**Exercise Files**

[[https://www.evernote.com/images/file-generic.png](https://www.evernote.com/shard/s555/res/ff1a2dcc-147d-4a08-b535-d7a1b1c672f4/lfcs-linux-service-management.zip)**lfcs-linux-service-management...**4.6 MB](https://www.evernote.com/shard/s555/res/ff1a2dcc-147d-4a08-b535-d7a1b1c672f4/lfcs-linux-service-management.zip)

PluralSight - LCFS Linux Service Management Course Notes

**Lab Environment**

**Service Management and the Associated Certification Module Introduction**

Demo

Using systemd and systemctl to manage services

Starting Fresh

As we are dealing with so many services in this course, we will start with clean machines. server1 and server2 will be installed a fresh

Revisit Linux Essentials course for instructions of installing clean vms

Services that will be presented

DNS/DHCP/PXE/FTP

Email

SELinux and Services

HTTP/PHP/MariaDB

Printing

Practical

**Using Systemd to Manage Services**

Requirements on VM, ensure the command have already been executed

# yum install -y net-tools bash-completion vim-enhanced

# ps -fp 1

List the processes entirely. Note that the app used is systemd instead of initd

Systemd normally uses more power and resources than initd

# systemctl status sshd

List details including the man pages, config files, PID pf the service, and the control group along with the latest logs

# systemctl disable sshd

This will disable the service on boot but the service remains on until the system is shut down or restarted

# systemctl status sshd

This verifies that the service is still running but the state is set to disabled

# systemctl disable sshd

# systemctl mask sshd

This will disable the service and prevents anyone from turning the service back on until the use of unmask

# systemctl unmask sshd

#systemctl enable sshd

This will turn the mask off on the sshd and enable the service

**Service Management and the Associated Certification Module Recap**

Notes were not necessary for this module

**Configuring a BIND DNS Service Module Introduction**

Exam Objectives

Configure a basic DNS server

Maintain a DNS zone

Configure a caching-only name server

Configure a caching-only name server to forward DNS queries

Demo

install and test BIND

DNS forwarding

Identify DNS files and locations

Configure forward lookup zones

Using DNS APIs

**DNS Introduction**

BIND

Berkeley Internet Name Domain. The most widely implemented Domain Name System (DNS) service

**Configuring a Caching Only Server**

# yum install -y bind bind-utils

The package is called bind but the service is called named (name daemon)

Installing the BIND package and the utilities

# systemctl enable named

# systemctl start named

This will start and enable the service to start on boot

Start with the basic configuration which provides at least caching only

# iptables -L

There is no firewall setup on the freshly new VM's. the iptables will be empty

If necessary open up port 53 and tcp port 53

# netstat -ltn

List the ports including the 953 and 53 ports, one is for the service and the other for controlling the server remotely

# dig [www.pluralsight.com](http://www.pluralsight.com) @127.0.0.1

Perform the dig lookup on the local machine

This displays a query to the DNS 127.0.0.1 and displays results

This process will create a caching only server with BIND and BIND-utils

**Configuring Forwarding and Security**

# netstat -ltn

Verifies that the 127.0.0.53 ip listed is listening on the local host. The DNS server has been created but not configured to listen outside the local network

# vim /etc/named.conf

Displays the listening on port 53 but the interface is on 127.0.0.1

Change the port 53 configuration to any

listen-on port 53 { any; };

Change the v6 configuration setting to none

listen-on-v6 port 53 { none; };:x

:

# named-checkconf

Verifying the configuration file with this command. If no result, means configs is ok

# systemctl restart named.service

# netstat -ltn

The ip should be changed to the ip of the main and public ip for v4 and no listings displayed for v6 which is correct

# vi /etc/named.conf

Change the configuration file to allow only the local and the 192.168.56.0/24 ip range. This way, only the local area and machines with the ip range can query the DNS, otherwise, would be open to all for allow-query

allow-query     { localhost;     192.168.56.0/24;     localnets; };

# named-checkconf -v

Checking the configuration

-v switch is for verbose

# systemctl restart named.service

# dig [www.pluralsight.com](http://www.pluralsight.com) @127.0.0.1

Restarting service and checking to see if we still get an answer. This should resolve correctly. This is still using caching only processes

# vi /etc/named.conf

Change the file to allow for forwarding and security

Make the changes underneath the last option before the end brace to add the google servers

forwarders { 8.8.8.8; 8.8.4.4; };

forward only;

