽ xảy ra? Nếu station lấy thông tin từ element thì cả 2 station sẽ có thông tin giống nhau. Nếu 2 station tác động đến cùng một element thì element sẽ đáp ứng cả 2 tác động theo thứ tự cái nào đến trước.

Ngoài ra còn có khái niệm *SNMP agent*. SNMP agent là một tiến trình (process) chạy trên network element, có nhiệm vụ cung cấp thông tin của element cho station, nhờ đó station có thể quản lý được element. Chính xác hơn là application chạy trên station và agent chạy trên element mới là 2 tiến trình SNMP trực tiếp liên hệ với nhau. Các ví dụ minh họa sau đây sẽ làm rõ hơn các khái niệm này:

+ Để dùng một máy chủ (= station) quản lý các máy con (= element) chạy HĐH Windows thông qua SNMP thì bạn phải: cài đặt một phần mềm quản lý SNMP (=application) trên máy chủ, bật SNMP service (= agent) trên máy con.

+ Để dùng một máy chủ (= station) giám sát lưu lượng của một router (= element) thì bạn phải: cài phần mềm quản lý SNMP (= application) trên máy chủ, bật tính năng SNMP (=agent) trên router.



Hình 3 Quan hệ giữa Network management station và Network Element

1.1.4. Object ID

Một thiết bị hỗ trợ SNMP có thể cung cấp nhiều thông tin khác nhau, mỗi thông tin đó gọi là một object. Ví dụ :

- + Máy tính có thể cung cấp các thông tin : tổng số ổ cứng, tổng số port nối mạng, tổng số byte đã truyền/nhận, tên máy tính, tên các process đang chạy,

- + Router có thể cung cấp các thông tin : tổng số card, tổng số port, tổng số byte đã truyền/nhận, tên router, tình trạng các port của router,

Mỗi object có một tên gọi và một mã số để nhận dạng object đó, mã số gọi là Object ID (OID). VD :

- + Tên thiết bị được gọi là sysName, OID là 1.3.6.1.2.1.1.5 .

- + Tổng số port giao tiếp (interface) được gọi là ifNumber, OID là 1.3.6.1.2.1.2.1.

- + Địa chỉ Mac Address của một port được gọi là ifPhysAddress, OID là 1.3.6.1.2.1.2.2.1.6.

- + Số byte đã nhận trên một port được gọi là ifInOctets, OID là 1.3.6.1.2.1.2.2.1.10.

Bạn hãy khoan thắc mắc ý nghĩa của từng chữ số trong OID, chúng sẽ được giải thích trong phần sau. Một object chỉ có một OID, chẳng hạn tên của thiết bị là một object. Tuy nhiên nếu một thiết bị lại có nhiều tên thì làm thế nào để phân biệt ? Lúc này người ta dùng thêm 1 chỉ số gọi là “scalar instance index” (cũng có thể gọi là “sub-id”) đặt ngay sau OID. Ví dụ :

- + Tên thiết bị được gọi là sysName, OID là 1.3.6.1.2.1.1.5; nếu thiết bị có 2 tên thì chúng sẽ được gọi là sysName.0 & sysName.1 và có OID lần lượt là 1.3.6.1.2.1.1.5.0 & 1.3.6.1.2.1.1.5.1.

- + Địa chỉ Mac address được gọi là ifPhysAddress, OID là 1.3.6.1.2.1.2.2.1.6; nếu thiết bị có 2 mac address thì chúng sẽ được gọi là ifPhysAddress.0 & ifPhysAddress.1 và có OID lần lượt là 1.3.6.1.2.1.2.2.1.6.0 & 1.3.6.1.2.1.2.2.1.6.1.

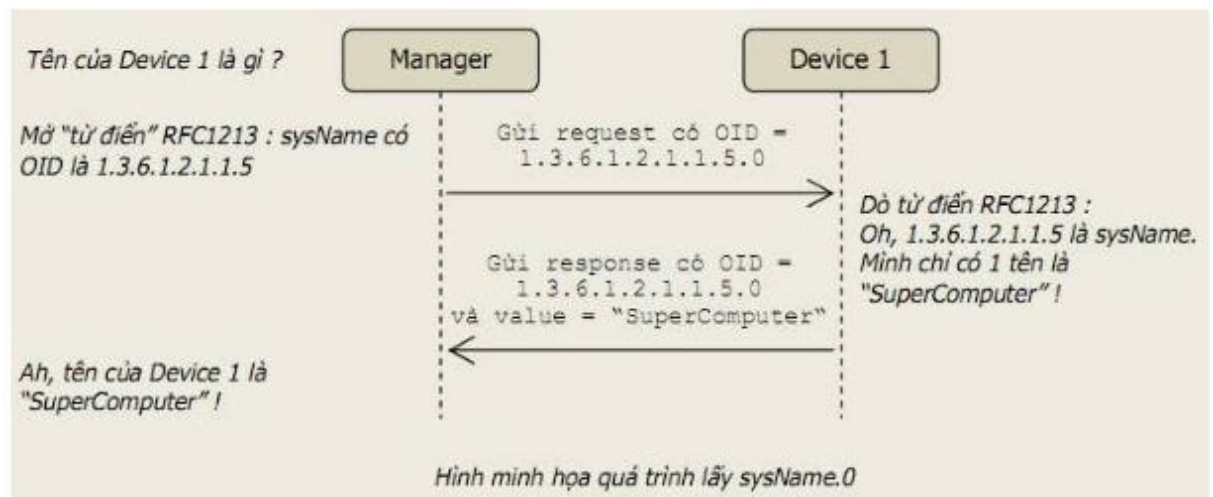
+ Tổng số port được gọi là ifNumber, giá trị này chỉ có 1 (duy nhất) nên OID của nó không có phân cấp con và vẫn là 1.3.6.1.2.1.2.1.

Ở hầu hết các thiết bị, các object có thể có nhiều giá trị thì thường được viết dưới dạng có sub-id. VD một thiết bị dù chỉ có 1 tên thì nó vẫn phải có OID là sysName.0 hay 1.3.6.1.2.1.1.5.0. Bạn cần nhớ quy tắc này để ứng dụng trong lập trình phần mềm SNMP manager.

Sub-id không nhất thiết phải liên tục hay bắt đầu từ 0. VD một thiết bị có 2 mac address thì có thể chúng được gọi là ifPhysAddress.23 và ifPhysAddress.125645.

OID của các object phổ biến có thể được chuẩn hóa, OID của các object do bạn tạo ra thì bạn phải tự mô tả chúng. Để lấy một thông tin có OID đã chuẩn hóa thì SNMP application phải gửi một bản tin SNMP có chứa OID của object đó cho SNMP agent, SNMP agent khi nhận được thì nó phải trả lời bằng thông tin ứng với OID đó.

VD : Muốn lấy tên của một PC chạy Windows, tên của một PC chạy Linux hoặc tên của một router thì SNMP application chỉ cần gửi bản tin có chứa OID là 1.3.6.1.2.1.1.5.0. Khi SNMP agent chạy trên PC Windows, PC Linux hay router nhận được bản tin có chứa OID 1.3.6.1.2.1.1.5.0, agent lập tức hiểu rằng đây là bản tin hỏi sysName.0, và agent sẽ trả lời bằng tên của hệ thống. Nếu SNMP agent nhận được một OID mà nó không hiểu (không hỗ trợ) thì nó sẽ không trả lời.



Hình 4 Hình minh họa quá trình lấy sysName.0

Một trong các ưu điểm của SNMP là nó được thiết kế để chạy độc lập với các thiết bị khác nhau. Chính nhờ việc chuẩn hóa OID mà ta có thể dùng một SNMP application để lấy thông tin các loại device của các hãng khác nhau.

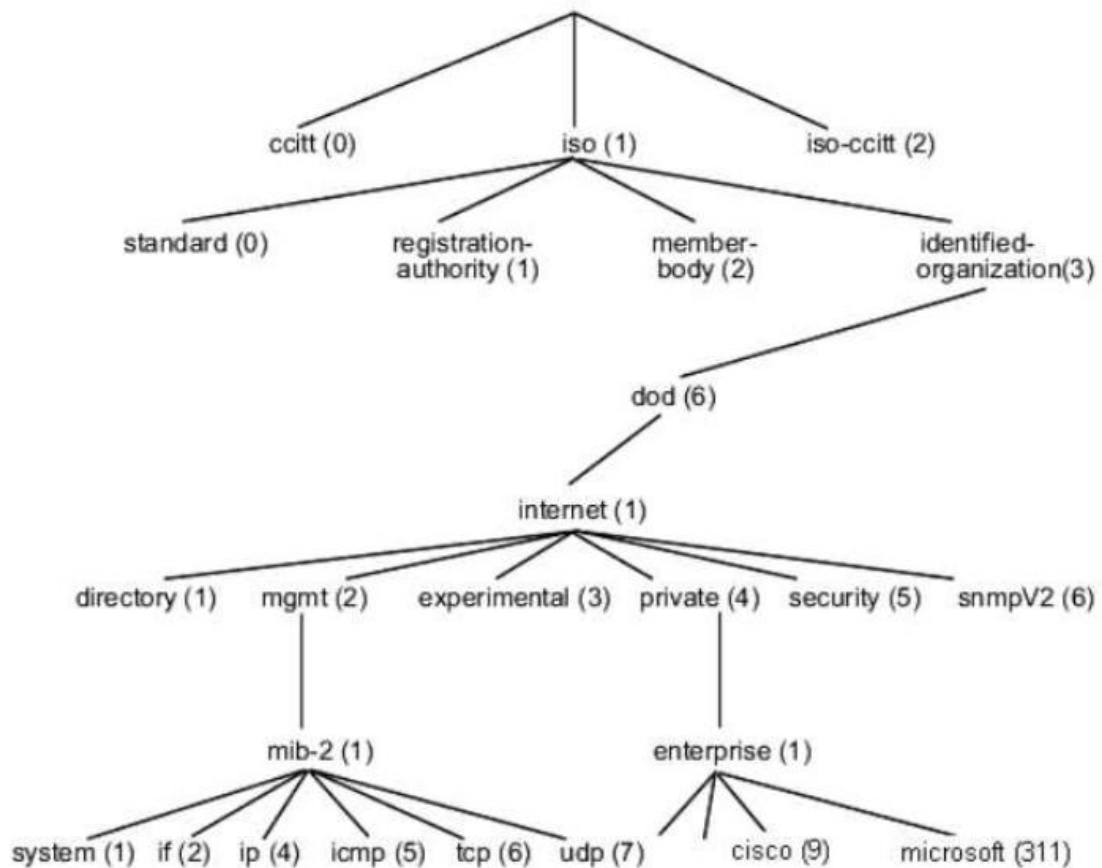
1.1.5. Object access

Mỗi object có quyền truy cập là READ_ONLY hoặc READ_WRITE. Mọi object đều có thể đọc được nhưng chỉ những object có quyền READ_WRITE mới có thể thay đổi được giá trị. VD : Tên của một thiết bị (sysName) là READ_WRITE, ta có thể

thay đổi tên của thiết bị thông qua giao thức SNMP. Tổng số port của thiết bị (ifNumber) là READ_ONLY, dĩ nhiên ta không thể thay đổi số port của nó.

