|  |
| --- |
| BAN CƠ YẾU CHÍNH PHỦ  **HỌC VIỆN KỸ THUẬT MẬT MÃ**  ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯  Description: Logo HvKTMM |
| **AN TOÀN ĐIỆN TOÁN ĐÁM MÂY**  **ĐỀ TÀI: TÌM HIỂU THỬ NGHIỆM VỀ COBBLER**  *Lớp:* **L02 - AT12** |
| *Giảng viên:* Bùi Việt Thắng  *Sinh viên thực hiện:*   1. Nguyễn Nhật Nam 2. Lê Tuấn Khải |
| Hà Nội, 2019 |

**NHẬN XÉT CỦA GIẢNG VIÊN**

**I. Tổng quan PXE và Kickstart**

# 1. PXE là gì

## **1.1. Mục đích ra đời**

Khi máy tính gặp sự cố không khởi động được, người ta thường nghĩ đến các đĩa CD/DVD hoặc USB có thể khởi động được để khắc phục sự cố, phục hồi dữ liệu. Có một giải pháp khá hiệu quả là xây dựng một PXE Server trên một máy tính khác trong cùng mạng LAN và trên máy gặp sự cố, sẽ khởi động bằng card mạng và truy cập vào PXE Server để sử dụng các chương trình được cài sẵn trên đó. Bài viết này sẽ hướng dẫn cách thức xây dựng một PXE Server với nhiều lựa chọn khi khởi động để hỗ trợ người quản trị khắc phục các máy tính khi gặp sự cố.

## **1.2. Giới thiệu**

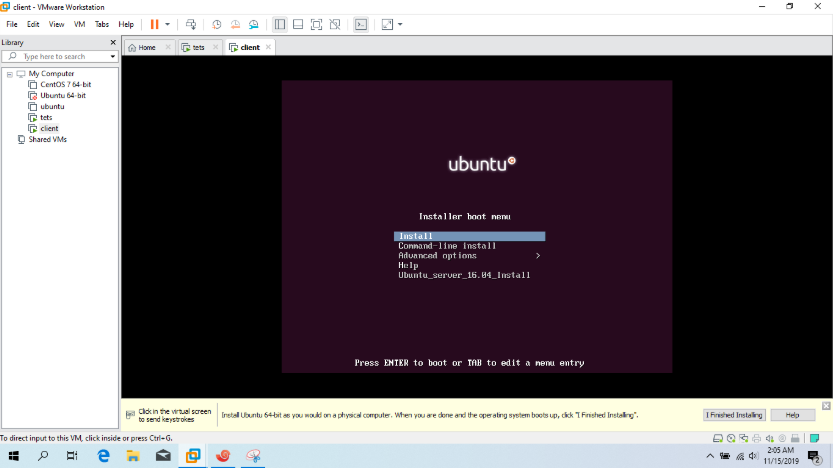
Preboot eXecution Environment - PXE là một chuẩn công nghiệp mở được phát triển bởi một số các nhà sản xuất phần mềm và phần cứng. Được thiết kế ban đầu bởi Intel, cùng với sự tham gia của các nhà sản xuất như HP, 3Com, Dell, Compaq, and Phoenix Technologies.

PXE là một môi trường cho phép khởi động máy tính bằng việc sử dụng card mạng cùng với RAM. Việc khởi động đó sẽ không phụ thuộc vào những thiết bị của máy tính như CD, harddisk và các hệ điều hành đã được cài đặt.PXE bao gồm 2 thành phần: PXE Client và PXE Server:

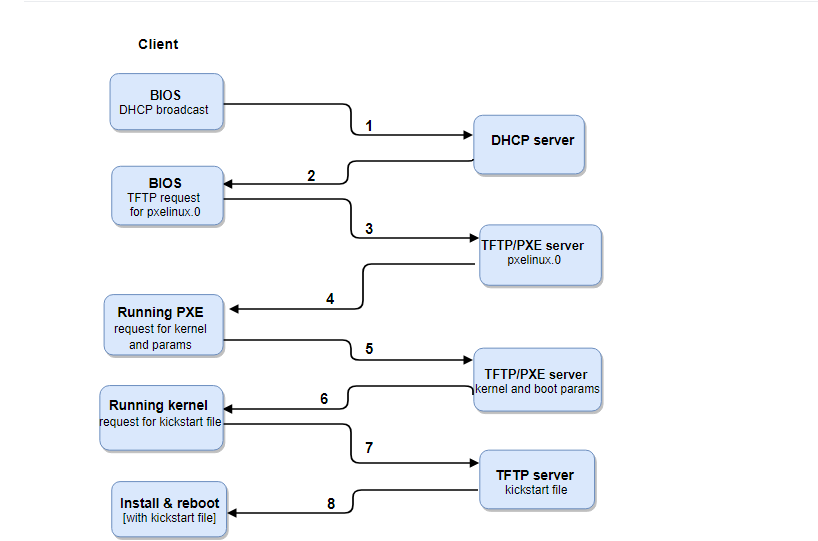
+/ Thành phần PXE Client (PXE boot code) nằm trên card mạng.

+/ PXE Server là phần mềm chạy trên một máy tính nào đó trong mạng LAN. Phần mềm này hỗ trợ các giao thức TFTP, DHCP.Trong đó dịch vụ DHCP có nhiệm vụ cấp phát địa chỉ IP để các máy client có thể liên lạc với các máy khác và với TFTP server để lấy các file boot và file cấu hình cần thiết. Dịch vụ TFTP cung cấp dịch vụ truyền các file boot và file cấu hình cần thiết cho Client.

Ví dụ minh họa



## **1.3. PXE workflow**



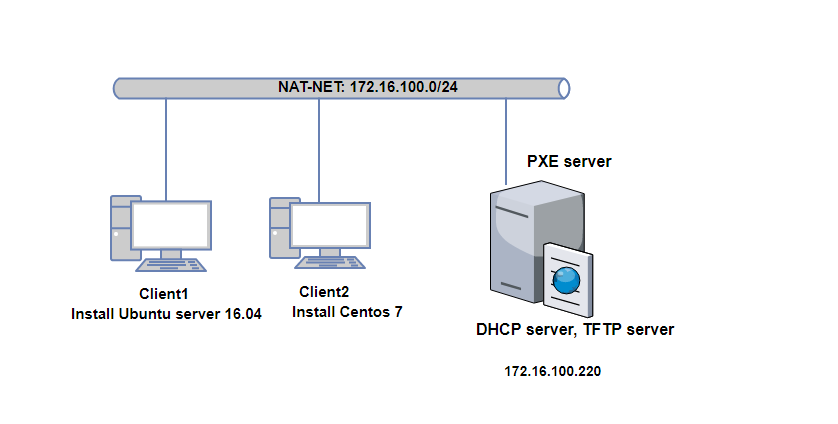
* Workflow của PXE gồm những bước sau:
  + 1.Máy client được khởi động (bật nguồn). NIC của client gửi đi bản tin broadcast để tìm DHCP server.
  + 2.DHCP server nhận được bản tin của client và gửi lại các thông tin cần thiết cho client (IP, subnet mask, gateway, DNS ...). Và cung cấp cho client thông tin về vị trí của TFTP server và boot image (pxelinux.0 - đây là file chứa các mã lệnh để cho biết loại kernel mà hệ thống sẽ boot).
  + 3.Khi client nhận được các thông tin này, nó sẽ liên lạc với TFTP server để lấy boot image.
  + 4.TFTP server gửi lại boot image (pxelinux.0), và client sẽ sử lý nó.
  + 5.Mặc định, boot image tìm được thư mục pxelinux.cfg trên TFTP server để lấy các file cấu hình.
  + 6,7,8:Client doownload tất cả các file cần thiết (filesystem, kickstart, ...) và tải về.
* Cuối cùng, Client install và reboot.