# named-checkconf -v

# systemctl restart named.service

# dig [www.pluralsight.com](http://www.pluralsight.com) @127.0.0.1

Check and restart the service again and do another dig. Response quicker

# dig [www.ford.com](http://www.ford.com) @127.0.0.1

It may be more useful to use another URL in case the information is cached

**Locating Files and DNS Logs**

# vi /etc/named.conf

The configuration for the logging resides in this file. Severity and file location can be adjusted here

Add the severity level to the logs by adding the print-severity line in the logging section, below the other configurations

print-severity yes;

# named-checkconf

Ensure no errors by running this command

# systemctl stop named

# > /var/named/data/named.run

Stop the named service and redirect the logging to named.run

# cat /var/named/data/named.run or # cat !$

Should not have data present at the moment

# systemctl start named

# cat /var/named/data/named.run

Should list logging information

Notice that all zones loaded correctly

# vi /etc/named.conf

Zone information is located below the logging braces (zone ".")

List the ip addresses of the root zone servers (top of the DNS tree)

# vi /etc/named.rfc1912.zones

File where references local zones

file "named.localhost";

Reverse lookup zones

file "named.loopback";

# ls /var/named

named.ca, named.loopback and named.localhost reside here

**Entering a Zone in the Named.Conf**

# vi /etc/named.conf

Configure the name.conf so that it knows about the zone

Add an entry in the configuration file above the existing zones for example.com

zone "example.com." {

     type master;

     file "db.example";

     allow-update { none; };

};

# named-checkconf

If error free, configurations have been entered in correctly for new zone. The zone still needs to be created as the configuration only references the zone

**Creating a DNS Zone**

# cd /var/named/

Database files live in this directory

# cp named.empty db.example

Copy the empty file and change it to the name used in the configuration file. This will copy an empty template that has the same permissions as the others

# chgrp named db.example

Change the group to named for db.example so that the group is named instead of root

# vi db.example

$TTL 3H - This indicates the length of time before expiry in cache

Variable called ORIGIN to append the origin if left blank

SOA (start of authority) line will need to be adjusted to indicate where to read the zone from and the mail authority

The serial will need to be adjusted to incrementally set the timeout for the configuration changes. Timeout cycles are according to time D-Day, W-Week, H-Hour

The file will look like the below

$TTL 3H

$ORIGIN example.com

example.com.     IN SOA     server1.example.com.     root.example.com. (

                                    dig                                            1          ;     serial

                                                                                1D      ;     refresh

                                                                                1H      ;     retry

                                                                                 1W     ;     expire

                                                                                3H  )     ;     minimum

example.com.     NS     server1.example.com.

server1                 A     192.168.56.107

# named-checkzone example.com db.example

Check to ensure the zone configurations are error free. Should result in "OK"

# systemctl restart named

# cat /var/named/data/named.run

Restart the system with the adjustments and check the logs to see if the new zone started correctly

**Using DNS Tools and APIs**

# dig server1.example.com @127.0.0.1

checking to see if the local connection resolves to the server1 correctly

# dig -t NS example.com @127.0.0.1

-t switch is type

Look for the name server records

# which python

# python --version

Checking the python location and version

# yum install python-dns

Install the python API to use dns from python

# vim /usr/share/doc/python-dns-1.12.0/examples/mx.py

Look at the mx file (mail exchange)

# python /usr/share/doc/python-dns-1.12.0/examples/mx.py

Run the script directly using python

# cp /usr/share/doc/python-dns-1.12.0/examples/mx.py /usr/share/doc/python-dns-1.12.0/examples/our.py

Copy the mail exchange file for a direct copy to use as a new file called our.py

# vim /usr/share/doc/python-dns-1.12.0/examples/our.py

Specify a resolver to query vi the local server. The file will look like the below

#! /usr/bin/env python

import dns.resolver

r = dns.resolver.Resolver()

r.nameservers = ['127.0.0.1']

answers = r.query('example.com', 'NS')

for rdata in answers:

          print rdata

# python /usr/share/doc/python-dns-1.12.0/examples/our.py

Execute the script after saving. Should resolve to the server1.example.com

**Configuring a BIND DNS Service Module Recap**

BIND DNS CentOS 7

# yum install bind bind-utils

service named

/etc/named.conf

listen-on

listen-on-v6

allow-query

forwarders

forward

directory

/var/named

named-checkconf

Zone Entry

zone "example.com." IN {

     type 'master';

     file 'db.example';

     allow-update { 'none'; };