1.1.6. Management Information Base – MIB

MIB (cơ sở thông tin quản lý) là một cấu trúc dữ liệu gồm các đối tượng được quản lý (managed object), được dùng cho việc quản lý các thiết bị chạy trên nền TCP/IP. MIB là kiến trúc chung mà các giao thức quản lý trên TCP/IP nên tuân theo, trong đó có SNMP. MIB được thể hiện thành 1 file (MIB file), và có thể biểu diễn thành 1 cây (MIB tree). MIB có thể được chuẩn hóa hoặc tự tạo.



Hình 5 : Minh họa MIB tree

Một node trong cây là một object, có thể được gọi bằng tên hoặc id. Ví dụ :

+ Node iso.org.dod.internet.mgmt.mib-2.system có OID là 1.3.6.1.2.1.1, chứa tất cả các object liên quan đến thông tin của một hệ thống như tên của thiết bị

+ Các OID của các hãng tự thiết kế nằm dưới iso.org.dod.internet.private.enterprise. Ví dụ : Cisco nằm dưới iso.org.dod.internet.private.enterprise.cisco hay 1.3.6.1.4.1.9, Microsoft nằm dưới iso.org.dod.internet.private.enterprise.microsoft hay 1.3.6.1.4.1.311. Số 9 (Cisco) hay 311 (Microsoft) là số dành riêng cho các công ty do IANA cấp. Nếu Cisco hay Microsoft chế tạo ra một thiết bị nào đó, thì thiết bị này có thể hỗ trợ các MIB chuẩn

đã được định nghĩa sẵn (như mib-2) hay hỗ trợ MIB được thiết kế riêng. Các MIB được công ty nào thiết kế riêng thì phải nằm bên dưới OID của công ty đó.

Các objectID trong MIB được sắp xếp thứ tự nhưng không phải là liên tục, khi biết một OID thì không chắc chắn có thể xác định được OID tiếp theo trong MIB. VD trong chuẩn mib-2 thì object ifSpecific và object atIfIndex nằm kề nhau nhưng OID lần lượt là 1.3.6.1.2.1.2.2.1.22 và 1.3.6.1.2.1.3.1.1.1.

Muốn hiểu được một OID nào đó thì bạn cần có file MIB mô tả OID đó. Một MIB file không nhất thiết phải chứa toàn bộ cây ở trên mà có thể chỉ chứa mô tả cho một nhánh con. Bất cứ nhánh con nào và tất cả lá của nó đều có thể gọi là một mib.

Một manager có thể quản lý được một device chỉ khi ứng dụng SNMP manager và ứng dụng SNMP agent cùng hỗ trợ một MIB. Các ứng dụng này cũng có thể hỗ trợ cùng lúc nhiều MIB.

1.1.7. Các phương thức của SNMP

Giao thức SNMPv1 có 5 phương thức hoạt động, tương ứng với 5 loại bản tin như sau:

Bản tin/phương thức	Mô tả tác dụng
GetRequest	Manager gửi GetRequest cho agent để yêu cầu agent cung cấp thông tin nào đó dựa vào ObjectID (trong GetRequest có chứa OID)
GetNextRequest	Manager gửi GetNextRequest có chứa một ObjectID cho agent để yêu cầu cung cấp thông tin nằm kế tiếp ObjectID đó trong MIB.
SetRequest	Manager gửi SetRequest cho agent để đặt giá trị cho đối tượng của agent dựa vào ObjectID.
GetResponse	Agent gửi GetResponse cho Manager để trả lời khi nhận được GetRequest/GetNextRequest
Trap	Agent tự động gửi Trap cho Manager khi có một sự kiện xảy ra đối với một object nào đó trong agent.

Hình 6 Bảng các phương thức cơ bản của SNMP

Mỗi bản tin đều có chứa OID để biết object mang trong nó là gì. OID trong GetRequest cho biết nó muốn lấy thông tin của object nào. OID trong GetResponse cho biết nó mang giá trị của object nào. OID trong SetRequest chỉ ra nó muốn thiết lập giá trị cho object nào. OID trong Trap chỉ ra nó thông báo sự kiện xảy ra đối với object nào.

GetRequest

Bản tin GetRequest được manager gửi đến agent để lấy một thông tin nào đó. Trong GetRequest có chứa OID của object muốn lấy. VD : Muốn lấy thông tin tên của Device1 thì manager gửi bản tin GetRequest OID=1.3.6.1.2.1.1.5 đến Device1, tiến trình SNMP agent trên Device1 sẽ nhận được bản tin và tạo bản tin trả lời.

Trong một bản tin GetRequest có thể chứa nhiều OID, nghĩa là dùng một GetRequest có thể lấy về cùng lúc nhiều thông tin.

GetNextRequest

Bản tin GetNextRequest cũng dùng để lấy thông tin và cũng có chứa OID, tuy nhiên nó dùng để lấy thông tin của object nằm kế tiếp object được chỉ ra trong bản tin.

Tại sao phải có phương thức GetNextRequest ? Như bạn đã biết khi đọc qua những phần trên : một MIB bao gồm nhiều OID được sắp xếp thứ tự nhưng không liên tục, nếu biết một OID thì không xác định được OID kế tiếp. Do đó ta cần GetNextRequest để lấy về giá trị của OID kế tiếp. Nếu thực hiện GetNextRequest liên tục thì ta sẽ lấy được toàn bộ thông tin của agent.

SetRequest

Bản tin SetRequest được manager gửi cho agent để thiết lập giá trị cho một object nào đó. Ví dụ :

- + Có thể đặt lại tên của một máy tính hay router bằng phần mềm SNMP manager, bằng cách gửi bản tin SetRequest có OID là 1.3.6.1.2.1.1.5.0 (sysName.0) và có giá trị là tên mới cần đặt.

- + Có thể shutdown một port trên switch bằng phần mềm SNMP manager, bằng cách gửi bản tin có OID là 1.3.6.1.2.1.2.2.1.7 (ifAdminStatus) và có giá trị là 2

- * ifAdminStatus có thể mang 3 giá trị là UP (1), DOWN (2) và TESTING (3).

Chỉ những object có quyền READ_WRITE mới có thể thay đổi được giá trị.

GetResponse

Mỗi khi SNMP agent nhận được các bản tin GetRequest, GetNextRequest hay SetRequest thì nó sẽ gửi lại bản tin GetResponse để trả lời. Trong bản tin GetResponse có chứa OID của object được request và giá trị của object đó.

Trap

Bản tin Trap được agent tự động gửi cho manager mỗi khi có sự kiện xảy ra bên trong agent, các sự kiện này không phải là các hoạt động thường xuyên của agent mà là các sự kiện mang tính biến cố. Ví dụ : Khi có một port down, khi có một người dùng login không thành công, hoặc khi thiết bị khởi động lại, agent sẽ gửi trap cho manager.

Tuy nhiên không phải mọi biến cố đều được agent gửi trap, cũng không phải mọi agent đều gửi trap khi xảy ra cùng một biến cố. Việc agent gửi hay không gửi trap cho biến cố nào là do hãng sản xuất device/agent quy định.

Phương thức trap là độc lập với các phương thức request/response. SNMP request/response dùng để quản lý còn SNMP trap dùng để cảnh báo. Nguồn gửi trap gọi là Trap Sender và nơi nhận trap gọi là Trap Receiver. Một trap sender có thể được cấu hình để gửi trap đến nhiều trap receiver cùng lúc.

Có 2 loại trap : trap phổ biến (generic trap) và trap đặc thù (specific trap). Generic trap được quy định trong các chuẩn SNMP, còn specific trap do người dùng tự định nghĩa (người dùng ở đây là hãng sản xuất SNMP device). Loại trap là một số nguyên chứa trong bản tin trap, dựa vào đó mà phía nhận trap biết bản tin trap có nghĩa gì.

Theo SNMPv1, generic trap có 7 loại sau : coldStart(0), warmStart(1), linkDown(2), linkUp(3), authenticationFailure(4), egpNeighborloss(5), enterpriseSpecific(6). Giá trị trong ngoặc là mã số của các loại trap. Ý nghĩa của các bản tin generic-trap như sau :

+ coldStart : thông báo rằng thiết bị gửi bản tin này đang khởi động lại (reinitialize) và cấu hình của nó có thể bị thay đổi sau khi khởi động.

+ warmStart : thông báo rằng thiết bị gửi bản tin này đang khởi động lại và giữ nguyên cấu hình cũ.

+ linkDown : thông báo rằng thiết bị gửi bản tin này phát hiện được một trong những kết nối truyền thông (communication link) của nó gặp lỗi. Trong bản tin trap có tham số chỉ ra ifIndex của kết nối bị lỗi.

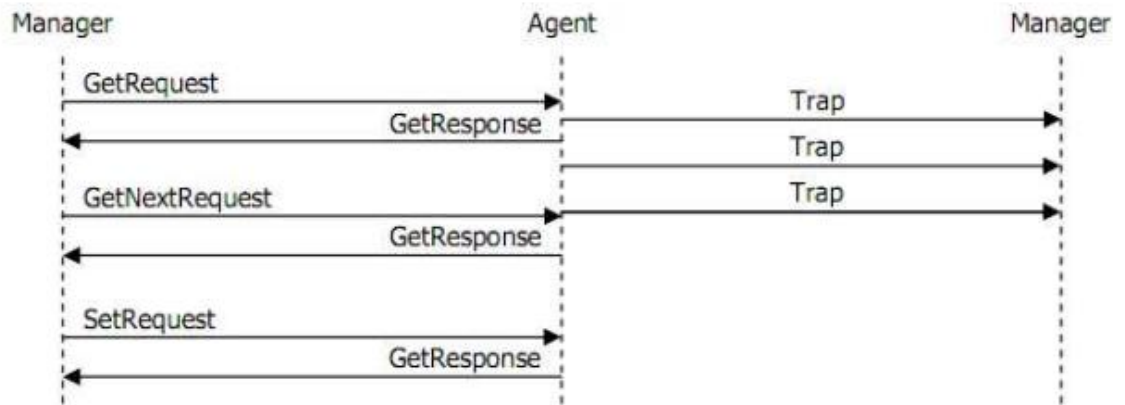
+ linkUp : thông báo rằng thiết bị gửi bản tin này phát hiện được một trong những kết nối truyền thông của nó đã khôi phục trở lại. Trong bản tin trap có tham số chỉ ra ifIndex của kết nối được khôi phục.

+ authenticationFailure : thông báo rằng thiết bị gửi bản tin này đã nhận được một bản tin không được chứng thực thành công (bản tin bị chứng thực không thành công có thể thuộc nhiều giao thức khác nhau như telnet, ssh, snmp, ftp, ...). Thông thường trap loại này xảy ra là do user đăng nhập không thành công vào thiết bị.

+ egpNeighborloss : thông báo rằng một trong số những “EGP neighbor” của thiết bị gửi trap đã bị coi là down và quan hệ đối tác (peer relationship) giữa 2 bên không còn được duy trì.

+ enterpriseSpecific : thông báo rằng bản tin trap này không thuộc các kiểu generic như trên mà nó là một loại bản tin do người dùng tự định nghĩa.

