ใบงาน

รหัสวิชา 30901-2004

วิชา ระบบปฏิบัติการเครื่องแม่ข่าย

ครูผู้สอน

นายวรกิจ วิริยะเกษามงคล

แผนกวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคนิคชลบุรี

ใบงาน

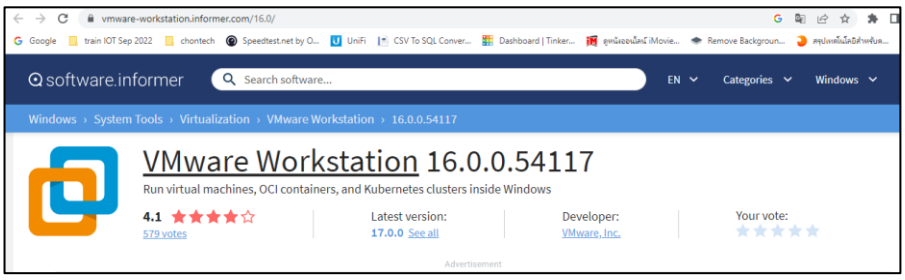
ชื่องาน ติดตั้งระบบปฏิบัติการบนโปรแกรมจำลองเครื่องคอมพิวเตอร์

วิชา 30901-2004 ชื่อวิชา ระบบปฏิบัติการเครื่องแม่ข่าย ทฤษฎี 1 ปฏิบัติ 4 หน่วยกิต 3

ขั้นตอนการทำงาน

1. เตรียมโปรแกรม vmware workstation

   Download ไฟล์ VMware-workstation-full-16.xxx.exe

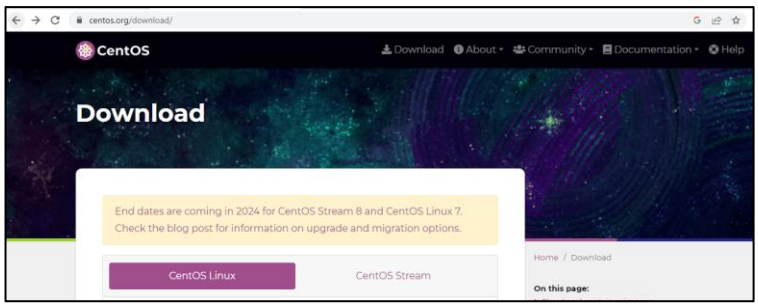


2. ติดตั้งโปรแกรม vmware workstation

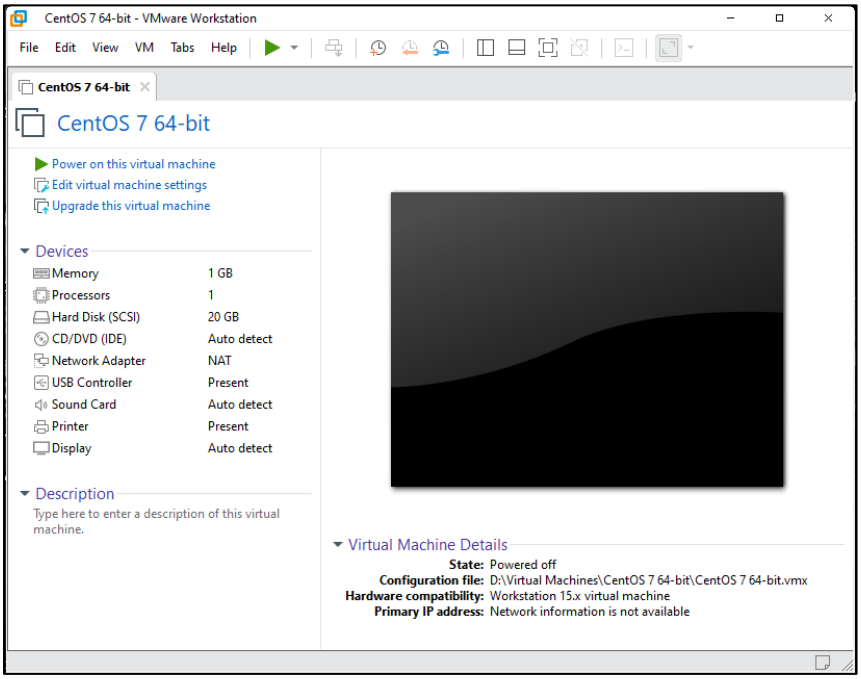
   https://www.youtube.com/watch?v=AMo77WFIZD4

3. เตรียมไฟล์โปรแกรม ISO สำหรับติดตั้ง centos 7 บันทึกข้อมูลไว้ในฮาร์ดดิสก์

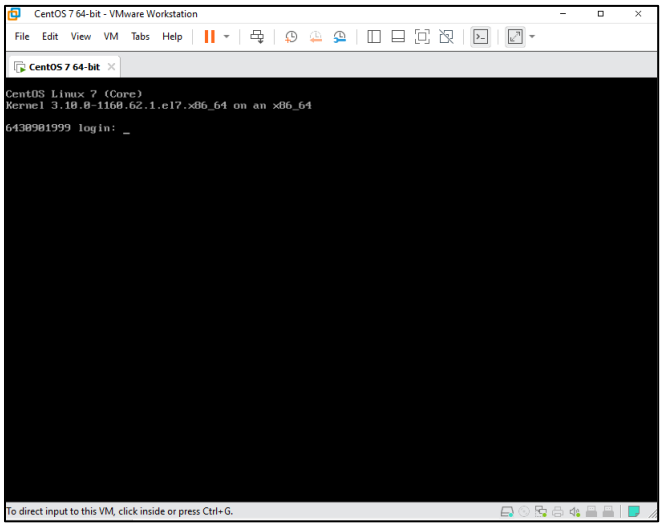
   Download ไฟล์ CentOS-7-x86_64-Minimal-1810.iso



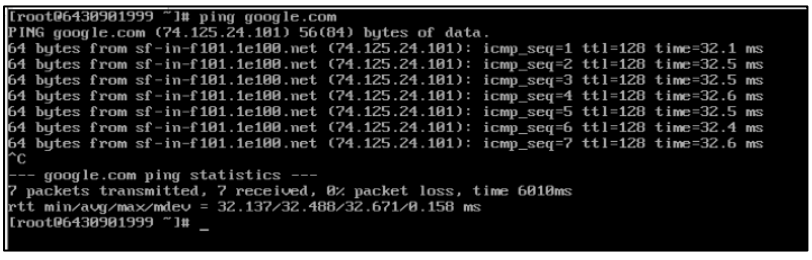
4. Boot ระบบปฏิบัติการจากไฟล์ ISO ที่เตรียมไว้ในฮาร์ดดิสก์ ด้วยโปรแกรม vmware



