

Contents

FreeIPA Server and FreeIPA Client Setup (RHEL 8 / 9)	2
Prepare the Client and Server	2
Set Up the FreeIPA Server	3
Set up the FreeIPA Client	4
Uninstall an IPA client	6
Reinstalling the IPA client	7
Uninstall FreeIPA Server	7
Checking Services	7
Setting up DNS Server and Client(s)	8
Network Configuration Files	8
Setting up a DNS Server and Client(s) in RHEL 8 / Oracle Linux 8	9
Adding More DNS Records	11
Adding More A Records and Testing with ping	13
Setting up NTP Server and Client(s)	14
Setting up Samba File Server(s)	16
Setting up Web Server	18

FreeIPA Server and FreeIPA Client Setup (RHEL 8 / 9)

https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/8/html-single/installing_identity_management/index#uninstalling-an-ipa-server_installing-identity-management

FreeIPA is a free and open source identity management system, it is the upstream open-source project for Red Hat Identity Management.

FreeIPA is the Linux version or implementation of Active Directory, which features the following:

- Integrated security information management solution combining Linux (Fedora), 389 Directory Server, MIT Kerberos, NTP, DNS, SSSD and others.
- Built on top of well-known Open Source components and standard protocols.
- Strong focus on ease of management and automation of installation and configuration tasks.
- Full multi master replication for higher redundancy and scalability.
- Extensible management interfaces (CLI, Web UI, XMLRPC and JSONRPC API) and Python SDK.

This guide assumes:

- Two fully functional Linodes equal to a 2GB Plan or greater must be created using CentOS 7 or later. One will host the FreeIPA server, while the other will host the client.
- FreeIPA requires that the user has possession of their own fully qualified domain name (FQDN) with an active subdomain for both the client and server. Before proceeding, ensure that each Linode has A/AAAA records configured using a Unique Subdomain for both the server and client Linode respectively.
- Set up Reverse DNS for each Linode using their full unique subdomain.

Prepare the Client and Server

First, the FreeIPA Server and Client Linodes must be prepared for the installation. Follow the pre-installation steps below on both the Client and Server Linodes:

Set the hostname to match the domain you will be using for the FreeIPA server or client:

```
sudo hostname ipa.example.com
```

Edit the systems hosts file to reflect the new hostname.

File: /etc/hosts

```
127.0.0.1 localhost.localdomain localhost
203.0.113.10 server.example.com example-hostname
```

FreeIPA requires access to the following ports for the services listed below:

Ports	Service	Protocol
80, 443	HTTP/HTTPS	TCP
389, 636	LDAP/LDAPS	TCP
88, 464	Kerberos	TCP/UDP
53	DNS	TCP/UDP
123	NTP	UDP

All of the above ports can be opened using the commands

```
firewall-cmd --permanent --add-  
port={80/tcp,443/tcp,389/tcp,636/tcp,88/tcp,464/tcp,53/tcp,88/udp,464/udp,53/udp,123/udp}
```

Reload the firewall rules to save and activate them:

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

Note: (Oracle 8.10)

if freeipa-server is not found,

```
sudo dnf module list idm  
sudo dnf module enable idm:DL1 -y  
sudo dnf install freeipa-server  
dnf distro-sync
```

Set Up the FreeIPA Server

On the server Linode, install and set up the FreeIPA server with the following commands:

Download the FreeIPA server software:

```
dnf install ipa-server
```

Begin the installation process by entering the following command:

```
sudo ipa-server-install
```

Respond to the prompts with your desired FreeIPA configuration.

Prompt	Response
Do you want to configure integrated DNS (BIND)?	Bind can be set up to provide additional DNS support to the FreeIPA sever. Since the FreeIPA configuration being used as part of this tutorial is relying on external DNS using Linode's DNS Manager, no is the recommended choice.
Server host name	ipaserver.example.com
Please confirm the domain name.	example.com
Please provide a realm name.	When used with Kerberos, a Realm represents the domain that the server has authority over. The realm name should be the same as the primary domain being used for the FreeIPA server.
Directory Manager Password	Enter a secure Password of your choice for the Directory Manager. The Directory Manager is an administrative user with full access permissions to the directory server. The password must be at least 8 characters long.
IPA Admin Password	The password of the administrative user account for the IPA server.
Continue to configure the system with these values?	After reviewing your settings, enter yes to continue.