# 2. Kickstart là gì?

* Kickstart file : đây là thành phần quan trọng nhất, kickstart file là file chứa tất cả những câu trả lời cần thiết cho việc cài đặt các distro Linux nhờ vào file trả lời này mà toàn bộ quá trình cài đặt sẽ được tự hóa hoàn toàn.
* Các file Kickstart có thể được lưu trong một server riêng biệt (thường trong cùng PXE hoặc TFTP server), và có thể lấy được bởi bất kì máy tình nào trong suốt quá trình máy đó cài đặt. Phương pháp cài đặtnày hỗ trợ sử dụng một file Kickstart để cài đặt Ubuntu trên nhiều máy Client khác nhau, là phương pháp tuyệt với cho các nhà quản trị mạng và hệ thống cấu hình mô hình mạng theo ý muốn của mình.
* Kickstart cho phép bạn cấu hình tự động các thông tin trong suốt quá trình cài đặt như:
  + Lựa chọn ngôn ngữ
  + Cấu hình cho mouse
  + Cấu hình keyboard
  + Cài đặt boot loader
  + Phân vùng ổ cứng.
  + Cấu hình mạng +......

**II. Cài đặt PXE**

# 1. Mô hình và chuẩn bị

  
Để sử dụng PXE bạn cần cài đặt hệ thống PXE boot-server mà cho phép client:

* + Request một địa chỉ IP (thông qua DHCP)
  + Download kernel (thông qua TFTP, vì giao thức này có thể bỏ qua quá trình xác thực, các máy chưa được cài Hệ điều hành sẽ không đáp ứng được yêu cầu xác thực)
* **Chuẩn bị**: Phần này thực hiện lab trên môi trường máy ảo Vmware workstation 15. Các máy ảo cấu hình như sau:
  + Máy server: Ubuntu server 16.04 cài các dịch vụ DHCP server và TFTP server. Có NIC dải NAT-NET, địa chỉ 192.168.74.129
  + Các máy client card mạng hỗ trợ PXE. Có ít nhất một card mạng dải NAT-NET
  + Client 1 cài OS Ubuntu server 16.04; Client 2 cài OS Centos 7  
    Sau đây là các bước cấu hình:

# 2. Cài đặt và cấu hình DHCP server

### Cài đặt gói phần mềm DHCP server cho máy server: apt-get install isc-dhcp-server -y

### Cấu hình

* Sửa file /etc/deefault/isc-dhcp-server để cấp DHCP qua card ens33 (của máy server)

INTERFACES="ens33"

* Cấu hình dải DHCP mà server cấp phát: sửa file /etc/dhcp/dhcpd.conf
  + Bỏ comment dòng sau: authoritative;
  + Thêm các dòng sau vào cuối file để cấp phát cho dải mạng:
  + subnet 192.168.74.0 netmask 255.255.255.0 {
  + range 192.168.74.120 192.168.74.128;
  + option domain-name-servers 8.8.8.8;
  + option routers 192.168.74.1;
  + option broadcast-address 192.168.74.255;
  + default-lease-time 600;
  + max-lease-time 7200;
  + }
* Cấu hình các thông tin sau cho PXE boot server: (thêm vào cuối file /etc/dhcp/dhcpd.conf)
* allow booting;
* allow bootp;
* option option-128 code 128 = string;
* option option-129 code 129 = text;
* next-server 192.168.74.129; #IP cua TFTP server
* filename "pxelinux.0";
* Khởi động lại dịch vụ DHCP để áp dụng thông tin vừa cấu hình:

service isc-dhcp-server restart

# 3. Cài đặt và cấu hình TFTP server

## **3.1. Cài đặt và cấu hình TFTP server**

### Cài đặt các gói phần mềm cần thiết cho TFTP server:

apt-get install ftpd-hpa inetutils-inetd -y

### Cấu hình:

* Sửa file /etc/default/tftpd-hpa chèn vào cuối dòng trên các dòng dưới
* RUN\_DAEMON="yes"
* OPTIONS="-l -s /var/lib/tftpboot"
* Sửa file /etc/inetd.conf, thêm vào cuối cùng dòng sau:

tftp dgram udp wait root /usr/sbin/in.tftpd /usr/sbin/in.tftpd -s /var/lib/tftpboot