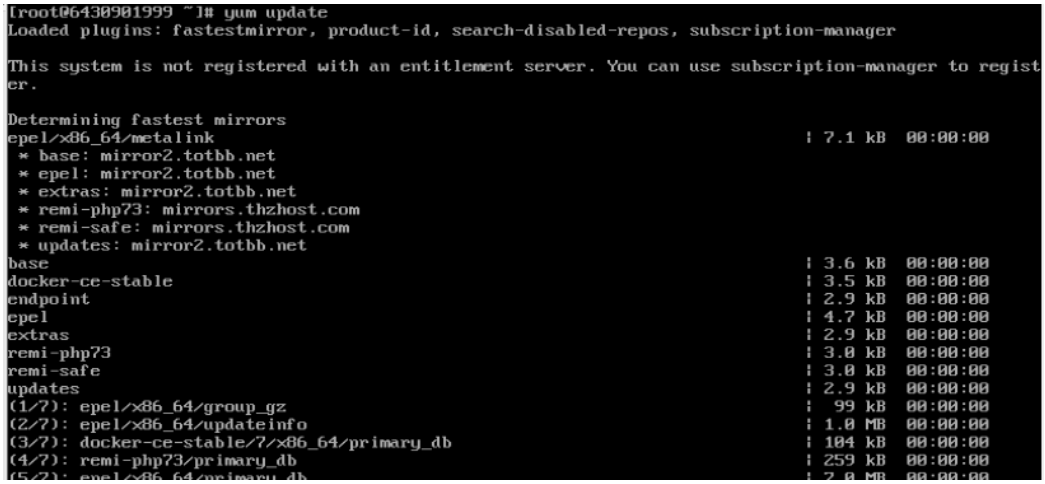
5. ติดตั้งระบบปฏิบัติการ CentOS



6. ทดสอบการเชื่อมต่อ internet



7. ทำการ update kernel



ใบงาน

ชื่องาน การตั้งค่าพื้นฐาน การให้บริการ Web Server

วิชา 30901-2004 ชื่อวิชา ระบบปฏิบัติการเครื่องแม่ข่าย ทฤษฎี 1 ปฏิบัติ 4 หน่วยกิต 3

คำสืบค้น (Key word) centos 7 setup web server

https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-centos-7

Introduction

The Apache HTTP server is the most widely-used web server in the world. It provides many powerful features including dynamically loadable modules, robust media support, and extensive integration with other popular software.

In this guide, you will install an Apache web server with virtual hosts on your CentOS 7 server.

Step 1 — Installing Apache

Apache is available within CentOS's default software repositories, which means you can install it with the `yum` package manager.

As the non-root sudo user configured in the prerequisites, update the local Apache `httpd` package index to reflect the latest upstream changes:

```
1. sudo yum update httpd
```

Once the packages are updated, install the Apache package:

```
1. sudo yum install httpd
```

After confirming the installation, `yum` will install Apache and all required dependencies.

If you completed the Additional Recommended Steps for New CentOS 7 Servers guide mentioned in the prerequisites section, you will have installed `firewalld` on your server and you'll need to open up port `80` to allow Apache to serve requests over HTTP. If you haven't already done so, you can do this by enabling `firewalld`'s `http` service with the following command:

```
1. sudo firewall-cmd --permanent --add-service=http
```

If you plan to configure Apache to serve content over HTTPS, you will also want to open up port `443` by enabling the `https` service:

```
1. sudo firewall-cmd --permanent --add-service=https
```

Next, reload the firewall to put these new rules into effect:

```
1. sudo firewall-cmd --reload
```

After the firewall reloads, you are ready to start the service and check the web server.

Step 2 — Checking your Web Server

Apache does not automatically start on CentOS once the installation completes. You will need to start the Apache process manually:

```
1. sudo systemctl start httpd
```

Verify that the service is running with the following command:

```
1. sudo systemctl status httpd
```

You will see an `active` status when the service is running:

```
Output

Redirecting to /bin/systemctl status httpd.service

● httpd.service - The Apache HTTP Server

   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled;
vendor preset: disabled)

   Active: active (running) since Wed 2019-02-20 01:29:08 UTC; 5s ago

     Docs: man:httpd(8)

           man:apachectl(8)

 Main PID: 1290 (httpd)

   Status: "Processing requests..."

   CGroup: /system.slice/httpd.service

           ├─1290 /usr/sbin/httpd -DFOREGROUND

           ├─1291 /usr/sbin/httpd -DFOREGROUND

           ├─1292 /usr/sbin/httpd -DFOREGROUND
```

```
           ├─1293 /usr/sbin/httpd -DFOREGROUND
           ├─1294 /usr/sbin/httpd -DFOREGROUND
           └─1295 /usr/sbin/httpd -DFOREGROUND
...
```

As you can see from this output, the service appears to have started successfully. However, the best way to test this is to request a page from Apache.

You can access the default Apache landing page to confirm that the software is running properly through your IP address. If you do not know your server's IP address, you can get it a few different ways from the command line.

Type this at your server's command prompt:

```
1. hostname -I
```

This command will display all of the host's network addresses, so you will get back a few IP addresses separated by spaces. You can try each in your web browser to see if they work.

