TÌM HIỂU VÀ TRIỂN KHAI KAMAILIO

Nhóm L52

1. **Danh sách thành viên và công việc**

|  |  |  |  |
| --- | --- | --- | --- |
| Họ và tên | Mã SV | Công việc | Tiến độ |
| Nguyễn Thế Vinh | 175A071562 | - Cài đặt Kamailio  - Hướng dẫn cài đặt  - Hướng dẫn sử dụng/quản trị Kamailio  - Hướng dẫn phát triển Kamailio | - Cài đặt hoàn thành  - Hoàn thành  - Hoàn thành  -Hoàn thành |
| Nguyễn Khắc Kỷ | 175A071312 | - Cài đặt Kamailio  - Tìm hiểu tài liệu về Kamailio : khái niệm, chức năng, hoạt động.  - Phân tích ưu, nhược điểm  - Hướng dẫn phát triển Kamailio | - Cài đặt hoàn thành  - Hoàn thành  - Hoàn thành  -Hoàn thành |

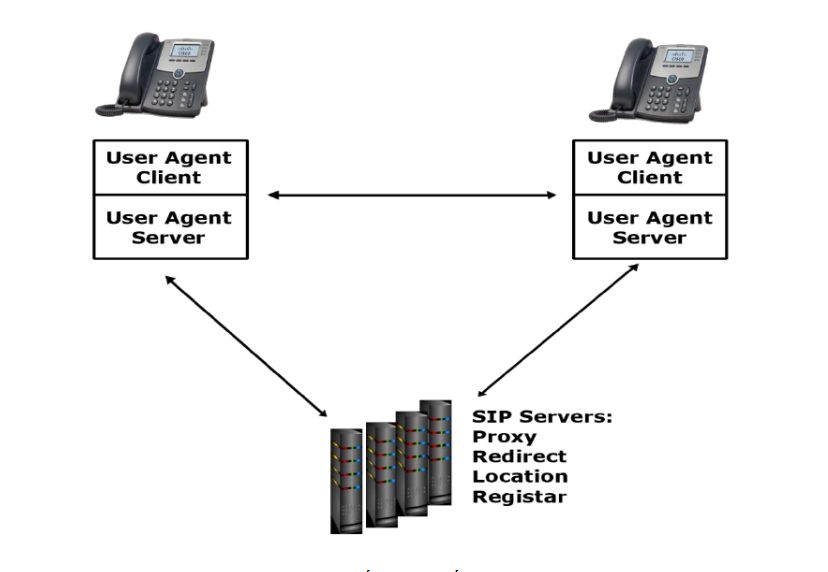
1. **Nội dung nghiên cứu**
   1. **Kamailio là gì?**

Kamailio (trước đây là OpenSER) là một máy chủ SIP được cấp phép theo giấy phép công cộng GNU.

* Nó có thể được thiết lập thành một nhà đăng ký dịch vụ SIP, hay là một máy chủ proxy hay một server chuyển hướng và và với những thứ đang được hỗ trợ như RADIUS / syslog để tính toán và ủy quyền, XML và JSON để điều hướng, SQL backend, …
* Kamailio là một từ Hawai có nghĩa là “giao tiếp”. Kamailio được chọn vì những đặc tính riêng biệt của riêng nó.
* Như vậy, SIP là gì ?
* SIP (Session Initiation Protocol - giao thức khởi tạo phiên) là một giao thức truyền thông để truyền tín hiệu và điều khiển các phiên truyền thông trong các ứng dụng điện thoại sử dụng internet cho các cuộc gọi có tiếng hoặc cả video trong hệ thống điện thoại, cũng như nhắn tin qua mạng Internet Protocol (IP).
* Giao thức này định dạng cụ thể các tin nhắn trao đổi với sự hợp tác của những người tham gia đa phương tiện.
* SIP được thiết kế độc lập với các giao thức ở tầng giao vận (Transport Layer ) nằm ở dưới và có thể được sử dụng chung với UDP, TCP hay SCTP.
* SIP thường dùng chung với UDP vì cùng chung mục đích truyền video cần tính nhanh chóng không nhất thiết phải bảo toàn dữ liệu
* Kiến trúc hệ thống SIP :