* Tạo thư mục chứa các file boot TFTP:

mkdir -p /var/lib/tftpboot/

* Khởi động lại dịch vụ TFTP server:

service tftpd-hpa restart

## **3.2. Bung các gói cần thiết từ file ISO**

Để setup install OS cho các client lấy các gói phần mềm cần thiết trong quá trình cài đặt từ TFTP server local ta thực hiện các bước như sau:

### **Bước 1**: Download các iso cần thiết cài đặt cho OS Ubuntu server 16.04 như sau:

wget http://releases.ubuntu.com/16.04/ubuntu-16.04.6-server-amd64.iso

### **Bước 2**: Mount các file iso

Để thực hiện mount các file iso vừa tải vào thư mục /mnt để copy tới tftpd root (/var/lib/tftpboot/) cho các client download file pxelinux.0 về để tiến hành cài đặt OS (pxelinux.0 là một đoạn mã của bootloader cho phép client lúc đó chưa có hệ điều hành có được khả năng boot OS thông qua card mạng)

mount –o loop /root/ubuntu-16.04.6-server-amd64.iso /mnt

* Copy các file cần thiết trong quá trình netboot cho OS linux:
* cd /mnt
* cp -fr install/netboot/\* /var/lib/tftpboot/

### **Bước 3**: Tạo thư mục chứa kenel và initrd cho từng OS:

* **Ubuntu**: đã có sẵn do trước đó copy toàn bộ từ iso của ubuntu sang thư mục tftpd root.

### **Bước 4**: Đưa các gói cần thiết trong quá trình cài đặt lên http server để client có thể download các gói phần mềm từ mạng local. Ở đây, cài http server lên luôn PXE server:

* Cài đặt apache2 để các máy client có thể download thông qua HTTP:

apt-get install apache2 -y

* Tạo các thư mục cho Ubuntu:
* mkdir -p /var/www/html/US/ # thư mục cho Ubuntu

# Copy các file trong thư mục /mnt/ vừa mount ở trên

# sudo cp -fr /mnt/\* /var/www/html/US/

# 4. Cấu hình PXE và kickstart

## **4.1. Cấu hình kickstart**

* Tạo file kickstart cho client cài đặt OS Ubuntu server 16.04 có nội dung tùy chỉnh như sau:
* # set language to use during installation
* lang en\_US
* langsupport en\_US
* # set keyboard layout for the system
* keyboard us
* # reboot the system after installation
* reboot
* # config repo source.list
* # url --url http://192.168.74.100/US/
* # Sets up the authentication options for the system.
* auth --useshadow --enablemd5
* bootloader --location=mbr
* zerombr yes
* clearpart --all
* # setup timezone
* timezone Asia/Ho\_Chi\_Minh
* # Set the system's root password
* rootpw rootpassword123
* # Creates a new user on the system
* user ttp --fullname=ttp --password=ttppassword123
* # create partition on the system with LVM
* part pv.01 --size 1 --grow
* volgroup ubuntu pv.01
* logvol swap --fstype swap --name=swap --vgname=ubuntu --size 1024
* logvol / --fstype ext4 --vgname=ubuntu --size=1 --grow --name=slash
* # hack around Ubuntu kickstart bugs
* preseed partman-lvm/confirm\_nooverwrite boolean true
* preseed partman-auto-lvm/no\_boot boolean true
* # Configures network information
* #network --bootproto=dhcp --device=ens34 --active
* network --bootproto=dhcp --device=ens33 --active
* # Do not configure the X Window System
* skipx
* ## Install packet for the system
* %packages --ignoremissing
* @ ubuntu-server
* openssh-server
* ## Run script after installation
* %post
* sed -i 's/192.168.74.100/vn.archive.ubuntu.com/g' /etc/apt/sources.list
* sed -i 's/US/ubuntu/g' /etc/apt/sources.list
* apt-get update -y && apt-get upgrade –y
* mkdir /root/test
  + Lưu file lại với tên kickstartForUS16.04.cfg
  + Chuyển file kickstartForUS16.04.cfg vào thư mục web root như sau:

mv kickstartForUS16.04.cfg /var/www/html/

## **4.2. Cấu hình PXE**

* Sửa file /var/lib/tftpboot/pxelinux.cfg/default để khai báo menu của PXE boot, nội dung giống như sau:
* # D-I config version 2.0
* # search path for the c32 support libraries (libcom32, libutil etc.)
* path ubuntu-installer/amd64/boot-screens/
* include ubuntu-installer/amd64/boot-screens/menu.cfg
* default ubuntu-installer/amd64/boot-screens/vesamenu.c32
* DISPLAY boot.txt
* LABEL Ubuntu\_server\_16.04\_Install
* kernel ubuntu-installer/amd64/linux
* append ks=http://172.16.100.220/kickstartForUS16.04.cfg ksdevice=ens33 vga=normal initrd=ubuntu-installer/amd64/initrd.gz --
* prompt 0

timeout 0

# 5. Client boot OS từ PXE

# Tạo một máy ảo Client1 mới chưa setup hệ điều hành như sau:

# 

# Sau khi có IP và download được file pxelinux.0 client sẽ tự động load menu boot lựa chọn boot như sau:

# 

# Lựa chọn OS cần cài đặt. Sau khi cài đặt xong, máy client1 có giao diện như sau:

# 

# Đăng nhập bằng user ttp với mật khẩu ttppassword123 như đã tạo trong file kickstart => thành công.

# 

**III. Tổng quan về Cobbler**

1. **Cobbler là gì?**

Cobbler là một gói công cụ tiện ích cho phép triển khai hoàn chỉnh một máy chủ PXE server với khả năng cài đặt tự động các phiên bản Linux thông qua môi trường mạng đồng thời hỗ trợ kết hợp tính năng của kickstart file cho phép tự động hóa hoàn toàn quy trình cài đặt loại bỏ hẳn các thao tác trả lời không cần thiết trong quá trình triển khai ngoài ra còn support deploy OS qua mạng mà không cần sự hỗ trợ của DHCP truyền thống thích hợp cho một số trường hợp triển khai khác nhau. Đây chưa phải là tất cả khả năng của Cobbler như tôi đã trình bày mà các bạn còn cần phải tìm hiểu thêm nhưn trong bài viết này tôi sẽ cố gắng mang đến các khái niệm và ứng dụng thông thường nhất cho một PXE server.

1. **Các thành phần của Cobbler**

Cobbler kết nối và tự động hóa nhiều công đoạn khác nhau trong quá trình cài đặt Linux, giúp cho người quản trị dễ dàng hơn trong việc cài đặt số lượng lớn hệ điều hành Linux với những cấu hình khác nhau. Cobbler quản lý những thành phần chính sau :

– **Kickstart file** : đây là thành phần quan trọng nhất, kickstart file là file chứa tất cả những câu trả lời cần thiết cho việc cài đặt các distro Linux nhờ vào file trả lời này mà toàn bộ quá trình cài đặt sẽ được tự hóa hoàn toàn.

– **TFTP, Web, FTP, rsync server** : đây là các giao thức mà Cobbler sử dụng để truyển tải các file cài đặt từ server đến các máy trạm để cài linux.