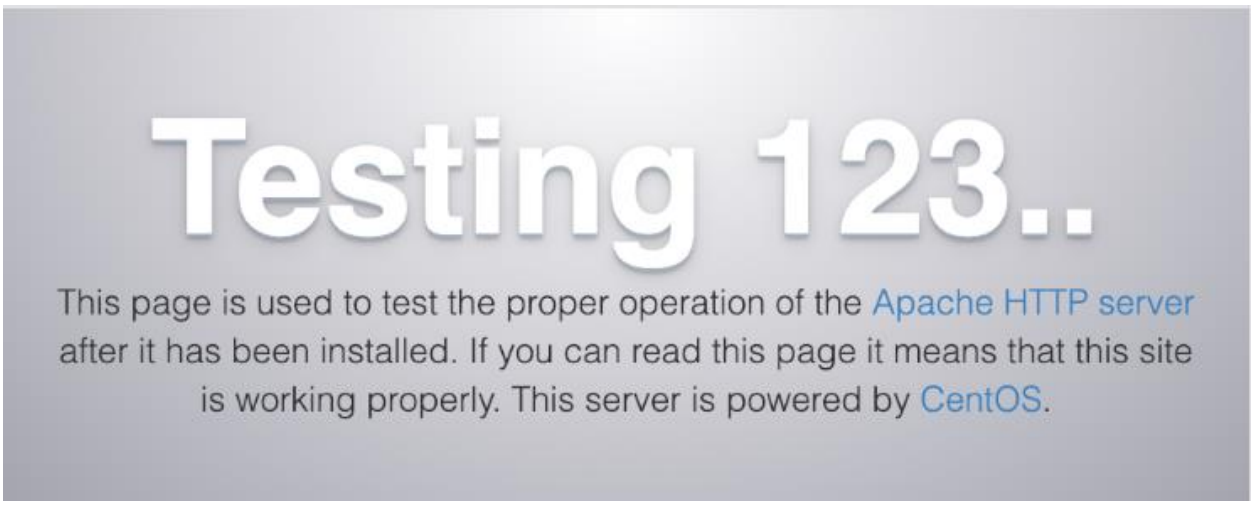
Alternatively, you can use `curl` to request your IP from `icanhazip.com`, which will give you your public IPv4 address as seen from another location on the internet:

```
1. curl -4 icanhazip.com
```

When you have your server's IP address, enter it into your browser's address bar:

```
http://your_server_ip
```

You'll see the default CentOS 7 Apache web page:



This page indicates that Apache is working correctly. It also includes some basic information about important Apache files and directory locations. Now that the service is installed and running, you can now use different `systemctl` commands to manage the service.

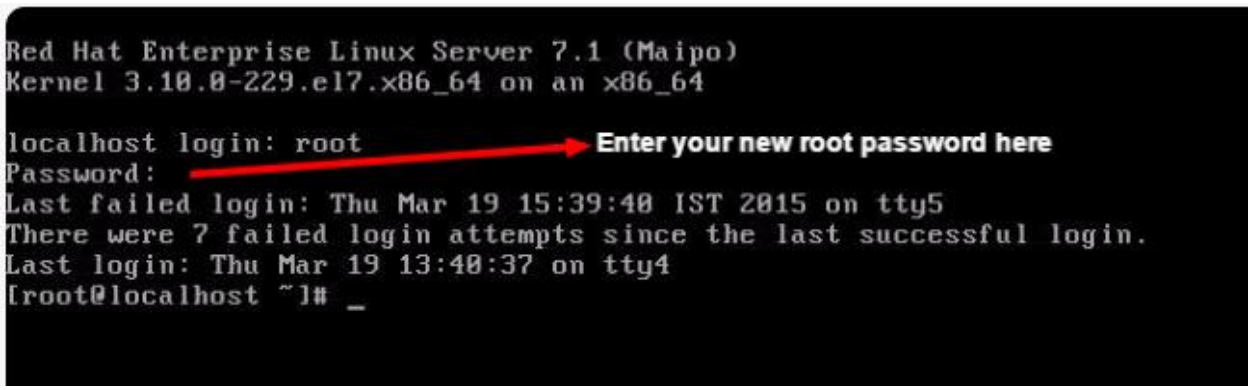
ใบงาน

ชื่องาน การ upload เว็บไซต์ขึ้น server

วิชา 30901-2004 ชื่อวิชา ระบบปฏิบัติการเครื่องแม่ข่าย ทฤษฎี 1 ปฏิบัติ 4 หน่วยกิต 3