The installation process will begin, and you should see every step being outlined in your terminal. This process can take between 3-5 minutes to complete.

Once the installation is complete, you will be provided with instructions on how to create a Kerberos ticket for the admin user, allowing you to begin working with Kerberos. This can be done by running the following command:

```
kinit admin
```

Note

The admin ticket created with kinit admin is set to expire in 24 hours following ticket creation. To re-create another admin ticket, enter kinit admin again at any time.
At the prompt, enter the IPA Admin Password to proceed.

To view all active Kerberos tickets along with statistics, enter the following command:

```
klist
```

Create a new user by entering the following command and following the prompts that appear in the terminal:

```
ipa user-add --password
```

New users can be created with the above command at any time.

Note

Kerberos tickets associated with these users will expire similarly to admin users. Basic syntax for creating new kerberos tickets is kinit username.

Set up the FreeIPA Client

On the client Linode, install and set up the FreeIPA client with the following commands:

Download the FreeIPA client software:

```
sudo dnf install freeipa-client
```

Begin the installation process by entering the following command:

```
sudo ipa-client-install --mkhomedir
```

Respond to the prompts with your desired FreeIPA client configuration. Below are explanations on the configuration options and what options should be entered.

Prompt	Response
Provide the domain name of your IPA server (ex: example.com)	The primary domain used for the server installation.
Provide your IPA server name (ex: ipa.example.com).	The full domain used for the server installation including the subdomain.
If you proceed with the installation, services will be configured to always access the	

discovered server for all operations and will not fail over to other servers in case of failure.	
Proceed with fixed values and no DNS discovery?	This option is informing the user that the server is not configured with high availability, and it is safe to proceed by entering yes. More information on high availability on FreeIPA can be found in FreeIPA's Official Documentation
Client Hostname	The full domain, including the subdomain of the Client server currently being configured.
Continue to configure the system with these values?	The values for this freeIPA client installation appear in the terminal. Review these configuration options and enter yes to approve them and proceed with the installation.
User authorized to enroll computers.	Enter the username of a Kerberos user able to enroll new computers. The admin user may be entered.
Password for user@example.com.	The Password for the Kerberos user entered at the previous step. This will be the same password set on the FreeIPA server.

FreeIPA is now successfully installed as both a client and server. To confirm this, authenticate to the server as a user created previously by entering the following command to switch to your user:

```
su - username
```

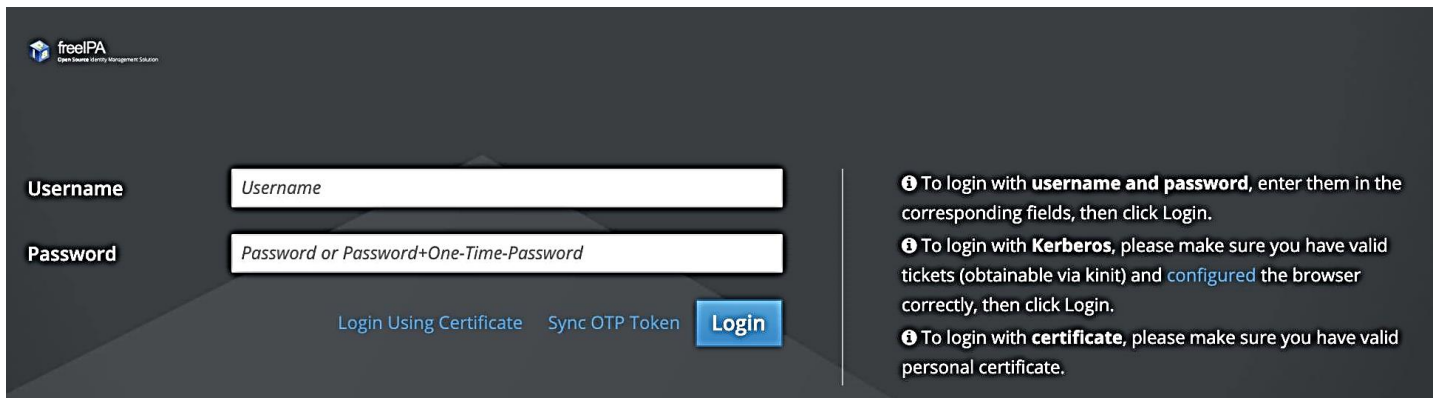
Provided a kerberos ticket has been created for the user using the kinit username command, you will additionally be able to authenticate to any client using kerberos tickets. Enter the following command to open an ssh session from the server Linode to the client to confirm:

```
ssh -k user@client.example.com
```

