

Material estudio Introducción a Cloud Computing (EA2).

Guía de Tareas Programas con cron en Linux.

Comandos:

crontab -e | Edita las tareas del cron para el usuario actual.

crontab -l | Lista las tareas programadas.

crontab -r | Elimina todas las tareas del cron.

sudo crontab -e | Edita las tareas del cron como superusuario.

sudo systemctl status cron | Verifica si el servicio está activo.

sudo systemctl start cron | Inicia el servicio si está detenido

sintaxis de cron: * * * * * comando a ejecutar

Cada “*” Significa una unidad de tiempo en particular.

El primer * Se utiliza para los minutos en un rango de 0-59.

El segundo es para la hora en un rango de 0-23.

El tercero es para los días del mes en un rango de 1-31.

El cuarto es para el mes en sí y va en un rango de 1-12.

Y el último es para los días de la semana y está en un rango entre 0,7, contando el 0 y 7 como día domingo.

Si quisiéramos ejecutar un script todos los días a las 23:00 hrs, seria:

0(0 minutos) 23(23 hrs) * * * [Ruta del script]

Si se deja el * significa que queremos todos los del rango, en este caso todos los días de todos los meses y todos los días de la semana.

Actividad principal : Registrar la hora cada 2 minutos.

1. Verificamos que tengamos nuestro gestor de paquetes actualizado (yum) e instalamos cron:

```
sudo yum update
```

```
sudo yum install cronie
```

Nota: yum es nuestro gestor de paquetes y de momento solo cronie me ha funcionado para realizar tareas programadas.

2. Crear directorio:

```
mkdir -p ~/cron_test
```

Nota: El “~” significa que tomará la ruta del directorio personal (home) por ejemplo si tu usuario se llama Juan , “~” se convierte en /home/juan en este caso ~ se convierte en ec2-user.

3. Creamos el script:

```
echo -e '#!/bin/bash echo "Hora actual:$(date)" >> ~/cron_test/horas.txt' >
~/cron_test/hora.sh
```

Nota: El parámetro -e activa el reconocimiento de caracteres especiales (como \n , \t, etc.) con la primera línea de código estamos diciéndole al sistema que use el intérprete de Bash para ejecutar el script. Luego, estamos imprimiendo la hora y fecha actuales con “>>” estamos redirigiendo la salida del archivo, agregandolo a línea final (como un append.) si usaramos un solo “>” solo estaríamos reescribiendo constantemente el archivo con 1 sola linea. Cómo lo hacemos al final que “>” redirige la salida del echo al archivo hora.sh

Recordemos usar comillas simples que son válidas para bash

4. Revisamos que se haya creado el archivo del script:

```
ls cron_test
```

5. Le otorgamos permisos al archivo creado:

```
chmod +x ~/cron_test/hora.sh
```

Nota: El permiso otorgado en este caso es el “x” que es el de ejecución podría ser el “r” o también el “w” que son para leer y escribir respectivamente.

6. Cambiamos el editor de cron:

```
export EDITOR=nano
export VISUAL =nano
```

Nota: Por defecto tenemos el editor vim , por comodidad se cambia a nano.

7. Agregamos la tarea al script:

```
crontab -e
*/2 * * * * ~/cron_test/hora.sh
```

Nota: Se nos abrirá el editor nano, escribimos la tarea y guardamos con CTRL + O y cerramos con CTRL + X. **El mensaje “no crontab for ec2-user - using an empty one crontab: installing new crontab” no quiere decir nada menos que no había ninguna tarea personalizada para el usuario ec2-user y se agregó una.**

8. Verificamos que se haya guardado correctamente:

```
crontab -l
```

9. Revisamos después de un tiempo.

```
cat ~/cron_test/horas.txt
```

POSIBLES ERRORES:

Si salta el error “cat: /home/ec2-user/cron_test/horas.txt: No such file or directory”, es porque horas.txt aun no se a creado no olvidemos que horas.txt se crea cuando se ejecuta el script asi que puede ser porque cron no esté funcionando correctamente. para esto verificaremos el status de cron:

```
systemctl status crond
```

Si nos aparece Active: inactive(DEAD) significa que cron no está activo en el sistema para activarlo utilizaremos:

```
sudo systemctl start crond  
sudo systemctl enable crond
```

para asegurarnos de que ahora si este funcionando ejecutaremos el script de forma manual con el comando:

```
bash ~/cron_test/hora.sh
```

y revisamos:

```
cat ~/cron_test/horas.txt
```

Tarea Extra: Simulación de reporte diario.