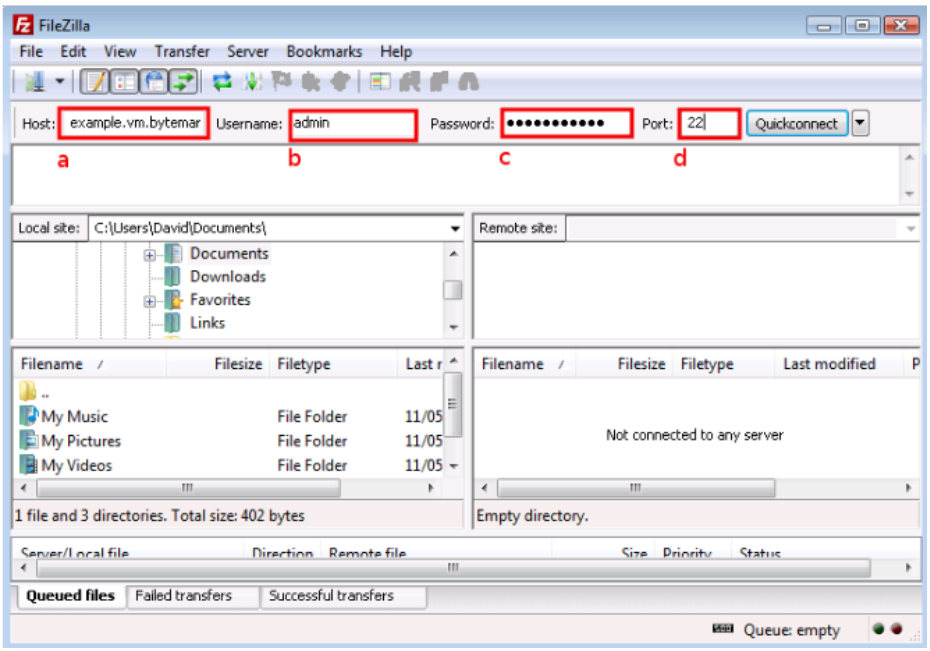
---

1. การ login ด้วย Root

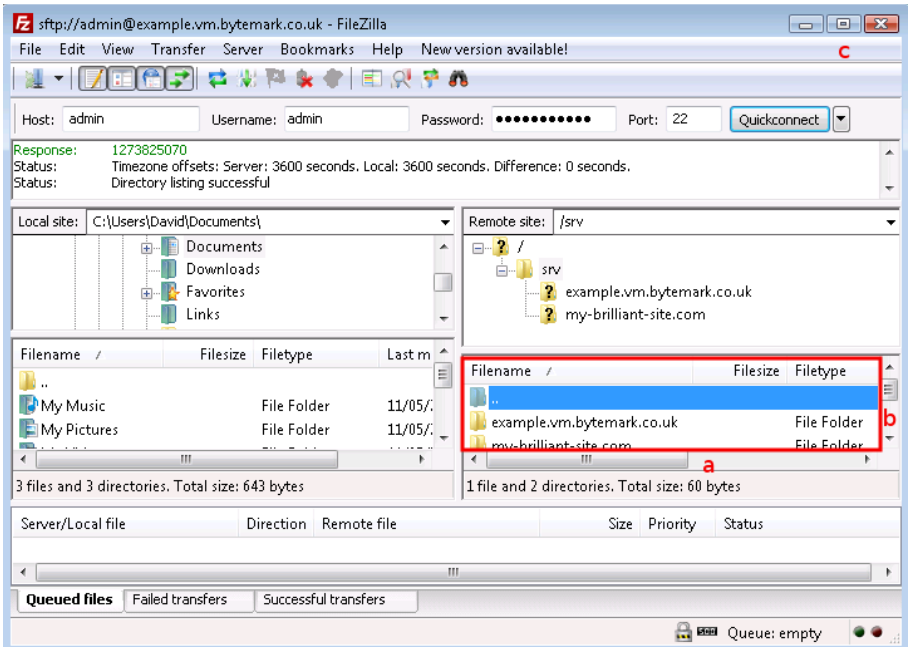


2. ทดสอบการทำงานของ web Server

3. ใช้โปรแกรม FileZilla เชื่อมต่อกับ Web Server ด้วย IP address



4. เตรียมข้อมูลเว็บไซต์ ที่สามารถเปิดทดสอบจากเครื่องคอมพิวเตอร์ laptop ได้
5. สร้างโฟลเดอร์บน Web Server แล้วทำการ upload ไฟล์ของเว็บไซต์ขึ้น Web Server



6. ทดสอบเว็บไซต์ จากเครื่องคอมพิวเตอร์ laptop ด้วยหมายเลข IP ของ Server

ใบงาน

ชื่องาน งานติดตั้งและตั้งค่าพื้นฐาน การให้บริการฐานข้อมูล

วิชา 30901-2004 ชื่อวิชา ระบบปฏิบัติการเครื่องแม่ข่าย ทฤษฎี 1 ปฏิบัติ 4 หน่วยกิต 3

---

1. ทำการติดตั้ง Service mariadb

Installing MariaDB Server

The RHEL 7 and CentOS 7 distributions include MariaDB Server 5.5 by default.

You can install MariaDB Server from the command-line:

```
yum install mariadb-server
```

```
systemctl start mariadb.service
```

2. ทำการตั้ง Password ให้กับ root

```
   mysql_secure_installation
```

3. ทดสอบทดสอบการทำงานของฐานข้อมูลจาก Command

We can verify our installation and get information about it by connecting with the `mysqladmin` tool, a client that lets you run administrative commands. Use the following command to connect to MariaDB as **root** (`-u root`), prompt for a password (`-p`), and return the version.

```
mysqladmin -u root -p
```

```
CREATE DATABASE db1;
Query OK, 1 row affected (0.18 sec)

CREATE DATABASE db1;
ERROR 1007 (HY000): Can't create database 'db1'; database exists

CREATE OR REPLACE DATABASE db1;
Query OK, 2 rows affected (0.00 sec)

CREATE DATABASE IF NOT EXISTS db1;
Query OK, 1 row affected, 1 warning (0.01 sec)
```

ใบงาน

ชื่องาน งานติดตั้งและตั้งค่าพื้นฐาน เครื่องมือจัดการฐานข้อมูล (PhpMyAdmin)

วิชา 30901-2004 ชื่อวิชา ระบบปฏิบัติการเครื่องแม่ข่าย ทฤษฎี 1 ปฏิบัติ 4 หน่วยกิต 3

คำสืบค้น (Key word) centos 7 setup phpmyadmin

https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-phpmyadmin-with-apache-on-a-centos-7-server

Introduction

Relational database management systems like MySQL and MariaDB are needed for a significant portion of web sites and applications. However, not all users feel comfortable administering their data from the command line.

To solve this problem, a project called phpMyAdmin was created in order to offer an alternative in the form of a web-based management interface. In this guide, we will demonstrate how to install and secure a phpMyAdmin configuration on a CentOS 7 server. We will build this setup on top of the Apache web server, the most popular web server in the world.

Step 1 — Install phpMyAdmin

With our LAMP platform already in place, we can begin right away with installing the phpMyAdmin software. Unfortunately, phpMyAdmin is not available in CentOS 7's default repository.

To get the packages we need, we'll have to add an additional repo to our system. The EPEL repo (**E**xtra **P**ackages for **E**nterprise **L**inux) contains many additional packages, including the phpMyAdmin package we are looking for.

The EPEL repository can be made available to your server by installing a special package called `epel-release`. This will reconfigure your repository list and give you access to the EPEL packages.

To install, just type:

```
sudo yum install epel-release
```

Now that the EPEL repo is configured, you can install the phpMyAdmin package using the `yum` packaging system by typing:

```
sudo yum install phpmyadmin
```

The installation will now complete. The installation included an Apache configuration file that has already been put into place. We will need to modify this a bit to get it to work correctly for our installation.

Open the file in your text editor now so that we can make a few changes:

```
sudo nano /etc/httpd/conf.d/phpMyAdmin.conf
```

Inside, we see some directory blocks with some conditional logic to explain the access policy for our directory. There are two distinct directories that are defined, and within these, configurations that will be valid for both Apache 2.2 and Apache 2.4 (which we are running).

Currently, this setup is configured to deny access to any connection not being made from the server itself. Since we are working on our server remotely, we need to modify some lines to specify the IP address of your *home* connection.