}

Zone File

$TTL 3H

$ORIGIN example.com

example.com.     IN SOA     server1.example.com.     root.example.com. (

                                                                                1          ;     serial

                                                                                1D      ;     refresh

                                                                                1H      ;     retry

                                                                                 1W     ;     expire

                                                                                3H  )     ;     minimum

example.com.     NS     server1.example.com.

server1                 A     192.168.56.107

**Configuring FTP Servers Module Introduction**

Exam Objectives

Configure an FTP server

Configure anonymous-only download on FTP servers

Demo

Configure DNS Client on Server 2

Install vsftpd on Server 1

Create FTP YUM repo on Server 1

Use FTP Repo from Server 2

**Configure DNS client on Server 2**

# cat /etc/resolv.conf

This will lis the current information acquired from the NetworkManager

# vi /etc/sysconfig/network-scripts/ifcfg-enp0s3

Edit the configuration file for the enp0s3 interface

PEERDNS=no

IPV6\_PRIVACY=no

# vi /etc/sysconfig/network-scripts/ifcfg-enp0s8

Add a line for DNS1 to point to the other server and for PRIVACY=no

DNS1=192.168.56.106

IPV6\_PRIVACY=no

# reboot

Ensure the changes are in effect

# cat /etc/resolv.conf

# ping server1

The result will see an additional nameserver listed with the ping of server1 location

**Installing the vsftpd Service**

VSFTPD defined

vsftpd, (or very secure FTP daemon),[1] is an FTP server for Unix-like systems, including Linux. It is licensed under the GNU General Public License. It supports IPv6 and SSL. vsftpd supports explicit (since 2.0.0) and implicit (since 2.1.0) FTPS. vsftpd is the default FTP server in the Ubuntu, CentOS, Fedora, NimbleX, Slackware and RHEL Linux distributions

# yum install vsftpd

Install the package for the ftp on server 1

 # systemctl enable vsftpd

# systemctl start vsftpd

Start and enable at boot the vsftpd service

# netstat -ltn

Check the list of ports. Should see that port 22 has been opened for vsftpd service

Use the browser to test by going on a browser outside the VM Machines but on the main computer that the vm machines reside on and type the url ftp://192.168.56.107

the ftp site with the pub file should be displayed. This file is from the machine on /var/ftp/pub/. The top level and cannot navigate higher. The user will also be anonymous user

**Configure FTP to Allow Only Anonymous Connections**

# cd /etc/vsftpd/

Directory for the vsftps service. The configuration file resides in this directory

# vi /etc/vsftpd/vsftpd.conf

File that contains detailed information regarding the configuration of the vsftpd service. Adjust the file for anonymous only by removing all the unwanted information and keep the file like below

anonymous\_enable=YES

local\_enable=NO

write\_enable=NO

local\_umask=022

dirmessage\_enable=YES

xferlog\_enable=YES

listen=YES

listen\_ipv6=NO

pam\_service\_name=vsftpd

userlist\_enable=YES

tcp\_wrappers=YES

anon\_world\_readable\_only=YES

# systemctl restart vsftpd

# netstat -ltn

Restart the service when completed and check the ports. Now should display listening on port 21 for ipv4 only

Check the ftp url from the local browser again to ensure that can access the ftp site after configurations. Now only anonymous users can only read

**Creating an FTP YUM Repository**

Create a yum repository and add the Linux Distribution package to simulate a real life situation where the ftp would be used for. A CentOS 7 DVD file is required to be set in the Oracle VirtualBox Manager for server 1 to extract the information to place in the FTP server

# mount /dev/sr0 /mnt

Mount the newly created drive for the DVD file

# df -h

Checking the amount of memory the CentOS distribution will take and free space on the VM

# mkdir /var/ftp/pub/centos72

Create a centos72 directory

cpio Defined

cpio copies files into or out of a cpio or tar archive. The archive can be another file on the disk, a magnetic tape, or a pipe

# mount /dev/sr0 /mnt

Add the CentOS 7 dvd to the machine then mount it

# cd /mnt

# find . | cpio -pmd /var/pub/centos72

Copy the files over using cpio by first finding all the files in the current directory and using cpio to copy them to the pub/centos72 directory. This will take some time to complete

