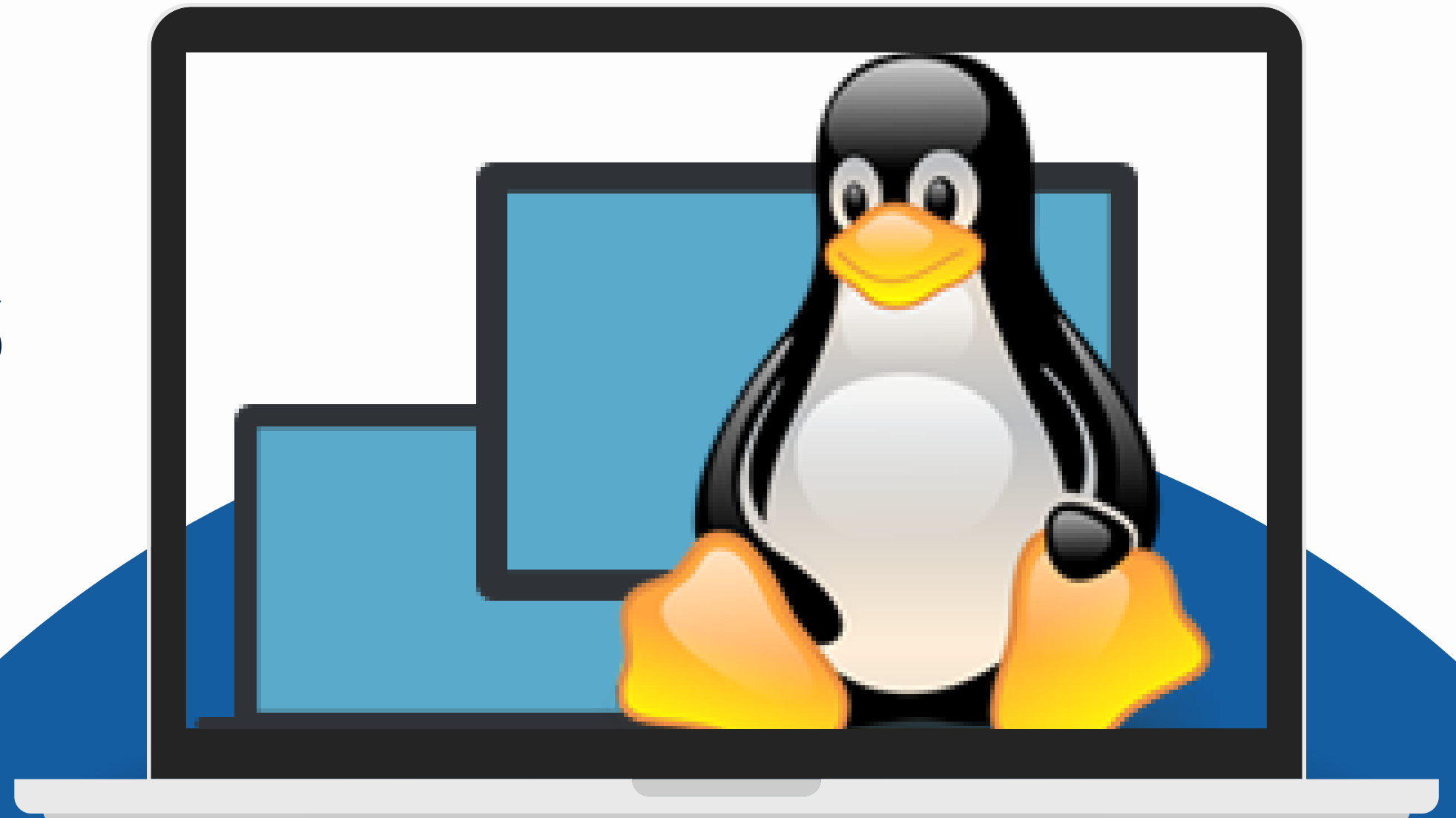





Lab - Procesos administrativos

Presentación realizada por Brendon Buriol



Objetivos

En este laboratorio:

- Crearemos un nuevo archivo de registro para listados de procesos
 - Utilizaremos el comando top
 - Estableceremos una tarea repetitiva que ejecute sus comandos de auditoría anteriores una vez al día.
- 

Tarea 1: Ejercicio: crear una lista de procesos

En este ejercicio, se creará un archivo de registro a partir del comando ps. Este archivo de registro debe agregarse a la sección SharedFolders:

Creearemos un archivo de registro llamado processes.csv desde ps -aux y omita cualquier proceso que contenga usuario root o contenga "["or"]" en la sección COMMAND.