Change any lines that read `Require ip 127.0.0.1` or `Allow from 127.0.0.1` to refer to your home connection's IP address. If you need help finding the IP address of your home connection, check out the next section. There should be four locations in the file that must be changed:

```
. . .

Require ip your_workstation_IP_address

. . .

Allow from your_workstation_IP_address

. . .

Require ip your_workstation_IP_address

. . .

Allow from your_workstation_IP_address

. . .
```

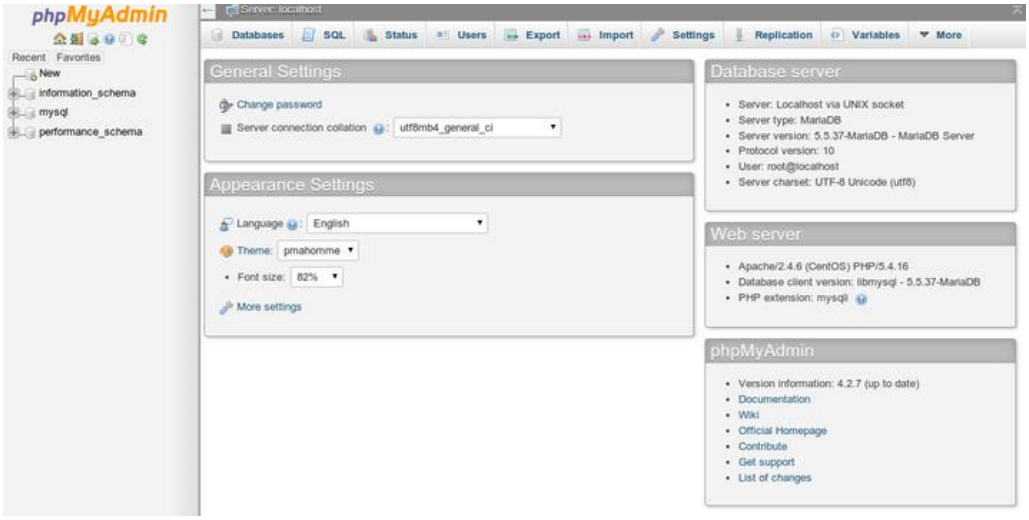
When you are finished, restart the Apache web server to implement your modifications by typing:

```
sudo systemctl restart httpd.service
```

With that, our phpMyAdmin installation is now operational. To access the interface, go to your server's domain name or public IP address followed by `/phpMyAdmin`, in your web browser:

```
http://server_domain_or_IP/phpMyAdmin
```

To sign in, use a username/password pair of a valid MariaDB user. The `root` user and the MariaDB administrative password is a good choice to get started. You will then be able to access the administrative interface:



ใบงาน

ชื่องาน งานตั้งค่าพื้นฐาน การให้บริการ Web Server กับ User

วิชา 30901-2004 ชื่อวิชา ระบบปฏิบัติการเครื่องแม่ข่าย ทฤษฎี 1 ปฏิบัติ 4 หน่วยกิต 3

การเพิ่ม user เข้าระบบ

How to Create a New User in Linux

To create a new user account, invoke the `useradd` command followed by the name of the user.

For example to create a new user named `username` you would run:

```
sudo useradd username
```

When executed without any option, `useradd` creates a new user account using the default settings specified in the `/etc/default/useradd` file.

The command adds an entry to the `/etc/passwd`, `/etc/shadow`, `/etc/group` and `/etc/gshadow` files.

To be able to log in as the newly created user, you need to set the user password. To do that run the `passwd` command followed by the username:

```
sudo passwd username
```

You will be prompted to enter and confirm the password. Make sure you use a strong password.

How To Enable Apache UserDir In CentOS 7/RHEL 7

Install apache:

```
yum install httpd -y
```

Enable Apache Userdirs

```
vi  /etc/httpd/conf.d/userdir.conf
```

```
<IfModule mod_userdir.c>

 #

 # UserDir is disabled by default since it can confirm the presence

 # of a username on the system (depending on home directory

 # permissions).

 #

UserDir enabled unixmenuser



 #

 # To enable requests to /~user/ to serve the user's public_html

 # directory, remove the "UserDir disabled" line above, and
uncomment

 # the following line instead:

 #

 UserDir public_html
```

```
</IfModule>


<Directory /home/*/public_html>
Options Indexes Includes FollowSymLinks
##For  apache 2.2,Please use:


 AllowOverride All
 Allow from all
 Order deny,allow
#For apache >= 2.4,Please  use :
  Require all granted


</Directory>
```

Restart apache

```
systemctl restart httpd.service
```

Then create user's **public_html** and assign permissions.

```
mkdir /home/unixmenuser/public_html
chmod 711 /home/unixmenuser
chown unixmenuser:unixmenuser /home/unixmenuser/public_html
```

```
chmod 755 /home/unixmenuser/public_html
```

Then here's the other new things, especially you are using SELinux

```
setsebool -P httpd_enable_homedirs true

chcon -R -t httpd_sys_content_t /home/unixmenuser/public_html
```

Run the test by navigating to the following URL from your browser.

http://ip/~username

ใบงาน

ชื่องาน งานติดตั้งและตั้งค่าพื้นฐาน การให้บริการ DHCP

วิชา 30901-2004 ชื่อวิชา ระบบปฏิบัติการเครื่องแม่ข่าย ทฤษฎี 1 ปฏิบัติ 4 หน่วยกิต 3

How to Install and Configure DHCP Server on CentOS7

1.Install DHCP Package

```
# yum install dhcp
```

2.Update /etc/sysconfig/dhcpd File

```
# nano /etc/sysconfig/dhcpd

DHCPDARGS=enp0s8
```

3.Configure DHCP Server copy the content of sample configuration file to the main configuration file.

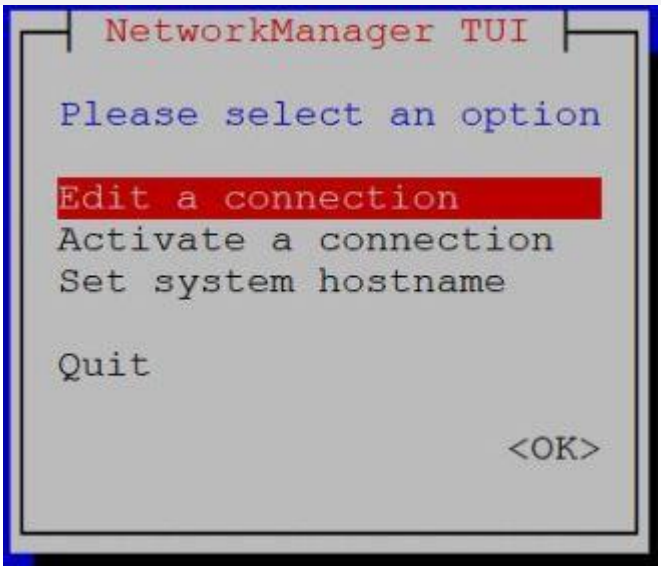
```
# cp /usr/share/doc/dhcp-4.2.5/dhcpd.conf.example /etc/dhcp/dhcpd.conf
```

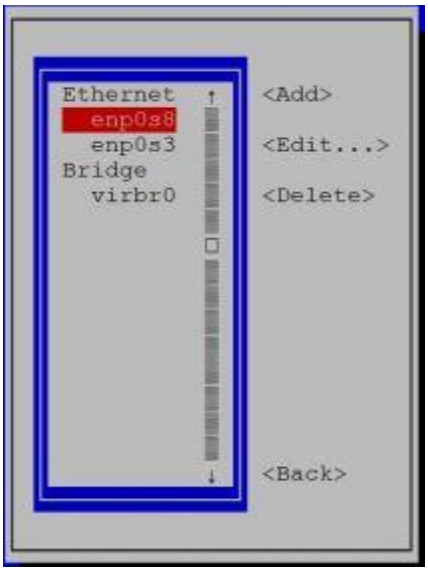
4.Edit dhcpd.conf file.