-p switch is for pass through mode

-m switch to maintain the last modified time of each file

-d switch to create the directories as required

#cd

# eject /mnt

Navigate out of mounted directory and eject DVD from VM

# df -h

# ls /var/ftp/pub/centos72/

Check the disk free after copy and list files in directory

**Connecting to the YUM Repo Using FTP**

# cd /etc/yum.repos.d

# ls

Directory where the default repos are located

# mv \* /root/

Move the existing files to the root users home directory

# vi ftp.repo

Create a new repo for the client machine (server2) and configuration. The file details are below

[ftpc7]

name=FTP CentOS 7.2

baseurl=ftp://server1.example.com/pub/centos72

enabled=1

gpgcheck=0

# yum clean all

Ensure the YUM is working correctly by cleaning cache

# yum install bash-completion

This has already been installed but can help us verify if server2 will check the repo as it is the only setup repo in server2

# yum repolist

Command to check which repos are listed

**Configuring FTP Servers Module Recap**

Configure DNS Client

First configured the DNS on server2

PEERDNS=no #External Card

PEERDNS=yes #Host-Only Card

DNS1=<ip address of server 1>

Install and Test FTP

On server1 we install the vsftpd FTP Server

# yum install -y vsftpd

# systemctl enable vsftpd && systemctl start vsftpd

ftp://<ip address of server 1>

/etc/vsftpd/vsftpd.conf

anonymous\_enable=YES

local\_enable=NO

write\_enable=NO

local\_umask=022

dirmessage\_enable=YES

xferlog\_enable=YES

connect\_from\_port\_20=YES

xferlog\_std\_format=YES

listen=YES

listen\_ipv6=NO

pam\_service\_name=vsftpd

userlist\_enable=YES

tcp\_wrappers=YES

anon\_world\_readable\_only=YES

Create Repo

On server1 will create an FTP repo by copying the full 7.2 DVD to the server

# mkdir /var/ftp/pub/centos72

# mount /dev/sr0 /mnt

# find /mnt/ | cpio -pmd /var/ftp/pub/centos72

Create Repo FIle

On server2, point to the YUM repo on server1. Remove all existing repos in /etc/yum.repos.d/ on server2. Create the file as ftp.repo as shown below

[ftpc7]

name=FTP CentOS 7.2

baseurl=ftp://server1.example.com/pub/centos72/

enabled=1

gpgcheck=0

**Configuring DHCP Module Introduction**

Exam Objectives

Configure a DHCP server

Demo

Configure static IP address on server1

Disable DHCP in VIrtualBox

Install DHCP on server1

Configure DHCP

Test DHCP Service

**The ISC DHCP Server**

Dynamic Host Configuration Protocol

Enables a server to automatically assign an IP Address and other network configuration to client devices

**Configure a static IP address**

# ip a s

Verify the ip address information on server1 on enp0s8

# vi /etc/sysconfig/network-scripts/ifcfg-enp0s8

Make adjustments to the file

BOOTPROTO=none

IPADDR='192.168.56.107'

NETMASK='255.255.255.0'

NETWORK='192.168.56.0'

DNS1='127.0.0.1'

PEERDNS=yes

# reboot

Reboot the machine or take down the connection and bring up with nmcli conn show

# ip a s

Display the IP information and verify the same ip for the interface. Take note of the interface and scope should be labeled as "scope global" instead of "scope global dynamic"

# cat /etc/resolv.conf

List the nameservers for the machine

# vi /etc/sysconfig/network-scripts/ifcfg-enp0s3

Adjust the config for the external interface so that the Peer DNS is set to no

PEERDNS=no

# nmcli connection show

List the two interfaces to see names of which connection to reload (enp0s3)

# nmcli connection reload enp0s3

Reload the interface that was adjusted

# cat /etc/resolv.conf

Now only one nameserver will be listed

**Disable VirtualBox DHCP and Install ISC DHCP Server**

File -> Preferences -> Network ->DHCP Server

Configure so the DHCP server is disabled in the Oracle VirtualBox. In the VirtualBox Preferences, disable the DHCP for the Host only Network adapter

