

# UBUNTU SERVER

**Administración**

# REGISTROS DE SUCEOS

Guardan detalle del funcionamiento del sistema: **/var/log**

Daemon: **syslog-ng** y **rsyslogd** (predeterminado en Ubuntu).

Registro	Tipo de eventos
auth.log	Mensajes relativos a la seguridad y a las autorizaciones.
boot.log	Mensajes de arranque del sistema.
dpkg.log	Mensajes relacionados con la instalación de paquetes.
kern.log	Mensajes relacionados con el núcleo.
syslog	Mensajes relacionados con el daemon de registro.

# MONITORIZACIÓN

## Herramientas:

**ps:** Informa sobre los procesos que se están ejecutando en el sistema.

**ps tree:** Muestra los procesos que se están ejecutando con una estructura de árbol.

**top** y **htop:** Información sobre el grado de ocupación de la CPU, de la memoria RAM, de la memoria de intercambio (swap) y los procesos que se están ejecutando.

**uptime:** Nos informa sobre la cantidad de tiempo que lleva funcionando el sistema.

**free:** Ofrece información detallada sobre la memoria (tanto física como de intercambio).

**pmap:** Muestra el mapa de memoria de un proceso en particular.

**who:** Informa sobre los usuarios que están en estos momentos autenticados en el sistema.

# GESTIÓN DE SERVICIOS

Los servicios (Daemon) son programas que se ejecutan en segundo plano para ofrecer una función concreta de forma independiente del usuario que pueda iniciar sesión en el sistema.

Servicio	Descripción	Servicio	Descripción
atd	Ejecuta las tareas programadas con <b>at</b> .	ntp	Sincroniza la hora en la red.
crond	Ejecuta las tareas programadas del sistema.	rpc	Permite la ejecución remota de procesos.
netfs	Monta el sistema de archivos en red.	rsyslog	Almacena los sucesos del sistema.
network	Activa las interfaces de red del sistema.	sshd	Shell segura.
nfs	Servicio de ficheros en red.	xinetd	Activa servicios en red.

# GESTIÓN DE SERVICIOS (II)

Servicios relacionados con la funcionalidad del servidor.

Servicio	Descripción	Servicio	Descripción
cups	Servidor de impresión.	mysqld	Servidor de base de datos MySQL.
dhcpd	Servidor DHCP.	named	Servidor DNS.
httpd	Servidor de páginas web Apache.	sendmail	Servidor de correo electrónico.
innd	Servidor de noticias.	smb	Compartir archivos con Windows.
iptables	Cortafuegos del sistema.	squid	Servidor proxy.

# GESTIÓN DE SERVICIOS (III)

Obtener todos los servicios en ejecución o que fallan:

```
$ systemctl --state running
```

```
$ systemctl --state failed
```

Iniciar, detener o reiniciar un servicio

```
$ systemctl start <service>
```

```
$ systemctl stop <service>
```

```
$ systemctl restart <service>
```

# GESTIÓN DE SERVICIOS (IV)

Nivel de ejecución: Modo de operación del sistema operativo (0 = Apagado; 6 = Reinicio)

\$ **runlevel**

La ejecución automática de servicios se organiza en base a los niveles de ejecución. Cada nivel de ejecución en particular iniciará o detendrá unos servicios en concreto.

Los servicios que se inician con cada nivel de ejecución se guardan en los directorios desde **/etc/rc0.d** a **/etc/rc6.d** y **/etc/rcS.d** (arranque del sistema).

Cada directorio contiene enlaces simbólicos a elementos del directorio **/etc/init.d**.

Los scripts que comienzan con la letra **K** finalizan servicios (**Kill**) y los que comienzan con la letra **S** los inician (**start**).

Si un servicio debe correr únicamente en un nivel de ejecución particular, dispondrá de un script de inicio, mientras que en las carpetas del resto de los niveles de ejecución deberá aparecer un script de finalización.

El número que sigue a la letra inicial hace que cada script se ejecute en un orden determinado.

# CUOTAS DE DISCO

Instalación de paquetes:

```
$ sudo apt-get install quota quotatool
```

Activar las cuotas en el sistema de archivos:

```
$ sudo nano /etc/fstab
```

```
# Añadimos las cuotas al directorio /home e indicamos que podrá  
# tener cuotas de usuario (usrquota) y de grupo (grpquota).  
UUID=0f953ff4 /home ext4 defaults,usrquota,grpquota 0 2
```

Para que el sistema de cuotas esté activo montaremos de nuevo la partición sobre la que se aplicarán las cuotas:

```
$ sudo mount -o remount,rw /home
```



# CUOTAS DE DISCO (II)

Crear los archivos de cuota y la tabla de uso de espacio compartido:

```
$ sudo quotacheck -cgu /home
```

Configurar cuotas para usuarios (-u) y grupos (-g):

```
$ sudo edquota -u juan
```

```
$ sudo edquota -g ventas
```

Blando: El sistema avisará cuando el límite sea sobrepasado.

Duro: El sistema impide que el límite sea sobrepasado.

Sist. arch.	bloques	blando	duro	inodos	blando	duro
/dev/sda6	64118	65000	66000	1299	0	0

bloques:  
Tamaño máximo que se puede ocupar.

inodos:  
Número de archivos que se pueden crear.

Para eliminar cualquier límite, basta con poner su valor a cero.

# TAREAS PROGRAMADAS: CRON

El servicio que se encarga de ejecutar tareas en intervalos regulares es **cron**.

Los usuarios pueden crear tareas programadas ejecutaremos el comando:

```
$ crontab -e
```

Cada entrada en el archivo de crontab del usuario tiene seis datos:

1. **Minuto:** Un valor entero entre 0 y 59.
2. **Hora:** Un valor entero entre 0 y 23.
3. **Día del mes:** Un valor entero entre 1 y 31.
4. **Mes del año:** Un valor entero entre 1 y 12.
5. **Día de la semana:** Un valor entero entre 0 y 7 (donde 0 = domingo). También se puede utilizar los valores: sun, mon, tue, wed, thu, fri y sat.
6. **Orden:** Será ejecutada por la Shell.

Un asterisco (\*) en un campo indica todos.

Ejemplo de tarea:

```
00 22 * 1-6 1,2,3,4,5 respaldo.sh
```

# TAREAS PROGRAMADAS: CRON (II)

Las tareas de usuario se guardan en: `/var/spool/cron`

Consultar las tareas programadas:

```
$ crontab -l
```

Eliminar las tareas programadas:

```
$ crontab -r
```

Si el archivo `/etc/cron.allow` existe, sólo los usuarios incluidos en él pueden utilizar **cron**. Si no existe se comprueba `/etc/cron.deny`, y los usuarios que no estén incluidos en él pueden hacer uso de **cron**.

# TAREAS PROGRAMADAS: CRON (III)

El root puede programar las tareas del sistema editando directamente el archivo **/etc/crontab**. Los campos de las tareas programadas son 7: Minuto, Hora, Día del Mes, Mes, Día de la Semana, Usuario, Comando. (Usuario -> El que ejecuta el comando).

Otra manera de usar cron es colocar un script en los directorios:

- /etc/cron.hourly
- /etc/cron.daily
- /etc/cron.weekly
- /etc/cron.monthly

Entonces el script se ejecutará automáticamente cada hora, cada día, cada semana o cada mes, dependiendo del directorio.

# TAREAS PROGRAMADAS: AT

El comando **at** nos permite ejecutar una tarea en un momento particular, una sola vez.

```
$ at 16:30
```

```
rm /home/juan/*.backup  
rm /home/juan/*.bak
```

**Ctrl + D**

Otras formas de usar **at**:

```
$ at 14:35 10/12/2015
```

```
$ at 9pm
```

```
$ at 4am Sep 15
```

```
$ at now + 4 days
```

```
$ at 8pm + 2 weeks
```

# TAREAS PROGRAMADAS: AT (II)

Consultar los trabajos pendientes:

```
$ at -l
```

Eliminar un trabajo indicado (En el ejemplo borra el trabajo número 4)

```
$ at -r 4
```

Si el archivo **/etc/at.allow** existe, sólo los usuarios incluidos en él pueden utilizar **at**. Si no existe se comprueba **/etc/at.deny** , y los usuarios que no estén incluidos en él pueden hacer uso de **at**.

# EJERCICIOS

1. Desde la terminal, averigua qué líneas del archivo **auth.log** contienen el texto “**FAILED LOGIN**” y trata de encontrarles una explicación.
2. Utiliza el archivo **auth.log** para averiguar la fecha y hora de del último inicio de sesión del usuario **alberto**.
3. Averigua el **PID** del proceso **systemd**.
4. Descubre quién es el proceso padre de **atd**.
5. Obtener la cantidad de memoria ram y swap del sistema.
6. Averigua el estado del servicio **ssh**.
7. Lista el estado de todos los servicios.
8. Obtén el nivel de ejecución actual del sistema y indica algún servicio que se active en este nivel.