Esto se hará a partir del siguiente conjunto de códigos:

```
[ec2-user@ip-10-0-10-194 ~]$ pwd
/home/ec2-user
[ec2-user@ip-10-0-10-194 ~]$ cd companyA
[ec2-user@ip-10-0-10-194 companyA]$ sudo ps -aux | grep -v root | sudo tee SharedFolders/processes.csv
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
dbus      1699  0.0  0.4  58248  3924 ?        Ss   16:01   0:00 /usr/bin/dbus-daemon --system --address=sys
temd: --nofork --nopidfile --systemd-activation
rpc       1700  0.0  0.3  67256  3304 ?        Ss   16:01   0:00 /sbin/rpcbind -w
libstor+  1702  0.0  0.1  12628  1856 ?        Ss   16:01   0:00 /usr/bin/lsmd -d
rngd      1722  0.0  0.4  94212  4628 ?        Ss   16:01   0:00 /sbin/rngd -f --fill-watermark=0 --exclude=
jitter
chrony    1731  0.0  0.3 120344  3208 ?        S    16:01   0:00 /usr/sbin/chronyd -F 2
postfix   2153  0.0  0.6  90388  6620 ?        S    16:01   0:00 pickup -l -t unix -u
postfix   2154  0.0  0.6  90464  6712 ?        S    16:01   0:00 qmgr -l -t unix -u
ec2-user  3168  0.0  0.4 148504  4476 ?        S    16:07   0:00 sshd: ec2-user@pts/0
ec2-user  3170  0.0  0.4 124736  3956 pts/0    Ss   16:07   0:00 -bash
[ec2-user@ip-10-0-10-194 companyA]$ cat SharedFolders/processes.csv
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
dbus      1699  0.0  0.4  58248  3924 ?        Ss   16:01   0:00 /usr/bin/dbus-daemon --system --address=sys
temd: --nofork --nopidfile --systemd-activation
rpc       1700  0.0  0.3  67256  3304 ?        Ss   16:01   0:00 /sbin/rpcbind -w
libstor+  1702  0.0  0.1  12628  1856 ?        Ss   16:01   0:00 /usr/bin/lsmd -d
rngd      1722  0.0  0.4  94212  4628 ?        Ss   16:01   0:00 /sbin/rngd -f --fill-watermark=0 --exclude=
jitter
chrony    1731  0.0  0.3 120344  3208 ?        S    16:01   0:00 /usr/sbin/chronyd -F 2
postfix   2153  0.0  0.6  90388  6620 ?        S    16:01   0:00 pickup -l -t unix -u
postfix   2154  0.0  0.6  90464  6712 ?        S    16:01   0:00 qmgr -l -t unix -u
ec2-user  3168  0.0  0.4 148504  4476 ?        S    16:07   0:00 sshd: ec2-user@pts/0
ec2-user  3170  0.0  0.4 124736  3956 pts/0    Ss   16:07   0:00 -bash
[ec2-user@ip-10-0-10-194 companyA]$
```

Tarea 2: Ejercicio: enumerar los procesos usando el comando top

En este ejercicio vamos a:

- Ejecutar el comando top para mostrar los procesos y subprocesos que están activos en el sistema.
- Observar los resultados del comando top.

El resultado del comando top proporciona el rendimiento del sistema y brinda la siguiente información: número total de tareas, cuántas se están ejecutando, cuántas están inactivas, cuántas están detenidas y en estado zombie. Proporciona el porcentaje de CPU utilizada, la memoria KiB utilizada y el intercambio de KiB.



```
top - 16:14:12 up 12 min, 1 user, load average: 0,00, 0,00, 0,00
Tasks: 86 total, 1 running, 47 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,0 us, 0,0 sy, 0,0 ni,100,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 966816 total, 400240 free, 72904 used, 493672 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 751988 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	123516	5448	3920	S	0,0	0,6	0:01.59	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H
5	root	20	0	0	0	0	I	0,0	0,0	0:00.08	kworker/u4:0
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0,0	0,0	0:00.03	ksoftirqd/0
8	root	20	0	0	0	0	I	0,0	0,0	0:00.04	rcu_sched
9	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_bh
10	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/0
11	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	watchdog/0
12	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/0
13	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/1
14	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	watchdog/1
15	root	rt	0	0	0	0	S	0,0	0,0	0:00.20	migration/1
16	root	20	0	0	0	0	S	0,0	0,0	0:00.02	ksoftirqd/1
18	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/1:0H
20	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs
21	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	netns
34	root	20	0	0	0	0	I	0,0	0,0	0:00.02	kworker/u4:2
110	root	20	0	0	0	0	S	0,0	0,0	0:00.00	khungtaskd
203	root	20	0	0	0	0	S	0,0	0,0	0:00.00	oom_reaper
204	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	writeback
205	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kcompactd0
207	root	25	5	0	0	0	S	0,0	0,0	0:00.00	ksmd
208	root	39	19	0	0	0	S	0,0	0,0	0:00.00	khugepaged
209	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	crypto
210	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kintegrityd
212	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kblockd
320	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	md

Tarea 3: Ejercicio: trabajar con el comando corn

En este ejercicio, creará un trabajo cron que creará un archivo de auditoría con ##### para cubrir todos los archivos csv:

Recuerde que cron es un comando que ejecuta una tarea de forma regular a una hora específica. Este comando mantiene la lista de tareas para ejecutar en un archivo crontab. Ejecutaremos un comando que crea el archivo de auditoría con ##### para cubrir todos los archivos .csv. Cuando ingresa el comando crontab -e, se lo lleva a un editor donde luego se ingresará una lista de pasos de lo que ejecutará el daemon corn.

En la imagen de la terminal, tenemos los comandos contenidos dentro del corn.

```
SHELL=/bin/bash
PATH=/usr/bin:/bin:/usr/local/bin
MAILTO=root
0 * * * * ls -la $(find .) | sed -e 's/..csv/#####.csv/g' > /home/ec2-user/companyA/SharedFolders/filteredAudit.csv
-- INSERT --
```

4,116 All

Para validar la creación, ingresaremos sudo crontab -l y presionaremos Enter. Inspeccionaremos el archivo crontab para asegurarnos de que coincida exactamente con el texto, como se muestra en el siguiente resultado:

```
[ec2-user@ip-10-0-10-194 companyA]$ sudo crontab -l ←  
SHELL=/bin/bash  
PATH=/usr/bin:/bin:/usr/local/bin  
MAILTO=root  
0 * * * * ls -la $(find .) | sed -e 's/..csv/####.csv/g' > /home/ec2-user/companyA/SharedFolders/filteredAudit.csv  
[ec2-user@ip-10-0-10-194 companyA]$
```

Fin ;).