# yum install -y dhcp

Install the dhcp package

**Configure an ISC DHCP Server**

# vi /etc/dhcp/dhcpd.conf

Configuration file for dhcp. Clear the file and make the following additions

option domain-name-servers 192.168.56.106;

option domain-search "example.vm";

default-lease-time 86400;

max-lease-time 86400;

ddns-update-style none;

authorative;

log-facility local4;

subnet 192.168.56.0 netmask 255.255.255.0 {

  range 192.168.56.151 192.168.56.254;

}

host server2 {

  hardware ethernet 08:00:27:12:0f:e1;

  fixed-address 192.168.56.120;

}

To get the hardware link/ether id, enter in command in the client server. For example, my Host is server1 and my client is server2, I will acquire the link from server2 to place in config for server1

# ip link show enp0s8

My results are 08:00:27:12:0f:e1

# dhcpd -t -cf /etc/dhcp/dhcpd.conf

-t switch for test

-cf for configuration file type

This command will test the dhcp to verify if its working. The error will be displayed if any occur. Even a missing brace can effect the outcome of the result, keep an eye for any missing characters

# systemctl start dhcpd

# systemctl enable dhcpd

# systemctl status dhcpd

Start and enable the service. There will be errors in the status indicating to configure enp0s3, this is normal and can be ignored for now

**Testing DHCP and DHCP Client**

# dhclient -r; dhclient

-r switch for release

Verify if the client can pick up the DHCP host. server1 is my host and server2 is my client. Execute the command on server2

# ip a s

# cat /var/lib/dhclient/dhclient.leases

File lists the leases including the newly configured enp0s8 interface details

**Configuring DHCP Module Recap**

Static IP on server1 (Host)

BOOTPROTO=none

IPADDR='192.168.56.106'

NETMASK='255.255.255.0'

NETWORK='192.168.56.0'

PEERDNS='yes'

DNS1='127.0.0.1'

Disable DHCP on VirtualBox

File > Preferences

Network > Host-Only Networks

Edit Settings > DHCP > Deselect Enable

Reboot host-system

Install DHCP

Installing the DHCP Server is stright forward on server1 (host)

# yum install -y dhcp

Configurations on server1 (host)

/etc/dhcp/dhcpd.conf

option domain-name-servers 192.168.56.106;

option domain-search "example.com";

default-lease-time 86400;

max-lease-time 86400;

ddns-update-style none;

authorative;

log-facility local4;

subnet 192.168.56.0 netmask 255.255.255.0 {

range 192.168.56.151 192.168.56.254;

}

host server2 {

hardware ethernet 08:00:27:12:0f:e1;

fixed-address 192.168.56.120;

}

Test DHCP Server

# dhclient -r enp0s8

# dhclient enp0s8

# cat /var/lib/dhclient/dhclient.leases

Create Repo FIle

On server2 (client) we will point to the YUM repo on server1 (host). We should remove all existing repo files in /etc/yum.repos.d/ on server2 (client). Then create this file as ftp.repo

[ftpc7]

name=FTP CentOS 7.2

baseurl=ftp://server1.example.com/pub/centos72/

enabled=1

gpgcheck=0

**Installing PXE Module Introduction**

Exam Objectives

Configure a PXE Boot Server

Demo

Install Syslinux and required packages

Configure TFTP and DHCP Server

Build PXE Options

Automate Installs with KickStart

Test PXE Installs

**Install SYSLINUX and Other Packages**

Syslinux Includes

SYSLINUX package

Boot from FAT filesystems

PXELINUX package

Network booting

ISOLINUX package

Bootable CD-ROMS and DVDs

EXTLINUX package

Boot from EXT or BTRFS filesystems

# yum install -y tftp tftp-server syslinux

Install the packages on host (server1) necessary for PXE

**Configuring TFTP and DHCP**

Ensure there is a DHCP server up and running

# vi /etc/dhcp/dhcpd.conf

Update the DHCP server configuration for TFTP server (server1) and the boot file "pxelinux.0" inside the subnet braces, underneath the range

next-server 192.168.56.107;

filename "pxelinux.0";

# dhcpd -t -cf /etc/dhcpd/dhcpd.conf

Test the new configuration changes. If the last command was updating the file, use # dhcpd -t -cf !$

# systemctl restart dhcpd

# systemctl status dhcpd

Restart and check the service. The errors regarding the listen can be ignored

# netstat -lun

-u switch for udp ports

-n switch for listen only

check the listening ports 67 and 68. 67 being the server and 68 being the client

# netstat -lu

This command can also verify using the "bootpc" and "bootps"