Kerberos tickets enable you to authenticate to any client using tickets instead of providing credentials. In this case, you will be able to log in to the FreeIPA client via SSH without providing any credentials, as the identity of both hosts has already been validated via FreeIPA and Kerberos.

Next Steps

The kerberos admin server will be freely accessible via it's domain in a web browser. Credentials created during installation can then be used to log in as the admin user via FreeIPA's web ui. Enter the admin server domain into your browser and you will see a page similar to the following:



Once logged in, you will have access to many of the tools and utilities available to FreeIPA from the command line directly on a more user friendly web interface.

Uninstall an IPA client

Run the `ipa-client-install --uninstall` command:

```
[root@client ~]# ipa-client-install --uninstall
```

Check that you cannot obtain a Kerberos ticket-granting ticket (TGT).

```
[root@client ~]# kinit admin
kinit: Client 'admin@EXAMPLE.COM' not found in Kerberos database while getting initial credentials
```

On the client, remove old Kerberos principals from each identified keytab other than `/etc/krb5.keytab`:

```
[root@client ~]# ipa-rmkeytab -k /path/to/keytab -r EXAMPLE.COM
```

On an IdM server, remove all DNS entries for the client host from IdM:

```
[root@server ~]# ipa dnsrecord-del
Record name: old-client-name
Zone name: idm.example.com
No option to delete specific record provided.
Delete all? Yes/No (default No): yes
-----
Deleted record "old-client-name"
```

On the IdM server, remove the client host entry from the IdM LDAP server. This removes all services and revokes all certificates issued for that host:

```
[root@server ~]# ipa host-del client.idm.example.com
```

Reinstalling the IPA client

Make sure the required ports are open on the server side.

Run the `ipa-client-install` command.

NOTE: To install the system with different values, run `ipa-client-install` and specify the required values by adding command-line options to `ipa-client-install`.

```
# ipa-client-install --force
```

NOTE: If the installation fails with the warning message below, it may be due to an empty or incorrect file that needs to be manually removed from the client side.

"WARNING Using existing certificate '/etc/ipa/ca.crt'

a). Remove the old `/etc/ipa/ca.crt` file, and try to reinstall.

```
# rm /etc/ipa/ca.crt
```

When the installation is finished, test if you can obtain a Kerberos ticket-granting ticket (TGT).

```
[root@client ~]# kinit -V admin
```