Người dùng có thể tự định nghĩa thêm các loại trap để làm phong phú thêm khả năng cảnh báo của thiết bị như : boardFailed, configChanged, powerLoss, cpuTooHigh, v.v.... Người dùng tự quy định ý nghĩa và giá trị của các specific trap này, và dĩ nhiên chỉ những trap receiver và trap sender hỗ trợ cùng một MIB mới có thể hiểu ý nghĩa của specific trap. Do đó nếu bạn dùng một phần mềm trap receiver bất kỳ để nhận trap của các trap sender bất kỳ, bạn có thể đọc và hiểu các generic trap khi chúng xảy ra; nhưng bạn sẽ không hiểu ý nghĩa các specific trap khi chúng hiện lên màn hình vì bản tin trap chỉ chứa những con số.



Hình 7 Minh họa các phương thức của SNMPv1

Đối với các phương thức Get/Set/Response thì SNMP Agent lắng nghe ở port UDP 161, còn phương thức trap thì SNMP Trap Receiver lắng nghe ở port UDP 162.

1.1.8. Các cơ chế bảo mật cho SNMP

Một SNMP management station có thể quản lý/giám sát nhiều SNMP element, thông qua hoạt động gửi request và nhận trap. Tuy nhiên một SNMP element có thể được cấu hình để chỉ cho phép các SNMP management station nào đó được phép quản lý/giám sát mình.

Các cơ chế bảo mật đơn giản này gồm có : community string, view và SNMP access control list.

Community string

Community string là một chuỗi ký tự được cài đặt giống nhau trên cả SNMP manager và SNMP agent, đóng vai trò như “mật khẩu” giữa 2 bên khi trao đổi dữ liệu. Community string có 3 loại : Read-community, Write-Community và Trap-Community.

Khi manager gửi GetRequest, GetNextRequest đến agent thì trong bản tin gửi đi có chứa Read-Community. Khi agent nhận được bản tin request thì nó sẽ so sánh Read-community do manager gửi và Read-community mà nó được cài đặt. Nếu 2 chuỗi này giống nhau, agent sẽ trả lời; nếu 2 chuỗi này khác nhau, agent sẽ không trả lời.

Write-Community được dùng trong bản tin SetRequest. Agent chỉ chấp nhận thay đổi dữ liệu khi write-community 2 bên giống nhau.

Trap-community nằm trong bản tin trap của trap sender gửi cho trap receiver. Trap receiver chỉ nhận và lưu trữ bản tin trap chỉ khi trap-community 2 bên giống nhau, tuy nhiên cũng có nhiều trap receiver được cấu hình nhận tất cả bản tin trap mà không quan tâm đến trap-community.

Community string có 3 loại như trên nhưng cùng một loại có thể có nhiều string khác nhau. Nghĩa là một agent có thể khai báo nhiều read-community, nhiều write-community.

Trên hầu hết hệ thống, read-community mặc định là “public”, write-community mặc định là “private” và trap-community mặc định là “public”.

Community string chỉ là chuỗi ký tự dạng cleartext, do đó hoàn toàn có thể bị nghe lén khi truyền trên mạng. Hơn nữa, các community mặc định thường là “public” và “private” nên nếu người quản trị không thay đổi thì chúng có thể dễ dàng bị dò ra. Khi community string trong mạng bị lộ, một người dùng bình thường tại một máy tính nào đó trong mạng có thể quản lý/giám sát toàn bộ các device có cùng community mà không được sự cho phép của người quản trị.

View

Khi manager có read-community thì nó có thể đọc toàn bộ OID của agent. Tuy nhiên agent có thể quy định chỉ cho phép đọc một số OID có liên quan nhau, tức là chỉ đọc được một phần của MIB. Tập con của MIB này gọi là view, trên agent có thể định nghĩa nhiều view. Ví dụ : agent có thể định nghĩa view interfaceView bao gồm các OID liên quan đến interface, storageView bao gồm các OID liên quan đến lưu trữ, hay AllView bao gồm tất cả các OID.

Một view phải gắn liền với một community string. Tùy vào community string nhận được là gì mà agent xử lý trên view tương ứng. Ví dụ : agent định nghĩa read-community “inf” trên view interfaceView, và “sto” trên storageView; khi manager gửi request lấy OID ifNumber với community là “inf” thì sẽ được đáp ứng do ifNumber nằm trong interfaceView; nếu manager request OID hrStorageSize với community “inf” thì agent sẽ không trả lời do hrStorageSize không nằm trong interfaceView; nhưng nếu manager request hrStorageSize với community “sto” thì sẽ được trả lời do hrStorageSize nằm trong storageView.

Việc định nghĩa các view như thế nào tùy thuộc vào từng SNMP agent khác nhau. Có nhiều hệ thống không hỗ trợ tính năng view.

SNMP access control list

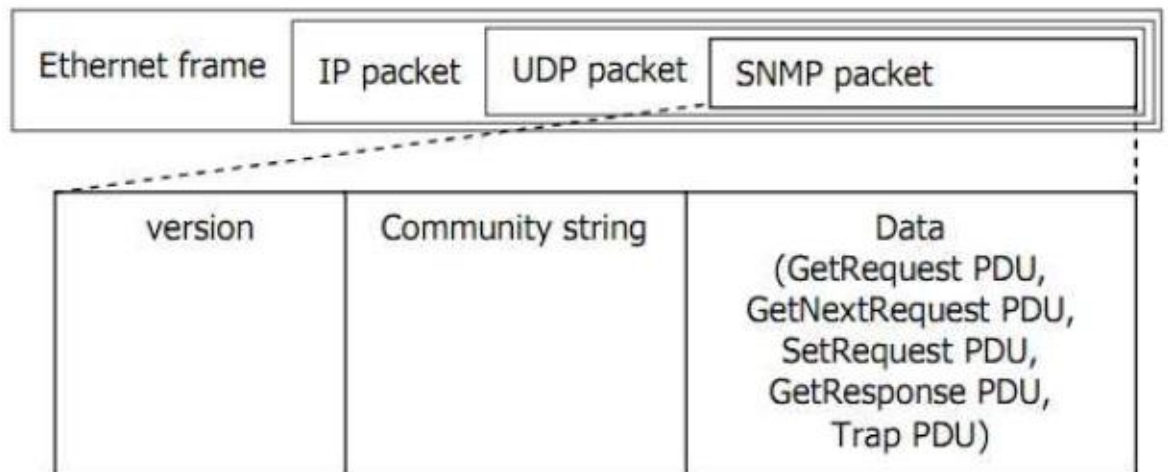
Khi manager gửi không đúng community hoặc khi OID cần lấy lại không nằm trong view cho phép thì agent sẽ không trả lời. Tuy nhiên khi community bị lộ thì một manager nào đó vẫn request được thông tin. Để ngăn chặn hoàn toàn các SNMP manager không được phép, người quản trị có thể dùng đến SNMP access control list (ACL).

SNMP ACL là một danh sách các địa chỉ IP được phép quản lý/giám sát agent, nó chỉ áp dụng riêng cho giao thức SNMP và được cài trên agent. Nếu một manager có IP không được phép trong ACL gửi request thì agent sẽ không xử lý, dù request có community string là đúng.

Đa số các thiết bị tương thích SNMP đều cho phép thiết lập SNMP ACL.

1.1.9. Cấu trúc bản tin SNMP

SNMP chạy trên nền UDP. Cấu trúc của một bản tin SNMP bao gồm : version, community và data.



+ Version : v1 = 0, v2c = 1, v2u = 2, v3 = 3.

+ Phần Data trong bản tin SNMP gọi là PDU (Protocol Data Unit). SNMPv1 có 5 phương thức hoạt động tương ứng 5 loại PDU. Tuy nhiên chỉ có 2 loại định dạng bản tin là PDU và Trap-PDU; trong đó các bản tin Get, GetNext, Set, GetResponse có cùng định dạng là PDU, còn bản tin Trap có định dạng là Trap-PDU.

CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1. Giới thiệu về phần mềm Zabbix

2.1.1. Khái niệm

Zabbix là công cụ được sáng sủa lập bởi Alexei Vladishev và hiện tại tổ chức Zabbix SIA đang phụ trách phát triển. Công cụ này giúp các doanh nghiệp quản lý hệ thống mạng cũng như thông tin trong các hệ thống mạng. Không những vậy nó còn góp phần to lớn trong việc quản lý hệ thống mạng của các công ty làm về lĩnh vực truyền thông và dịch vụ công nghệ thông tin.

Zabbix là một hệ thống giám sát được phát triển nhiều tính năng vượt trội trong việc giám sát thông tin kể cả những thông tin do khách hàng cung cấp.

Bằng cách giải quyết thông minh và linh hoạt, Zabbix sẽ phát hiện các sự cố và gửi báo cáo về cho máy chủ hoặc quản trị viên bằng email, sms, OTT app. Không những vậy công cụ này còn báo cáo một cách chính xác những thông tin đã thu thập được.

2.1.2. Ưu điểm

- Giám sát server và các thiết bị mạng được sử dụng trong hệ thống nội bộ
- Giúp dễ dàng thực hiện thao tác với cấu hình đơn giản dễ sử dụng
- Hỗ trợ hoạt động của các máy chủ như Linux, Solaris, FreeBSD,...
- Đáng tin cậy trong việc nhận diện xác minh người dùng
- Linh hoạt trong việc phân định quyền của mỗi người sử dụng
- Giao diện web được thiết kế tinh tế, đẹp mắt
- Nhanh chóng thông báo sự cố qua SMS, email cho quản trị viên
- Lập biểu đồ theo dõi báo cáo thông minh
- Mã nguồn mở và chi phí dịch vụ thấp
- Sở hữu nhiều mức độ plugin hỗ trợ dịch vụ hệ thống khác nhau
- Hỗ trợ monitor các máy client với nhiều loại OS khác nhau

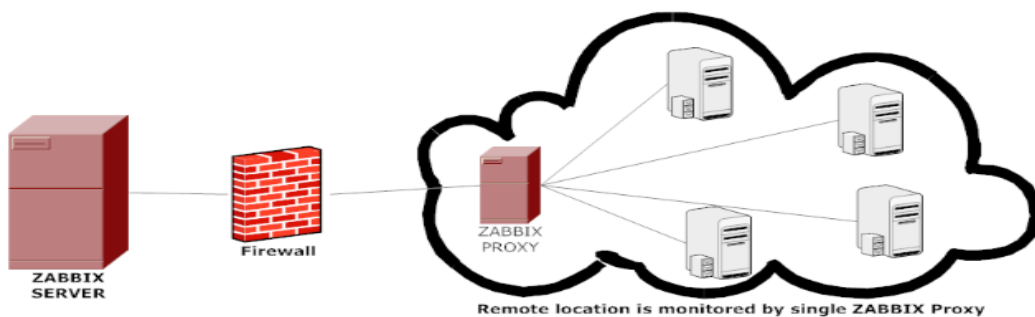
2.1.3. Kiến trúc của hệ thống giám sát Zabbix

Server Zabbix: Chịu trách nhiệm kiểm tra hoạt động dịch vụ từ xa, thu thập thông tin, lưu trữ dữ liệu,... làm nguồn để cài đặt và phát triển các thao tác giám sát hệ thống, đưa ra báo cáo