Kiến trúc hệ thống SIP gồm 2 thành phần cơ bản :

* User Agent (UA)
* Network Server (NS)



Hình 1 : Kiến trúc hệ thống SIP

User Agent là thiết bị đầu cuối của SIP gồm 2 loại :

* User Agent Client (UAC): gửi các yêu cầu SIP
* User Agent Server (UAS): tiếp nhận và giải quyết yêu cầu SIP

Network Server :

* Proxy Server (PS): nhận và xử lý các yêu cầu trước khi tiếp tục truyền
* Redirect Server (RS): trả địa chỉ người nhận cho người tạo yêu cầu
* Location Server (LS): định vị, cung cấp các địa chỉ miền, địa chỉ người dùng
* Registrar Server (RS): Server đăng kí, thường được cài đặt cùng RS hoặc PS
* Các phương thức của SIP gồm :
* INVITE = thiết lập phiên
* ACK = xác nhận yêu cầu từ INVITE
* BYE = kết thúc phiên
* CANCLE : hủy bỏ việc thiết lập phiên
* REGISTER : trao đổi thông tin địa điểm người dùng (tên máy, IP)
* OPTIONS : trao đổi các thông tin về khả năng của các điện thoại SIP gọi và nghe trong phiên

Tóm lại : SIP là một giao thức khởi tạo phiên nó được dùng để truyền và điều khiển các đa phương tiện (giọng nói, video, ...) bằng cách thức nào ? : qua mạng IP.

* 1. **Kamailo có chức năng cụ thể là gì ?**
* Kamailio có thể được thiết lập là một nhà đăng ký dịch vụ SIP, cũng có thể là một máy chủ proxy vậy chúng ta xem nó hoạt động như thế nào, hay chức năng của nó ra sao?

Vì Kamailio là giao thức sử dụng SIP → ta cần biết cơ bản về SIP proxy

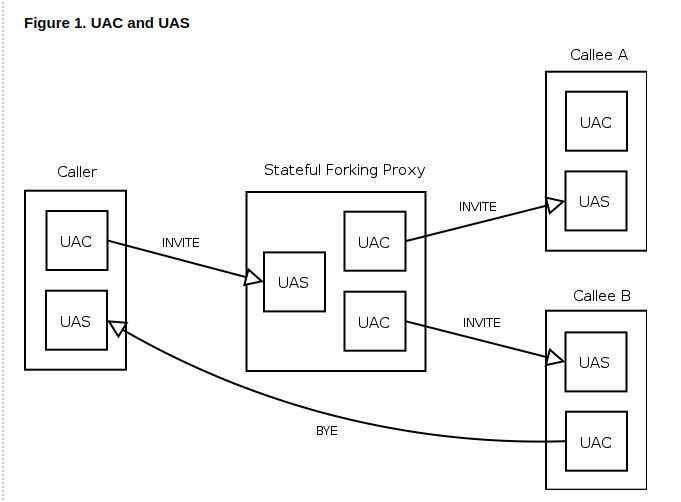
* Proxy là gì?

Giả sử ta có 2 người dùng A gọi cho B, lời gọi sẽ không được chuyển trực tiếp đến B mà ta sẽ chuyển qua trung gian 1 máy chủ proxy để điều hướng các phiên và cũng như thuận lợi cho các vấn để xác thực người dùng.

Ta hãy xem với một ví dụ về cấu trúc mạng proxy Server

1. User Agent : thường được biết đến như User Agent Server (UAS) và User Agent Client (UAC)

* Với mỗi người dùng (UserAgent) đều có 1 UAC và UAS
* UserAgent hành động như là một UAC khi gửi yêu cầu (request) và nhận phản hồi (response)
* UserAgent thay đổi như là UAS khi người được gọi gửi lại tín hiệu BYE và hủy phiên



