



## 253-[LX]-Lab - [Reto] Ejercicio Bash

### Datos Generales:

**Nombre:** Tomás Alfredo Villaseca Constantinescu

**País:** Chile

**Fecha:** 09/09/2023

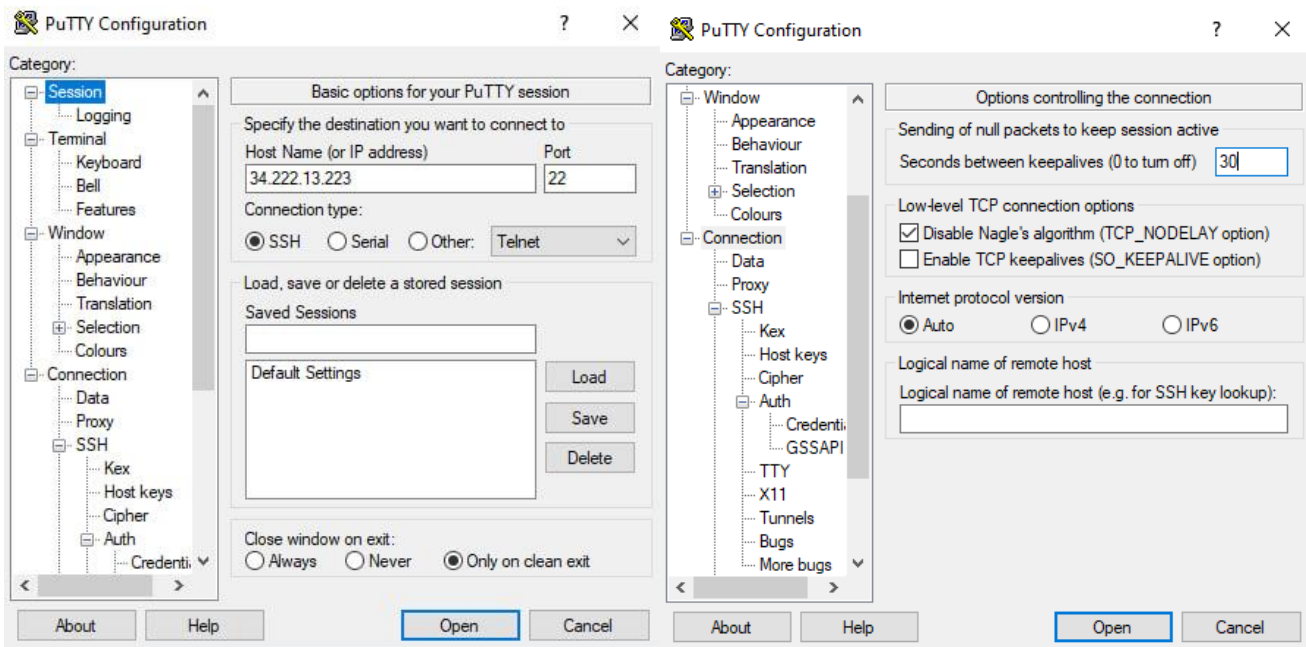
**Contacto:** [tomas.villaseca.c@gmail.com](mailto:tomas.villaseca.c@gmail.com)

En este desafío, hará lo siguiente:

- Crear un directorio utilizando un script de bash.

# Tarea 1: conectarse a una instancia EC2 de Amazon Linux mediante SSH

1. Abrir Putty.exe: Se ingresa dirección IPv4 de la instancia EC2 en la sección Session.
2. En la sección Connection → SSH → Auth → Credentials se ingresa el archivo PPK descargado anteriormente.
3. En la sección Connection se establece **Seconds between keepalive en 30 (el valor predeterminado es 0).**



4. Se hace click en "Open" para validar y conectarse al Host.



# Desafío

Escriba un script bash y siga los siguientes requisitos:

- Cree 25 archivos vacíos (0 KB).
- Los nombres de archivo deben ser **<yourName><number>**, **<yourName><number+1>**, **<yourName><number+2>** y así sucesivamente.
- Diseñe el script de manera que cada vez que lo ejecute, este cree el siguiente lote de 25 archivos con números crecientes comenzando por el último número o número máximo que ya existe.
- No fije la codificación de estos números. Necesita generarlos mediante la automatización.

Un script es un archivo de texto que contiene un conjunto de comandos.

- Para crear un script en bash, simplemente cree un nuevo archivo de texto y escriba los comandos de bash que desea ejecutar. Luego, guarde el archivo con la extensión **.sh**
- **./script.sh** = ejecutar script

**pwd** = muestra el directorio de trabajo actual

**ls** = muestra el contenido de un directorio

**touch** = crear un archivo

ec2-user@ip-10-0-10-219:~

```
[ec2-user@ip-10-0-10-219 ~]$ pwd
/home/ec2-user
[ec2-user@ip-10-0-10-219 ~]$ touch script.sh
[ec2-user@ip-10-0-10-219 ~]$ ls
companyA  script.sh
[ec2-user@ip-10-0-10-219 ~]$
```

**chmod** = cambia los permisos de un archivo o directorio

ec2-user@ip-10-0-10-219:~

```
[ec2-user@ip-10-0-10-219 ~]$ sudo chmod 755 script.sh
[ec2-user@ip-10-0-10-219 ~]$ ls -l
total 4
drwxr-xr-x 11 ec2-user Personnel 184 Sep 10 01:19 companyA
-rwxr-xr-x  1 ec2-user ec2-user   767 Sep 10 01:21 script.sh
[ec2-user@ip-10-0-10-219 ~]$
```