Uninstall FreeIPA Server

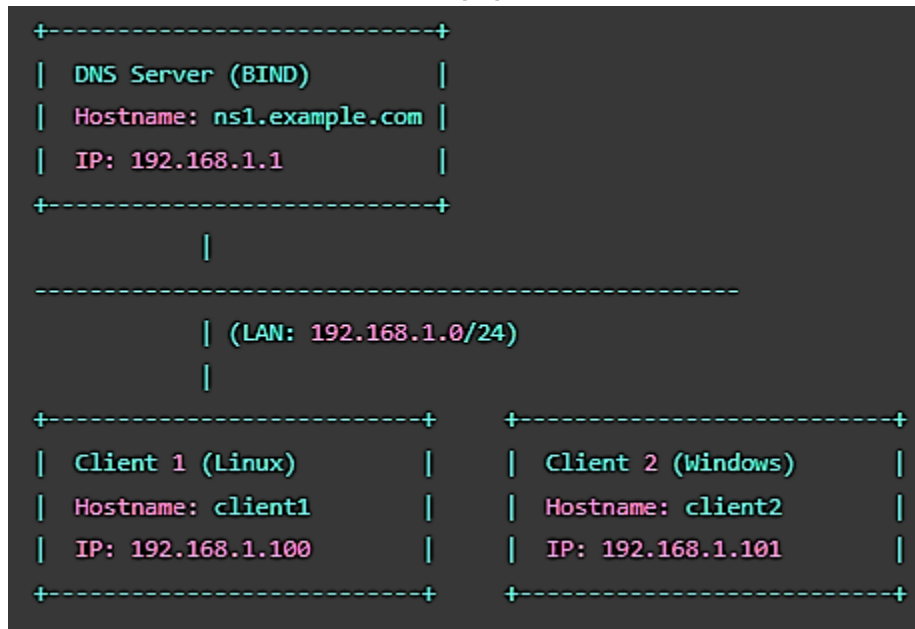
If you no longer need the FreeIPA server, you can uninstall it as shown.

```
$ sudo ipa-server-install --uninstall
```

Checking Services

Command	Description
<code>systemctl list-units --type=service --all</code>	Show all services
<code>systemctl list-units --type=service --state=running</code>	Show only running services
<code>systemctl status <service-name></code>	Check the status of a specific service
<code>systemctl list-unit-files --type=service</code>	List all installed services
<code>systemctl is-enabled <service-name></code>	Check if a service is enabled at boot
<code>ss -tulpn</code>	Show active ports and their services

Setting up DNS Server and Client(s)



Network Configuration Files

1. Configure Static IP on the DNS Server

Modify the network configuration file for your DNS Server (ns1.example.com).

Find your network interface name:

```
nmcli device status
```

*Assume it's ens192.

Edit the network configuration file:

```
sudo vi /etc/sysconfig/network-scripts/ifcfg-ens192
```

Set the following:

```
BOOTPROTO=static
DEVICE=ens192
ONBOOT=yes
IPADDR=192.168.1.1
NETMASK=255.255.255.0
GATEWAY=192.168.1.254
DNS1=127.0.0.1
```

Restart the network service:

```
sudo systemctl restart NetworkManager
```


2. Configure Static IP on Client Machines

For Linux Clients (Example: client1 - 192.168.1.100):

Edit network configuration:

```
sudo vi /etc/sysconfig/network-scripts/ifcfg-ens192
```

Set:

```
BOOTPROTO=static
DEVICE=ens192
ONBOOT=yes
IPADDR=192.168.1.100
NETMASK=255.255.255.0
GATEWAY=192.168.1.254
DNS1=192.168.1.1
```

Restart the network:

```
sudo systemctl restart NetworkManager
```

For Windows Clients (Example: client2 - 192.168.1.101):

- Go to Control Panel → Network and Sharing Center → Change adapter settings.
- Right-click your Ethernet/WiFi connection → Properties.
- Select Internet Protocol Version 4 (TCP/IPv4) → Properties.

Set:

```
IP Address: 192.168.1.101
Subnet Mask: 255.255.255.0
Default Gateway: 192.168.1.254
Preferred DNS Server: 192.168.1.1
```

Setting up a DNS Server and Client(s) in RHEL 8 / Oracle Linux 8

A DNS (Domain Name System) server resolves domain names to IP addresses. In this setup, we'll use BIND (Berkeley Internet Name Domain) to configure a DNS server.