1. Proxy Servers (đây là thứ chúng ta quan tâm)

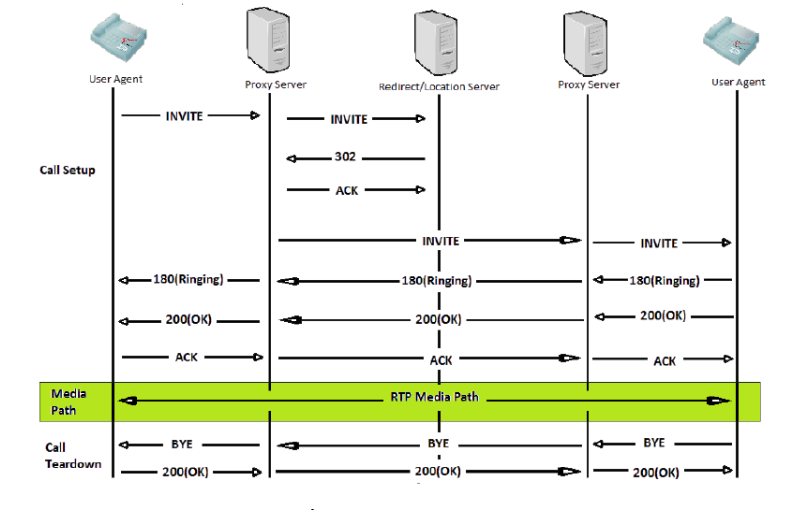
SIP cho phép chúng ta tạo một hệ thống mạng lưới hosts gọi là proxy servers. UserAgents có thể gửi tín hiệu tới proxy servers .Proxy servers góp phần quan trọng trong hệ thống SIP . Nó có trách nhiện điều hướng một lời mời gọi phiên (session invitation) thẳng tới người được nhận và người nhận sẽ quyết định xem chấp nhận hay từ chối cuộc gọi.

Có 2 loại SIP proxy servers cơ bản:

* Stateless Servers: đối với mô hình này proxy chỉ làm nhiệm vụ chuyển tiếp đơn giản. Nó chỉ thực hiện để chuyển tiếp mỗi request và loại bỏ hoàn toàn các thông tin về request mỗi khi đã chuyển đi.
* Stateful Servers : Stateful proxy thì phức tạp hơn. Mỗi lần server nhận được yêu cầu, stateful proxy khởi tao một trạng thái và giữ trạng thái đó cho đến khi chuyển giao kết thúc .Một vài chuyển giao đặc biệt là INVITE, có thể kết thúc muộn (chờ đến khi người nhận chấp thuận hay từ chối). Bởi vì thế stateful proxy phải duy trì trạng thái trong lúc chuyển giao cho dù có 1 vài giới hạn.

Một tính năng đặc biệt của stateful proxy là có thể lựa chọn “fork” 1 request có nghĩa là định tuyến request đến nhiều đích khác nhau.