– **DHCP server** : để phục vụ cho việc cài đặt qua mạng, máy trạm phải kết nối được đến server và được cấp 1 địa chỉ IP. Quá trình cấp địa chỉ này được thực hiện bởi DHCP server

– **DNS server** : để tiện cho việc quản lý, ta có thể gán địa chỉ IP với 1 tên miền dịch vụ này không phải yêu cầu bắt buộc như DHCP.

– **Web server** : Cobbler cung cấp giao diện web cho phép người quản trị thông qua đó, quản lý các profile cũng như các máy trạm được cài đặt.

– Các khái niệm cơ bản cần biết của Cobbler

– **Distribution** : chứa các thông tin về kernel và initrd nào được sử dụng, bao gồm cả các dữ liệu dùng để cài đặt. Hiểu 1 cách đơn giản. Đấy chính là đĩa cài đặt của Linux của ta.

– **Profile** = **Distribution** + **kickstart file** + **các gói cài đặt thêm** (nếu cần)

– **System**= **Profile** + **MAC addr** (IP addr, hostname)

– **Repo** : là nơi chứa các gói cài đặt thêm.

Như vậy, có thể thấy, trình tự cấu hình Cobbler sẽ như sau :

• Tạo distribution  
• Tạo profle  
• Tạo repo.  
• Add system.  
• Boot máy trạm và chờ kết quả

**IV. Cài đặt Cobbler**

# 1. Mô hình triển khai

Chuẩn bị: bài lab thực hiện trên server đã cài đặt KVM và quản lý bởi Webvirtmgr. Trong đó:

* **Cobbler server**: máy ảo KVM cài đặt sẵn hệ điều hành Centos 7. Có một card mạng ens3 (thuộc dải mạng default của server vật lý, có thể ra ngoài Internet) địa chỉ: 192.168.122.110/24
* Các máy **Client 1** và **Client 2**: là các máy ảo KVM được tạo ra với một card mạng thuộc dải default-net. Chưa được cài đặt Hệ điều hành.
* Cấu hình trên server vật lý: Tắt DHCP của mạng default-net như sau:

# 2. Các bước cài đặt và cấu hình Cobbler

Thực hiện trên máy Cobbler đã cài đặt sẵn Centos 7.

## **Bước 1: Cài đặt EPEL-repo trên Centos**

* Epel-repo (Extra Packages for Enterprise Linux) là một dự án repository từ Fedora team cung cấp rất nhiều gói add-on package mà chúng ta thường dùng cho các bản Linux bao gồm CentOS, RHEL (Red Hat Enterprise Linux) và Scientific Linux.
* Cài đặt Epel-repo thực hiện lệnh sau:
* yum install epel-release

## **Bước 2: Cài đặt Cobbler và một số gói cần thiết**

yum install cobbler cobbler-web dnsmasq syslinux xinetd bind bind-utils dhcp debmirror pykickstart fence-agents-all -y

Trong đó:

- cobbler, cobbler-web: các gói phần mềm cài đặt chạy dịch vụ cobbler và giao diện web của cobbler.

- dnsmasq, bind, bind-utils, dhcp : các gói phần mềm chạy dịch vụ quản lý DNS và quản lý DHCP cho các máy client boot OS từ cobbler.

- syslinux : là một chương trình bootloader và tiện ích cho phép đẩy vào client cho phép client boot OS qua mạng. (trong trường hợp này nó được gọi là pxelinux)

- xinetd: chịu trách nhiệm tạo socket kết nối với máy client. Dựa vào cổng và giao thức (tcp hay udp) nó biết được phải trao đổi dữ liệu mà nó nhận được với back-end nào dựa vào thuộc tính server trong file cấu hình. Được sử dụng để quản lý và tạo socket cho TFTP server truyền file boot cho client.

- debmirror: gói phần mềm cài đặt cho phép tạo một mirror server chứa các gói phần mềm cài đặt của các distro trên một server local (ở đây cài luôn lên cobbler)

- pykickstart : thư việc python cho phép đọc và chỉnh sửa nội dung file kickstart, hỗ trợ cobbler chỉnh sửa file kickstart thông qua giao diện web.