# ls /var/lib/tftpboot/

The root location of the tftp server

# cd /var/lib/tftpboot/

# cp /usr/share/syslinux/pxelinux.0 .

Navigate to tftpboot and copy the tiny network boot file from the share directory to current directory

# cp /usr/share/syslinux/menu.c32 .

Copy the menu file from the share directory to current

# cp /var/ftp/pub/centos72/isolinux/vmlinuz .

# cp /var/ftp/pub/centos72/isolinux/initrd.img

Copy the ram disk and kernel from the CentOS DVD

# ls

Verify the files initrd.img, menu.c32, pxelinux.0, vmlinuz

# systemctl start tftp.socket

# systemctl enable tftp.socket

Start up the tftp socket

# netstat -lun

Verify the ports for DHCP Server: 67, DHCP Client: 68 TFTP Server: 69

**The Default PXE File**

# cd /var/lib/tftpboot

# yum install tree

# tree .

List the items in tree view

# mkdir pxelinux.cfg

Create a new subdirectoy to cater different clients. The director labeled "pxelinux.cfg"

# vi pxelinux.cfg/default

Create the default file inside the pxelinux.cfg directory with the following information. Add enough information to verify it's working

default menu.c32

prompt 0

timeout 1000

ontimeout local

menu title Pluralsight Boot Menu

label local

  menu label Boot from local disk

  localboot 0xffff

In the Oracle VirtualBox, create a new machine called "test" and leave all defaults

In Settings > System, deselect Floppy and Optical from boot order and select the network. Ensure the Network is above Hard Disk for boot order

In Settings > Network, change the Adapter 1 Settings to Host Only Adapter

Start the machine and cancel the prompt regarding the Hard Drive. The PXE boot firmware will start with the default menu file options displayed

**MAC Specific Configuration**

# cd /var/lib/tftpboot

# tree

List the current files including the default file in the pxelinux.cfg directory. This default file will automatically work but if we require specific boot processes dependant on the machine, we could use MAC UUID to specify which configuration goes with what machine

Acquire the MAC address for the test machine in Oracle VirtualBox Manager. Network > Advanced, copy the MAC Address

# cd pxelinux.cfg

# cp default 01-08-00-27-08-42-ca

Copy the default file to a new file and cal it the name of the MAC address including "01" in the beginning to signify the network type being Ethernet. Ensure the letters are set as lower case

# vi 01-08-00-27-08-42-ca

Add in a new stanza to the file underneath the Local label. This will add an additional option to teh menu to boot manually using the ftp server that was created prior. The ftp and the tftp servers need to be active. Ensure the paths are correct for the ftp server and the variables are set correctly for kernel type and initrd

label install

  menu label Manual Install CentOS 7.2

  kernel vmlinuz

  append initrd=initrd.img ip=dhcp repo=ftp://192.168.56.107/pub/centos72

# cd ..

# tree

Verify the file is listed in the proper place

Go back to the test machine and boot it. The new menu will appear with the new option to install manually. Choose to install with CedntOS 7.2 to ensure the install menu appears

**Editing a KickStart File to Automate CentOS Installs**

# ls

# cp anaconda-ks.cfg /var/ftp/pub/install.ks

# chmod 644 /var/ftp/pub/install.ks

Copy the anaconda kernel file from the root home directory to the ftp directory and adjust the permissions

# vim /var/ftp/pub/install.ks

Edit the file to use the ftp and repo servers and reference the paths while removing redundant information to automate the install and adding a new user

auth --enableshadow --passalgo=sha512

install

url --url="ftp://192.168.56.106/pub/centos72"

text

firstboot --disable

ignoredisk --only-use=sda

keyboard --vckeymap=gb --xlayouts='gb'

lang en\_GB.UTF-8

network --bootproto=dhcp --device=enp0s3 --ipv6=auto --activate

network --hostname=server3.example.vm

rootpw --iscrypted $6$XsKFqwNbBwMHHVWr$cCK3NQb11Ke8p2HtQIAfB6dB7kAaPWuGthgyrF6UUacyneH269oQEQmWI7vymVV/qEGlQ7jjRIIU.YkTs9TBx/

timezone Europe/London --isUtc

user --groups=wheel --name=dansaf --password=$6$b2ZByT9yjc3lid2t$AeBHa4/ifyGKjigXyPKvHk72tWr1Zwli00Voc.QTBS37d0V8j7mFwjDga9GrRXei6OcUu5NQ3CKI62aS2kvky. --iscrypted

bootloader --append=" crashkernel=auto" --location=mbr --boot-drive=sda

autopart --type=lvm

clearpart --all --initlabel

repo --name=server1 --baseurl="ftp://192.168.56.106/pub/centos72/"