Install BIND (DNS Server)

```
sudo dnf install -y bind bind-utils
```

Configure BIND

Edit the main configuration file:

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

Modify or add the following settings:

```
options {  
    listen-on port 53 { 127.0.0.1; 192.168.1.1; }; # Replace with your server IP  
    listen-on-v6 port 53 { none; };  
    directory "/var/named";  
    allow-query { localhost; 192.168.1.0/24; }; # Replace with your subnet  
    recursion yes;  
};  
  
zone "example.com" IN {  
    type master;  
    file "example.com.zone";  
};
```

Create a Forward Zone File

```
sudo vi /var/named/example.com.zone
```

Add the following:

```
$TTL 86400  
@ IN SOA ns1.example.com. root.example.com. (  
    2024022701 ; Serial  
    3600      ; Refresh  
    1800      ; Retry  
    604800    ; Expire  
    86400 )   ; Minimum TTL  
  
@   IN NS ns1.example.com.  
ns1 IN A  192.168.1.1  
www IN A  192.168.1.100
```

Set Permissions

```
sudo chown named:named /var/named/example.com.zone  
sudo chmod 640 /var/named/example.com.zone
```

Start and Enable BIND

```
sudo systemctl enable --now named  
sudo systemctl status named
```

Configure the DNS Client

On the client machine, edit:

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

Add:

```
nameserver 192.168.1.1
```

Test the DNS Server

On the client, use:

```
nslookup www.example.com
dig www.example.com
```

Adding More DNS Records

Edit the zone file:

```
sudo vi /var/named/example.com.zone
```

New Entries to Add

```
$TTL 86400
@ IN SOA ns1.example.com. root.example.com. (
    2024022702 ; Serial (increment when updating)
    3600      ; Refresh
    1800      ; Retry
    604800    ; Expire
    86400 )   ; Minimum TTL

; Nameservers
@ IN NS ns1.example.com.
ns1 IN A 192.168.1.1

; A Records (Host-to-IP mappings)
www IN A 192.168.1.100
mail IN A 192.168.1.101
ftp IN A 192.168.1.102
db IN A 192.168.1.103

; CNAME Records (Aliases)
web IN CNAME www.example.com.
files IN CNAME ftp.example.com.

; MX Records (Mail Exchange)
@ IN MX 10 mail.example.com.

; TXT Records (Useful for testing)
@ IN TXT "This is a test DNS server"
spf IN TXT "v=spf1 mx -all"

; PTR Record (For Reverse Lookup)
1.1.168.192.in-addr.arpa. IN PTR ns1.example.com.
```

Apply and Restart BIND

After updating the zone file, change ownership and restart BIND.

```
sudo chown named:named /var/named/example.com.zone
sudo chmod 640 /var/named/example.com.zone
sudo systemctl restart named
sudo systemctl status named
```

Testing the New DNS Records

On a client machine (Linux or Windows), run the following commands.

```
nslookup www.example.com
dig www.example.com
```

Expected output:

```
www.example.com. IN A 192.168.1.100
```

```
nslookup ftp.example.com
dig ftp.example.com
```

Expected output:

```
ftp.example.com. IN A 192.168.1.102
```

Test CNAME Records

```
nslookup web.example.com
dig web.example.com
```

Expected output:

```
web.example.com. IN CNAME www.example.com.
www.example.com. IN A 192.168.1.100
```

Test MX Records (Mail Server)

```
nslookup -type=MX example.com
dig example.com MX
```

Expected output:

```
example.com. IN MX 10 mail.example.com.
```

Test TXT Records

```
nslookup -type=TXT example.com
dig example.com TXT
```

Expected output:

```
example.com. IN TXT "This is a test DNS server"
```

Test PTR Records (Reverse DNS Lookup)

```
nslookup 192.168.1.1
```

Expected output:

```
1.1.168.192.in-addr.arpa. IN PTR ns1.example.com.
```

Adding More A Records and Testing with ping

To add additional hostnames to your DNS server, follow these steps:

Modify the Zone File

Edit the zone file on your DNS server:

```
sudo vi /var/named/example.com.zone
```

Add More A Records (Host-to-IP Mappings)

```
; Additional A Records
server1 IN A 192.168.1.104
server2 IN A 192.168.1.105
server3 IN A 192.168.1.106
storage IN A 192.168.1.107
backup IN A 192.168.1.108
proxy IN A 192.168.1.109
```

Increment the Serial Number

Find this line:

```
2024022702 ; Serial
```

Increment it (change to a new number, e.g., 2024022703).

Apply Changes and Restart BIND

Save the file and restart the named (BIND) service:

```
sudo systemctl restart named
sudo systemctl status named
```

Verify that there are no syntax errors:

```
sudo named-checkconf
sudo named-checkzone example.com /var/named/example.com.zone
```

*If there are no errors, your changes are active.

Test the New A Records

From a Client Machine

Check DNS Resolution for the New Hosts

```
nslookup server1.example.com
dig server1.example.com
```

Expected output:

```
server1.example.com. IN A 192.168.1.104
```

Repeat for the other hosts:

```
nslookup server2.example.com
dig server2.example.com
nslookup storage.example.com
dig storage.example.com
```

Test with ping

Now, check if the clients can ping the new DNS records:

```
ping server1.example.com
```

Expected output:

```
PING server1.example.com (192.168.1.104) 56(84) bytes of data.
64 bytes from 192.168.1.104: icmp_seq=1 ttl=64 time=0.5 ms
```

Try with other records:

```
ping server2.example.com
ping storage.example.com
ping proxy.example.com
```

Setting up NTP Server and Client(s)

Network Time Protocol (NTP) ensures accurate time synchronization across systems. In RHEL 8 / Oracle Linux 8, we use Chrony instead of the older ntpd service.

Install Chrony

On the server (e.g., 192.168.1.1), install Chrony:

```
sudo dnf install -y chrony
```

Configure Chrony as an NTP Server

Edit the Chrony configuration file:

```
sudo vi /etc/chrony.conf
```

Modify or add the following lines:

```
# Allow clients in the 192.168.1.0/24 network to sync time
allow 192.168.1.0/24

# Set the local system as the master time source
local stratum 10

# Enable logging for debugging (optional)
logdir /var/log/chrony
```

(Optional) Use Public NTP Servers for Accurate Time Sync Add/Uncomment:

```
server 0.pool.ntp.org iburst
server 1.pool.ntp.org iburst
server 2.pool.ntp.org iburst
server 3.pool.ntp.org iburst
```

Enable and Start the Chrony Service

```
sudo systemctl enable --now chronyd
sudo systemctl status chronyd
```

Verify NTP Server is Working

Check if Chrony is synchronized:

```
chronyc tracking
```

Check connected NTP sources:

```
chronyc sources -v
```

Expected output:

MS Name/IP address	Stratum	Poll	Reach	LastRx	Last sample
=====					
^* time.google.com	1	10	377	21	-141us[-341us] +/- 12ms

Configure NTP Clients

Install Chrony on Clients

On the client machines (192.168.1.100, 192.168.1.101, etc.), install Chrony:

```
sudo dnf install -y chrony
```

Configure the Clients to Use the NTP Server

Edit the Chrony config file:

```
sudo vi /etc/chrony.conf
```

Replace or add:

```
server 192.168.1.1 iburst
```

*(Remove or comment out any public NTP servers.)

Restart Chrony on Clients

```
sudo systemctl restart chronyd
sudo systemctl enable chronyd
```

Verify Time Sync on Clients

Check the synchronization status:

```
chronyc tracking
```

Check the NTP source:

```
chronyc sources -v
```

Expected output:

```
MS Name/IP address      Stratum Poll Reach LastRx Last sample
```

```
=====
```

```
^* 192.168.1.1          10 10 377 20 +4us[ +24us] +/- 1ms
```

Test time accuracy:

```
date
```

Setting up Samba File Server(s)

Samba is an open-source implementation of the SMB protocol that allows Linux to share files with Windows and Linux clients.