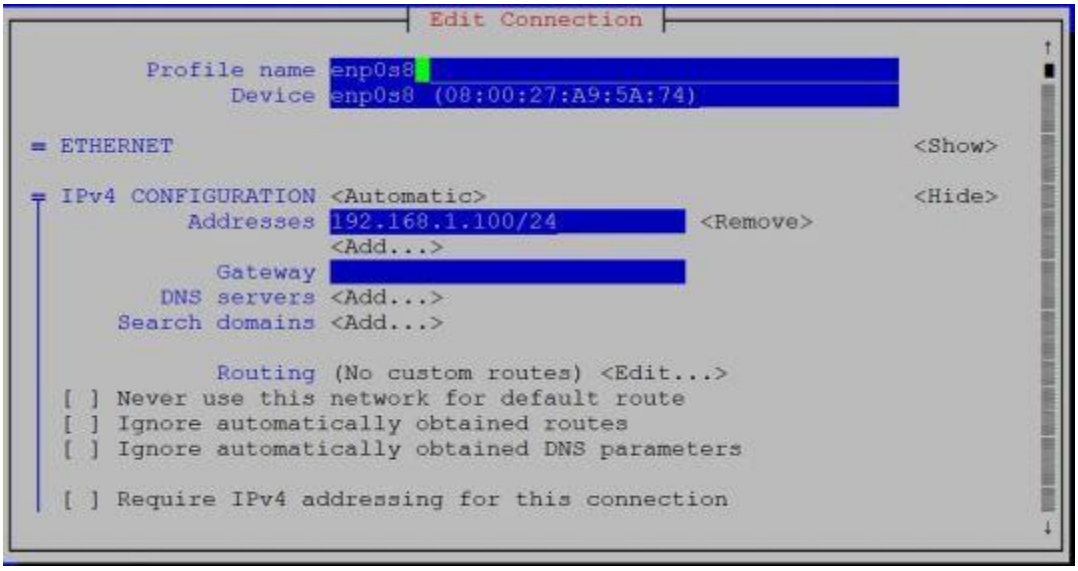
```
# nano /etc/dhcp/dhcpd.conf
option domain-name "example.com";
authoritative;
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.110 192.168.1.130;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
option domain-name-servers 8.8.8.8;
default-lease-time 600;
max-lease-time 7200;
}
```

4.1 config-network nmtui

```
# nmtui
```







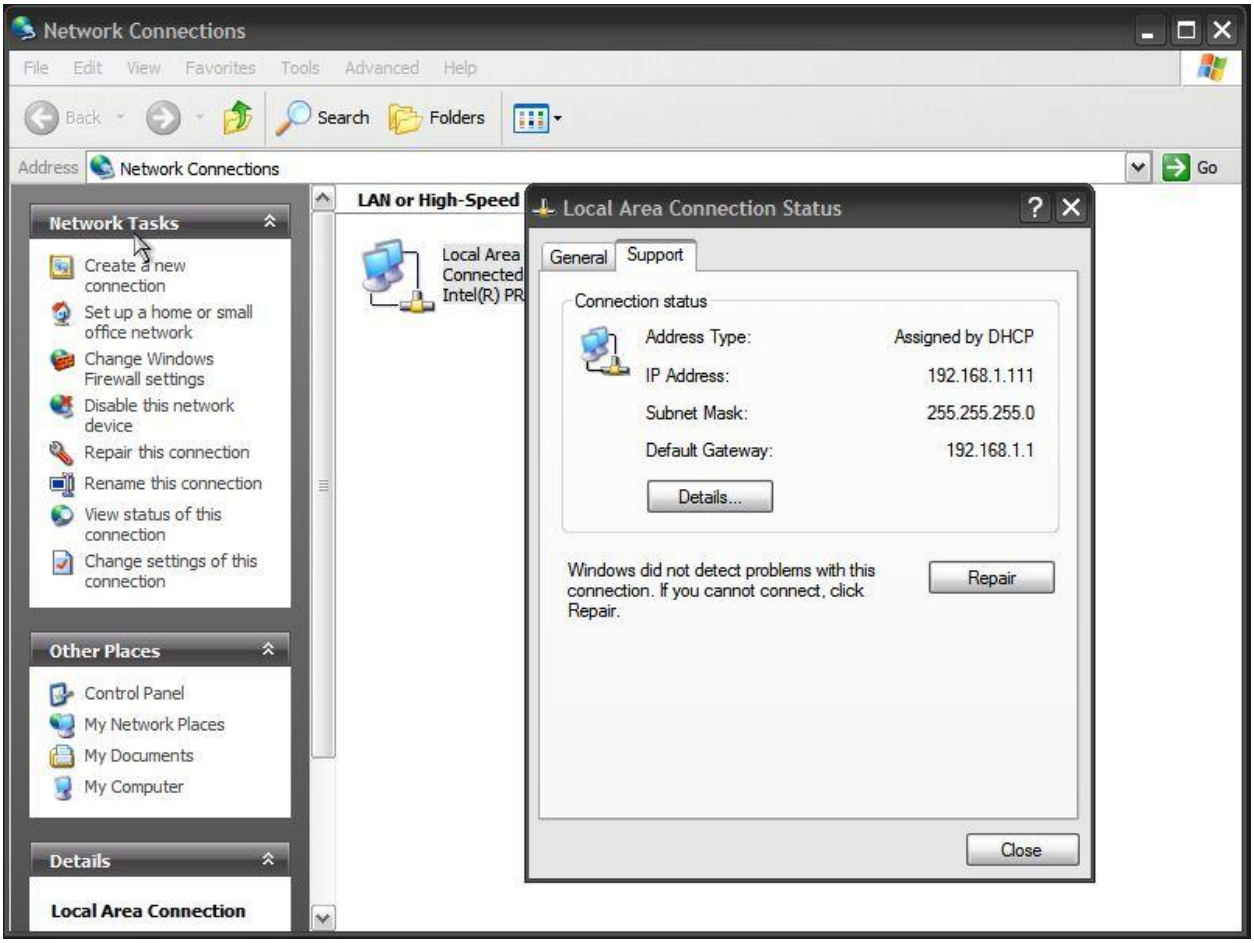
4.2 configure IP Address in CentOS 7 DHCP Mode Client (internal network)

```
cd /etc/sysconfig/network-scripts/

nano ifcfg-enp0s8

HWADDR=00:0C:29:76:96:A8

TYPE=Ethernet

BOOTPROTO=dhcp  ##Assigning IP from DHCP
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=eno16777736
UUID=e5a5d8e9-b5d4-4b5e-bd1e-6ebcd0144dfa
ONBOOT=yes   ## Interface enabled
```