## **Bước 3: Kích hoạt các dịch vụ**

* Kích hoạt và khởi động các dịch vụ **cobblerd** và **httpd**:
* systemctl start cobblerd
* systemctl enable cobblerd
* systemctl start httpd
* systemctl enable httpd
* Disable SELinux:
  + Thực hiện các lệnh sau để disable tính năng của SELinux:
  + sed -i 's/\(^SELINUX=\).\*/\SELINUX=disabled/' /etc/sysconfig/selinux
  + sed -i 's/\(^SELINUX=\).\*/\SELINUX=disabled/' /etc/selinux/config
  + setenforce 0
  + Khởi động lại máy và thực hiện bước tiếp theo.
* Thực hiện các lệnh sau nếu OS chạy firewall:
* firewall-cmd --add-port=80/tcp --permanent
* firewall-cmd --add-port=443/tcp --permanent
* firewall-cmd --add-service=dhcp --permanent
* firewall-cmd --add-port=69/tcp --permanent
* firewall-cmd --add-port=69/udp --permanent
* firewall-cmd --add-port=4011/udp --permanent
* firewall-cmd --reload

## **Bước 4: Cấu hình Cobbler**

Thực hiện sửa file cấu hình của Cobbler để cho phép sử dụng DHCP server của Cobbler cấp phát IP cho các client, quản lý DNS, cung cấp địa chỉ IP của Cobbler và của TFTP server (dùng để lấy các file boot về client, thường tích hợp chung vào Cobbler luôn nên nó cũng là IP của Cobbler).

### Sửa file cấu hình Cobbler /etc/cobbler/settings

* Cấu hình password mặc định để tăng bảo mật cho hệ thống.
* Sử dụng openssl để sinh ra mật khẩu đã được mã hóa như sau:
* openssl passwd -1
* Password: <enter\_password\_here>
* Verifying - Password: <reenter\_password\_here>
* $1$1unDtJqr$6THfdcVTmF58L35aigs7N/
* Sửa file /etc/cobbler/settings với các thông số default\_password\_crypted với password đã được mã hóa vừa sinh ra ở trên, và cập nhật các thông số của DHCP, DNS, PXE từ 0 lên 1 như sau:
  + Thay thế mật khẩu vừa tạo vào mật khẩu mặc định: (sửa tại dòng 101)
  + default\_password\_crypted: "$1$1unDtJqr$6THfdcVTmF58L35aigs7N/"

Đoạn password này được sử dụng để làm password mặc định cho client khi được cấu hình trong file kickstart sử dụng với tùy chọn --iscrypted

Ví dụ:

#Root password

rootpw --iscrypted $default\_password\_crypted

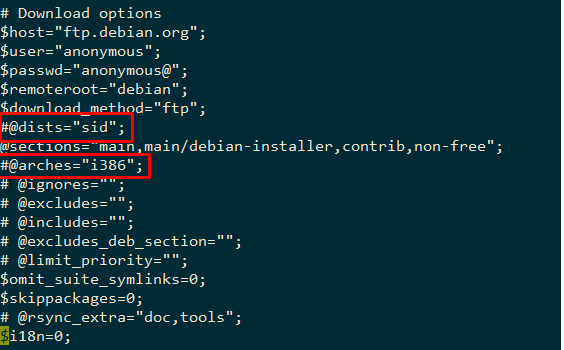
Khi đó, các client khi boot lên sẽ có password như đã cấu hình. Việc này nhằm mục đích tăng tính bảo mật, không cho người khác thấy rõ được password của bạn.