Thiết lập cuộc gọi qua SIP Proxy Server

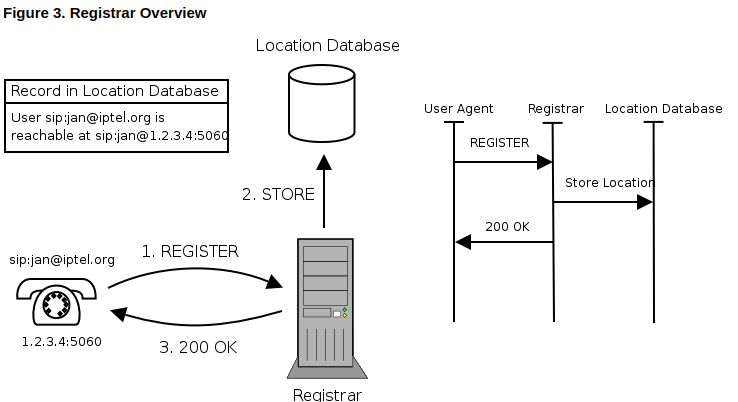


* Bước 1 : UAC gửi bản tin INVITE tới PS gần nhất và PS đó chuyển thông điệp INVITE đó đến RS
* Bước 2 : RS liên hệ với LS để lấy thông tin đường truyền đến UAS và gửi trả lại thông điệp 302 cho PS quản lý UAC
* Bước 3 : PS quản lý UAC gửi thông điệp ACK báo nhận được phản hồi cuối cùng của bản tin INVITE nó gửi đi trước đó
* Bước 4 : Sau khi có thông tin đường truyền đến UAS, PS quản lý UAC gửi thông điệp INVITE đến PS quản lý UAS và PS đó sẽ gửi thông điệp đó đến UAS
* Bước 5 : Sau khi nhận được thông điệp INVITE, USA gửi trả thông điệp 180(Ringing) và sau đó là thông điệp 200(OK) cho PS quản lý nó, và PS sẽ chuyển tiếp cho PS quản lý UAC và tiếp đó là đến UAC
* Bước 6 : Sau khi nhận được thông điệp 180 và 200 thì UAC sẽ gửi thông điệp ACK (báo nhận được phản hồi cuối cùng của bản tin INVITE nó gửi đi trước đó) cho PS quản lý nó. PS sẽ chuyển tiếp đến PS quản lý UAS và chuyển tiếp đến UAS
* Bước 7 : Thiếp lập 1 đường dẫn Media Path giữa UAC và UAS sử dụng giao thức RTP
* Bước 8 : UAC hoặc UAS gửi thông điệp BYE sau đó bên còn lại xác nhận bằng thông điệp 200 và kết thúc cuộc gọi qua proxy SIP

1. SIP Registrar Server

Chúng ta đã đề cập tới cách server SIP điều hướng nhưng chúng ta chưa biết làm sao proxy có thể biết được địa chỉ hiện tại của người dùng.

Như trong hình Figure 3 một SIP phone muốn đăng ký với nhà đăng kí (registrar). Nhà đăng kí là một thực thể đặc biệt của SIP nhận những đơn đăng ký của người dùng (user), giải nén những thông tin về địa chỉ hiện tại của họ (như là địa chỉ IP ,cổng đang dùng và tên người dùng) và lưu chúng trên cơ sở dữ liệu.



* Figure 3 thể hiện một đăng ký SIP thông thường. Một tín hiệu đăng ký chứa địa chỉ là sip:jan@iptel.org và có địa chỉ là 1.2.3.4:5060 được gửi đến nhà đăng kí. Nhà đăng kí này giải nén tất cả những thông tin ấy và lưu nó trong cơ sở dữ liệu . Nếu tất cả ổn thỏa thì nhà đăng kí sẽ gửi tín hiểu 200:OK về điện thoại và tiến trình kết thúc.

Mỗi lần đăng ký đều có thời gian sống nhất định. Trường Expires Header sẽ quyết định lần đăng ký này sống bao lâu. User Agent phải làm mới đăng ký của họ nếu không sẽ hết hạn và người dùng sẽ không còn khả dụng.

* Những tính năng khác mà Kamailio hỗ trợ:
* Hỗ trợ giao tiếp thông qua UDP, TCP , TLS và SCTP.
* Ipv4 và Ipv6.
* Xử lý tiến trình không đồng bộ TCP.
* Bảo mật giao tiếp IP và Network Authentication…

Tóm lại : Kamailio sử dụng giao thức SIP để :

* Điều hướng
* Tạo phiên
* Đăng kí SIP
* Truyền và điều hướng đa phương tiện (video, giọng nói, cuộc gọi…)
  1. **Hoạt động của Kamailio hoặc So sánh Ưu/Nhược điểm**
* Ưu/Nhược điểm :
* Kamailio Server dựa trên SIP (giao thức khởi tạo phiên): giao thức này là một giao thức gửi tín hiệu để khởi tạo duy trì, ngắt phiên trong thế giới thực những thứ như :giọng nói ,video, và các ứng dụng tin nhắn.
* SIP có thể được sử dụng để khởi tạo phiên cho nhóm-2(unicast) hay nhóm-nhiều (multicast).
* Cho phép tùy chỉnh những cuộc gọi đang tồn tại. Sự điều chỉnh này có thể liên quan đến địa chỉ hoặc cổng nguồn, mời nhiều người tham dự.
* Kamailio bạn có một cộng đồng người dùng hoạt động rộng rãi, phát triển nhanh chóng, những nhà phát triển thân thiện và sẵn sàng giúp
* So sánh với một vài server SIP khác như Opensips thì Kamailio có kho modules nhiều hơn hẳn.

