

# Guía Instalación del Opennebula en el front-end1

## Contenido de la guía

Contenido de la guía.....	1
Objetivos: .....	2
Nomenclatura de la guía: .....	2
Indicaciones: .....	2
Recursos requeridos:.....	2
Consideraciones:.....	3
I. Instalación de Opennebula 4.4 .....	4
1.1 Información de Opennebula .....	4
1.2 Instalación Opennebula 4.4 .....	7
Paso 1 – Agregue los repositorios de Opennebula para opensuse .....	7
Paso 2 – Instalar los servidores one y sunstone. ....	8
Paso 3 – Definir las credenciales para la ejecución de Opennebula .....	9
Paso 4 – Crear el archivo para almacenar el PID .....	10
Paso 5 – Inicie el servicio one .....	10
1.3 Modificación de los ID de las credenciales de oneadmin.....	11
Paso 1 – Modifique el ID para el usuario y el grupo .....	11
Paso 2 – Cambie los permisos de propietario en todos los objetos del usuario oneadmin .....	12
Paso 3 – Verifique que funcione el servidor Opennebula .....	12
Paso 4 – Configuración de Opennebula para iniciar con el arranque del sistema .....	15
1.4 Configuración de la NAS para los hipervisores .....	17
Paso 1 – Ingresar a una sesión de consola como el usuario root .....	17
Paso 2 – Instalar el servidor nfs-server (si no estuviera instalado).....	17
Paso 3 – Publique el directorio a compartir con los siguientes datos .....	17
Paso 4 – Habilitar los permisos para la ejecución del servicios NFS .....	17
Paso 5 – Permitir acceso en el firewall .....	18

## Objetivos:

### Objetivo general de la guía.

- Instalar Opennebula en un equipo con openSUSE.

### Objetivos específicos.

- Instalar opennebula desde los repositorios.
- Ejecutar servicio por consola.
- Ejecutar la herramienta web sunstone-server
- Instalar y configurar el servicios NAS con NFS

## Nomenclatura de la guía:

En esta guía se ha utilizado el siguiente formato:

- Comandos con fuente courier en negrita, (además se indica que usuario que debe realizarlos).  
`root@front-end:~# psaux |grep sshd`
- Texto con resaltado en amarillo para la información que debe visualizar cuando realice algún procedimiento o comando. Por ejemplo:  
`root@front-end:~# mcedit /etc/resolv.conf`  
`search empresay.com.sv`  
`nameserver 192.168.2.1`
- Las notas o consideraciones se destacan con: **Nota:**
- Los comandos que deben digitarse en una sola línea y que por efectos del ancho del texto ocupan más de un renglón tienen un fondo en todo el ancho de la página, por ejemplo:  
`front-end1:~ # zypperar -f -n packman`  
`http://packman.inode.at/suse/openSUSE_13.1 packman`

## Indicaciones:

- Se escogió openSUSE porque presenta mayor facilidad de uso cuando no se tiene mucha experiencia en Linux, inicialmente para este módulo se había considerado Centos 6.4 u Ubuntu Server 12.04 LTS, debido a que OpenSUSE consume más recursos de RAM y proceso cundo se ejecuta con el escritorio.
- **Si se instala opensuse en como una máquina virtual como VirtualBox o VM Workstation, no se tendrá la opción de virtualización completa**, la cual es necesaria para ejecutar sistemas operativos sin modificación alguna, como Windows (todas las versiones) y los Linux cuyo kernel no sea similar al del sistema operativo instalado (en este caso opensuse).
- **Si se instala dentro de un programa de virtualización solo se podrá realizar paravirtualización**, aunque esto es una ventaja en el rendimiento, el inconveniente es modificar los kernel para ejecutar las máquinas virtuales. Tenga en cuenta que si el hipervisor estuviera instalado en un equipo real, el rendimiento sería mucho mayor.

## Recursos requeridos:

- Equipo o MV con Opensuse 13.X con KVM instalado (hipervisor1 – para verificar conexión)
- La imagen en formato ISO o el CD del sistema operativo openSUSE 13.X64 bits (enlace para descarga: <http://software.opensuse.org/131/es>). **Se utilizará esta distribución porque tiene una herramienta gráfica (Yast) el cual facilita la administración ocupa muy poco espacio de almacenamiento y RAM.**
- Conexión a Internet.
- Equipo o MV con una tarjeta en modo bridge o red interna si desea la comunicación más directa con otros equipos virtuales configurados previamente.

## Consideraciones:

- En esta guía se definirá como equipos virtuales al datastore1, front-end1 e hipervisores y como Máquinas virtuales a los sistemas operativos (Windows 2003 server, TinyCore y ttylinux) que se ejecutan en los hipervisores.
- Escriba en un papel todas las direcciones IPv4 de su red, utilice el valor de Y con el número de grupo asignado, por ejemplo: Y=grupo1 192.168.20~~Y~~.1 = 192.168.168.20~~1~~.1
- Los dos equipos deben poder enviar mensajes ICMP utilizando los nombres y dominio. Los nombres a utilizar son hipervisor1 y datastore1, el dominio es empresay.com.sv (Y número de grupo)
- Los dos equipos o MV deben ejecutarse en modo texto para ahorrar memoria RAM.
- La máquina virtual o equipo que se utilizará para el router debe tener dos interfaces de red.

# I. Instalación de Opennebula 4.4

## 1.1 Información de Opennebula

La documentación para conocer los elementos fundamentales de Opennebula así como su diseño e implantación puede descargarse de: [http://docs.Opennebula.org/pdf/4.4/Opennebula\\_4.4\\_design\\_and\\_installation\\_guide.pdf](http://docs.Opennebula.org/pdf/4.4/Opennebula_4.4_design_and_installation_guide.pdf). La información más completa se encuentra en: <http://docs.Opennebula.org/stable/index.html>

### ¿Qué es opennebula?

Es un proyecto de software libre para proporcionar un servicio de infraestructura de Cloud Computing para centros de datos, el cual pretende convertirse en la solución estándar para la industria; más centrado en plataformas privadas tanto estándares como privativas, aunque soporta drivers para EC2 de Amazon. Es decir Opennebula está diseñado para empresas que quieran instalar y administrar su propia nube (privada, pública o híbrida), ya que existen otras tecnologías que permiten a las empresas montar nubes para ofrecer a otras empresas el servicio IaaS (una especie de competencia a Amazon AWS)

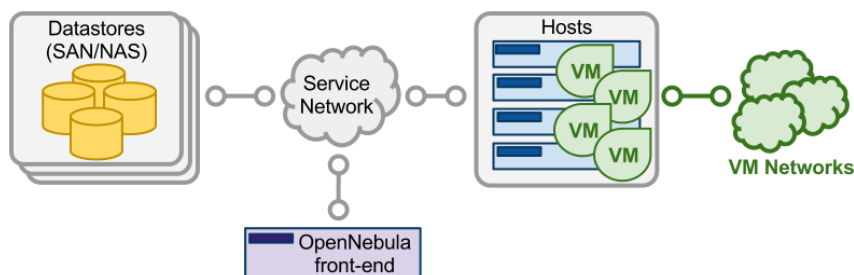
### Objetivos de Opennebula:

- Crear una solución adaptable, avanzada y escalable para construir nubes IaaS.
- Proporcionar un sistema modular para crear nubes usando distintos tipos de arquitecturas cloud y que interaccionen con otros servicios.
- Ofrecer la posibilidad de elegir interfaces desde software libre hasta estándares de facto.
- Asegurar la estabilidad y calidad del software distribuido.

La arquitectura de Opennebula supone que en la red de la empresa existen 4 elementos:

1. Datastores: equipos que brindan el servicio de almacenamiento ya sea por SAN (iSCSI) o NAS (NFS)
2. Red de datos: se encarga de comunicar a los equipos, crear políticas de seguridad u ofrecer servicios que prioricen determinado tipo de paquetes (bond, bridge, GigaEthernet, FDDI, NAT, Firewall, ebttables, VLAN, VPN, QoS, etc.)
3. Hipervisores: equipos o cluster de equipos que brindan los recursos a las máquinas virtuales (vCPU, RAM, vNET, drivers, etc.)
4. Front-end: equipo, equipos o máquinas virtuales que ejecutan los servicios de Opennebula.

En la siguiente imagen se ilustra el escenario de la arquitectura de Opennebula, (la imagen fue copiada del sitio de Opennebula y es de uso público, enlace [http://docs.Opennebula.org/stable/design\\_and\\_installation/building\\_your\\_cloud/plan.html](http://docs.Opennebula.org/stable/design_and_installation/building_your_cloud/plan.html))



*Imagen 1 – Arquitectura de una nube (Opennebula)*

En la imagen 2 se ilustran los componentes de Opennebula (imagen tomada del archivo en formato pdf de acceso público “Opennebula 4.4 Design and Installation Guide” disponible en [http://docs.Opennebula.org/pdf/4.4/Opennebula\\_4.4\\_design\\_and\\_installation\\_guide.pdf](http://docs.Opennebula.org/pdf/4.4/Opennebula_4.4_design_and_installation_guide.pdf)).

Así que para que se pueda instalar Opennebula se debe tener los siguientes elementos fundamentales:

**a. Tecnologías de virtualización:**

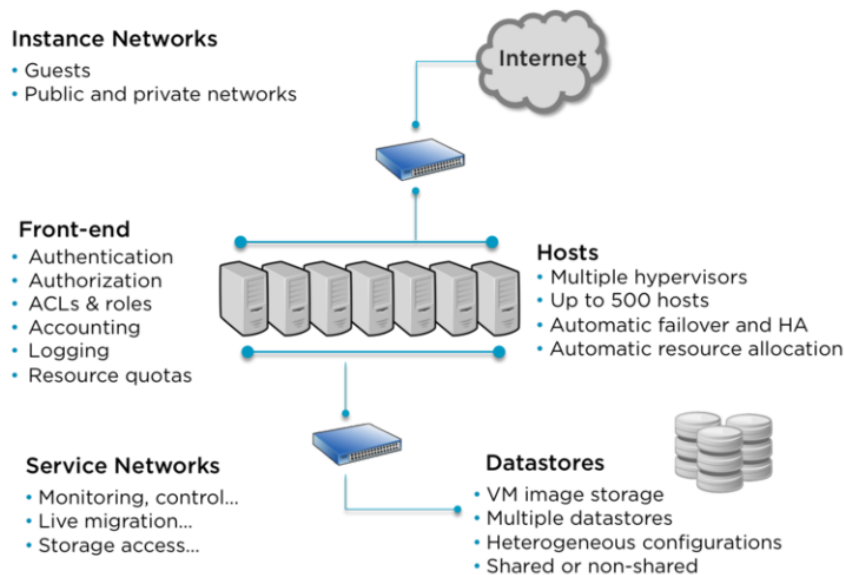
- Es el subsistema encargado de comunicarse con el hipervisor de cada nodo y llevar las tareas en cada paso del ciclo de vida las MVs.
- Proporciona soporte para los siguientes hipervisores: Xen, KVM y ESX

**b. Almacenamiento de datos empresariales: DataStore**

- DataStore: un medio de almacenamiento de imágenes de disco para MV (repositorio de imágenes).
- Diversos tipos de DataStore y métodos de transferencia: NFS, iSCSI, ssh, vmfs (VMWARE)

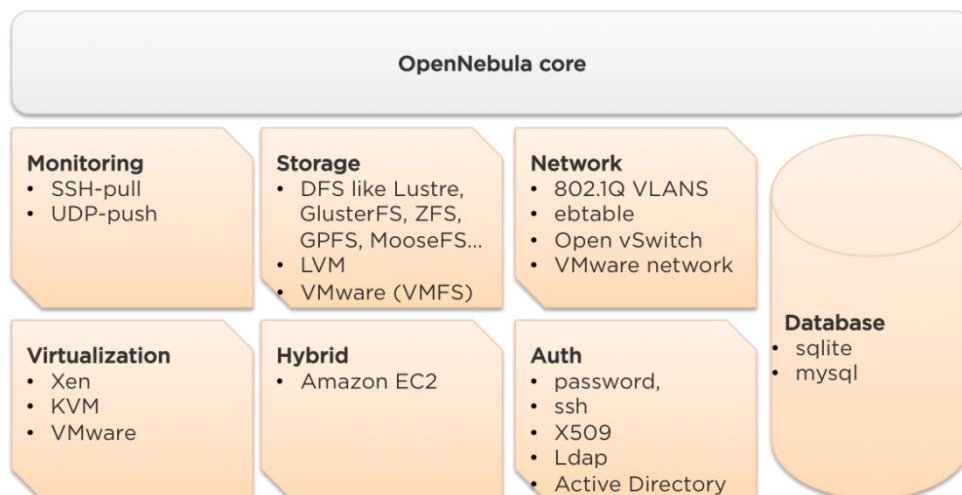
**c. Servicios de Networking**

- El sistema encargado de conectar los vNIC de las imágenes a las redes (públicas o privadas) oportunas.
- Se ofrece compatibilidad con diversos driver: Firewall, Openswitch, 802.1Q, VMWARE



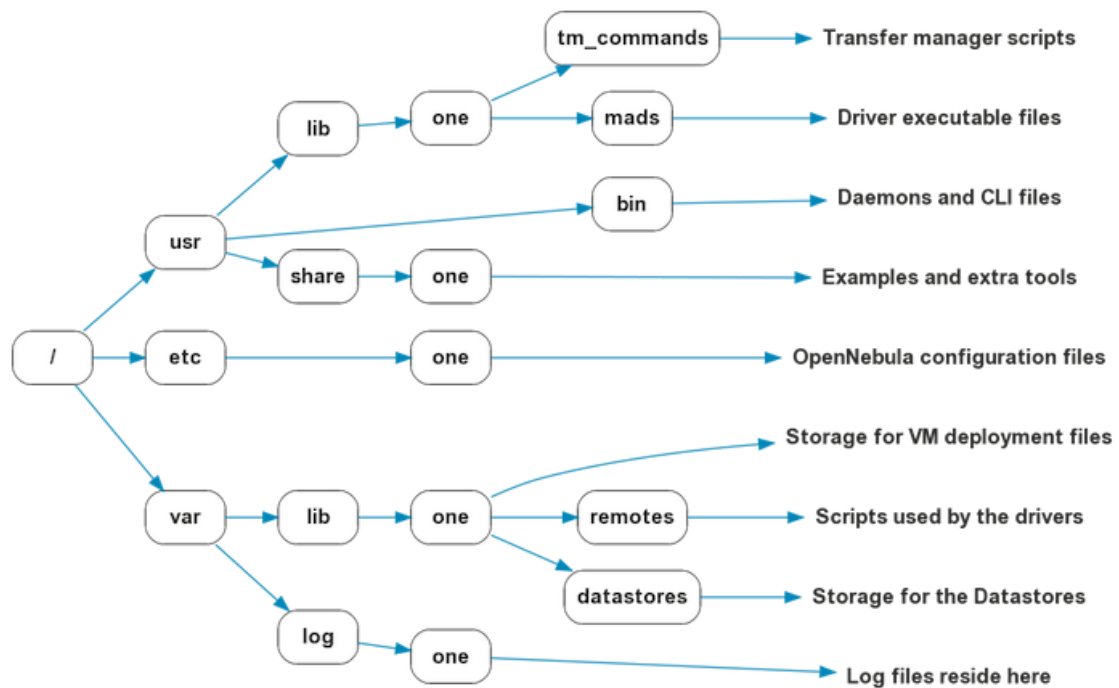
*Imagen 2 – Componentes de Opennebula*

Opennebula contiene una serie de módulos que permiten de forma conjunta brindar servicios de nube, tal como lo muestra la siguiente imagen (de acceso público y tomada de [http://docs.Opennebula.org/stable/design\\_and\\_installation/building\\_your\\_cloud/plan.html](http://docs.Opennebula.org/stable/design_and_installation/building_your_cloud/plan.html) )



*Imagen 3 – Módulos de la tecnología Opennebula 4.4*

Cuando se instala Opennebula se crea la estructura de archivos que se muestran en la siguiente figura (imagen tomada de [http://docs.Opennebula.org/stable/design\\_and\\_installation/building\\_your\\_cloud/ignc.html#step-1-front-end-installation](http://docs.Opennebula.org/stable/design_and_installation/building_your_cloud/ignc.html#step-1-front-end-installation))



## 1.2 Instalación Opennebula4.4

Métodos disponibles:

- Instalar desde paquetes genéricos: descargar, desempaquetar, instalar dependencias, crear objetos, etc.
- Instalar desde repositorios para la versión de openSUSE, crear repositorios, instalar paquetes, modificar algunas opciones.

### Paso 1 – Agregue los repositorios de Opennebula para opensuse

#### 1.1 Ingrese en una consola de texto como usuario root.

#### 1.2 Agregue los repositorios

```
front-end1:~ # zypperar -f -n packman
http://packman.inode.at/suse/openSUSE_13.1 packman
```

```
Añadiendo el repositorio 'packman' .....[hecho]
El repositorio 'packman' se ha añadido correctamente
Habilitado: Si
Actualización automática: Si
Comprobación GPG: Si
URI: http://packman.inode.at/suse/openSUSE_13.1
```

```
front-end1:~ # zypperaddrepo --no-gpgcheck --refresh -t YUM
http://downloads.Opennebula.org/repo/openSUSE/12.3/stable/x86_64 Opennebula
```

```
Añadiendo el repositorio 'Opennebula' .....[hecho]
El repositorio 'Opennebula' se ha añadido correctamente
Habilitado: Si
Actualización automática: Si
Comprobación GPG: No
URI: http://downloads.Opennebula.org/repo/openSUSE/12.3/stable/x86_64
```

#### 1.3 Actualice los contenidos de los repositorios

```
front-end1:~ # zypperrefresh
```

```
Obteniendo los metadatos del repositorio 'Opennebula' .....[hecho]
Construyendo el caché del repositorio 'Opennebula' .....[hecho]
Obteniendo los metadatos del repositorio 'packman' -----[\\]
```

```
Se ha recibido una nueva clave de firma para el repositorio o paquete:
ID de clave: 45A1D0671ABD1AFB
Nombre de la clave: PackMan Project (signingkey) <packman@links2linux.de>
. . .
```

```
¿Desea (r)echazar la clave, confiar (t)emporalmente o confiar (s)iempre? [r/t/s/?
mostrar todas las opciones] (r): s presione (s)
```

```
Obteniendo los metadatos del repositorio 'packman' .....[hecho]
Construyendo el caché del repositorio 'packman' .....[hecho]
```

```
. . .
Todos los repositorios han sido actualizados.
```

## Paso 2 – Instalar los servidores one y sunstone.

### 2.1 Instale el servicio one (texto) y la herramienta web (sunstone-server)

```
front-end1:~ # zypperinstallopennebulaopennebula-sunstone
```


```
Obteniendo los datos del repositorio...
Leyendo los paquetes instalados...
Resolviendo dependencias...
```

```
Los siguientes 8 paquetes NUEVOS van a ser instalados:
OpennebulaOpennebula-sunstonepwgenrubygem-mini_portilerubygem-nokogiri
rubygem-sqlite3 sqlite3 xmlrpc-c
8 nuevos paquetes a instalar.
Tamaño total a descargar: 21,9 MiB. Después de la operación, se utilizarán 43,6
MiB adicionales.
¿Desea continuar? [s/n/? mostrar todas las opciones] (s): s
Descargando paquetes . . .
```

```
(1/8) Instalando xmlrpc-c-1.33.06-2.9.....[hecho]
(2/8) Instalando pwgen-2.06-101.1.2 .....[hecho]
(3/8) Instalando rubygem-mini_portile-0.5.1-2.1.2 .....[hecho]
(4/8) Instalando sqlite3-3.7.17-2.1.2 .....[hecho]
(5/8) Instalando rubygem-nokogiri-1.6.0-3.1.3 .....[hecho]
Salida adicional del rpm:
rm: cannot remove '/usr/bin/nokogiri': No such file or directory
update-alternatives: using /usr/bin/nokogiri-1.6.0 to provide
/usr/bin/nokogiri (nokogiri) in auto mode

(6/8) Instalando rubygem-sqlite3-1.3.7-2.1.3 .....[hecho]
(7/8) Instalando Opennebula-4.4.0-1 .....[hecho]
Salida adicional del rpm:
Creating oneadmin user

(8/8) Instalando Opennebula-sunstone-4.4.0-1 .....[hecho]
```


 **Nota:** En este proceso se ha creado el usuario oneadmin y el grupo cloud

```
front-end1:~ # geminstallsinatra
```

```
Fetching: rack-1.5.2.gem (100%)
Successfully installed rack-1.5.2
Fetching: tilt-1.4.1.gem (100%)
Successfully installed tilt-1.4.1
Fetching: rack-protection-1.5.2.gem (100%)
Successfully installed rack-protection-1.5.2
Fetching: sinatra-1.4.4.gem (100%)
Successfully installed sinatra-1.4.4
Parsing documentation for rack-1.5.2
Installing grid documentation for rack-1.5.2
Parsing documentation for tilt-1.4.1
Installing grid documentation for tilt-1.4.1
Parsing documentation for rack-protection-1.5.2
Installing grid documentation for rack-protection-1.5.2
. . .
Installing grid documentation for sinatra-1.4.4
4 gems installed
```



## 2.2 Verificar que se haya instalado correctamente Opennebula

 **Nota:** El ID de usuario para este ejemplo es 1001 y el ID de grupo es 1000, si existen otros usuarios el UID y GID serán diferentes.

```
front-end1:~ # cat /etc/passwd |grep oneadmin
oneadmin:x:1001:1000:Opennebula Cloud Admin:/var/lib/one:/bin/bash
```

```
front-end1:~ # cat /etc/group |grep cloud
cloud:x:1000:
```

```
front-end1:~ # ls /var/lib/one/
.bash_history bindata stores .fonts .local public_html .ssh
.xim.template.bashrc .config .emacs .inputrc .profilere motes
vms .xinitrc.template
```

## 2.3 Defina la contraseña al usuario oneadmin

```
front-end1:~ # passwd oneadmin
```

```
Nueva contraseña: 123456
CONTRASEÑA INCORRECTA: es demasiado simple/sistemática
CONTRASEÑA INCORRECTA: es demasiado sencilla
Vuelva a escribir la nueva contraseña: 123456
passwd: contraseña actualizada correctamente
```


# Paso 3 – Definir las credenciales para la ejecución de Opennebula

## 3.1 Crear el directorio para almacenar el archivo con las credenciales

```
front-end1:~ # mkdir /var/lib/one/.one
```

```
front-end1:~ # chown oneadmin:cloud /var/lib/one/.one
```

## 3.2 Crear el archivo con las credenciales para one

 **Nota:** El usuario para el acceso web será: oneadmin y La contraseña: opennebula, todo en minúsculas. Si tiene problemas puede utilizar mcedit y digitar: oneadmin:opennebula

```
front-end1:~ # echo "oneadmin:opennebula" > /var/lib/one/.one/one_auth
```

## 3.3 Verifique que se hayan creado correctamente.

```
front-end1:~ # cat /var/lib/one/.one/one_auth
oneadmin:opennebula
```

## 3.4 Defina los permisos del archivo one\_auth

```
front-end1:~ # chown oneadmin:cloud /var/lib/one/.one/one_auth
```


```
front-end1:~ # chmod 600 /var/lib/one/.one/one_auth
```

### 3.5 Verifique los permisos del archivo que tiene las credenciales del usuario web

```
front-end1:~ # ls -l /var/lib/one/.one/one_auth
-rw----- 1 oneadmincloud 20 Jan 24 20:41 /var/lib/one/.one/one_auth
```

## Paso 4 – Crear el archivo para almacenar el PID

### 4.1 Crear el directorio donde se almacena el archivo con el PID

 **Nota:** Esto es algo que debería hacer el instalador de Opennebula, ya que si no se crea hay un error al ejecutar one

```
front-end1:~ # mkdir /var/run/one
```

### 4.2 Defina los permisos al directorio

```
front-end1:~ #
```

## Paso 5 –Inicie el servicio one

Digite los siguientes comandos:

### 5.1 Inicie sesión en consola con el usuario oneadmin

```
localadmin@front-end1:~>su oneadmin -l
Contraseña: 123456
```

### 5.2 Verifique algunos parámetros

```
oneadmin@front-end1:~>pwd
/var/lib/one
```

```
oneadmin@front-end1:~>id
uid=1001(oneadmin) gid=1000(cloud) grupos=1000(cloud)
```

### 5.3 Inicie el servicio

```
oneadmin@front-end1:~>onestart
```

### 5.4 Verifique que funciona el servicio

```
oneadmin@front-end1:~>onehostlist
```

ID	NAME	CLUSTER	RVM	ALLOCATED_CPU	ALLOCATED_MEM	STAT
----	------	---------	-----	---------------	---------------	------

```
oneadmin@front-end1:~>onedatastorelist
```

ID	NAME	SIZE	AVAIL	CLUSTER	IMAGES	TYPE	DS	TM
0	system	0M	-	-	0	sys	-	shared
1	default	10.1G	59%	-	0	imgfsshared		
2	files	10.1G	59%	-	0	fil	fssh	

```
oneadmin@front-end1:~>onevmlist
```

ID	USER	GROUP	NAME	STAT	UCPU	UMEM	HOST	TIME
----	------	-------	------	------	------	------	------	------

### 5.5 Detenga el servicio one


```
oneadmin@front-end1:~>one stop
```

### 5.6 Finalice la sesión del usuario oneadmin

```
oneadmin@front-end1:~>exit
```

## 1.3 Modificación de los ID de las credenciales de oneadmin

Debido a que el usuario oneadmin y grupo cloud fueron creados por el archivo de instalación rpm, el sistema operativo asignó el primer número disponible para usuario; será necesario definir un nuevo id de forma que cuando el front-end se conecte a los hipervisores no haya conflictos de ID

 **Nota:** Para este paso es necesario que el servicio: one, one\_scheduler y sunstone estén detenidos.

### Paso 1 – Modifique el ID para el usuario y el grupo

#### 1.1 Inicie sesión en consola con el usuario root

```
localadmin@front-end1:~>su root -l
```

#### 1.2 Identifique las credenciales del usuario oneadmin

```
front-end1:~ # id oneadmin
```

```
uid=1001(oneadmin) gid=1000(cloud) grupos=1000(cloud)
```

Es posible que usted tenga otro número, si ha creado otros usuarios o grupos.

#### 1.3 Cambie el ID del usuario (UID)

```
front-end1:~ # usermod -u 9869 oneadmin
```

Verifique que el cambio UID 1001 -> 9869 se haya realizado

```
front-end1:~ # cat /etc/passwd | grep one
```

```
oneadmin:x:9869:9869:Administrador de Opennebula:/var/lib/one:/bin/bash
```

#### 1.4 Cambie el ID del grupo (GID)

```
front-end1:~ # groupmod -g 9869 cloud
```

Verifique que el cambio UID 100 -> 9869 se haya realizado

```
front-end1:~ # mcedit /etc/group | grep cloud
```

```
cloud:x:9869:oneadmin
```

Si no apareciera oneadmin dentro de cloud, agregue el usuario oneadmin al grupo cloud con editando el archivo /etc/group con el comando mcedit /etc/group

#### 1.5 Verifique las credenciales finales del usuario oneadmin


```
front-end1:~ # id oneadmin
```

```
uid=9869(oneadmin) gid=9869(cloud) grupos=9869(cloud)
```

## Paso 2 – Cambie los permisos de propietario en todos los objetos del usuario oneadmin

```
front-end1:~ # find / -user 1001 -exec chown -h 9869 {} \;
```

```
front-end1:~ # find / -group 1000 -exec chgrp -h 9869 {} \;
```

 **Nota:** Para listar los objetos con permisos se utiliza el comando `find / -user oneadmin`. Digite los siguientes comandos línea por línea:

```
front-end1:~ # chown -R oneadmin:cloud /var/lib/one/
front-end1:~ # chown -R oneadmin:cloud /var/spool/mail/oneadmin
front-end1:~ # chown -R oneadmin:cloud /var/run/one
front-end1:~ # chown -R oneadmin:cloud /var/run/lock/one
front-end1:~ # chown -R oneadmin:cloud /var/lock/one
front-end1:~ # chown -R oneadmin:cloud /var/log/one
front-end1:~ # chown -R oneadmin:cloud /run/one
front-end1:~ # chown -R oneadmin:cloud /run/lock/one
```

## Paso 3 – Verifique que funcione el servidor Opennebula

Digite los siguientes comandos:

### 3.1 Ingrese como el usuario oneadmin

```
localadmin@front-end1:~>su oneadmin -l
```

### 3.2 Verifique que haya ingresado en el directorio \$HOME = /var/lib/one

```
oneadmin@front-end1:~>pwd
/var/lib/one
```

### 3.3 Verifique que el usuario tenga el UID y GID asignado

```
oneadmin@front-end1:~>id
uid=9869(oneadmin) gid=9869(cloud) grupos=9869(cloud)
```

### 3.4 Ejecute el servicio one

```
oneadmin@front-end1:~>onestart
```

Si todo está bien no parecerá mensaje en la pantalla.

### 3.5 Ejecute el servicio Web sunstone

```
oneadmin@front-end1:~>sunstone-server start
VNC proxy started
sunstone-serverstarted
```

Si desea ver la herramienta Web deberá digitar en el navegador del front-end1 **http://127.0.0.1:9869**

De manera predeterminada sunstone-server solo permite el acceso desde el propio equipo (front-end)

 **Nota:** No es necesario iniciar en runlevel 5 ya que puede utilizar putty + xming y ejecutar el navegador web.



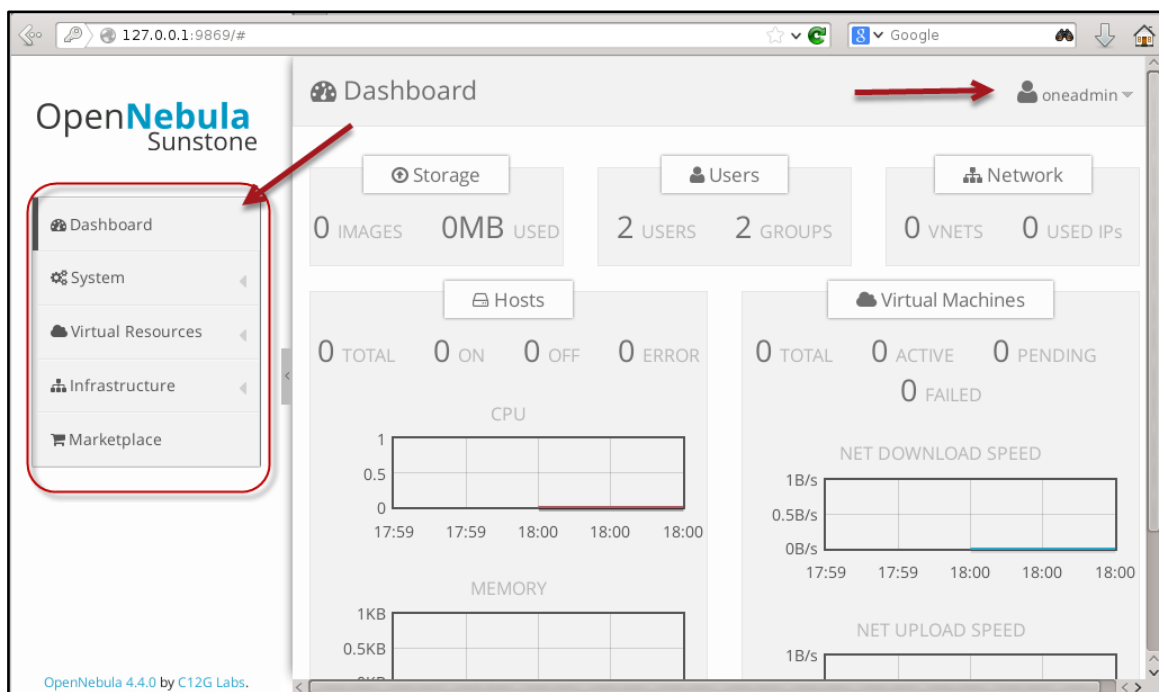
Para permitir el acceso desde otro equipo, sub-red o cualquier equipo es necesario modificar el archivo de configuración del servidor sunstone-server ubicado en `/etc/one/stone-server.conf`, en donde en la línea **:host** se define el valor para el acceso.

```
# Server Configuration
#
:host: 127.0.0.1
:port: 9869
```

Para permitir cualquier acceso  
**:host: 0.0.0.0**

Para permitir solo el acceso de la red de empresay.com.sv  
**:host: 192.168.200.0**

Al ingresar se deberá observar la pantalla del dashborad



Ejemplo de pantalla para agregar el tipo de hipervisor y definir tipo de red.

### 3.6 Apague los servicios one y sunstone

```
oneadmin@front-end1:~>sunstone-server stop
sunstone-serverstopped
```

```
oneadmin@front-end1:~>one stop
oned and schedulerstopped
```

## Paso 4 – Configuración de Opennebula para iniciar con el arranque del sistema

Los servicios que utiliza Opennebula son:

- **one**→ servicio principal se utiliza por consola
- **one\_scheduler**→ servicio para agendar eventos automáticamente (inicio, stop, pausa de MV, etc.) se ejecuta junto con one (oned)
- **sunstone-server**→ web service para ejecutar una herramienta de administración Web, es opcional, pero presenta una gran ayuda cuando se comienza a trabajar con Opennebula
- **ozone**→ es un servicio para administrar varios Opennebulas, es decir administrar varias nubes, esto es útil por ejemplo cuando hay que administrar muchas nubes provenientes de varias empresas

Consideraciones para este paso:

- Sólo el usuario oneadmin tiene derechos para ejecutar los servicios de Opennebula
- El servicio se inicia manualmente.
- Si el equipo se reinicia no se iniciaría Opennebula
- En openSUSE desde la versión 12 ya no se utiliza el sysV para iniciar los servicios según el runlevel que arranque el sistema, ahora se utiliza systemd. Para mayor información consulte los siguientes sitios: <http://news.opensuse.org/2011/12/22/systemd---boot-faster-and-cleaner-with-opensuse-12-1/> y [http://en.opensuse.org/openSUSE:Systemd\\_tips](http://en.opensuse.org/openSUSE:Systemd_tips)
- one (start/stop) y sunstone-server(start/stop) son scripts que inician los servicios oned, scheduler y sunstone-server, cuando active los servicios por systemd ya no se podrán ejecutar dichos scripts (porque los servicios no fueron arrancados por esos scripts, sino que por systemd)
- Cuando utilice systemd sólo el root podrá iniciar (start), ver (status) y finalizar (stop) los servicios.

### 4.1 Verifique que esté apagado one y sunstone

```
oneadmin@front-end1:~> sunstone-server stop
sunstone-server stopped
```

```
oneadmin@front-end1:~> one stop
oned and scheduler stopped
```

### 4.2 Ingrese en una consola de texto como el usuario root

### 4.3 Cree la variable ONE\_AUTH con la ruta al archivo con el usuario y contraseña

```
front-end1:~ # sudo -u oneadmin env ONE_AUTH=/var/lib/one/.one/one_auth
/usr/sbin/onedsetup
```

### 4.4 Habilite para el arranque el servicio one (oned) y scheduler

```
front-end1:~ # systemctl enable one.service
```

```
ln -s '/usr/lib/systemd/system/one.service' '/etc/systemd/system/multi-
user.target.wants/one.service'
```

```
front-end1:~ # systemctl enable one_scheduler.service
```

```
ln -s '/usr/lib/systemd/system/one_scheduler.service' '/etc/systemd/system/multi-
user.target.wants/one_scheduler.service'
```

#### 4.5 Active el servicio one (oned) y scheduler en el arranque

```
front-end1:~ #systemctlstartone.service
```


```
front-end1:~ #systemctlstartone_scheduler.service
```

#### 4.6 Habilite y active el servicio sunstone

```
front-end1:~ #systemctlenablesunstone.service
```

```
ln -s '/usr/lib/systemd/system/sunstone.service' '/etc/systemd/system/multi-  
user.target.wants/sunstone.service'
```

```
front-end1:~ #systemctlstartsunstone.service
```

 **Nota:** Tenga en cuenta que cuando haya conectado el datastore y los hipervisores deberá tener una secuencia de arranque, es decir debe arrancar primero el datastore1 para cuando se inicie el front-end1 el target iSCSI y servidor NFS estén listos para ofrecer el recurso al sistema operativo del front-end1

#### 4.7 Reinicie el equipo para comprobar el arranque de Opennebula

```
front-end1:~ #init 6 o reboot
```

#### 4.8 OPCIONAL: Si desea detener (stop) o ver estado (status) los servicios de Opennebula

```
front-end1:~ # systemctl status sunstone.service
```

```
sunstone.service - OpenNebula Web UI Server  
Loaded: loaded (/usr/lib/systemd/system/sunstone.service; enabled)  
Active: active (running) since sáb 2014-01-25 23:33:04 CST; 52s ago  
Process: 4346 ExecStop=/bin/kill -INT $MAINPID (code=exited, status=0/SUCCESS)  
Main PID: 4427 (bash)  
CGroup: /system.slice/sunstone.service  
└─4427 /bin/bash -c sleep 5; /usr/bin/ruby /usr/lib/one/sunstone/sunstone-  
server.rb> /var/log/one/sunstone.log 2>...  
└─4432 /usr/bin/ruby /usr/lib/one/sunstone/sunstone-server.rb
```

```
ene 25 23:33:04 eq1 systemd[1]: StartingOpenNebula Web UI Server...  
ene 25 23:33:04 eq1 systemd[1]: StartedOpenNebula Web UI Server.
```

```
front-end1:~ #systemctl stop sunstone.service
```

**Para activar si estuviera apagado Opennebula se utilizan los siguientes comandos**

```
front-end1:~ #systemctlstartone.service  
front-end1:~ #systemctlstartone_scheduler.service  
front-end1:~ #systemctlstartsunstone.service
```

**Para activar si estuviera apagado Opennebula se utilizan los siguientes comandos**

```
front-end1:~ #systemctl stop sunstone.service  
front-end1:~ #systemctl stop one_scheduler.service  
front-end1:~ #systemctl stop one.service
```



## 1.4 Configuración de la NAS para los hipervisores

Para compartir el directorio /var/lib/one se utilizará el protocolo NFS

### Paso 1 – Ingresar a una sesión de consola como el usuario root

### Paso 2 – Instalar el servidor nfs-server (si no estuviera instalado)

```
front-end1:~ # zypperinstallnfs-kernel-server
```

### Paso 3 – Publique el directorio a compartir con los siguientes datos

- Directorio a compartir = /var/lib/one
- Equipos y usuarios = \* (todos), para mayor seguridad se podría colocar 192.168.20Y.0
- Permisos a los usuarios = no\_subtree\_check,rw,root\_squash,sync

Digite en una sola línea los parámetros del directorio a exportar

```
front-end1:~ # mcedit /etc/exports
```

```
/var/lib/one *(no_subtree_check,rw,root_squash,sync)
```

### Paso 4 – Habilitar los permisos para la ejecución del servicios NFS

#### 4.1 Active el servicio rpc

```
front-end1:~ # systemctlenablerpcbind.service
```

```
ln -s '/usr/lib/systemd/system/rpcbind.service' '/etc/systemd/system/multi-  
user.target.wants/rpcbind.service'  
ln -s '/usr/lib/systemd/system/rpcbind.socket' '  
'/etc/systemd/system/sockets.target.wants/rpcbind.socket'
```

#### 4.2 Active el servicio nfs

```
front-end1:~ # systemctlenablenfsserver.service
```


nfsserver.service is not a native service, redirecting to /sbin/chkconfig.  
Executing /sbin/chkconfig nfsserver on  
The unit files have no [Install] section. They are not meant to be enabled  
using systemctl.  
Possible reasons for having this kind of units are:  
1) A unit may be statically enabled by being symlinked from another unit's  
.wants/ or .requires/ directory.  
2) A unit's purpose may be to act as a helper for some other unit which has  
a requirement dependency on it.  
3) A unit may be started when needed via activation (socket, path, timer,  
D-Bus, udev, scripted systemctl call, ...).

#### 4.3 Inicie el servicio rpc

```
front-end1:~ # systemctlstart rpcbind.service
```

#### 4.4 Inicie el servicio nfs

```
front-end1:~ # systemctlstart nfsserver.service
```

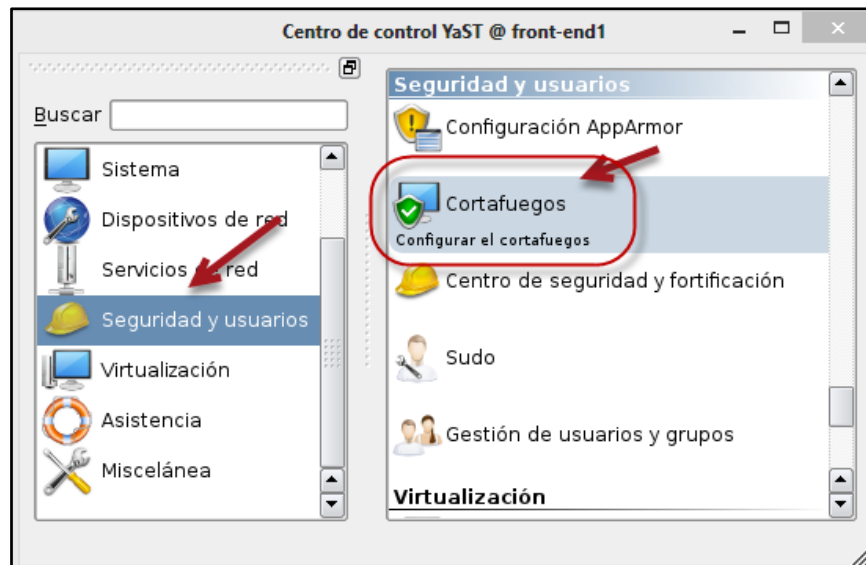
 **Nota:** Cada vez que se modifique el archivo /etc/exports se debe digitar el comando exports

## Paso 5 – Permitir acceso en el firewall

**5.1 Abra yast**

**5.2 Seleccione seguridad y usuarios**

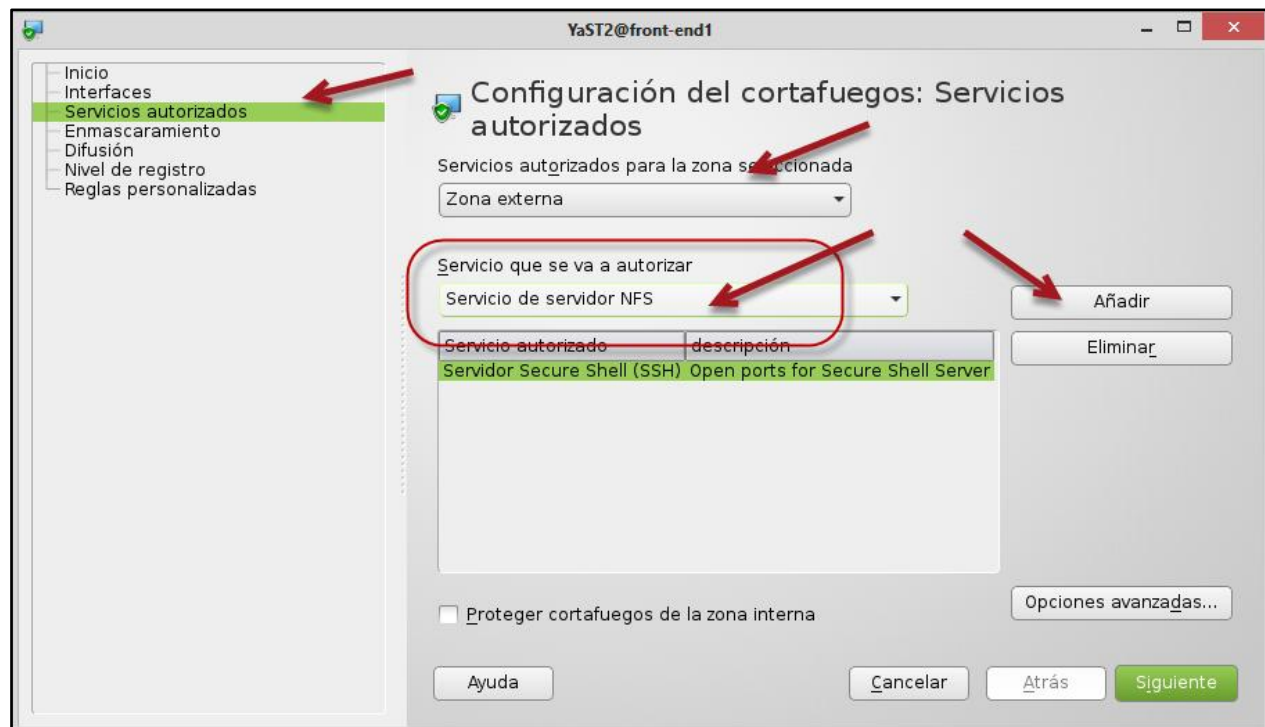
**5.3 Abra Cortafuegos**



**5.4 Seleccione "Servicios autorizados" del panel de la izquierda.**

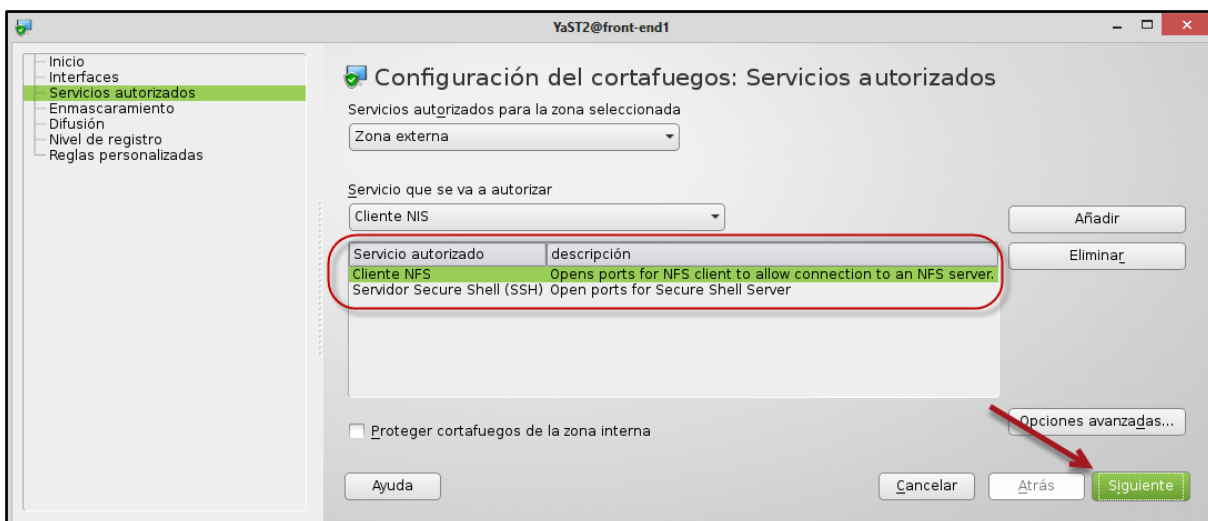
**5.5 Seleccione zona externa**

**5.6 Seleccione "Servicio de servidor NFS"**

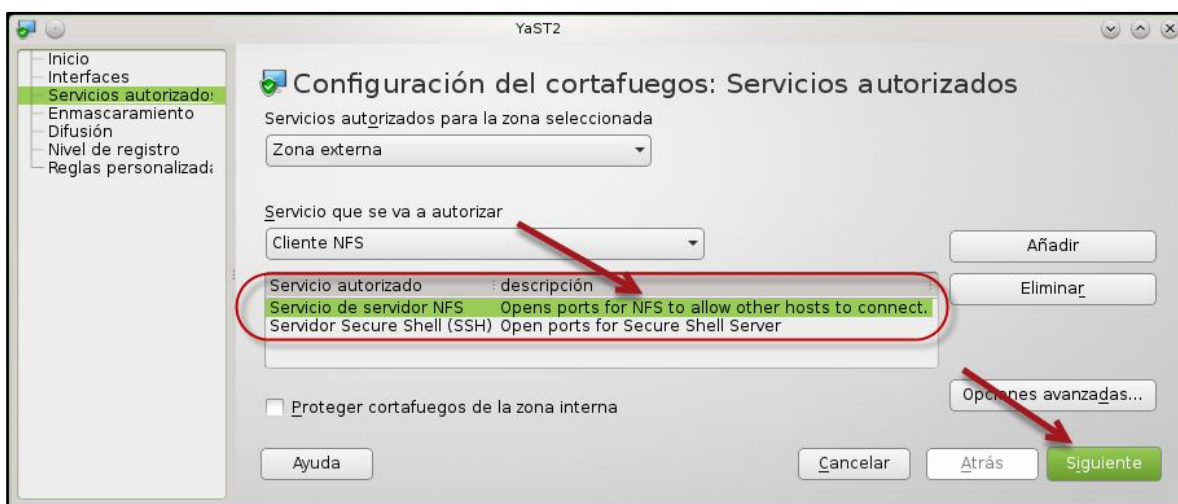


**5.7 Dé un clic en el botón "Añadir"**

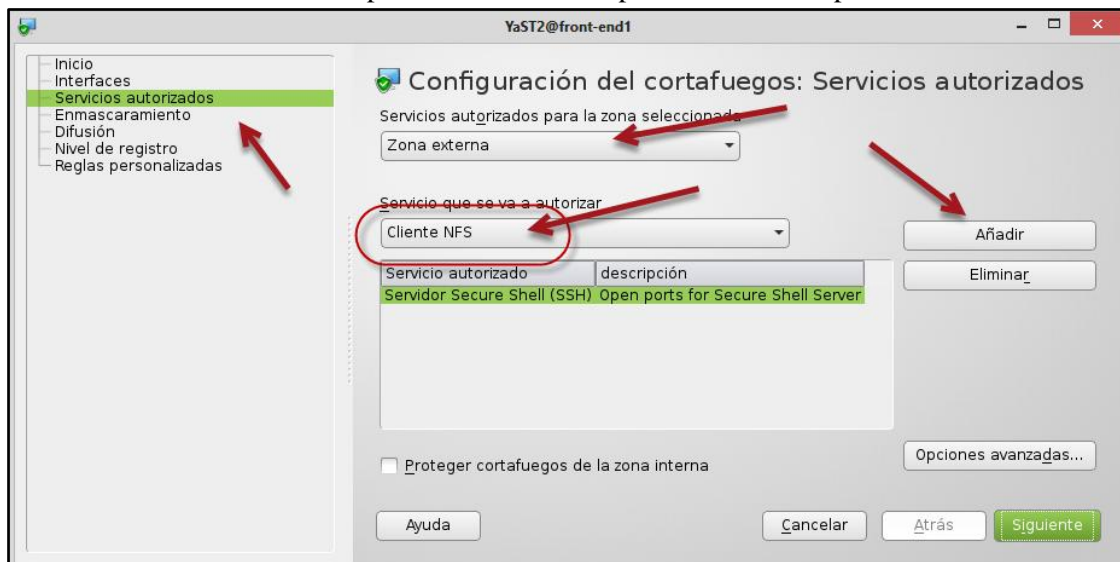
## 5.8 Compruebe que se haya agregado el puerto para el servidor NFS



## 5.9 Dé un clic en el botón "Siguiente"



**Nota:** Si conecta un cliente NFS en openSUSE deberá dar permiso al firewall para el cliente NFS



## Paso 6 – Verifique que el servidor publique los recursos


### 6.1 Verifique que el servidor RPC está funcionando

Digite el siguiente comando:

```
front-end1:~ # rpcinfo -p
programversprotoportservice
100000      4      tcp      111    portmapper
100000      4      udp       111    portmapper
100005      1      udp    20048  mountd
100005      1      tcp    20048  mountd
100005      2      udp    20048  mountd
100024      1      udp   48458  status
100024      1      tcp   52972  status
100003      4      tcp    2049  nfs
100227      3      tcp    2049  nfs_acl
100003      4      udp    2049  nfs
100227      3      udp    2049  nfs_acl
100021      4      udp   50502  nlockmgr
100021      4      tcp   40008  nlockmgr
. . .
```

### 6.2 Verifique los recursos que publica el servidor NFS

```
front-end1:~ # showmount -e 127.0.0.1
Exportlistfor 127.0.0.1:
/var/lib/one *
```

 **Nota:** Los comandos para activar y desactivar el firewall son:

- Apagar el firewall  
/sbin/SuSEfirewall2 off
- Encender el firewall  
/sbin/SuSEfirewall2 on
- Desactivar temporalmente el firewall  
/sbin/SuSEfirewall2 stop