5.Now, start the dhcpd service and make it start automatically on system reboot.

```
# service dhcpd start

  Redirecting to /bin/systemctl start dhcpd.service

# systemctl restart dhcpd.service

# service dhcpd status
```

```
Redirecting to /bin/systemctl status dhcpd.service
● dhcpd.service - DHCPv4 Server Daemon
   Loaded: loaded (/usr/lib/systemd/system/dhcpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2019-09-24 13:26:59 EDT; 14min ago
     Docs: man:dhcpd(8)
           man:dhcpd.conf(5)
 Main PID: 12485 (dhcpd)
   Status: "Dispatching packets..."
    Tasks: 1
   CGroup: /system.slice/dhcpd.service
           └─12485 /usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -group dhcpd --no-pid

Sep 24 13:40:16 localhost.localdomain dhcpd[12485]: DHCPREQUEST for 192.168.1.110 from 08:00:27:a9:
Sep 24 13:40:16 localhost.localdomain dhcpd[12485]: DHCPACK on 192.168.1.110 to 08:00:27:a9:5a:74 v
Sep 24 13:40:34 localhost.localdomain dhcpd[12485]: DHCPREQUEST for 192.168.1.110 from 08:00:27:a9:
Sep 24 13:40:34 localhost.localdomain dhcpd[12485]: DHCPACK on 192.168.1.110 to 08:00:27:a9:5a:74 v
Sep 24 13:40:45 localhost.localdomain dhcpd[12485]: DHCPREQUEST for 192.168.1.110 from 08:00:27:a9:
Sep 24 13:40:45 localhost.localdomain dhcpd[12485]: DHCPACK on 192.168.1.110 to 08:00:27:a9:5a:74 v
Sep 24 13:40:48 localhost.localdomain dhcpd[12485]: DHCPDISCOVER from 08:00:27:a9:5a:74 via enp0s8
Sep 24 13:40:48 localhost.localdomain dhcpd[12485]: DHCPOFFER on 192.168.1.110 to 08:00:27:a9:5a:74
Sep 24 13:40:48 localhost.localdomain dhcpd[12485]: DHCPREQUEST for 192.168.1.110 (192.168.1.110) f
Sep 24 13:40:48 localhost.localdomain dhcpd[12485]: DHCPACK on 192.168.1.110 to 08:00:27:a9:5a:74 v
```


Network Connections
File Edit View Favorites Tools Advanced Help
Back Search Folders
Address Network Connections
Go
Network Tasks
Create a new connection
Set up a home or small office network
Change Windows Firewall settings
Disable this network device
Repair this connection
Rename this connection
View status of this connection
Change settings of this connection
Other Places
Control Panel
My Network Places
My Documents
My Computer
Details
Local Area Connection
LAN or High-Speed
Local Area
Connected
Intel(R) PR
Local Area Connection Status
General
Support
Connection status
Address Type:
Assigned by DHCP
IP Address:
192.168.1.111
Subnet Mask:
255.255.255.0
Default Gateway:
192.168.1.1
Details...
Windows did not detect problems with this connection. If you cannot connect, click Repair.
Repair
Close

URL: https://www.unixmen.com/setting-dns-server-centos-7/

Setting Up DNS Server On CentOS 7

DNS stands for "**D**omain **N**ame **S**ystem", translates hostnames or URLs into IP addresses. For example, if we type **www.unixmen.com** in the browser, the DNS server translates the domain name into its associated IP address. Since the IP addresses are hard to remember all time, DNS servers are used to translate the hostnames like www.unixmen.com to 173.xxx.xx.xxx. So it makes it easy to remember the domain names instead of their IP address.

This detailed tutorial will help you to set up a local DNS server on your CentOS 7 system. However, the steps are applicable for setting up DNS server on RHEL and Scientific Linux 7 too.

DNS Server Installation

Scenario

For the purpose of this tutorial, I will be using three nodes. One will be acting as Master DNS server, the second system will be acting as Secondary DNS, and the third will be our DNS client. Here are my three systems details.

Primary (Master) DNS Server Details:

```
Operating System      : CentOS 7 minimal server
```

```
Hostname              : masterdns.unixmen.local

IP Address            : 192.168.1.101/24
```

Secondary (Slave) DNS Server Details:

```
Operating System      : CentOS 7 minimal server

Hostname              : secondarydns.unixmen.local

IP Address            : 192.168.1.102/24
```

Client Details:

```
Operating System      : CentOS 6.5 Desktop

Hostname              : client.unixmen.local

IP Address            : 192.168.1.103/24
```