**d. Hướng dẫn cài đặt Kamailio trên Ubuntu**

**Bước 1 : Cài đặt máy chủ Apache2 HTTP**

* Cài đặt Apache2 HTTP

|  |
| --- |
| #sudo apt update    #sudo apt install apache2 |

* Sau khi cài đặt Apache2 thì khởi động 3 lệnh dừng , khởi động và cho phép dịch vụ Apache2 luôn khởi động với máy chủ khởi động.

|  |
| --- |
| #sudo systemctl stop apache2.service  #sudo systemctl start apache2.service  #sudo systemctl enable apache2.service |

* Để kiểm tra thiết lập Apache2, hãy mở trình duyệt của bạn và duyệt đến tên máy chủ hoặc địa chỉ IP của máy chủ

http: //localhost

|  |
| --- |
|  |

**Bước 2 : Cài đặt máy chủ cơ sở dữ liệu MariaDB**

* Cài đặt MariaDB Database Server

|  |
| --- |
| #sudo apt install mariadb-server mariadb-client |

* Sau khi cài đặt MariaDB thì khởi động 3 lệnh dừng , khởi động và cho phép dịch vụ MariaDB luôn khởi động khi máy chủ khởi động

|  |
| --- |
| #sudo systemctl stop mariadb.service  #sudo systemctl start mariadb.service  #sudo systemctl enable mariadb.service |

* Sau đó, chạy các lệnh bên dưới để bảo mật máy chủ MariaDB bằng cách tạo mật khẩu gốc và không cho phép truy cập root từ xa

|  |
| --- |
| #sudo mysql\_secure\_installation |

Khi được nhắc, hãy trả lời các câu hỏi dưới đây bằng cách làm theo :

* + Enter current password for root (enter for none): Just press the **Enter**

(Nhập mật khẩu hiện tại cho root (không nhập): chỉ cần Enter)

* + Set root password? [Y/n]: **Y**

(Đặt mật khẩu root? : Y)

* + New password: **Enter password**

(Nhập mật khẩu mới mà bạn muốn đặt)

* + Re-enter new password: **Repeat password**

(Nhập lại mật khẩu mà bạn đã đặt)

* + Remove anonymous users? [Y/n]: **Y**

(Xóa người dùng ẩn danh? : Y)

* + Disallow root login remotely? [Y/n]: **Y**

(Không cho phép đăng nhập root từ xa ? : Y)

* + Remove test database and access to it? [Y/n]:  **Y**

(Xóa cơ sở dữ liệu kiểm tra và truy cập vào nó? Y)

* + Reload privilege tables now? [Y/n]: **Y**

(Tải lại bảng đặc quyền bây giờ? Y)

|  |
| --- |
|  |

* Để kiểm tra xem MariaDB đã được cài đặt chưa, hãy nhập các lệnh bên dưới để đăng nhập vào máy chủ MariaDB

|  |
| --- |
| #sudo mysql -u root -p |

**Bước 3 : Cài đặt PHP các Modules liên quan**

* Cập nhật

|  |
| --- |
| #sudo apt update |

* Cài đặt PHP 7.2 và các modules liên quan

|  |
| --- |
| #sudo apt install php7.2 libapache2-mod-php7.2 php7.2-common php7.2-curl php7.2-intl php7.2-mbstring php7.2-xmlrpc php7.2-mysql php7.2-gd php7.2-imagick php-pear php7.2-xml php7.2-cli php7.2-zip php7.2-sqlite |

* Mở tệp cấu hình mặc định PHP cho Apache2

|  |
| --- |
| #sudo nano /etc/php/7.2/apache2/php.ini |

* Thay đổi các giá trị như bảng rồi lưu

|  |
| --- |
| * file\_uploads = On      * allow\_url\_fopen = On      * short\_open\_tag = On      * memory\_limit = 256M      * upload\_max\_filesize = 100M      * max\_execution\_time = 360      * max\_input\_vars = 1500      * date.timezone = Asia/HoChiMinh |