* + Để thực hiện boot PXE, người quản trị cần một DHCP server để cấp phát IP và chuyển hướng trực tiếp client boot tới TFTP server nơi mà nó có thể download các file boot. Cobbler có thể quản lý và thực hiện việc này, và đồng thời quản lý dịch vụ DNS (nếu có, tuy nhiên vẫn cần cấu hình) thông qua sửa đồi thông số manage\_dhcp và manage\_dns (cho phép dịch vụ DHCP chạy local trên máy server). Thực hiện sửa đổi bằng lệnh như sau:
  + sed -i 's/manage\_dhcp: 0/manage\_dhcp: 1/g' /etc/cobbler/settings
  + sed -i 's/manage\_dns: 0/manage\_dns: 1/g' /etc/cobbler/settings
  + Kích hoạt cho phép boot các file cấu hình cài đặt OS qua card mạng
  + sed -i 's/pxe\_just\_once: 0/pxe\_just\_once: 1/g' /etc/cobbler/settings
  + Chỉnh sửa IP của TFTP server (next\_server) và IP của Cobbler (server). Thực hiện các lệnh sau:
  + sed -i 's/next\_server: 127.0.0.1/next\_server: 192.168.122.110/g' /etc/cobbler/settings
  + sed -i 's/server: 127.0.0.1/server: 192.168.122.110/g' /etc/cobbler/settings

Trong đó, server là địa chỉ IP của cobbler server (lưu ý: không sử dụng địa chỉ 0.0.0.0, nên sử dụng địa chỉ IP mà bạn muốn các client sử dụng để liên lạc với cobbler server với các giao thức như http, tftp), next\_server là địa chỉ IP của TFTP server mà các file boot (kernel, initrd) được lấy về. Thường thì sẽ thiết lập cho cùng là Cobbler server.

### Cập nhật file cấu hình DHCP và DNSMASQ

* Phần này thực hiện cấu hình dải DHCP cho phép Cobbler cấp phát cho Client, và thông tin về các file pxelinux.0 gửi về cho client . Ở đây, cho phép trong dải từ 192.168.122.100 tới 192.168.122.200.
* Sửa file cấu hình của DHCP như sau vi /etc/cobbler/dhcp.template
* [...]
* subnet 192.168.122.0 netmask 255.255.255.0 {
* option routers 192.168.122.1;
* option domain-name-servers 8.8.8.8;
* option subnet-mask 255.255.255.0;
* range dynamic-bootp 192.168.122.100 192.168.122.200; #cau hinh dai dia chi IP cap phat cho Client
* default-lease-time 21700;
* max-lease-time 43100;
* next-server $next\_server; #vi tri cua TFTP server (cung la Cobbler luon) de Client lay cac file boot can thiet
* class "pxeclients" {
* match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";
* if option pxe-system-type = 00:02 {
* filename "ia64/elilo.efi";
* } else if option pxe-system-type = 00:06 {
* filename "grub/grub-x86.efi";
* } else if option pxe-system-type = 00:07 {
* filename "grub/grub-x86\_64.efi";
* } else {
* filename "pxelinux.0";
* }
* }
* }
* Cập nhật dải địa chỉ IP được cấp phát cho client trong file /etc/cobbler/dnsmasq.template như sau:
* [...]
* dhcp-range=192.168.122.100, 192.168.122.200
* Thực hiện comment @dists và @arches trong file /etc/debmirror.conf để hỗ trợ các distro debian:



* Khởi động lại rsyncd, cobbler và xinetd, sau đố đồng bộ lại cobbler dùng các lệnh sau:
* systemctl enable rsyncd.service
* systemctl restart rsyncd.service
* systemctl restart cobblerd
* systemctl restart xinetd
* systemctl enable xinetd
* systemctl restart dhcpd
* systemctl restart tftp
* cobbler get-loaders
* cobbler check
* cobbler sync

Như vậy là cấu hình xong cho Cobbler. Tham khảo các ghi chép tiếp theo để sử dụng Cobbler.

## **Bước 5: Truy cập vào giao diện Web xác nhận lại**

* Sau khi hoàn thành các bước trên, truy cập vào giao diện web của Cobbler như sau (***lưu ý: sử dụng https***):

https://192.168.122.110/cobbler\_web/

* Được giao diện như sau là thành công: