Hospital Puerto Montt
Dr. Eduardo Schütz Schroeder

Documentación Sistemas Linux

Departamento Tecnologías de la Información y Comunicación

23 de Febrero 2018

Versión 0.9.12

Documento basado en los archivos presentes en https://github.com/rlienlaf/SistemasHPM y en la correspondiente Wiki del proyecto ubicada en https://github.com/rlienlaf/SistemasHPM/wiki

Índice

1. Configuración básica de Fedora

1.1. Instalación básica Fedora

1.1.1. NTP

- Instalar NTP y configurar firewall:

```
root$  dnf install ntp
root$  systemctl start ntpd
root$  systemctl enable ntpd
root$  firewall-cmd --permanent --add-service=ntp
root$  firewall-cmd --reload
```

- Agregar al archivo `/etc/ntp.conf` las siguientes lineas:

  - `/etc/ntp.conf`

```
server 3.cl.pool.ntp.org iburst
server 1.south-america.pool.ntp.org iburst
server 2.south-america.pool.ntp.org iburst
```

  Debería quedar algo como esto:

  - `/etc/ntp.conf`

```
driftfile /var/lib/ntp/ntp.drift

restrict default nomodify notrap nopeer noquery kod limited

restrict 127.0.0.1
restrict ::1

server 3.cl.pool.ntp.org iburst
server 1.south-america.pool.ntp.org iburst
server 2.south-america.pool.ntp.org iburst
```

- Reiniciar ntpd

```
root$  systemctl restart ntpd
```

1.1.2. Modificar los repositorios

En el directorio `/etc/yum.repos.d/`, modificar el archivo `fedora.repo`, en donde descomentamos la primera línea de `baseurl` y la cambiamos para que quede (ajustar dependiendo la versión de Fedora instalada):

- `/etc/yum.repos.d/fedora.repo`

```
baseurl=http://ftp.inf.utfsm.cl/fedora/linux/releases/27/Everything/x86_64/os/
```

Además, comentar la línea mirrorlist en caso de existir.
Para finalizar, tirar el comando:

```
root$  yum clean all
```

1.1.3. Actualizar

Finalmente instalamos unos paquetes básicos y actualizamos todos los paquetes del sistema:

```
root$  dnf -y install vim
root$  dnf update
```

2. Entorno Gráfico

2.1. Instalar Escritorio estilo Windows 10

Basado en la documentación de Gnome Layout Manager[1].

2.1.1. Instalación estilo Windows 10 en GNOME

Para instalar el entorno de Windows en Linux, primero necesitamos que el computador tenga instalado las siguientes dependecias (se necesita root para esto):

```
root$  dnf install unity-gtk-module-common zenity wget curl unzip
```

Después ejecutamos los siguientes comandos para instalarlo (esto se ejecuta como el usuario):

```
usuario$  wget https://raw.githubusercontent.com/\
          bill-mavromatis/gnome-layout-manager/\
          master/layoutmanager.sh
usuario$  chmod +x layoutmanager.sh
usuario$  ./layoutmanager.sh --windows
```

2.2. Crear accesos en escritorio

Basado en la documentación en Creando enlaces en GNOME[2].

2.2.1. Activar iconos en escritorio

Por defecto los iconos en el escritorio están desactivados, por lo que hay que ejecutar un comando para activarlos:

```
usuario$  gsettings set org.gnome.desktop.background show-desktop-icons true
```

2.2.2. Creando un Acceso directo

Para crear un icono, en la carpeta Escritorio (o Desktop) del usuario, creamos un archivo `nombre.desktop`. Por ejemplo, para crear un acceso a la Intranet podemos crear un archivo `intranet.desktop`, en donde escribimos lo siguiente:

- `intranet.desktop`

```
[Desktop Entry]
# Nombre que tendra el Acceso directo
Name=Intranet
# Ruta al icono que tendra el acceso directo
Icon=/usr/share/icons/hicolor/256x256/apps/firefox.png
Type=Application
# Comando que ejecutara el acceso directo
Exec=firefox "10.7.120.68/hpm_intro"
# Indica que no se necesita abrir una terminal para ejecutarlo
Terminal=false
```

2.2.3. Activar Acceso directo

Para finalmente activar el icono, es necesario abrir el `nombre.desktop` que aparecerá en el escritorio y darle permisos de confianza.

[1] https://github.com/vmavromatis/gnome-layout-manager
[2] http://diocesanos.es/blogs/equipotic/2014/10/02/creando-enlaces-en-el-escritorio-gnome-de-linux/

2.3. Cambiar color de letras

2.3.1. Cambiar color de letras en iconos escritorio

Para cambiar el color de las letras para un usuario, es necesario crear un archivo `gtk.css` en la carpeta `~/.config/gtk-3.0/`

```
usuario$  cd ~/.config/gtk-3.0/
usuario$  vim gtk.css
```

En donde escribimos lo siguiente:

- `/home/usuario/.config/gtk-3.0/gtk.css`

```
.nautilus-desktop.nautilus-canvas-item {
color: white;
}
```

Podemos reemplazar `white` por el color que queramos.
Es necesario reiniciar para aplicar los cambios.

2.4. Desactivar Lista de Usuarios logeados

Documentación basada en Login userlist disable[3].

2.4.1. Crear perfil para GNOME

Primero necesitamos crear un perfil para el daemon de GNOME Entramos a la carpeta `/etc/dconf/profile/` y creamos el archivo `gdm`:

```
root$  cd /etc/dconf/profile/
root$  vim gdm
```

En el archivo escribimos lo siguiente:

- `/etc/dconf/profile/gdm`

```
user-db:user
system-db:gdm
file-db:/usr/share/gdm/greeter-dconf-defaults
```

Luego, entramos a la carpeta `/etc/dconf/db/gdm.d/` y creamos el archivo `00-login-screen`:

```
root$  cd /etc/dconf/db/gdm.d/
root$  vim 00-login-screen
```

En donde escribimos:

- `/etc/dconf/db/gdm.d/00-login-screen`

```
[org/gnome/login-screen]
# Do not show the user list
disable-user-list=true
```

Finalmente, cargamos la configuración:

```
root$  dconf update
```

[3] https://help.gnome.org/admin/system-admin-guide/stable/login-userlist-disable.html.en

3. Cloud y Docker

3.1. Instalación Docker y Docker compose

3.1.1. Docker

- **Instalación**

  Primero hay que configurar los repos:

```
root$  dnf -y install dnf-plugins-core
root$  dnf config-manager --add-repo https://download.docker.com/linux/fedora/docker-ce.repo
```

  Luego, instalamos:

```
root$  dnf install docker-ce
```

  Finalmente iniciamos y habilitamos Docker:

```
root$  systemctl start docker
root$  systemctl enable docker
```

- **Prueba**

  Para probar si Docker se instaló correctamente, podemos ejecutar el comando:

```
root$  docker run hello-world
```

3.1.2. Docker-compose

- **Instalación**
  Ejecutamos los siguientes comandos:

```
root$  curl -L https://github.com/docker/compose/releases/\
   download/1.19.0/docker-compose-`uname -s`-`uname -m` \
   -o /usr/local/bin/docker-compose
root$  chmod +x /usr/local/bin/docker-compose
```

- **Prueba**
  Para testear que docker-compose se instaló correctamente, podemos ejecutar:

```
root$  docker-compose --version
```

3.2. Instalación Nextcloud y Onlyoffice

3.2.1. Git

Primeramente necesitamos instalar Git:

```
root$  dnf install git
```

3.2.2. Descargamos el repositorio

Bajamos el repositorio con Git:

```
root$  git clone --recursive https://github.com/ONLYOFFICE/\
   docker-onlyoffice-owncloud
root$  cd docker-onlyoffice-owncloud
root$  git submodule update --remote
```

3.2.3. Configuración

El repositorio originalmente instala Owncloud, pero podemos cambiar esto modificando el archivo `docker-compose.yml`, cambiamos la línea:

```
      image: owncloud:fpm
```

por la línea:

```
      image: nextcloud:fpm
```

Además, por defecto los puertos para acceder a Nextcloud serán 80 (http) y 443 (https). De ser necesario podemos cambiar esto al modificar las líneas:

```
    ports:
      - 80:80
      - 443:443
```

por: (para este caso al server se accederá por el puerto 10080 y 10443)

```
    ports:
      - 10080:80
      - 10443:443
```

3.2.4. Iniciar Containers

Iniciamos los containers de Docker. Cada vez que queramos levantar los containers ejecutamos:

```
root$  docker-compose up --build -d
```

3.2.5. Creando el administrador

Una vez creados todos los containers, accedemos a través del navegador al servidor: `<ip-del-servidor>:10080` (el puerto depende si se cambió en el paso anterior o no). Luego, seguimos los pasos en la página web para crear un administrador.

3.2.6. Configurando los servidores

Una vez creado el administrador, volvemos a la terminal, y dentro de la carpeta donde nos encontrábamos ejecutamos el siguiente comando:

```
root$  ./set_configuration.sh
```

4. Automatización

4.1. Automatización con Ansible

4.1.1. Instalación

Primeramente, instalamos Ansible en el servidor que sera el Master:

```
root$  dnf install ansible
```

4.1.2. Configurar llave SSH

Para mandar ordenes a los equipos esclavos, es necesario crear un par de llave SSH para que Ansible se autentifique.
Para crear un par de llaves SSH ejecutamos el comando:

```
root$  ssh-keygen -t rsa
```

Al ejecutar el comando pedirá que ingresemos unos datos, por facilidad de uso es mejor dejar estos tres campos vacíos, así que con pulsar 'enter' tres veces bastará para completar el comando.
Al finalizar el comando aparecerá una imagen con extraños caracteres, esto significa que terminó correctamente.

4.1.3. Deshabilitar chequeo de ssh-host-key

Basado en la documentación en Disable SSH host key checking.
Modificamos el archivo `/etc/ssh/ssh_config`, y agregamos las siguientes líneas al inicio del archivo:

- `/etc/ssh/ssh_config`

```
Host 10.7.*.*
   StrictHostKeyChecking no
   UserKnownHostsFile=/dev/null
```

4.1.4. Configurar un equipo automáticamente

Para agregar un equipo a la lista de esclavos de Ansible, solo basta con descargar el script `agregar_equipo.sh` desde la carpeta `ansible` en el repositorio SistemasHPM:

```
root$  wget https://raw.githubusercontent.com/rlienlaf/SistemasHPM/master/ansible/agregar_equipo.sh
root$  chmod +x agregar_equipo.sh
```

Luego, basta con ejecutarlo:

```
root$  sh agregar_equipo.sh
```

Este script nos pedirá la **IP del equipo a agregar**. Enviará nuestra configuración hacia el equipo dado y agregará el equipo a la configuración de Ansible.

4.1.5. Configurando manualmente

En caso de ocurrir algún problema, se puede utilizar el script nuevamente para agregar el equipo a las configuraciones necesarias.
El archivo en donde están listados los equipos esclavos es `/etc/ansible/hosts`.
La estructura del archivo `/etc/ansible/hosts` es de la siguiente forma:

```
 ...
[nombre_grupo]
ip_server_1
ip_server_2
ip_server_3
 ...
```

Entre corchetes "[ ]"va el nombre del grupo al que pertenecerán los equipos listados a continuación, ejemplo:

```
...

[carlitos]
10.7.164.22
10.7.164.145

[H1_farmacia]
10.7.164.188
10.7.164.190
10.7.164.189

...
```

Para este ejemplo los equipos `10.7.164.22` y `10.7.164.145` pertenecen al grupo `carlitos`, mientras que `10.7.164.188`, `10.7.164.189` y `10.7.164.190` pertenecen al grupo `H1_farmacia`

4.1.6. Uso de Ansible

Ansible tiene tres formas de utilizarse para ejecutar comandos a uno o múltiples equipos.

1. La primera forma es utilizar el comando `ansible` para mandar un comando a un equipo, un grupo o a todos los equipos:

```
$  ansible {objetivo} -a "{comando}"
```

Ejemplo:

```
root$  ansible 10.7.164.190 -a "mkdir prueba"
root$  ansible H1_farmacia -a "mkdir prueba"
root$  ansible all -a "mkdir prueba"
```

El primero comando creará un directorio `prueba/` en la IP `10.7.164.190`, el segundo lo hará en todos los equipos listados en el grupo `H1_farmacia`, y el tercero lo hará en todos los equipos agregados en Ansible.

2. La segunda forma es utilizar el comando `ansible` para realizar comandos sobre un equipo, un grupo o a todos los equipos:

```
$  ansible {objetivo} -m {comando}
```

Ejemplo:

```
root$  ansible 10.7.164.190 -m ping
root$  ansible H1_farmacia -m ping
root$  ansible all -m ping
```

El primero comando realizará un ping al equipo con IP `10.7.164.190`, el segundo a todos los equipos listados en el grupor `H1_farmacia` y el tercero hará ping a todos los equipos agregados en Ansible.

Esta forma de utilizar el comando Ansible está basada en módulos (de ahí el `-m` en el comando), existe una gran variedad de módulos implementados en Ansible, es cosa de buscar en Google si alguno en particular está implementado.

De hecho, existe un módulo que nos permite simular la primera forma en de usar Ansible mencionada más arriba:

```
$  ansible {objetivo} -m shell -a "{comando}"
```

Ejemplo:

```
root$  ansible all -m shell -a "yum update"
```

Este comando tirará `yum update` (actualizar todos los paquetes instalados) en todos los equipos agregados a Ansible.

3. La tercera forma es utilizar los llamados “Playbooks” en Ansible

   Los Playbooks son un conjunto de órdenes con cierta estructura que permiten relizar diversas y complejas tareas hacia los equipos. El rango de tareas va desde actualizar los equipos hasta instalar y configurar distintas bases de datos por equipo.

   Una amplia documentación sobre cómo escribir Playbooks se puede encontrar en Documentación Playbooks[4].

   Un ejemplo de un Playbook es el siguiente:

   - `actualizar_equipos.yml`

```
- hosts: carlitos
  tasks:
  - name: Actualiza todos los paquetes ("*") a su version mas nueva (latest)
    yum:
       name="*"
       state=latest
```

   Este Playbook lo que hace es ejecutar `yum update` en todos los equipos del grupo 'carlitos' (indicado en la línea de hosts).

   Para correr un Playbook, solo es necesario ejecutar el comando:

```
root$  ansible-playbook {nombre_del_playbook.yml}
```

   Para el caso del Playbook de ejemplo de arriba, el comando a ejecutar sería:

```
root$  ansible-playbook actualizar_equipos.yml
```

[4] http://docs.ansible.com/ansible/latest/playbooks.html

5. Autenticación y montado de cuentas

Basado en la documentación de LDAP and NFS Server[5].

LDAP es el servicio para autenticar usuarios, mientras que NFS es el servicio para transferir archivos. Por lo mismo, un equipo cualquiera (cliente) debe tener ambos servicios instalados, mientras que para el Servidor se pueden usar uno para LDAP server y otro para NFS server, sin embargo, es recomendable que ambos servidores sean el mismo.

Importante: Recordar que cualquier servidor que no sea el dedicado a LDAP y NFS se considera como cliente. Por ejemplo: para el servidor de la nube, se debe seguir la documentación de LDAP client y NFS client, y no la de servidor.

5.1. LDAP Server

5.1.1. Instalando

Instalamos las dependencias:

```
root$  dnf -y install openldap-servers openldap-clients
```

Luego, ejecutamos los siguientes comandos para iniciar una configuración default:

```
root$  cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
root$  chown ldap. /var/lib/ldap/DB_CONFIG
```

Iniciamos y habilitamos el servicio:

```
root$  systemctl start slapd
root$  systemctl enable slapd
```

5.1.2. Creando una política para SELinux

Instalamos:

```
root$  dnf -y install checkpolicy policycoreutils-python-utils
```

Creamos un archivo `slapd.te` y escribimos:

- `slapd.te`

```
module slapd 1.0;

require {
        type slapd_t;
        type slapd_db_t;
        class file map;
}

#============= slapd_t ==============
allow slapd_t slapd_db_t:file map;
```

Activamos la configuración:

```
root$  checkmodule -m -M -o slapd.mod slapd.te
root$  semodule_package --outfile slapd.pp --module slapd.mod
root$  semodule -i slapd.pp
```

5.1.3. Configurando el admin de LDAP

Ejecutamos el comando:

```
root$  slappasswd
```

Aquí podemos ingresar una contraseña y nos la devolverá encriptada, de la forma `{SSHA}xxxxxxxxxxxxxxxxxxx`, copiamos esta contraseña encriptada.

Creamos el archivo (reemplazar la clave donde sea necesario):

[5] https://www.server-world.info/en/note?os=Fedora_27&p=openldap

- `chrootpw.ldif`

```
dn: olcDatabase={0}config,cn=config
changetype: modify
add: olcRootPW
# Aqui va la contrasena encriptada que obtuvimos anteriormente:
olcRootPW: {SSHA}xxxxxxxxxxxxxxxxxxxxxxxx
```

Cargamos la configuración recién hecha:

```
root$  ldapadd -Y EXTERNAL -H ldapi:/// -f chrootpw.ldif
```

Luego, importamos algunos esquemas para nuestro servicio de LDAP:

```
root$  ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/cosine.ldif
root$  ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/nis.ldif
root$  ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/inetorgperson.ldif
```

5.1.4. Configurando la base de datos de LDAP

Generamos una clave para la base de datos (podemos usar la misma que para el admin):

```
root$  slappasswd
```

Recibimos una clave encriptada nuevamente, de la forma: {SSHA}xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.
Creamos un archivo `chdomain.ldif`, y escribimos (reemplazar la clave donde sea necesario):

- `chdomain.ldif`

```
dn: olcDatabase={1}monitor,cn=config
changetype: modify
replace: olcAccess
olcAccess: {0}to * by dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth"
  read by dn.base="cn=Manager,dc=hpm,dc=cl" read by * none

dn: olcDatabase={2}mdb,cn=config
changetype: modify
replace: olcSuffix
olcSuffix: dc=hpm,dc=cl

dn: olcDatabase={2}mdb,cn=config
changetype: modify
replace: olcRootDN
olcRootDN: cn=Manager,dc=hpm,dc=cl

dn: olcDatabase={2}mdb,cn=config
changetype: modify
add: olcRootPW
# Aqui agregamos la clave encriptada:
olcRootPW: {SSHA}xxxxxxxxxxxxxxxxxxxxxxxx

dn: olcDatabase={2}mdb,cn=config
changetype: modify
add: olcAccess
olcAccess: {0}to attrs=userPassword,shadowLastChange by
  dn="cn=Manager,dc=hpm,dc=cl" write by anonymous auth by self write by * none
olcAccess: {1}to dn.base="" by * read
olcAccess: {2}to * by dn="cn=Manager,dc=hpm,dc=cl" write by * read
```

Cargamos la configuración:

```
root$  ldapmodify -Y EXTERNAL -H ldapi:/// -f chdomain.ldif
```

5.1.5. Configurando el dominio

Creamos el archivo `basedomain.ldif` y escribimos:

- `basedomain.ldif`

```
dn: dc=hpm,dc=cl
objectClass: top
objectClass: dcObject
objectclass: organization
o: Hospital Puerto Montt
dc: hpm

dn: cn=Manager,dc=hpm,dc=cl
objectClass: organizationalRole
cn: Manager
description: Directory Manager

dn: ou=People,dc=hpm,dc=cl
objectClass: organizationalUnit
ou: People

dn: ou=Group,dc=hpm,dc=cl
objectClass: organizationalUnit
ou: Group
```

Cargamos nuestra configuración:

```
root$  ldapadd -x -D cn=Manager,dc=srv,dc=world -W -f basedomain.ldif
```

5.1.6. Creando certificados SSL para LDAP

Entramos a la carpeta `/etc/pki/tls/certs`

```
root$  cd /etc/pki/tls/certs
```

Ejecutamos el siguiente comando para generar una llave, el cual nos pedirá ingresar una contraseña:

```
root$  make server.key
```

Para que no nos pidan esta contraseña cada vez que ingresemos, ejecutamos el siguiente comando:

```
root$  openssl rsa -in server.key -out server.key
```

Luego, generamos un certificado con la llave que creamos:

```
root$  make server.csr
```

Este comando nos pedirá ingresar unos datos, los cuales en orden son:

```
Country Name (2 letter code) [XX]:      # El codigo del pais, en nuestro caso: CL
State or Province Name (full name) []:      # La region, en nuestro caso: Los Lagos
Locality Name (eg, city) [Default City]:       # La ciudad, en nuestro caso: Puerto Montt
Organization Name (eg, company) [Default Company Ltd]:        # La empresa, en nuestro caso:
                                                              #Hospital Puerto Montt
Organizational Unit Name (eg, section) []:      # El departamento, en nuestro caso: TI
Common Name (eg, your name or your server's hostname) []:         # El nombre del servidor,
                                                                  # en este caso: ldap.hpm.cl
Email Address []:         # El email del administrador, puede ser: cmoncada@ssdr.gob.cl

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:     # Una contrasena
An optional company name []:     # Un nombre opcional, puede ser: HPM
```

Finalmente, firmamos nuestro certificado ejecutando el comando:

```
root$  openssl x509 -in server.csr -out server.crt -req -signkey server.key -days 3650
```

5.1.7. Configurando LDAP con los certificados

Copiamos los certificados que recién hicimos:

```
root$  cp /etc/pki/tls/certs/server.key /etc/pki/tls/certs/server.crt \
          /etc/pki/tls/certs/ca-bundle.crt /etc/openldap/certs/
```

Les asignamos los permisos correspondientes:

```
root$  chown ldap. /etc/openldap/certs/server.key /etc/openldap/certs/server.crt \
                   /etc/openldap/certs/ca-bundle.crt
```

Creamos el archivo `mod_ssl.ldif`, en donde escribimos:

- `mod_ssl.ldif`

```
dn: cn=config
changetype: modify
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/openldap/certs/ca-bundle.crt
-
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/openldap/certs/server.crt
-
replace: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/openldap/certs/server.key
```

Cargamos nuestra configuración:

```
root$  ldapmodify -Y EXTERNAL -H ldapi:/// -f mod_ssl.ldif
```

Finalmente, reiniciamos el servicio para cargar todo y agregamos los servicios al firewall:

```
root$  systemctl restart slapd
root$  firewall-cmd --add-service={ldap,ldaps} --permanent
root$  firewall-cmd --reload
```

5.2. LDAP Client

Instalar dependecias:

```
root$  dnf -y install openldap-clients sssd sssd-ldap
```

Luego nos sincronizamos con el servidor:

```
root$  authconfig --enableldap --enableldaptls \
       --enableldapauth --ldapserver=10.7.164.145 \
       --ldapbasedn="dc=hpm,dc=cl" --enablemkhomedir --update
```

Importante: `10.7.164.145` es la IP del servidor en donde se configuró LDAP, cambiar dicha IP en el comando a la IP del servidor de LDAP que se esté utilizando.

Luego, habilitamos las conexiones externas, para ello editamos el archivo `/etc/sssd/sssd.conf`, en donde buscamos la sección `[domain/default]` y le agregamos la línea:

```
ldap_tls_reqcert = allow
```

Debería quedar algo parecido a:

- `/etc/sssd/sssd.conf`

```
[domain/default]

id_provider = ldap
autofs_provider = ldap
auth_provider = ldap
chpass_provider = ldap
ldap_uri = ldap://10.7.164.145/
ldap_search_base = dc=hpm,dc=cl
ldap_id_use_start_tls = True
ldap_tls_cacertdir = /etc/openldap/certs
cache_credentials = True
ldap_tls_reqcert = allow
[sssd]
  ...
```

Importante: `10.7.164.145` es la IP del servidor en donde se configuró LDAP, cambiar dicha IP en el archivo a la IP del servidor de LDAP que se esté utilizando.

Guardamos el archivo y reiniciamos el servicio sssd para cargar los cambios:

```
root$  systemctl restart sssd
```

Además modificamos el archivo (en caso de existir) `/etc/nsswitch.conf`, en donde cambiamos la línea `automount` a:

- `/etc/nsswitch.conf`

```
automount:  ldap files
```

5.3. NFS Server

Instalamos los paquetes:

```
root$  dnf -y install nfs-utils
```

Luego, modificamos el archivo `/etc/exports`, en donde escribimos:

- `/etc/exports`

```
/home 10.7.0.0/16(rw,sync,no_root_squash)
```

Después, iniciamos y habilitamos los servicios:

```
root$  systemctl start rpcbind autofs
root$  systemctl enable rpcbind autofs
```

Finalmente, agregamos el servicio al firewall:

```
root$  firewall-cmd --add-service=nfs --permanent
root$  firewall-cmd --reload
```

5.4. NFS Client

Instalar dependencias:

```
root$  dnf -y install nfs-utils autofs
```

Reiniciamos y habilitamos el servicio:

```
root$  systemctl restart rpcbind
root$  systemctl enable rpcbind
```

Luego, modificamos el archivo `/etc/auto.master`, en donde agregamos al final la siguiente línea (y dejamos una línea vacía al final):

- `/etc/auto.master`

```
/home /etc/auto.home --timeout=300
```

Es **importante** que el archivo `/etc/auto.master` tenga al final una línea vacía.
Luego, creamos el archivo `/etc/auto.home`, en donde escribimos la línea:

- `/etc/auto.home`

```
* -fstype=nfs,rw,nosuid,soft   10.7.164.145:/home/&
```

Importante: `10.7.164.145` es la IP del servidor en donde se configuró NFS, cambiar dicha IP en la línea a la IP del servidor de NFS que se esté utilizando.
Finalmente, recargamos el servicio de montado `autofs`:

```
root$  systemctl restart autofs
```

5.5. Crear Usuarios

Para crear usuarios primeramente es necesario descargar los siguientes archivos desde la carpeta `ldap/` en el repositorio SistemasHPM: `configurar_escritorio_usuarios.sh`, `crear_usuario.sh`, `cambiar_password.sh`, `ldapuser.ldif` y `user_id.conf` en el servidor de LDAP-NFS. Además es necesario descargar toda la carpeta `skel/` desde el mismo repositorio:

```
root$  git clone https://github.com/rlienlaf/SistemasHPM.git
root$  rm -rf django/ nginx/
root$  cd ldap/
```

5.5.1. Configurar escritorio usuarios

Primeramente, es necesario que se configuren los directorios para los usuarios nuevos, para ello utilizaremos el script `configurar_escritorio_usuarios.sh`:

```
root$  sh configurar_escritorio_usuarios.sh
```

5.5.2. Crear nuevos usuarios

Dentro de la carpeta `SistemasHPM/ldap/` que descargamos arriba en la sección 5.5. ejecutamos el script `crear_usuario.sh`:

```
root$  sh crear_usuario.sh
```

Al ejecutar el script nos pedirá un **nombre de usuario** y una **contraseña** para el nuevo usuario.

5.5.3. Cambiar contraseña usuario

Dentro de la carpeta `SistemasHPM/ldap/` que descargamos en la sección 5.5. ejecutamos el script `cambiar_password.sh`:

```
root$  sh cambiar_password.sh
```

Al ejecutar el script nos pedirá un **nombre de usuario existente** y una **nueva contraseña** para el usuario.

6. Problemas Generales

Guía de soluciones a problemas comunes.

6.1. Errores en usuarios

6.1.1. Errores al conectarse vía SSH

A veces puede haber problemas con el servicio que permite las conexiones SSH, para lo mismo hay un par de cosas que podemos revisar:

1. Primero podemos revisar si esta instalado:

```
root$  dnf install openssh openssh-server
```

2. Segundo podemos revisar si el servicio esta corriendo:

```
root$  systemctl restart sshd
root$  systemctl enable sshd
```

3. Tercero, podemos revisar si falta alguna regla en el firewall:

```
root$  firewall-cmd --add-service=ssh --permanent
root$  firewall-cmd --reload
```

6.1.2. Errores cargando escritorio gráfico

Puede ser un problema con el servicio que muestra los gráficos, el cual podemos reiniciar ejecutando el siguiente comando:

```
root$  systemctl restart gdm
```

6.1.3. Errores montando cuentas

Hay cuatro tipos de errores que pueden pasar si es que un equipo no monta cuentas:

1. **Error de red**

   Ya que el montado de cuentas y autenticación es por red, puede que haya algún problema con la conexión o configuración de la red.

   También se puede probar a reiniciar el servicio de red, ejecutando el comando:

```
root$  systemctl restart NetworkManager
```

2. **Error de configuración**

   Puede que alguna configuración este mala, para arreglarlo hay que revisar la documentación de NFS Client y LDAP Cliente y revisar que las configuraciones hechas coincidan.

3. **Error de servicios**

   Es un error bastante común, puede que algún servicio haya fallado, para solucionarlo se pueden reiniciar todos los servicios correspondientes al montado de cuentas. Ejecutamos:

```
root$  systemctl restart autofs rpcbind sssd NetworkManager
```

4. **Error de SELinux**

   Es un error bastante común, para probarlo se puede ejecutar el comando:

```
root$  setenforce 0
```

   Esto desactiva SELinux hasta que se reinicie el equipo, por mientras se puede ocupar normalmente. Si dicho comando arregló el problema es recomendable revisar la documentación de la Sección 6.3 para solucionar correctamente el problema.

6.2. Errores en Servidores

6.2.1. Servidor LDAP y NFS

1. **Error de red**

   Ya que el montado de cuentas y autenticación es por red, puede que haya algún problema con la conexión o configuración de la red.

   También se puede probar a reiniciar el servicio de red, ejecutando el comando:

```
root$  systemctl restart NetworkManager
```

2. **Error de configuración**

   Puede que alguna configuración este mala, para arreglarlo hay que revisar la documentación de NFS Client y LDAP Cliente y revisar que las configuraciones hechas coincidan.

3. **Error de servicios**

   Es un error bastante común, puede que algún servicio haya fallado, para solucionarlo se pueden reiniciar todos los servicios correspondientes al montado de cuentas. Ejecutamos:

```
root$  systemctl restart rpcbind sssd slapd nfs NetworkManager nfs-server rpc-statd nfs-idmapd
```

4. **Error de SELinux**

   Es un error bastante común, para probarlo se puede ejecutar el comando:

```
root$  setenforce 0
```

   Esto desactiva SELinux hasta que se reinicie el equipo, por mientras se puede ocupar normalmente. Si dicho comando arregló el problema es recomendable revisar la documentación de la Sección 6.3 para solucionar correctamente el problema.

6.3. Arreglando problemas de SELinux

Es bastante común que ocurran problemas con el servicio de seguridad SELinux, pero dichos problemas suelen ser bastante fáciles de resolver, aunque pueden requerir un par de pasos.

6.3.1. Instalar herramientas

Es recomendable instalar herramientas que ayuden a debugear SELinux:

```
root$  dnf install setroubleshoot setools
```

6.3.2. Revisar logs

Con las herramientas instaladas revisar los logs se vuelve muy fácil, y hasta las herramientras entregan los comando a ejecutar para solucionar los problemas. Para revisar los logs ejecutamos:

```
root$  sealert -a /var/log/audit/audit.log
```

Este comando entregará un X número de alertas, lo cual se verá escrito en la primera línea:

```
  100% done found X alerts in /var/log/audit/audit.log
```

Para cada alerta, las herramientas instaladas ofrecen de 1 a 3 soluciones distintas, con un respectivo porcentaje % de confiabilidad de la solución. Cada solución esta separada por una línea de asteriscos (**********) en donde sale escrito el porcentaje de confianza, ejemplo:

```
    **********  Plugin catchall_labels (83.8 confidence) suggests **********
```

Finalmente, la solución que se da esta compuesta por una breve explicación seguida de uno o dos comandos a ejecutar, los cuales estarán escritos explícitamente. Ejemplo:

```
    **********  Plugin catchall (17.1 confidence) suggests **********

    If you believe that httpd should be allowed getattr access on the index.html file by
    default.
    Then you should report this as a bug.
    You can generate a local policy module to allow this access.
    Do
    allow this access for now by executing:
    # grep httpd /var/log/audit/audit.log | audit2allow -M mypol
    # semodule -i mypol.pp
```

En donde los comandos a ejecutar en el ejemplo serían `grep httpd /var/log/audit/audit.log | audit2allow -M mypol` seguido por `semodule -i mypol.pp`.

De todas las soluciones que se dan, las primeras son las mejores, mientras que la última (y quizás la única que aparezca) siempre funcionará.

Finalmente, repetir el proceso por cada alerta que aparezca.

6.3.3. Plan B

En caso de que nada lo solucione, siempre se puede desactivar momentáneamente SELinux, ejecutando el comando:

```
root$  setenforce 0
```

SELinux volverá a iniciarse al reiniciar el equipo.

Disclaimer

La mayoría de los tildes en secciones de código fueron omitidos por problemas de compatibilidad.