Objetivo: Crear un reporte de todos los días a las 9:00 AM.

1. Como ya tenemos creado un directorio en donde guardar los scripts, lo crearemos directamente.

```
echo -e '#!/bin/bash  
echo "Reporte generado:$(date)" >> ~/cron_test/reporte_diario.txt' > ~/cron_test/reporte.sh
```

2. Otorgamos permisos de ejecución

```
chmod +x ~/cron_test/report.sh
```

3. Abrimos nano y añadimos el comando a utilizar:

```
crontab -e  
0 9 * * * ~/cron_test/reporte.sh
```

4. Verificar después de las 9:00 AM:

```
crontab -l  
bash ~/cron_test/reporte.sh  
cat ~/cron_test/reporte_diario.txt
```

Nota: Para saltarnos el tener que esperar ejecutaremos el script como lo hicimos anteriormente y así podemos verificar que funcionó con crontab -l verificamos si se guardó correctamente dentro del editor.

Tarea extra : Montar un disco virtual utilizando EFS de AWS.

1. Crear EFS.

Para crear el disco virtual vamos a Amazon Elastic file System y le damos click donde dice crear un sistema de archivos.

Crear un sistema de archivos

Cree un sistema de archivos de EFS con la configuración recomendada.

Crear un sistema de archivos

Virtual Private Cloud (VPC)

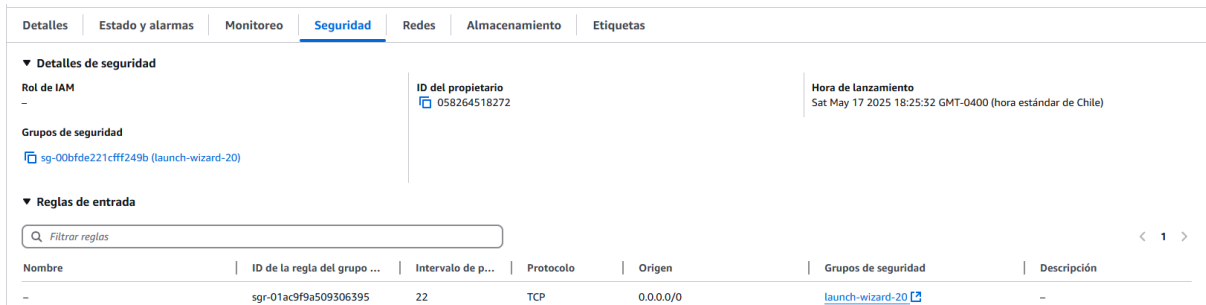
Elija la VPC en la que desea que las instancias EC2 se conecten a su sistema de archivos.

vpc-0210d228738a2b393	predeterminado	▲
vpc-0210d228738a2b393	predeterminado	✓

Nota: Nos debemos fijar que el VPC sea el mismo de nuestra máquina virtual, el tener creada solo una nos aparecerá una lógicamente.

2. Configurar Security Group:

Para este paso vamos a el panel de nuestra instancia previamente creada y pinchamos donde dice /Security/ y hacemos click en launch-wizard.



Luego seleccionaremos el security group con el que fue creada la máquina de EC2 y modificaremos su reglas de entrada.

Añadiremos una regla nueva de tipo NFS y ipv4. (0.0.0.0/o) y guardaremos.

Editar reglas de entrada

Las reglas de entrada controlan el tráfico entrante que puede llegar a la instancia.

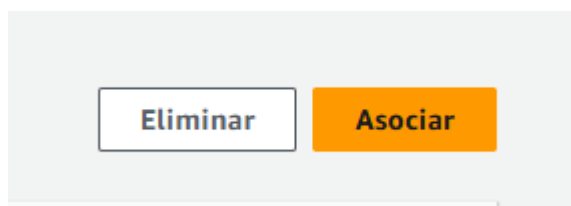
The screenshot shows the 'Editar reglas de entrada' form. It lists existing rules and allows adding a new one. The existing rules are:

ID de la regla del grupo de seguridad	Tipo	Protocolo	Intervalo de puertos	Origen	Descripción: opcional	Acción
sgr-01ac9f9a509306395	SSH	TCP	22	Person...		Eliminar
sgr-05f559c07b01f7dee	NFS	TCP	2049	Person...		Eliminar

At the bottom, there is an 'Agregar regla' button and a 'Previsualizar los cambios' button.

3. Montamos el disco:

Con la instancia previamente creada, vamos al panel de EFS , click en ver sistema de archivos y clickeamos en “Asociar.”



Seleccionamos montar via IP y copiamos el comando que genera.

Asociar

Monte el sistema de archivos de Amazon EFS en una instancia de Linux.
[Más información](#)

☐ Montaje a través de DNS
☒ Montaje a través de IP

Zona de disponibilidad

us-east-1a

Mediante el cliente de NFS:

```

sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport 172.31.21.221:/ efs

```

Consulte nuestra guía del usuario para obtener más información.
[Más información](#)

Cerrar

4. Añadir Security Group al EFS:

Nos volvemos al panel del EFS y le damos click a RED, hay que fijarse en que la Availability Zone sea la misma en la que estamos trabajando y que el security group sea el mismo que modificamos en el punto anterior.

Debe eliminar todos los destinos de montaje existentes para cambiar la VPC del sistema de archivos.

Destinos de montaje

Un destino de montaje proporciona un punto de enlace NFSv4 en el que puede montar un sistema de archivos de Amazon EFS. Le recomendamos que cree un destino de montaje por zona de disponibilidad. [Más información](#)

Zona de disponibilidad	ID de la subred	Dirección IP	Grupos de seguridad	
us-east-1a	subnet-04e9bd57258533dec	172.31.21.221	<div>Elegir grupos de seguridad</div> <div>sg-00bfde221cfff249b ✕</div> <div>launch-wizard-20</div>	Eliminar
us-east-1b	subnet-0861e92656b7aaed2	172.31.38.194	<div>Elegir grupos de seguridad</div> <div>sg-00bfde221cfff249b ✕</div> <div>launch-wizard-20</div>	Eliminar
us-east-1c	subnet-01327e950a59b8321	172.31.6.195	<div>Elegir grupos de seguridad</div> <div>sg-00bfde221cfff249b ✕</div> <div>launch-wizard-20</div>	Eliminar
us-east-1d	subnet-022b1795c628d28b9	172.31.90.210	<div>Elegir grupos de seguridad</div> <div>sg-00bfde221cfff249b ✕</div> <div>launch-wizard-20</div>	Eliminar
us-east-1e	subnet-060935b09f41e8e85	172.31.61.229	<div>Elegir grupos de seguridad</div> <div>sg-00bfde221cfff249b ✕</div> <div>launch-wizard-20</div>	Eliminar
us-east-1f	subnet-0ca4a14b6b49fe4e6	172.31.64.25	<div>Elegir grupos de seguridad</div> <div>sg-00bfde221cfff249b ✕</div> <div>launch-wizard-20</div>	Eliminar

Añadir destino de montaje

5. Añadimos el disco por la terminal de la máquina virtual:

Creamos una carpeta como punto de montaje

```
mkdir efs
```

pegamos el mensaje que copiamos anteriormente

```
sudo mount -t nfs4 -o  
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport  
172.31.21.221:/ efs
```

y cambiamos la ruta a la que necesitamos en este caso /home/ec2-user/efs

quedaría así

```
sudo mount -t nfs4 -o  
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport  
172.31.21.221: /home/ec2-user/efs
```

chequeamos con df -h

POSIBLES PROBLEMAS:

si al momento de montar no cargaba nunca y se mantiene una línea para escribir probablemente sea porque el puerto 2049 no estaba abierto y cargaba de manera infinita, eso es porque configuramos mal los grupos de seguridad así que como recomendación puedes cambiar el grupo de seguridad de todas las zonas disponibles en el panel de EFS al grupo en donde pusiste la regla 2049.