Editar script.sh utilizando nano:

```
[ec2-user@ip-10-0-10-219 ~]$ nano script.sh
```

```
ec2-user@ip-10-0-10-219:~  
GNU nano 2.9.8 script.sh  
#!/bin/bash  
  
# Obtener nombre de usuario  
username=tomasvc  
  
# Obtener el número máximo de archivos existentes.  
# ls -l para listar los archivos en el directorio actual uno por línea.  
# grep -E para buscar un patrón que coincide con cualquier cadena de texto que comience con username seguido de cualquier número de dígitos.  
# cut -c 8= para cortar extraer solamente los valores numéricos del output anterior.  
# sort -n para ordenar por valor numérico.  
# tail -n 1 para mostrar la ultima línea del output anterior, es decir, el numero mas grande.  
  
max_number=$(ls -l | grep -E "${username}[0-9]+" | cut -c 8- | sort -n | tail -1)  
  
# Bucle para crear los archivos  
for i in {1..25}; do  
  
    # Generar el número del archivo  
    number=$(expr $max_number + $i)  
  
    # Generar un nombre de archivo único  
    filename="${username}${number}"  
  
    # Crear el archivo  
    touch "$filename"  
  
done
```

Se define la variable username=tomasvc

Se define variable max-number:

```
max_number=$(ls -l | grep -E "${username}[0-9]+" | cut -c 8- | sort -n | tail -1)
```

### Explicación de variable max\_number para obtener el número máximo de los archivos existentes:

Paso 1: **ls -l** lista los archivos y directorios en el directorio actual por línea.

Paso 2: **grep -E "\${username}[0-9]+"** permite filtrar por un patrón que coincide con un texto que comience con la variable **username** (tomasvc) seguido de cualquier número de dígitos (tomasvc1, tomasvc2, etc.).

Paso 3: **cut -c 8-** permite extraer solamente los caracteres desde la posición 8 en adelante, es decir, solamente los valores numéricos (tomasvc25 → 25).

Paso 4: **sort -n** permite ordenar el output anterior en orden número ascendente (1, 2, 3, etc.)

Paso 5: **tail -n 1** muestra solamente la última línea del output anterior, es decir, el número más grande.

El output final de esta cadena de comandos es el número máximo de los archivos existentes.

## Explicación del bucle for:

```
for i in {1..25}; do

    # Generar el número del archivo
    number=$(expr $max_number + $i)

    # Generar un nombre de archivo único
    filename="${username}${number}"

    # Crear el archivo
    touch "$filename"

done
```

Para la variable “i” en una secuencia de números desde el 1 al 25, realizar:

Paso 1: Se define la variable **number** que genera el número siguiente a utilizar para el primer nuevo archivo vacío (si el número máximo era el 25, el siguiente número a utilizar será el 25+1=26).

Paso 2: Se define variable **filename** indicando que debe tener formato "\${username}\${number}" de nombre.

Paso 3: Se crea un archivo con el comando **touch** utilizando las variables anteriormente definidas, entregando un nuevo archivo de nombre único.

El bucle itera hasta terminar la secuencia de números preestablecida desde el 1 al 25, generando 25 nuevos archivos vacíos con nombre único y creciente en la parte numérica.

Primera ejecución del script:

```
ec2-user@ip-10-0-10-219:~  
[ec2-user@ip-10-0-10-219 ~]$ ./script.sh  
[ec2-user@ip-10-0-10-219 ~]$ ls  
companyA  tomasvc1  tomasvc11  tomasvc13  tomasvc15  tomasvc17  tomasvc19  tomasvc20  tomasvc22  tomasvc24  tomasvc3  tomasvc5  tomasvc7  tomasvc9  
script.sh tomasvc10 tomasvc12  tomasvc14  tomasvc16  tomasvc18  tomasvc2  tomasvc21  tomasvc23  tomasvc25  tomasvc4  tomasvc6  tomasvc8  
[ec2-user@ip-10-0-10-219 ~]$
```

Segunda ejecución del script:

```
ec2-user@ip-10-0-10-219:~  
[ec2-user@ip-10-0-10-219 ~]$ ./script.sh  
[ec2-user@ip-10-0-10-219 ~]$ ls  
companyA  tomasvc1  tomasvc12  tomasvc15  tomasvc18  tomasvc20  tomasvc23  tomasvc26  tomasvc29  tomasvc31  tomasvc34  tomasvc37  tomasvc4  tomasvc42  tomasvc45  tomasvc48  tomasvc50  tomasvc8  
script.sh tomasvc10 tomasvc13  tomasvc16  tomasvc19  tomasvc21  tomasvc24  tomasvc27  tomasvc3  tomasvc32  tomasvc35  tomasvc38  tomasvc40  tomasvc43  tomasvc46  tomasvc49  tomasvc6  tomasvc9  
tomasvc  tomasvc11 tomasvc14  tomasvc17  tomasvc2  tomasvc22  tomasvc25  tomasvc28  tomasvc30  tomasvc33  tomasvc36  tomasvc39  tomasvc41  tomasvc44  tomasvc47  tomasvc5  tomasvc7  
[ec2-user@ip-10-0-10-219 ~]$
```

Se verifica que el script funciona correctamente.

Laboratorio Completado