Install Samba Server

On the server (192.168.1.1), install Samba:

```
sudo dnf install -y samba samba-client samba-common
```

Enable and start the Samba service:

```
sudo systemctl enable --now smb nmb
sudo systemctl status smb nmb
```

Configure Samba Shares

Edit the Samba configuration file:

```
sudo vi /etc/samba/smb.conf
```

Example Configuration:

Add the following at the bottom of the file:

```
[public]
comment = Public Share
path = /srv/samba/public
browsable = yes
writable = yes
guest ok = yes
create mask = 0777
directory mask = 0777
force user = nobody

[secure]
comment = Secure Share
path = /srv/samba/secure
browsable = yes
```



```
writable = yes
valid users = @smbgroup
create mask = 0770
directory mask = 0770
```

Create the Shared Directories

```
sudo mkdir -p /srv/samba/public
sudo chmod 777 /srv/samba/public
sudo mkdir -p /srv/samba/secure
sudo chown root:smbgroup /srv/samba/secure
sudo chmod 770 /srv/samba/secure
```

Create Samba User Accounts

Create a user group for Samba:

```
sudo groupadd smbgroup
```

Create a Samba user and add it to the group:

```
sudo useradd -M -s /sbin/nologin smbuser
sudo passwd smbuser
sudo smbpasswd -a smbuser
sudo usermod -aG smbgroup smbuser
```

Restart Samba:

```
sudo systemctl restart smb nmb
```

Configure Firewall

Allow Samba traffic through the firewall:

```
sudo firewall-cmd --permanent --add-service=samba
sudo firewall-cmd --reload
```

Connect to the Samba Share

Linux Client (192.168.1.100)

List shares from the server:

```
smbclient -L //192.168.1.1 -U smbuser
```

Mount the Samba share:

```
sudo mkdir /mnt/smbshare
sudo mount -t cifs //192.168.1.1/secure /mnt/smbshare -o username=smbuser,password=yourpassword
```

Windows Client

- Open Run (Win + R).
- Type \\192.168.1.1 and press Enter.
- You should see Public and Secure shares.

Test File Transfer

On a client machine, create a test file:

```
echo "Hello Samba" > /mnt/smbshare/testfile.txt
```

Verify it on the server:

```
ls -l /srv/samba/secure/
```

Setting up Web Server

A web server allows you to host websites and serve HTTP/HTTPS requests. In this example, we'll use Apache (httpd) to set up a web server.

Install Apache Web Server

On your server (192.168.1.1), install Apache (httpd):

```
sudo dnf install -y httpd
```

Enable and start the Apache service:

```
sudo systemctl enable --now httpd
sudo systemctl status httpd
```

Configure Firewall for Web Traffic

Allow HTTP (port 80) and HTTPS (port 443) traffic through the firewall:

```
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --permanent --add-service=https
sudo firewall-cmd --reload
```

Check open ports:

```
sudo firewall-cmd --list-all
```

Configure Apache Virtual Hosts (Optional)

If you want to host multiple websites, create virtual hosts.

Edit the Apache configuration file:

```
sudo vi /etc/httpd/conf.d/example.com.conf
```

Add the following:

```
<VirtualHost *:80>
    ServerAdmin webmaster@example.com
    DocumentRoot "/var/www/html/example.com"
    ServerName example.com
    ServerAlias www.example.com

    <Directory "/var/www/html/example.com">
        AllowOverride All
```

```
Require all granted
</Directory>
```

```
ErrorLog "/var/log/httpd/example.com_error.log"
CustomLog "/var/log/httpd/example.com_access.log" combined
</VirtualHost>
```

Create a Website Directory and Test Page

Create a directory for the website:

```
sudo mkdir -p /var/www/html/example.com
sudo chown -R apache:apache /var/www/html/example.com
sudo chmod -R 755 /var/www/html/example.com
```

Create an index.html test page:

```
echo "<h1>Welcome to Example.com</h1>" | sudo tee /var/www/html/example.com/index.html
```

Restart Apache:

```
sudo systemctl restart httpd
```

Test the Web Server

From a Web Browser

On any client machine, open a browser and go to:

```
http://192.168.1.1
```

or

```
http://example.com
```

*You should see the "Welcome to Example.com" page.

From a Linux Client

Use curl:

```
curl http://192.168.1.1
```