* Khởi động lại Apache2

|  |
| --- |
| #sudo systemctl restart apache2.service |

* Kiểm tra cài đặt PHP 7.2 với Apache2, hãy tạo tệp phpinfo.php trong thư mục gốc Apache2

|  |
| --- |
| #sudo nano /var/www/html/phpinfo.php |

* Nhập nội dung vào tệp và lưu

|  |
| --- |
| <?php phpinfo( ) ; ?> |

* Sau đó duyệt đến máy chủ của bạn : <http://localhost/phpinfo.php>

|  |
| --- |
|  |

**Bước 4 : Tải Kamailio**

* Tải xuống và thêm khóa GPG cho kho lưu trữ của nó

|  |
| --- |
| # wget -O- http://deb.kamailio.org/kamailiodebkey.gpg | sudo apt-key add - |

* Tạo tệp lưu trữ

|  |
| --- |
| #sudo nano /etc/apt/sources.list.d/kamailio.list |

* Chép, dán và lưu vào tệp

|  |
| --- |
| #deb http://deb.kamailio.org/kamailio52 bionic main  #deb-src http://deb.kamailio.org/kamailio52 bionic main |

* Cập nhật các gói Ubuntu và cài đặt Kamailio

|  |
| --- |
| * Cập nhật các gói Ubuntu   #sudo apt update     * Cài đặt Kamailio   #sudo apt install kamailio kamailio-mysql-modules kamailio-websocket-modules |

* Sau khi cài đặt Kamailio, kiểm tra xem nó đã được cài đặt và sẵn sàng chưa bằng cách :

|  |
| --- |
|  |

* Mở tệp cấu hình Kamailio

|  |
| --- |
| #sudo nano /etc/kamailio/kamctlrc |

* Chỉnh sửa trong tệp và lưu

|  |
| --- |
| SIP\_DOMAIN = baitaplon.kamailio.com  DBENGINE = MYSQL |

* Tạo cơ sở dữ liệu, người dùng và các bảng cần thiết cho Kamailio

|  |
| --- |
| #kamdbctl create |

* Nếu quyền truy cập bị từ chối bởi **root@localhost**

|  |
| --- |
|  |

thì :

|  |
| --- |
| * Đăng nhập vào máy chủ MariaDB chạy lệnh :   #sudo mysql -u root   * Tắt xác thực plugin cho người dùng root   use mysql;  update user set plugin='' where User='root';   * Tải lại tất cả các đặc quyền :   flush privileges;   * Thoát :   exit;     * Khởi động lại   #sudo systemctl restart mariadb.service |

* Bây giờ hãy chạy tập lệnh Kamailio để tạo cơ sở dữ liệu và người dùng

và trả lời các cài đặt bên dưới

|  |
| --- |
| * Enter character set name:   **latin1**   * INFO: creating database kamailio ... * INFO: granting privileges to database kamailio ... * INFO: creating standard tables into kamailio ... * INFO: Core Kamailio tables succesfully created. * Install presence related tables? (y/n): **y** * INFO: creating presence tables into kamailio ... * INFO: Presence tables succesfully created. * Install tables for imc cpl siptrace domainpolicy carrierroute   drouting userblacklist htable purple uac pipelimit mtree sca mohqueue rtpproxy rtpengine? (y/n): **y**   * INFO: creating extra tables into kamailio ... * INFO: Extra tables succesfully created. * Install tables for uid\_auth\_db uid\_avp\_db uid\_domain   uid\_gflags uid\_uri\_db? (y/n): **y**   * INFO: creating uid tables into kamailio ... * INFO: UID tables succesfully created. |