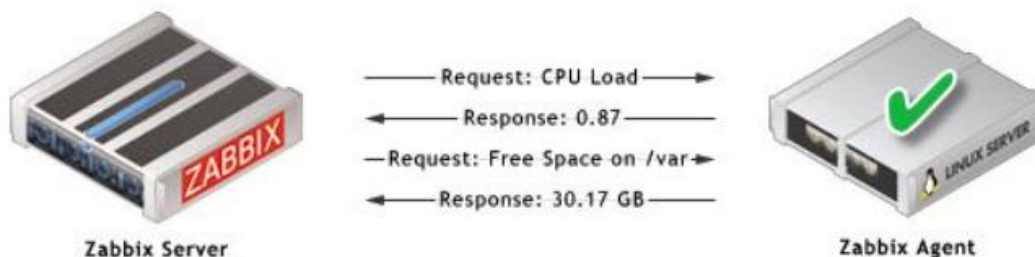
Hình 8 Zabbix Server

Zabbix Proxy: là một server có vai trò quản lý các lớp mạng và hệ thống từ xa. Bằng cách thu thập các thông tin thiết bị mạng rồi chuyển tiếp cho máy chủ chính của Zabbix



Hình 9 Zabbix Proxy

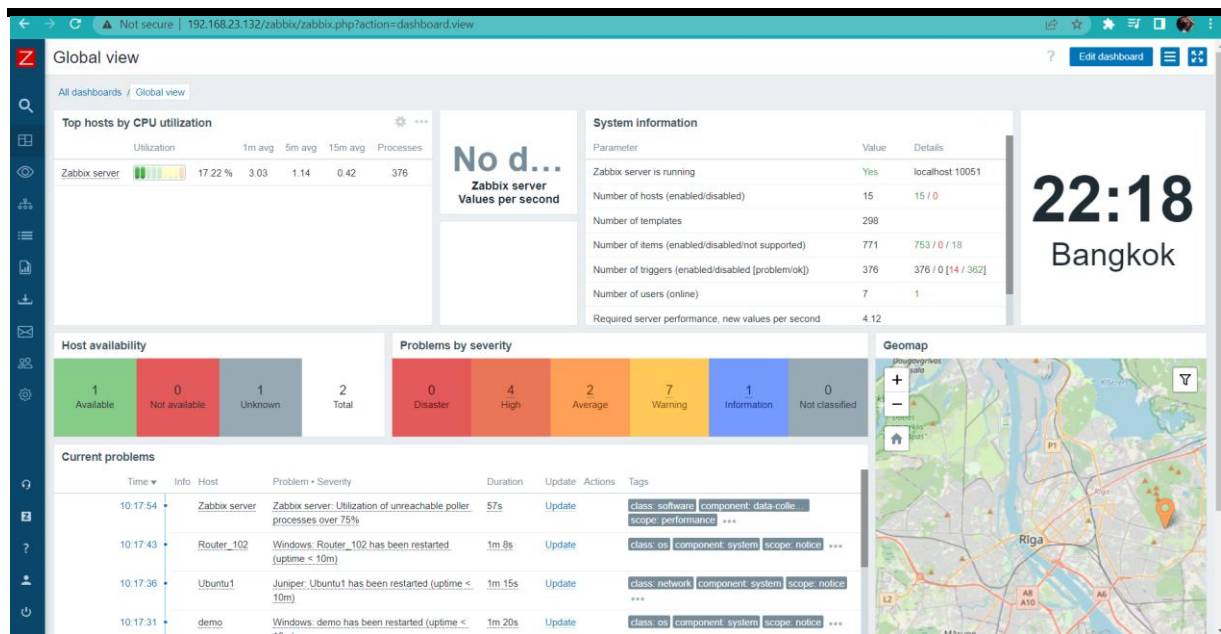
Zabbix Agent: chương trình để cài đặt lên các máy chủ hoặc thiết bị phía client. Qua đó, hệ thống sẽ lấy các thông tin cần thiết từ máy chủ khách hàng qua kết nối của máy chủ chính Zabbix, nhằm kiểm tra các tình trạng hệ thống hoặc theo nhu cầu quản trị viên.



Hình 10 Zabbix agent

Giao diện Web: được phát triển trên nền tảng mã nguồn PHP, giao diện metro. Theo như nhận định của người sử dụng, giao diện của Zabbix tinh tế, dễ sử dụng, bắt mắt, các số liệu được sắp xếp một cách trật tự, dễ chịu với người xem.

PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



Hình 11 Giao diện Web

2.1.4. Tính năng của Zabbix

Các chức năng của Zabbix rất linh hoạt, nó có thể được cấu hình để theo dõi, giám sát thiết bị mạng, máy chủ theo cách ta muốn. Nó cũng có một cơ chế để tự động phản ứng với các vấn đề, và một hệ thống cảnh báo mạnh. Tất cả điều này được dựa trên một hệ thống định nghĩa các đối tượng rõ ràng.

Khả năng giám sát: Zabbix có cấu hình tập trung, các thông tin giám sát được tập trung vào một cơ sở dữ liệu. Zabbix có khả năng sử dụng các proxy với số lượng không giới hạn, số nút đó có thể lên tới hàng ngàn.

Khả năng mở rộng: Các thí nghiệm cho thấy nó có khả năng xử lý quản trị tới 100.000 thiết bị và máy chủ. Số lượng thông tin, dịch vụ giám sát có thể lên tới 1.000.000

- Hỗ trợ giám sát thời gian thực: Zabbix có thể cảnh báo ngay tới người quản trị viên

khi hệ thống được giám sát có sự cố gì thông qua mail, SMS... Hơn nữa Zabbix còn có hồ sơ về các thông tin giám sát

Khả năng hiển thị kết quả bằng đồ thị, biểu đồ giúp người dùng có thể dễ dàng giám sát.

- Khả năng nhập và xuất cơ sở dữ liệu thông qua XML.

Khả năng tự động phát hiện: Người dùng có thể tạo ra các luật dựa trên nó Zabbix có thể tự động phát hiện ra các địa chỉ IP, các dịch vụ hoặc các thiết bị SNMP để thực hiện việc giám sát.

Tính linh hoạt: Zabbix hỗ trợ cả IPv4 và IPv6, các Zabbix agent có khả năng cài đặt trên nhiều nền tảng khác nhau.

Khả năng giám sát các thiết bị không hỗ trợ cài đặt Zabbix agent: Zabbix có khả năng giám sát các thiết bị hỗ trợ IPMI, SNMP v1,2,3,4.

- Khả năng bảo mật: Zabbix hỗ trợ người dùng một số tính năng, nó cung cấp khả năng chứng thực của địa chỉ IP.

Quản trị các chức năng: Ta có thể chạy lệnh ping, traceroute trên một chuỗi các máy chủ, các thiết bị được quản trị.

2.1.5. Cấu trúc thư mục

- docs: Thư mục chứa file hướng dẫn pdf

src: Thư mục chứa tất cả source cho các tiến trình Zabbix.

+ src/zabbix_server: Thư mục chứa file tạo và source cho zabbix_server.

+ src/zabbix_agent: Thư mục chứa file tạo và source cho zabbix_agent và zabbix_agentd.

+ src/zabbix_get: Thư mục chứa file tạo và source cho zabbix_get.

+ src/zabbix_sender: Thư mục chứa file tạo và source cho zabbix_sender.

include: Thư mục chứa các thư viện Zabbix.

misc

+ misc/init.d: Thư mục chứa các tập lệnh khởi động trên các nền khác nhau.

frontends

+ frontends/php: Thư mục chứa các file PHP.

- create: Thư mục chứa các tập lệnh SQL để tạo cơ sở dữ liệu ban đầu.

+ create/schema: Thư mục tạo biểu đồ cơ sở dữ liệu.

+ create/data: Thư mục chứa dữ liệu cho việc tạo cơ sở dữ liệu ban đầu.

- upgrades: thư mục chứa các thủ tục nâng cấp cho phiên bản khác nhau của Zabbix.

2.2. Phân tích yêu cầu:

Phần mềm giám sát các thiết bị mạng dựa trên Zabbix cần đáp ứng những yêu cầu sau:

- Tích hợp với Zabbix API:

- Phần mềm cần hỗ trợ kết nối và tương tác với Zabbix API để truy xuất dữ liệu giám sát và thực hiện các thao tác quản lý.

- Giám sát mạng và các thành phần liên quan:

- Hỗ trợ giám sát các thiết bị mạng như máy chủ, router, switch, firewall, thiết bị lưu trữ, ứng dụng và các thành phần khác.

- Theo dõi trạng thái kết nối mạng, tình trạng hoạt động, tài nguyên hệ thống (CPU, bộ nhớ, đĩa cứng), băng thông và các chỉ số quan trọng khác.
- **Cấu hình và quản lý thiết bị:**
 - Cung cấp giao diện để thêm, sửa đổi và xóa thiết bị từ Zabbix. Tạo danh sách thiết bị cần được giám sát và phân nhóm chúng để quản lý dễ dàng.
 - Hỗ trợ cấu hình các thuộc tính của thiết bị như địa chỉ IP, tên miền, cổng kết nối, giao thức, phương pháp xác thực và thông tin liên quan khác.
- **Cấu hình giám sát:**
 - Cho phép cấu hình các mục giám sát để thu thập dữ liệu từ các host.
 - Cung cấp khả năng tùy chỉnh các mục giám sát để thu thập các thông số cụ thể theo nhu cầu.
- **Lưu trữ dữ liệu và báo cáo:**
 - Lưu trữ dữ liệu giám sát trong một cơ sở dữ liệu để phân tích và xem lại.
- **Giao diện người dùng:**
 - Cung cấp giao diện trực quan và dễ sử dụng để quản lý và giám sát mạng
- **Bảo mật và phân quyền:**
 - Đảm bảo an ninh dữ liệu và thông tin giám sát mạng.
 - Hỗ trợ việc xác thực và ủy quyền người dùng để giới hạn quyền truy cập và thao tác..

2.3. Phân tích các thành phần của hệ thống:

Các thành phần chính của phần mềm giám sát các thiết bị mạng dựa trên Zabbix bao gồm:

- **Zabbix Server:**
 - Đây là thành phần trung tâm của hệ thống, nơi quản lý và xử lý dữ liệu giám sát. Máy chủ Zabbix thu thập dữ liệu từ các thiết bị mạng và lưu trữ chúng trong cơ sở dữ liệu. Nó cũng xử lý các cảnh báo và cung cấp giao diện người dùng để tương tác với hệ thống.
- **SNMP Agents:**
 - Đây là phần mềm hoặc phần cứng được cài đặt trên các thiết bị mạng để thu thập thông tin và hỗ trợ quản lý mạng thông qua giao thức SNMP.
 - Các SNMP Agents thu thập thông tin về tình trạng và hiệu suất của thiết bị mạng, chẳng hạn như thông tin về tài nguyên hệ thống (bộ nhớ, CPU). Các agent này cung cấp các dữ liệu này cho Zabbix Server.
- **Cơ sở dữ liệu:**
 - Cơ sở dữ liệu là nơi lưu trữ dữ liệu giám sát thu thập từ các thiết bị mạng. Zabbix sử dụng cơ sở dữ liệu để lưu trữ thông tin về host, item, trigger, cảnh báo, lịch sử giám sát và các dữ liệu khác.

MySQL là cơ sở dữ liệu phổ biến được sử dụng với Zabbix, nhưng cũng có thể sử dụng PostgreSQL hoặc Oracle.