%packages

@^minimal

@core

kexec-tools

%end

%addon com\_redhat\_kdump --enable --reserve-mb='auto'

%end

# cd /var/lib/tftpboot

# tree

# vim pxelinux.cfg/01-08-00-27-08-42-ca

Verify and edit the 01 file specific to the test machine boot. Change the localboot and repo and path to the below

localboot 0

append initrd=initrd.img ip=dhcp ks=ftp://192.168.56.107/pub/install.ks

**Automating Installs and Test**

Boot the test machine and choose Manual Install option and press Tab key to see the path and execute the automation.

Once the install has been completed, the machine will reboot to the same PXE menu. choose the first option to boot from local and press Tab key.

**Installing PXE Module Recap**

Install

# yum install tftp tftp-server syslinux

Populate tftpboot

# cd /var/lib/tftpboot

# cp /usr/share/syslinux/pxelinux.0 .

# cp /usr/share/syslinux/menu.c32 .

# cp /var/ftp/pub/centos72/isolinux/vmlinuz .

# cp /var/ftp/pub/centos72/isolinux/initrd.img .

/etc/dhcp/dhcpd.conf

subnet 192.168.56.0 netmask 255.255.255.0 {

range 192.168.56.151 192.168.56.254;

next-server 192.168.56.106;

filename “pxelinux.0”;

}

systemctl restart dhcpd

systemctl enable tftp.socket

systemctl start tftp.socket

PXELINUX.CFG

# mkdir /var/lib/tftpboot/pxelinux.cfg

# vim /var/lib/tftpboot/pxelinux.cfg/default

default menu.c32

prompt 0

timeout 1000

ontimeout localmenu

title Pluralsight Boot Menu

label local

menu label Boot from local disk

localboot 0

Manual Install

label install

menu label Manual Install CentOS 7.2

kernel vmlinuz

append initrd=initrd.img ip=dhcp repo=ftp://s1/pub/centos72

Auto Install

label ks

menu label Automated Install CentOS 7.2

kernel vmlinuz

append initrd=initrd.img ip=dhcp ks=ftp://s1/pub/install.ks

KickStart File

You may work with the /root/anaconda-ks.cfg as a start. The above is an extract only

auth --enableshadow --passalgo=sha512

install

url --url="ftp://192.168.56.106/pub/centos72“

text

firstboot --disable

ignoredisk --only-use=sda

keyboard --vckeymap=gb --xlayouts='gb'

**Configuring Email in CentOS 7 Module Introduction**

Exam Objectives

Configure an SMTP service

Restrict access to an SMTP service

Configure a mail transfer agent (MTA) to accept inbound email from other systems

Configure a MTA forward (relay) email through a smart host

Configure email aliases

Configure an IMAP and IMAPS service

Demo

Install and verify Postfix

Postfix configuration files

Create DNS MX records

Forward email from SMTP servers

Install Dovecot IMAP server

Install Mutt email client

**Introduction to Email in CentOS 7**

Outgoing Mail Process

Incoming Mail Process

Components

SMTP Server

Simple Mail Transfer Protocol Server

SMTP defined

Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (email) transmission. First defined by RFC 821 in 1982, it was last updated in 2008 with the Extended SMTP additions by RFC 5321—which is the protocol in widespread use today. SMTP by default uses TCP port 25

Also referred to as outgoing mail server which will send it to the SMTP server of the destination

Using Postfix

IMAP / POP3 Server

Internet Message Access Protocol

IMAP defined

In computing, the Internet Message Access Protocol (IMAP) is an Internet standard protocol used by e-mail clients to retrieve e-mail messages from a mail server over a TCP/IP connection. IMAP is defined by RFC 3501

Post Office Protocol Version 3

POP3 defined

Post Office Protocol version 3 (POP3) is a standard mail protocol used to receive emails from a remote server to a local email client. POP3 allows you to download email messages on your local computer and read them even when you are offline

Routed through to the mx record from the domain to the SMTP server of the domain, then to the IMAP /POP3 server. The client that is supposed to receive the message will log into the IMAP / POP3 to see the message they recieved

Using Dovecot

Email Client

Using Mutt

**Install and Verify Postfix MTA**

# yum install postfix

# alternatives --display mta

Should prompt link currently points to /usr/sbin/sendmail.postfix

# systemctl status postfix.service

Check status of the postfix service

# netstat -ltn

Check for port 25 which should be listed and is the necessary port for smtp. In order for the message to be sent out, it is required that we have External addresses listed

**Working with Out of the Box Configuration**

# netstat -ltn

Lists the local addresses are working

# yum install mailx

Install a local mail package to test the local send and receive. May have been installed

# mail

Lists the mail but at this moment should be none

# mail root

subject test

This is a test message

.

EOT

# mail

Now there should be one listed for the root user. Press the "Q" key to quit

Ensure that you can also test this on a different virtual machine with postfix package installed and send a message to the root mail of the machine The steps above need to applied to a new vm. If all works well, try it from vm to vm

# mail [root@server2.example.com](mailto:root@server2.example.com)

Address points to the first vm that was tested. The first vm should now have a mail from server1. Errors will occur the first time around as the configurations have not been set yet to receive internally from another machine or externally.

# tail /var/log/maillog

Log file can also give some information regarding the error

**Receiving SMTP Messages from Other Clients and Hosts**

postconf Defined

The postconf command displays the actual values of configuration parameters, changes configuration parameter values, or displays other configuration information about the Postfix mail system

# postconf

This will list all configurations

# postconf -d

-d switch to see defaults

-n switch to see configurations that were explicit set

# postconf -n config\_directory

Command will display the config directory location

# ls /etc/postfix

Lists the configuration directory

#vi /etc/postfix/main.cf

Postfix configuration file

# cd /etc/postfix

# cp main.cf main.cf.$(date +%F)

Create a copy of the main.cf file using a date variable for the extension so that there is a sample to refer to if necessary

# postconf -n inet\_interfaces

List the current interfaces in the Postfix configuration that are being listened on

#postconf -n inet\_protocols

Lists the ipv4

# postconf -e inet\_protocols=ipv4

-e switch for edit

Change the ipv4 protocol from all to ipv4

# postconf -e inet\_interfaces=all

Change the listening to all interfaces

# systemctl restart postfix.service

Restart the service for changes to be in effect

# netstat -ltn

Check to see if the port is listening with the changes for all on ipv4

**Configuring DNS MX Records**

# vi /var/named/db.example

MX or mail exchange record. Add the MX records for all associated clients

server2     A     192.168.56.120

server3     A     192.168.56.153

example.com.     MX     10     server1.example.com.

# systemctl restart named

Restart the service to verify

# dig -t MX example.com

In the ANSWER section the MX will be listed

**Configuring Postfix to Send and Receive Messages**

# postconf mydestination

Look at the configuration for the destination settings to check for the $mydomain variable

# postconf -e 'mydestination=localhost,$myhostname,localhost.$mydomain,$mydomain'

# postconf mydestination

Verify the adjustment was added correctly

# postconf myorigin

Verify when a message is sent, its going to be delivered based on the host name

# postconf -e 'myorigin=$mydomain'

# postconf myorigin

Adjust the origin to $mydomain, verify the change to the myorigin variable myorigin=mydomain

# postconf mydomain

This should be set to the domain of the server in this case 'example.com'

# postfix check

Verify no syntax errors will occur in restart of services

# systemctl restart postfix.service

**Configuring SMTP Relay**

Forwarding email would be referred to the same process as relaying email. With the SMTP protocols, the relay of email would be the more precise term

# cd /etc/postfix/

# cp main.cf main.cf.$(date +%F)

Copy the main.cf file of the client before adjusting

# postconf -e inet\_interfaces=all

Set interfaces to all

# postconf -e inet\_protocols=ipv4

Set protocols to ipv4

# postconf -e relayhost=mailserver1.example.com

Set the relayhost to equal the mail server

# postfix check

# systemctl restart postfix.service

Verify no syntax errors before restarting service

Repeat the process for any other clients so that all clients will relay to the host before sending out which minimizes footprint

Run a test mail from one server to another with the relay configured

# tail /var/log/maillog

Check the logs to verify the relay worked properly

# tail /var/log/maillog

Check the server with the relay configured to verify that it did in fact receive the message and sent it to the proper place