* Mở tệp cấu hình cho kamailio : /etc/kamailio/kamailio.cfg bằng cách chạy lệnh

|  |
| --- |
| #sudo nano /etc/kamailio/kamailio.cfg |

* Thêm và lưu vào tệp sau #! KAMAILIO

Để cho phép MySQL phụ trợ máy chủ, xác thực người dùng và vị trí người dùng liên tục

|  |
| --- |
| #!define WITH\_MYSQL  #!define WITH\_AUTH  #!define WITH\_USRLOCDB  #!define WITH\_ACCDB |

* Áp dụng các lệnh thay đổi

|  |
| --- |
| #sudo systemctl restart kamailio |

**Bước 5 : Cài đặt và cấu hình Siremis**

* Cài đặt XML\_RPC :

( XML-RPC là một cài đặt RPC (Remote Procedure Call) trên nền giao thức HTTP, sử dụng XML để mã hoá và trao đổi dữ liệu )

|  |
| --- |
| #sudo pear install XML\_RPC2 |

* Thay đổi thư mục thành root web Apache, cài đặt git và tải xuống các gói Siremis.

|  |
| --- |
| #cd / var / www /  #sudo apt install git    #sudo git clone <https://github.com/asipto/siremis> |

* Vào thư mục Siremis và chuẩn bị môi trường

|  |
| --- |
| #cd /var/www/siremis  #sudo make prepare24  #sudo make chown |

* Định cấu hình tệp cấu hình trang Apache2 cho Siremis

|  |
| --- |
| #sudo nano /etc/apache2/sites-available/siremis.conf |

* Chèn vào tập tin và lưu. Thay thế tên miền (ServerName) và vị trí thư mục gốc (ServerAlias)

|  |
| --- |
| <VirtualHost \*:80>  ServerAdmin admin@example.com  DocumentRoot /var/www/siremis/siremis  Alias /siremis "/var/www/siremis/siremis"  ServerName baitaplon.kamailio.com  ServerAlias www.baitaplon.kamailio.com  <Directory "/var/www/siremis/siremis">  Options Indexes FollowSymLinks MultiViews  AllowOverride All  Require all granted  <FilesMatch "\.xml$">  Require all denied  </FilesMatch>  <FilesMatch "\.inc$">  Require all denied  </FilesMatch>  </Directory>  ErrorLog ${APACHE\_LOG\_DIR}/error.log  CustomLog ${APACHE\_LOG\_DIR}/access.log combined  </VirtualHost> |

**Bước 6 : Kích hoạt Modules Siremis**

* Sau khi định cấu hình Virtualhost ở trên, bật nó :

|  |
| --- |
| #sudo a2ensite siremis.conf  (kích hoạt trang)    #sudo a2enmod rewrite  (kích hoạt rewrite\_module : cho phép viết lại URL )    #sudo systemctl restart apache2.service  (khởi động lại apache) |

* Cấp quyền truy cập cho người dùng Siremis vào cơ sở dữ liệu của nó

|  |
| --- |
| * Đăng nhập mysql :   #sudo mysql -u root -p |

* Chạy lệnh SQL

|  |
| --- |
| * grant all privileges on siremis.\* to siremis@localhost identified by 'siremisrw';      * flush privileges; * exit; |

* Sau đó mở trình duyệt và duyệt đến tên miền máy chủ hoặc tên máy chủ

<http://baitaplon.kamailio.com/>

|  |
| --- |
|  |

* Xác thực rằng tất cả các yêu cầu được đáp ứng, sau đó tiếp tục với trình hướng dẫn.

|  |
| --- |
|  |

* Nhập thông tin cấu hình cơ sở dữ liệu

|  |
| --- |
|  |

* Thiết lập hoàn tất và cung cấp cho bạn tên người dùng và mật khẩu quản trị viên. Sử dụng thông tin đăng nhập và bắt đầu thiết lập môi trường của bạn

|  |
| --- |
|  |

* Tên đăng nhập : admin

Mật khẩu : admin

|  |
| --- |
|  |

* Hoàn thành thiết lập

|  |
| --- |
|  |

* 1. **Hướng dẫn sử dụng/quản trị Kamailio**

Siremis : Giao diện quản lí web nguồn mở cho máy chủ Kamailio SIP

**Quản trị (Administration)**

|  |
| --- |
|  |

* **Người dùng(Users)**

Quản lí tài khoản người dùng trong ứng dụng :

|  |
| --- |
| Thêm người dùng(Add)    Sửa thông tin người dùng(Edit) :    Xóa thông tin người dùng(Delete) |

* **Vai trò (Roles)**

Quản lí vai trò người dùng trong ứng dụng để kiểm soát sự cho phép của người dùng :

|  |
| --- |
|  |

Thêm vai trò : với tên, trang, miêu tả vai trò, trạng thái(hoạt động, không hoạt động), cho phép(không cho phép)

|  |
| --- |
|  |

Sửa vai trò :

|  |
| --- |
|  |

Xóa vai trò :

|  |
| --- |
|  |

* **Nhóm (Groups)**

Quản lí nhóm : Thêm, sửa, xóa, xuất file danh sách nhóm

Thêm nhóm : thêm tên nhóm, mô tả nhóm và lưu

|  |
| --- |
|  |

Sửa thông tin nhóm : chỉnh sửa tên nhóm và mô tả nhóm sau đó lưu

|  |
| --- |
|  |

* **Nhật kí sự kiện (Event Log)**

Quản lí nhật kí sự kiện ghi lại các hoạt động quan trọng, chẳng hạn như đăng nhập…

|  |
| --- |
|  |

**SIP Admin Menu**

* **Dịch vụ thuê bao (Subscriber Services)**
* Quản lí thuê bao( Subcriber Management ) đã đăng kí dưới dạng danh sách (Subcriber List) với tên người dùng và tên miền.

|  |
| --- |
| Thêm thuê bao : thêm tên người dùng và tên miền  **[root@serverhost:]~$ kamctl add <username> <password>**    Sau khi thêm thuê bao thành công :    Xem các thông tin của thuê bao :   * Một thuê bao   **[root@serverhost:]~$ kamctl show <username>**     * Tất cả thuê bao trong cơ sở dữ liệu   **[root@serverhost:]~$ kamctl db show subscriber**    Sửa thông tin thuê bao : đổi mật khẩu của thuê bao  **[root@serverhost:]~$ kamctl passwd <username> <passwd>**    Sau khi đổi mật khẩu thành công :    Xoá thuê bao :  **[root@serverhost:]~$ kamctl rm <username>**    Sau khi xoá thành công :    Xuất (Export) danh sách thuê bao ra Exel :    Tìm kiếm thuê bao : nhập thông tin thuê bao cần tìm kiếm |

* Bí danh cơ sở dữ liệu (Aliases Db) : đổi tên một người dùng tạm thời bằng một tên khác nhưng không làm thay đổi trong cơ sở dữ liệu

|  |
| --- |
| Thêm mới bí danh cơ sở dữ liệu  **[root@serverhost:]~$ kamtcl alias\_db add <alias> <sip-id>**    Sau khi thêm thành công    Hiển thị thông tin bí danh :  **[root@serverhost:]~$ kamtcl alias\_db show <alias>**    Sửa bí danh :    Xóa bí danh :  **[root@serverhost:]~$ kamtcl alias\_db rm <alias>**    Sau khi xóa thành công    Tìm kiếm bí danh cơ sở dữ liệu : Nhập thông tin cần tìm kiếm    Tìm kiếm thành công |

* **Dịch vụ máy chủ (Server Services)**
* Quản lí domain (Domain Managenment) : thêm tên miền cho server kamailio với mặc định là địa chỉ localhost hoặc địa chỉ IP của máy bạn (tên miền, id tên miền và thời gian chỉnh sửa cuối cùng.

|  |
| --- |
| Thêm tên miền mới vào cơ sở dữ liệu :  **[root@serverhost:]~$ kamctl domain add <domain>**    Sau khi thêm thành công :    Hiển thị thông tin tên miền trong cơ sở dữ liệu  **[root@serverhost:]~$ kamctl domain showdb**    Xóa tên miền :  **[root@serverhost:]~$ kamctl domain rm <domain>**    Sau khi xoá tên miền thành công    Tìm kiếm : Nhập thông tin tên miền cần tìm kiếm |