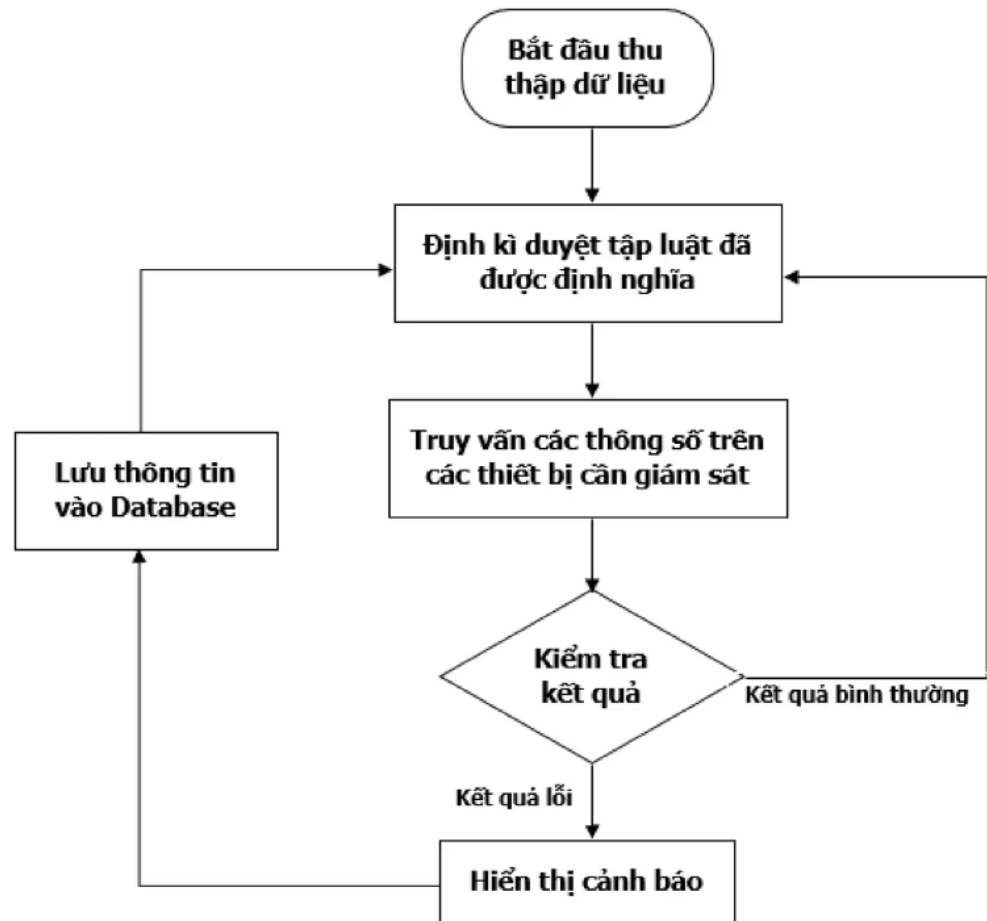
- **Web Interface:**

- Giao diện người dùng cung cấp một giao diện đồ họa để người dùng tương tác với hệ thống giám sát. Qua giao diện người dùng, người dùng có thể xem thông tin về host, item, trigger, cảnh báo, và các báo cáo giám sát. Giao diện người dùng cũng cung cấp các chức năng quản lý và cấu hình hệ thống.

2.4. Phân tích cơ chế hoạt động của hệ thống:

Cơ chế hoạt động của hệ thống có thể mô tả như sau:

- **Thu thập dữ liệu:** SNMP Agent sẽ thu thập dữ liệu từ host cần giám sát theo các thiết lập cấu hình. Dữ liệu này bao gồm thông tin về tình trạng hoạt động, tài nguyên sử dụng, băng thông và các chỉ số khác liên quan đến mạng. Dữ liệu thu thập từ SNMP Agent sẽ được gửi đến máy chủ Zabbix thông qua giao thức truyền thông như TCP/IP hoặc HTTP.
- **Xử lý dữ liệu:** Máy chủ nhận dữ liệu từ SNMP Agent và tiến hành xử lý. Quá trình này bao gồm:
 - Lưu trữ dữ liệu: Dữ liệu thu thập được sẽ được lưu trữ trong cơ sở dữ liệu của Zabbix. Zabbix sử dụng một cơ sở dữ liệu quan hệ như MySQL, PostgreSQL hoặc Oracle để lưu trữ dữ liệu.
 - So sánh với ngưỡng báo cáo: Zabbix so sánh dữ liệu thu thập được với các ngưỡng cảnh báo đã được định nghĩa trước. Nếu dữ liệu vượt quá hoặc dưới ngưỡng cảnh báo, Zabbix sẽ tạo ra các cảnh báo tương ứng.
- **Hiển thị dữ liệu:** Dữ liệu được lưu trữ trong cơ sở dữ liệu sẽ được hiển thị dưới dạng báo cáo và biểu đồ trên giao diện web. Người dùng có thể xem các báo cáo và biểu đồ này để phân tích và đưa ra các quyết định về việc quản lý hệ thống.
- **Hiển thị cảnh báo:** Người dùng có thể xem các cảnh báo được tạo ra trên giao diện web.



Hình 12 Cơ chế hoạt động của hệ thống

2.5. Các khái niệm cơ bản của hệ thống giám sát:

- Item: là thành phần cơ bản được sử dụng để thu thập dữ liệu từ các thiết bị, mang một số thuộc tính cơ bản sau:
 - o Name: tên của item
 - o Type: loại item, bao gồm Zabbix agent, SNMP agent,...
 - o Type of information: loại dữ liệu của dữ liệu thu thập được, bao gồm các loại: Numeric float, Numeric unsigned, Character (ký tự), Log (nhật ký), Text (văn bản)
 - o Units: đơn vị của dữ liệu thu thập được
 - o Update interval: thời gian cập nhật dữ liệu
 - o History storage period: thời gian lưu trữ dữ liệu (chi tiết)
 - o Trends storage period: thời gian lưu trữ dữ liệu
- Trigger: là một thành phần quan trọng được sử dụng để xác định và cảnh báo về các sự kiện hoặc trạng thái không mong muốn trong hệ thống giám

sát. Trigger kiểm tra các giá trị từ các item trong Zabbix để xác định liệu có sự việc xảy ra ngoài mong đợi hay không, và nếu có, nó sẽ kích hoạt một cảnh báo. Một trigger mang một số thuộc tính cơ bản sau:

- Name: tên của trigger
 - Expression: là biểu thức điều kiện được định nghĩa để xác định trạng thái không mong muốn. Biểu thức này sử dụng các giá trị từ các item và các toán tử logic để kiểm tra điều kiện.
 - Các toán tử logic được sử dụng trong expression bao gồm: =, !=, >, >=, <, <=, and, or, not,...
 - Các hàm được sử dụng trong expression bao gồm:
 - last (item:key): lấy giá trị mới nhất của item
 - last (item:key, #d): lấy giá trị mới thứ d của item
 - last (item:key, #m): lấy giá trị mới nhất của item trong khoảng thời gian m
 - Severity: mức độ nghiêm trọng của trigger, bao gồm 5 mức độ: not classified, information, warning, average, high, disaster
- Template: Là một mẫu chuẩn, đã được định nghĩa sẵn các item, triggers. Nó vô cùng thuận tiện khi triển khai giám sát nhiều host có những thành phần cần giám sát giống nhau. Vì vậy chỉ cần tạo 1 template là có thể áp dụng cho nhiều host khác nhau.
 - Template Group: được sử dụng để tổ chức và quản lý các mẫu (templates) của các thiết bị hoặc ứng dụng khác nhau. Template Group cho phép bạn nhóm các mẫu lại với nhau dựa trên một tiêu chí chung, chẳng hạn như nhóm các mẫu cho các máy chủ Windows, các mẫu cho các thiết bị mạng, các mẫu cho các ứng dụng cụ thể, v.v.
 - Host: đại diện cho một thiết bị, máy chủ hoặc ứng dụng cần giám sát. Host là đơn vị cơ bản để thu thập dữ liệu và theo dõi trạng thái của các thành phần trong mạng hoặc hệ thống. Các thuộc tính cơ bản của host bao gồm:
 - Host name: tên của host, đây là định danh duy nhất để xác định host trong hệ thống giám sát
 - Template: danh sách các template được áp dụng cho host
 - Items: danh sách các item được áp dụng cho host
 - Triggers danh sách các trigger được áp dụng cho host
 - IP address: địa chỉ ip của host để kết nối và thu thập dữ liệu từ đó
 - Port: cổng kết nối

- Host group: danh sách các host group mà host thuộc về
- Host Group: Là một tập hợp các host, host group có thể chứa các host group khác, tạo thành một cấu trúc cây. Host Group giúp cho việc quản lý các host dễ dàng hơn.

2.6. Các chức năng cơ bản của hệ thống giám sát:

- Tạo và quản lý các host
- Tạo và quản lý các host group
- Tạo và quản lý các item
- Tạo và quản lý các trigger
- Tạo và quản lý các template
- Tạo và quản lý các template group
- Tạo và quản lý các user (Super Admin)
- Cảnh báo khi có sự cố xảy ra
- Hiển thị các biểu đồ thống kê

CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

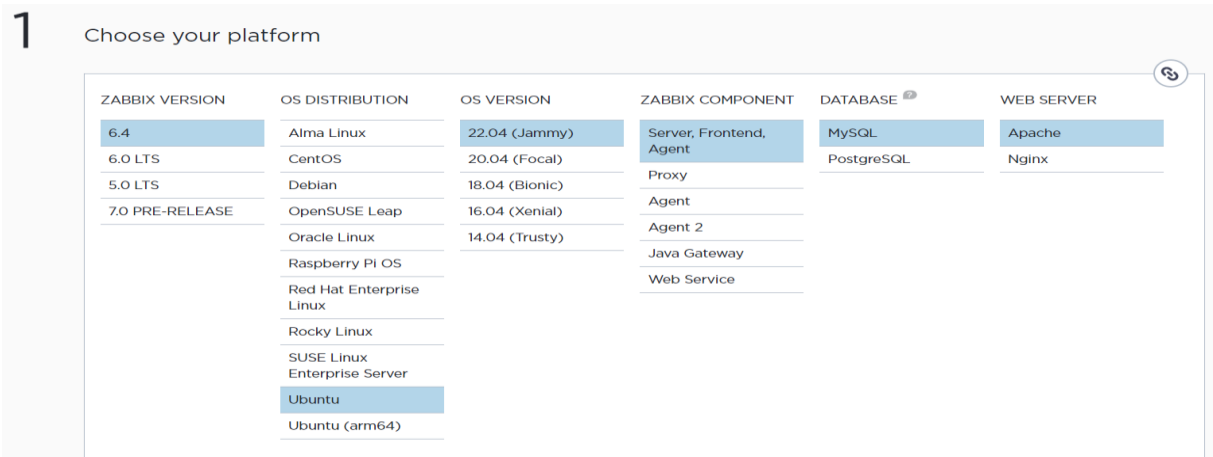
3.1. Môi trường triển khai

STT	Host name	IP	OS	Note
1.	Zabbixserver	174.31.38.247	Ubuntu 22.04	Máy chủ giám sát thiết bị mạng, ...
2.	Pc1	192.168.7.1	Window 11	Máy tính cá nhân, đối tượng bị giám sát
3.	Router1	192.168.137.1	Cisco IOS	Thiết bị mạng, đối tượng bị giám sát

3.2. Triển khai cấu hình

3.2.1. Cấu hình Zabbix Server trên Zabbixsrv

Ở đây Em cài đặt Zabbix trên hệ điều hành Ubuntu nên chọn nền tảng như ảnh dưới đây:



B1: Mở terminal ở Ubuntu và cấu hình theo hướng dẫn

B2 Cài đặt Zabbix repository

```
# wget https://repo.zabbix.com/zabbix/6.4/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.4-1+ubuntu22.04_all.deb
# dpkg -i zabbix-release_6.4-1+ubuntu22.04_all.deb
# apt update
```

B3 Cài đặt Zabbix server, frontend, agent

```
# apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-sql-scripts zabbix-agent
```

B4 Tạo cơ sở dữ liệu ban đầu:

Điều kiện: đã cài đặt MariaDB, mySql (nếu chưa cài đặt thì cài đặt theo hướng dẫn [ở đây](#))

PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

Chạy phần sau trên máy chủ cơ sở dữ liệu của bạn.

```
# mysql -uroot -p
password
mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
mysql> create user zabbix@localhost identified by 'password';
mysql> grant all privileges on zabbix.* to zabbix@localhost;
mysql> set global log_bin_trust_function_creators = 1;
mysql> quit;
```

Trên máy chủ Zabbix, nhập lược đồ và dữ liệu ban đầu. Bạn sẽ được nhắc nhập mật khẩu mới tạo.

```
# zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4 -uzabbix -p zabbix
```

Tắt tùy chọn log_bin_trust_function_creators sau khi nhập lược đồ cơ sở dữ liệu.

```
# mysql -uroot -p
password
mysql> set global log_bin_trust_function_creators = 0;
mysql> quit;
```

B5 Cấu hình cơ sở dữ liệu cho máy chủ Zabbix

Chỉnh sửa tệp /etc/zabbix/zabbix_server.conf

```
DBPassword=password
```

B6 Kích hoạt quy trình Zabbix server và agent

Khởi động các quy trình của agent và server và khởi động khi khởi động hệ thống

```
# systemctl restart zabbix-server zabbix-agent apache2
# systemctl enable zabbix-server zabbix-agent apache2
```

B7 Mở trang web giao diện người dùng Zabbix.

URL mặc định ho giao diện người dùng Zabbix khi sử dụng máy chủ web Apache là <http://host/zabbix>

PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

Host: là địa chỉ ip trên Server Ubuntu (gõ lệnh :ip addr show)

3.2.2. *Cấu hình deploy Web Interface tương tác với Zabbix Server trên Zabbixsrv*

3.2.2.1 Cài đặt NodeJS và Npm (nếu chưa có):

Bước 1: Kích hoạt kho lưu trữ NodeSource

```
curl -sL https://deb.nodesource.com/setup_20.x | sudo -E bash -
```

Bước 2: Cài đặt NodeJs

```
sudo apt install nodejs -y
```

3.2.2.2 Cài đặt Apache2 (nếu chưa có):

```
sudo apt-get install apache2
```

B1. Lấy source code của Web Interface từ Github về:

```
git clone https://github.com/dinhvan2310/pbl4\_snmp\_ui.git
```

B2. Cài đặt các gói phụ thuộc:

```
cd pbl4_snmp_ui  
npm install
```

B3. Deploy Web Interface lên Apache2:

```
// build source code  
sudo npm run build  
  
// copy source code vào thư mục /var/www/html/  
sudo cp -r build/* /var/www/html/
```

B4. Cấu hình Apache2:

Bước 4.1: Try cập vào file 000-default.conf

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Bước 4.2: Thay đổi DocumentRoot thành /var/www/html/build

```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/build
```

Bước 4.3: Truy cập vào file apache2.conf

```
sudo nano /etc/apache2/apache2.conf
```

Bước 4.4: Thêm vào cuối file

```
<Directory /var/www/html>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

Bước 4.5: Tạo file .htaccess trong thư mục /var/www/html/build với nội dung sau:

```
Options -MultiViews
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^ index.html [QSA,L]
```

Bước 4.6: Khởi động lại Apache2

```
sudo systemctl restart apache2
```

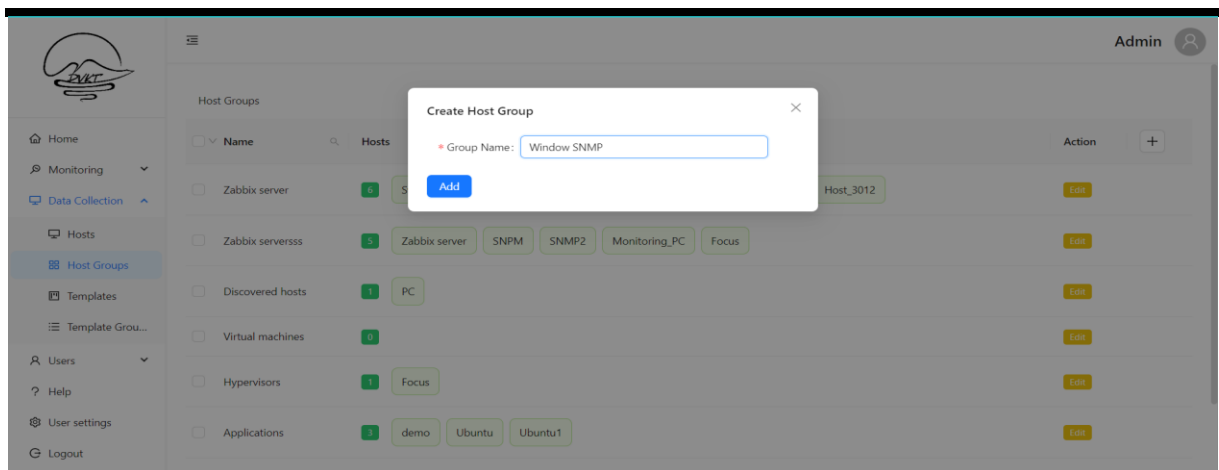
3.2.3. Cấu hình Web Interface thiết lập các tham số để giám sát PC1 và Router1

Khi hệ thống hoạt động, việc theo dõi trạng thái các máy chủ đang trong tình trạng như thế nào, các dịch vụ quan trọng có được chạy hay không? Hay trạng thái các interfaces cần thiết trên các thiết bị đầu cuối ấy là 1 điều bắt buộc cần giám sát. Thiết lập từng bước giám sát hệ thống mạng: tạo host, host group và template cho các host trong hệ thống

3.2.3.1 Cấu hình giám sát PC1:

B1. Tạo host group

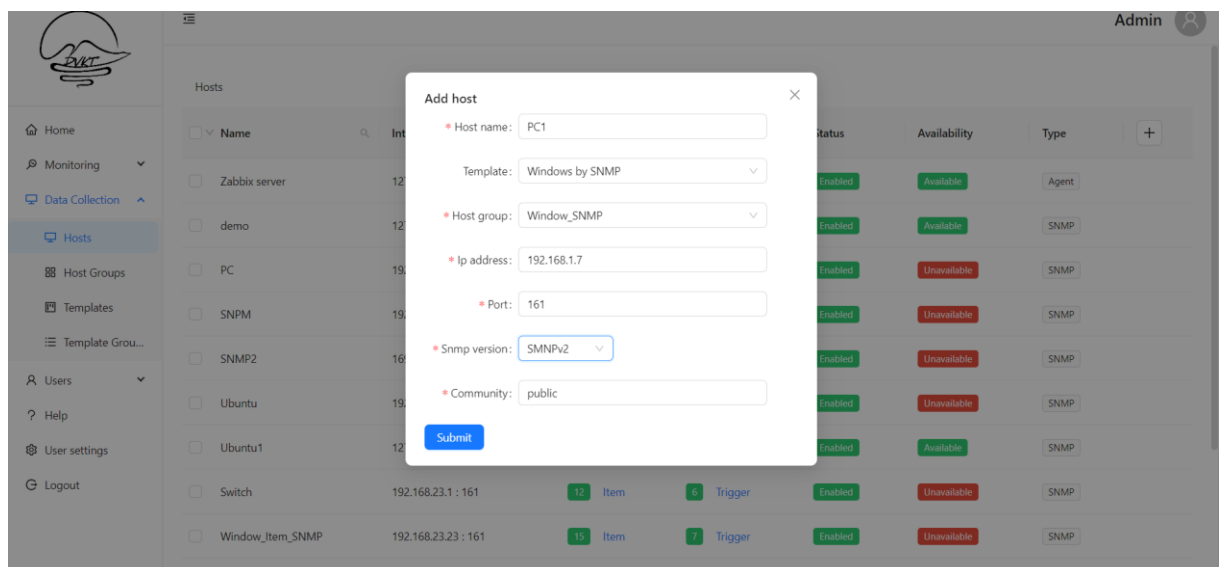
PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



Hình 13 : Tạo host group Window SNMP

B2. Tạo host và thêm vào host group, ở đây host PC1 được add vào host group Window SNMP vừa tạo, điền các tham số như hình bên dưới:

- Host name: PC1
- Template: Window by SNMP
- Host group: Window SNMP
- Ip address: 192.168.1.7
- Port: 161
- Snmp version: SNMPv2
- Community: public



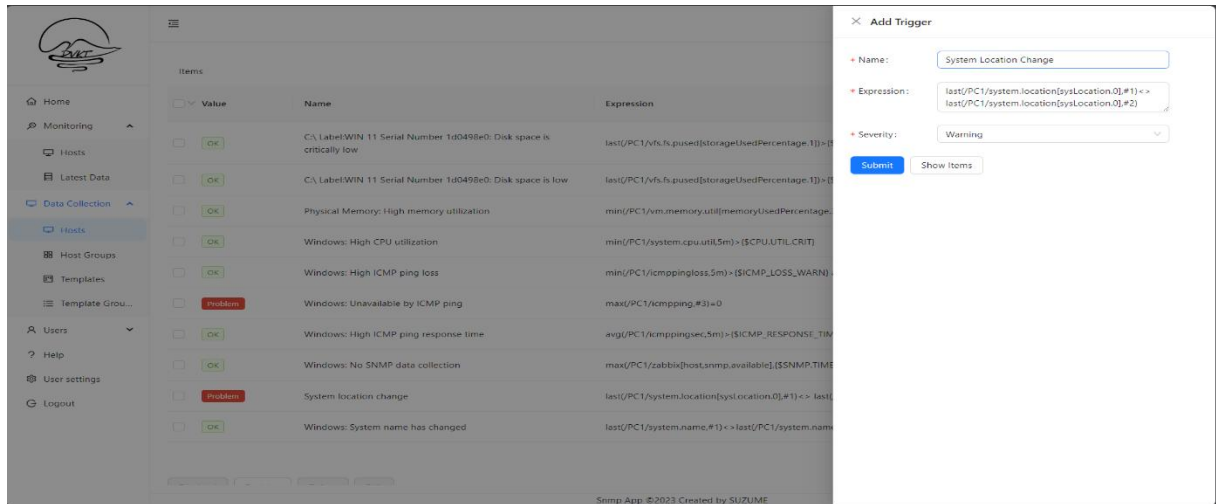
Hình 14 Tạo host PC1

B3. Tạo trigger sự kiện System location của host thay đổi với các cấu hình sau:

- Name: System Location Change

PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

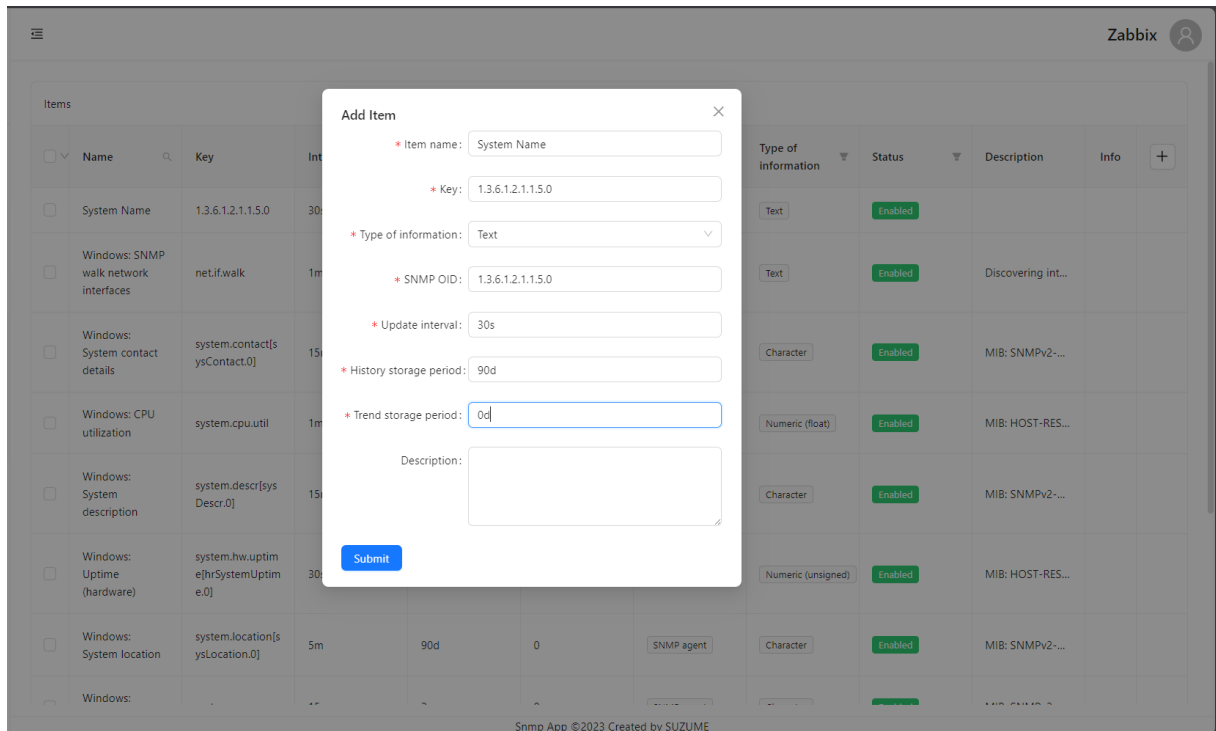
- Expression:
`last(/PC1/system.location[sysLocation.0],#1)<>last(/PC1/system.location[sysLocation.0],#2)`
- Severity: Warning



Hình 15 Tạo mới một Trigger

B4. Tạo mới item giám sát tên của host với cấu hình sau:

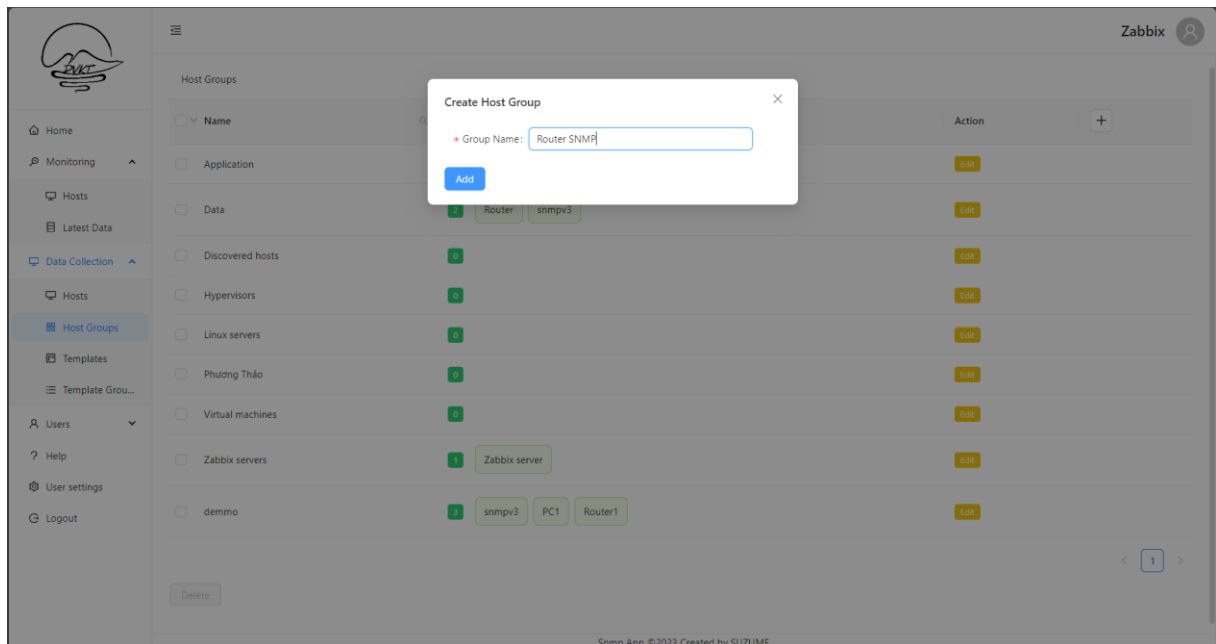
- Item name: System Name
- Key: 1.3.6.1.2.1.1.5.0
- Type of information: Text
- SNMP OID: 1.3.6.1.2.1.1.5.0
- Update Interval: 30s
- History storage period: 90d
- Trend storage period: 0d



Hình 16 Tạo mới một Item

3.2.3.2 Cấu hình giám sát Router1:

B1. Tạo host group: Router SNMP



Hình 17 : Tạo host groups Router SNMP

PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

B2. Tạo host và thêm vào host group, ở đây host Router1 được tạo và được thêm vào host group Router SNMP vừa tạo, điền các tham số như hình bên dưới

Add host

* Host name: Router1

Template: Cisco IOS by SNMP

* Host group: Router SNMP

* Ip address: 192.168.137.1

* Port: 161

* Snmp version: SMNPv3

Security level: authPriv

Context name:

Security name: user1

Auth protocol: MD5

Auth pass: password1

Priv protocol: AES128

Priv pass: privpassword1

Submit

Hình 8 Tạo host Router1

3.3. Triển khai giám sát

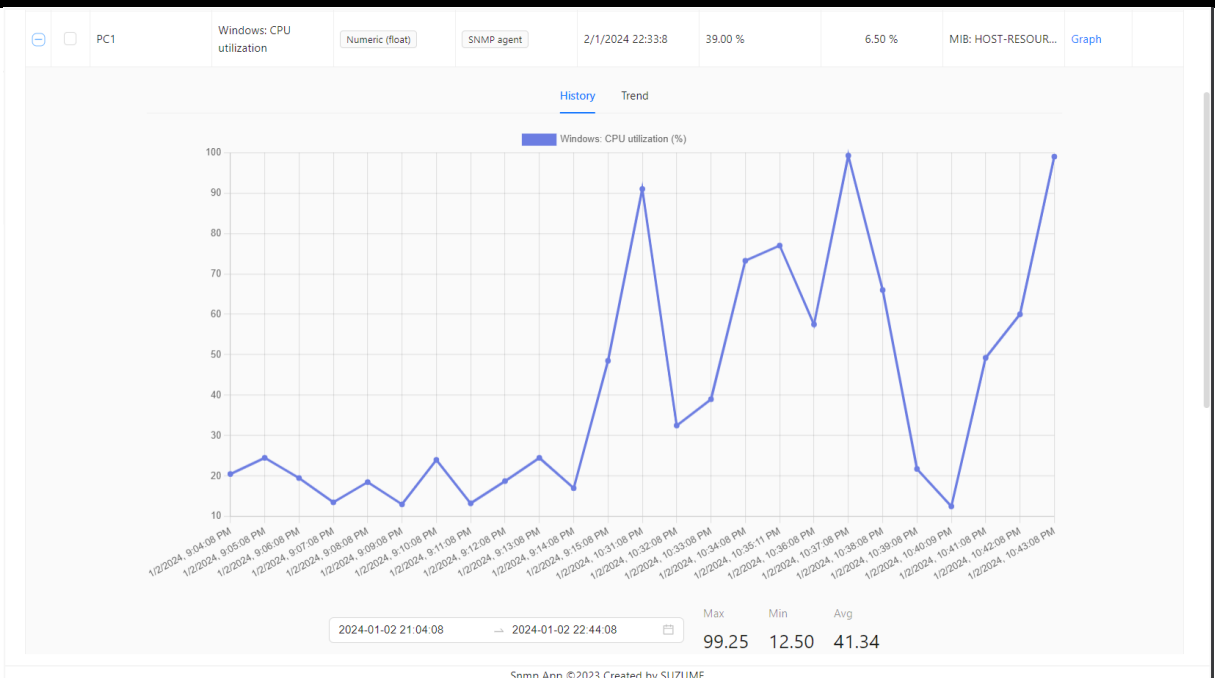
3.3.1. Giám sát PC1 thông qua Web Interface

Host	Name	Type of information	Type	Last check	Last value	Change	Description	Graph	Info
PC1	Windows: ICMP ping	Numeric (unsigned)	Simple check	2/1/2024 15:58:8	0	0		Graph	
PC1	Windows: ICMP loss	Numeric (float)	Simple check	2/1/2024 15:58:8	100.00 %	0.00 %		Graph	
PC1	Windows: ICMP response time	Numeric (float)	Simple check	2/1/2024 15:58:8	0.00 s	0.00 s		Graph	
PC1	Windows: SNMP walk network interfaces	Text	SNMP agent				Discovering interfac...		
PC1	Windows: SNMP traps (fallback)	Log	SNMP TRAP				The item is used to ...		
PC1	Windows: System contact details	Character	SNMP agent	2/1/2024 12:31:8	VANN		MIB: SNMPv2-MIB ...	Graph	
PC1	Windows: CPU utilization	Numeric (float)	SNMP agent	2/1/2024 15:58:8	21.00 %	14.75 %	MIB: HOST-RESOUR...	Graph	
PC1	Windows: System description	Character	SNMP agent	2/1/2024 12:31:8	Hardware: Intel64 F...		MIB: SNMPv2-MIB ...	Graph	
PC1	Windows: Uptime (hardware)	Text	SNMP agent	2/1/2024 15:58:8	0d 5h 51m 3s		MIB: HOST-RESOUR...	Graph	
PC1	Windows: System location	Character	SNMP agent	2/1/2024 12:31:8	QUANGNAM		MIB: SNMPv2-MIB ...	Graph	

Hình 18 Bảng thống kê các Items của PC1

- Giám sát tài nguyên CPU của PC1:

PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



Hình 19 Đồ thị dung lượng CPU đã sử dụng

Dựa vào biểu đồ trên chúng ta quan sát được lúc 9h04 ngày 1/2/2024 CPU sử dụng 20%, đến 10h41 ngày 1/2/2024 CPU sử dụng là 50%.

○ Giám sát tài nguyên RAM:

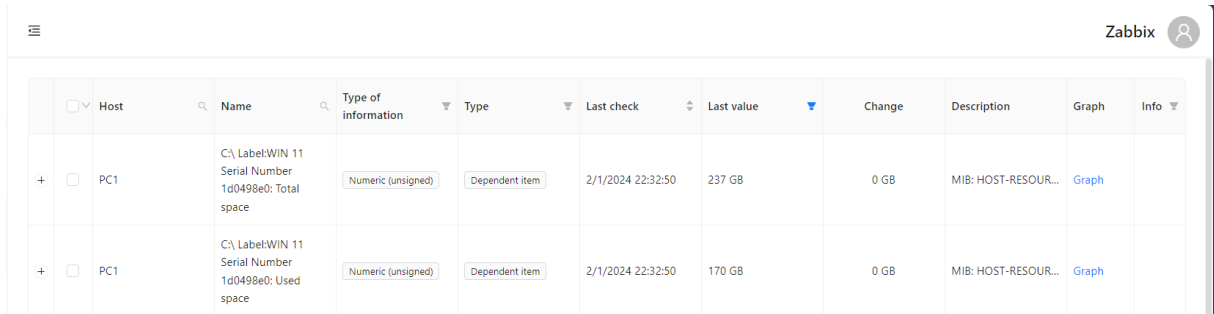


Hình 20 Đồ thị dung lượng Ram

PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

Dựa vào biểu đồ trên chúng ta quan sát được lúc 8h51 ngày 1/2/2024 dung lượng ram đang dùng là 6.735GB/ 8GB

- Giám sát tài nguyên Disk:



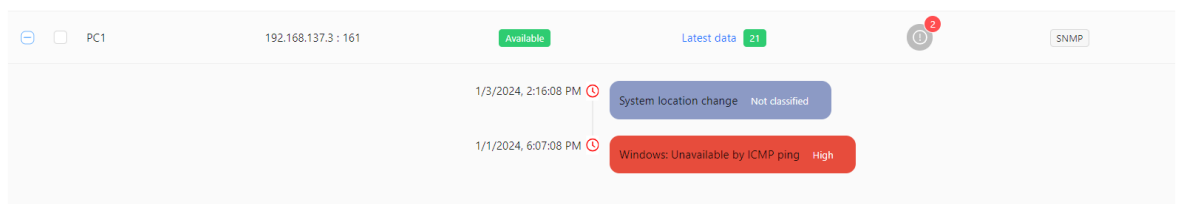