Setup Primary (Master) DNS Server

Install bind9 packages on your server.

```
yum install bind bind-utils -y
```

1. Configure DNS Server

Edit **'/etc/named.conf'** file.

```
vi /etc/named.conf
```

Add the lines as shown in bold:

```
//

// named.conf
```

```
//

// Provided by Red Hat bind package to configure the ISC BIND
named(8) DNS

// server as a caching only nameserver (as a localhost DNS resolver
only).

//

// See /usr/share/doc/bind*/sample/ for example named configuration
files.

//



options {

    listen-on port 53 { 127.0.0.1; 192.168.1.101;}; ### Master DNS
IP ###

#    listen-on-v6 port 53 { ::1; };

    directory     "/var/named";

    dump-file     "/var/named/data/cache_dump.db";

    statistics-file "/var/named/data/named_stats.txt";

    memstatistics-file "/var/named/data/named_mem_stats.txt";

    allow-query     { localhost; 192.168.1.0/24;}; ### IP Range ###

    allow-transfer{ localhost; 192.168.1.102; };   ### Slave DNS IP
###



    /*
```

```
     - If you are building an AUTHORITATIVE DNS server, do NOT
enable recursion.

     - If you are building a RECURSIVE (caching) DNS server, you
need to enable

       recursion.

     - If your recursive DNS server has a public IP address, you
MUST enable access

       control to limit queries to your legitimate users. Failing
to do so will

       cause your server to become part of large scale DNS
amplification

       attacks. Implementing BCP38 within your network would
greatly

       reduce such attack surface

    */

    recursion yes;



    dnssec-enable yes;

    dnssec-validation yes;

    dnssec-lookaside auto;



    /* Path to ISC DLV key */

    bindkeys-file "/etc/named.iscdlv.key";
```

```
    managed-keys-directory "/var/named/dynamic";


    pid-file "/run/named/named.pid";
    session-keyfile "/run/named/session.key";
};


logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};


zone "." IN {
    type hint;
    file "named.ca";
};


zone "unixmen.local" IN {
type master;
file "forward.unixmen";
```

```
allow-update { none; };

};

zone "1.168.192.in-addr.arpa" IN {

type master;

file "reverse.unixmen";

allow-update { none; };

};



include "/etc/named.rfc1912.zones";

include "/etc/named.root.key";
```

2. Create Zone files

Create forward and reverse zone files which we mentioned in the **'/etc/named.conf'** file.

2.1 Create Forward Zone

Create **forward.unixmen** file in the **'/var/named'** directory.

```
vi /var/named/forward.unixmen
```

Add the following lines:

```
$TTL 86400

@   IN  SOA     masterdns.unixmen.local. root.unixmen.local. (

        2011071001  ;Serial

        3600        ;Refresh
```

```
        1800        ;Retry
        604800      ;Expire
        86400       ;Minimum TTL
)
@       IN  NS          masterdns.unixmen.local.
@       IN  NS          secondarydns.unixmen.local.
@       IN  A           192.168.1.101
@       IN  A           192.168.1.102
@       IN  A           192.168.1.103
masterdns       IN  A   192.168.1.101
secondarydns    IN  A   192.168.1.102
client          IN  A   192.168.1.103
```

2.2 Create Reverse Zone

Create **reverse.unixmen** file in the **'/var/named'** directory.

```
vi /var/named/reverse.unixmen
```

Add the following lines:

```
$TTL 86400
@   IN  SOA     masterdns.unixmen.local. root.unixmen.local. (
        2011071001  ;Serial
        3600        ;Refresh
        1800        ;Retry
```

```
        604800      ;Expire
        86400       ;Minimum TTL
)
@       IN  NS          masterdns.unixmen.local.
@       IN  NS          secondarydns.unixmen.local.
@       IN  PTR         unixmen.local.
masterdns       IN  A   192.168.1.101
secondarydns    IN  A   192.168.1.102
client          IN  A   192.168.1.103
101     IN  PTR         masterdns.unixmen.local.
102     IN  PTR         secondarydns.unixmen.local.
103     IN  PTR         client.unixmen.local.
```

3. Start the DNS service

Enable and start DNS service:

```
systemctl enable named
systemctl start named
```

4. Firewall Configuration

We must allow the DNS service default port 53 through firewall.

```
firewall-cmd --permanent --add-port=53/tcp
firewall-cmd --permanent --add-port=53/udp
```

5. Restart Firewall

```
firewall-cmd --reload
```

6. Configuring Permissions, Ownership, and SELinux

Run the following commands one by one:

```
chgrp named -R /var/named
```

```
chown -v root:named /etc/named.conf
```

```
restorecon -rv /var/named
```

```
restorecon /etc/named.conf
```

7. Test DNS Configuration and Zone Files for any Syntax Errors

Check DNS default configuration file:

```
named-checkconf /etc/named.conf
```

If it returns nothing, your configuration file is valid.

Check Forward zone:

```
named-checkzone unixmen.local /var/named/forward.unixmen
```

Sample output:

```
zone unixmen.local/IN: loaded serial 2011071001

OK
```

Check reverse zone:

```
named-checkzone unixmen.local /var/named/reverse.unixmen
```

Sample Output:

```
zone unixmen.local/IN: loaded serial 2011071001

OK
```

Add the DNS Server details in your network interface config file.

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

TYPE="Ethernet"

BOOTPROTO="none"

DEFROUTE="yes"

IPV4_FAILURE_FATAL="no"

IPV6INIT="yes"

IPV6_AUTOCONF="yes"

IPV6_DEFROUTE="yes"

IPV6_FAILURE_FATAL="no"

NAME="enp0s3"

UUID="5d0428b3-6af2-4f6b-9fe3-4250cd839efa"

ONBOOT="yes"

HWADDR="08:00:27:19:68:73"

IPADDR0="192.168.1.101"

PREFIX0="24"

GATEWAY0="192.168.1.1"

DNS="192.168.1.101"
```

```
IPV6_PEERDNS="yes"

IPV6_PEERROUTES="yes"
```

Edit file **/etc/resolv.conf**,

```
vi /etc/resolv.conf
```

Add the name server ip address:

```
nameserver      192.168.1.101
```

Save and close the file.

Restart network service:

```
systemctl restart network
```

8. Test DNS Server

```
dig masterdns.unixmen.local
```

Sample Output:

```
; <<>> DiG 9.9.4-RedHat-9.9.4-14.el7 <<>> masterdns.unixmen.local

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25179

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2,
ADDITIONAL: 2



;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:
```

```
;masterdns.unixmen.local.    IN    A


;; ANSWER SECTION:

masterdns.unixmen.local. 86400    IN    A    192.168.1.101


;; AUTHORITY SECTION:

unixmen.local.        86400    IN    NS    secondarydns.unixmen.loc
al.

unixmen.local.        86400    IN    NS    masterdns.unixmen.local.


;; ADDITIONAL SECTION:

secondarydns.unixmen.local. 86400 IN    A    192.168.1.102


;; Query time: 0 msec

;; SERVER: 192.168.1.101#53(192.168.1.101)

;; WHEN: Wed Aug 20 16:20:46 IST 2014

;; MSG SIZE  rcvd: 125

nslookup unixmen.local
```

Sample Output:

```
Server:        192.168.1.101

Address:    192.168.1.101#53
```

```
Name:    unixmen.local

Address: 192.168.1.103

Name:    unixmen.local

Address: 192.168.1.101

Name:    unixmen.local

Address: 192.168.1.102
```

Now the Primary DNS server is ready to use.