The screenshot shows the Zabbix interface for monitoring disk space. It displays a table with columns for Host, Name, Type of information, Type, Last check, Last value, Change, Description, Graph, and Info. Two rows are visible, both for host PC1. The first row shows 'Total space' as 237 GB. The second row shows 'Used space' as 170 GB. Both rows indicate a change of 0 GB and are described as 'MIB: HOST-RESOUR...'. Each row has a 'Graph' link.

	Host	Name	Type of information	Type	Last check	Last value	Change	Description	Graph	Info
+	PC1	C:\ Label:WIN 11 Serial Number 1d0498e0: Total space	Numeric (unsigned)	Dependent item	2/1/2024 22:32:50	237 GB	0 GB	MIB: HOST-RESOUR...	Graph	
+	PC1	C:\ Label:WIN 11 Serial Number 1d0498e0: Used space	Numeric (unsigned)	Dependent item	2/1/2024 22:32:50	170 GB	0 GB	MIB: HOST-RESOUR...	Graph	

Hình 21 Bảng giám sát dung lượng ổ cứng

Hình trên cho ta biết tổng dung lượng ổ C là 237GB, dung lượng ổ C đã được sử dụng là 170GB

- Cảnh báo sự cố:



Hình 22 Màn hình hiển thị cảnh báo của Host PC1

Hình trên cho thấy vị trí của PC1 đã thay đổi nên cảnh báo xuất hiện.

3.3.2. Giám sát Router1 thông qua Web Interface

Home

Monitoring

Hosts

Latest Data

Data Collection

Users

Help

User settings

Logout

Host

Name

Type of information

Type

Last check

Last value

Change

Description

Graph

Info

Router1	Cisco IOS: ICMP loss	Numeric (float)	Simple check	2/1/2024 15:59:11	100.00 %	0.00 %		Graph	
Router1	I/O Cont Inlet: Temperature status	Numeric (unsigned)	Dependent item	2/1/2024 14:50:23	1	0	MIB: CISCO-ENVM...	Graph	
Router1	I/O Cont Outlet: Temperature status	Numeric (unsigned)	Dependent item	2/1/2024 14:50:23	1	0	MIB: CISCO-ENVM...	Graph	
Router1	NPE Inlet: Temperature status	Numeric (unsigned)	Dependent item	2/1/2024 14:50:23	1	0	MIB: CISCO-ENVM...	Graph	
Router1	NPE Outlet: Temperature status	Numeric (unsigned)	Dependent item	2/1/2024 14:50:23	1	0	MIB: CISCO-ENVM...	Graph	
Router1	I/O Cont Inlet: Temperature	Numeric (float)	Dependent item	2/1/2024 14:50:23	22.00 °C	0.00 °C	MIB: CISCO-ENVM...	Graph	
Router1	I/O Cont Outlet: Temperature	Numeric (float)	Dependent item	2/1/2024 14:50:23	22.00 °C	0.00 °C	MIB: CISCO-ENVM...	Graph	
Router1	NPE Inlet: Temperature	Numeric (float)	Dependent item	2/1/2024 14:50:23	22.00 °C	0.00 °C	MIB: CISCO-ENVM...	Graph	
Router1	NPE Outlet: Temperature	Numeric (float)	Dependent item	2/1/2024 14:50:23	22.00 °C	0.00 °C	MIB: CISCO-ENVM...	Graph	
Router1	#1: CPU utilization	Numeric (float)	Dependent item	2/1/2024 14:50:26	13.00 %	0.00 %	MIB: CISCO-PROCE...	Graph	

1

2

3

Snmp App ©2023 Created by SUZUME

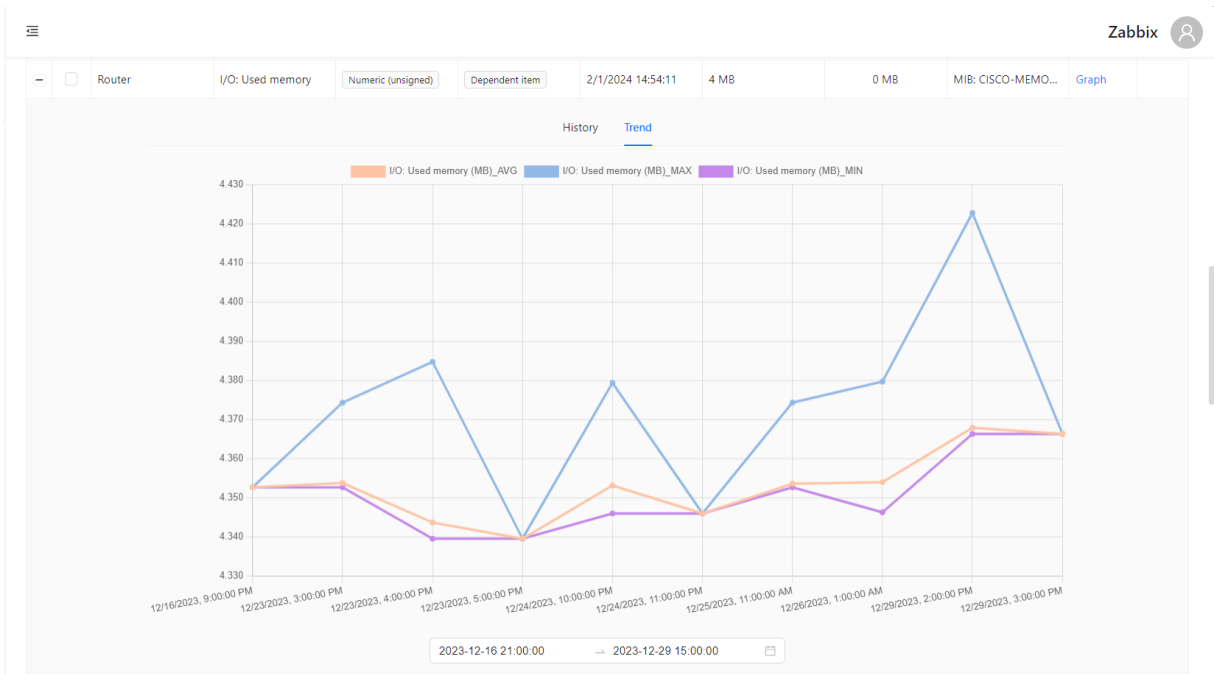
Hình 23 Bảng thống kê các Items của Router1

○ Giám sát tài nguyên CPU của Router



Hình 24 Đồ thị dung lượng CPU đã sử dụng

○ Giám sát tài nguyên RAM



Hình 25 Đồ thị dung lượng RAM đã sử dụng

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong đồ án này em đã nghiên cứu, tìm hiểu xây dựng hệ thống giám sát mạng dựa trên phần mềm nguồn mở Zabbix.

Đồ án thực hiện thành công mô hình giám sát mạng sử dụng phần mềm Zabbix đáp ứng được cơ bản yêu cầu quản trị mạng của 1 hệ thống bao gồm:

+ Quản lý được dữ liệu quan trọng, và các thông tin cơ bản của các thiết bị trong hệ thống kịp thời chính xác.

+ Xây dựng thành công cơ chế cảnh báo online 1 và hiện thị cảnh báo trên màn hình tiện ích quan trọng cho việc quản trị.

Em đã áp dụng các kiến thức của các môn học như điện toán đám mây, mạng máy tính và Hệ điều hành vào mô hình và thực tế. Đã học hỏi được thêm nhiều kinh nghiệm về cách thức tổ chức, xây dựng hệ thống giám sát cũng như quy hoạch hệ thống . Tuy nhiên, do thời gian và khả năng có hạn, nên em chưa đi sâu tìm hiểu được thêm những vấn đề cần thiết của hệ thống. Em đã cố gắng nhưng mô hình mới chỉ dừng ở mức độ theo dõi, giám sát máy chủ như giám sát tài nguyên máy, dung lượng bộ nhớ, tình trạng của host.

Trong thời gian tới em sẽ phát triển và nghiên cứu sâu hơn về hệ thống giám sát mạng Zabbix và các công cụ hỗ trợ giám sát mạng, giám sát sâu hơn những vấn đề cần thiết của hệ thống. Phát triển các chức năng trên Zabbix như: giám sát hạ tầng mạng bao gồm các thiết bị router, switch, firewall,... Cảnh báo qua SMS.

TÀI LIỆU THAM KHẢO

- [1] <https://vinahost.vn/snmp-la-gi/>
- [2] [Tên chủ sở hữu, Tên bài viết, url, ngày truy cập](#)
- [3] <https://mdungblog.wordpress.com/2020/01/06/ly-thuyet-giao-thuc-snmp-toan-tap/>
- [4] <https://www.zabbix.com/documentation/current/en/manual/api>
- [5] <https://thegioifirewall.com/huong-dan-cau-hinh-giam-sat-switch-cisco-voi-giao-thuc-snmp-tren-zabbix/>

Link source code : https://github.com/dinhvan2310/pbl4_snmp_ui