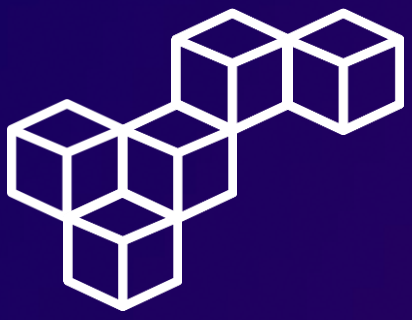
AWS re:Start LAB

Bash Shell Scripting

WEEK 2

aws re/start



Overview

Shell scripting empowers users on Unix-based systems to automate tasks. Imagine a series of commands you perform frequently; a shell script lets you bundle them into a single file. Executing this script saves time and eliminates errors from repetitive tasks. Beyond automation, shell scripting tackles complex operations that would be cumbersome manually.

The versatility of shell scripting shines in various areas. Scripts can manipulate files and directories, process text efficiently, and even manage system administration tasks. Want to write simple programs? Shell scripting offers that capability too. For anyone working with Unix systems, mastering shell scripting unlocks a world of automation, efficiency, and streamlined workflows.

Note: This lab was made using Windows Subsystem for Linux.

Topics covered

- Create a bash script that will automate the backup of a folder

Task 1

Use SSH to connect to an Amazon Linux EC2 instance

Initial Preparations

In the AWS Management Console, select the EC2 instance and make note of the **Public IPv4 address**.

Download the **private key file** labsuser.pem. Change to the Downloads directory and modify the permissions on the key to be read-only (r--------).

Connect to the instance using SSH

Establish a connection to the EC2 instance using the ssh command, the key and the instance's public IPv4 address.

```
support@HP-Pavilion-Laptop:~/Downloads$ ssh -i labsuser.pem ec2-user@35.93.116.4
The authenticity of host '35.93.116.4 (35.93.116.4)' can't be established.
ED25519 key fingerprint is SHA256:KdKdUCOF9zzFgiWid/PlnmXrP08srRrzmhePH+PJ9fA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '35.93.116.4' (ED25519) to the list of known hosts.
       #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-10-106 ~]$
```

aws re/start

Task 2

Write a shell script

Step 1: Create a generic shell script

Create a generic shell script called **backup.sh**. Then, change the file privileges to make **backup.sh** be executable, and use your preferred text editor to open the file for editing.

```
[ec2-user@ip-10-0-10-106 ~]$ pwd
/home/ec2-user
[ec2-user@ip-10-0-10-106 ~]$ touch backup.sh
[ec2-user@ip-10-0-10-106 ~]$ sudo chmod 755 backup.sh
[ec2-user@ip-10-0-10-106 ~]$ vi backup.sh
[ec2-user@ip-10-0-10-106 ~]$
```

Step 2: Write the shell script

Write a Bash shell script that automates the creation of a compressed archive backup for the **CompanyA** folder. Then, save your script and exit from the editor.

```
#!/bin/bash

DAY="$(date +%Y_%m_%d)"
BACKUP="/home/$USER/backups/$DAY-backup-CompanyA.tar.gz"
tar -csvpzf $BACKUP /home/$USER/CompanyA
~
~
~
~
~
-- INSERT --                                          4,40          All
```

aws re/start

Task 2

Write a shell script

Step 3: Execute the shell script

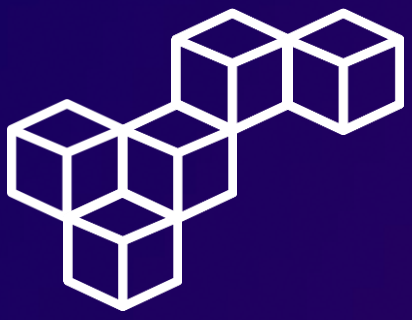
To run the **backup.sh** script, enter ./backup.sh in the terminal.

```
[ec2-user@ip-10-0-10-106 ~]$ ./backup.sh
tar: Removing leading `/' from member names
/home/ec2-user/CompanyA/
/home/ec2-user/CompanyA/Management/
/home/ec2-user/CompanyA/Management/Sections.csv
/home/ec2-user/CompanyA/Management/Promotions.csv
/home/ec2-user/CompanyA/Employees/
/home/ec2-user/CompanyA/Employees/Schedules.csv
/home/ec2-user/CompanyA/Finance/
/home/ec2-user/CompanyA/Finance/Salary.csv
/home/ec2-user/CompanyA/Finance/Hourly.csv
/home/ec2-user/CompanyA/HR/
/home/ec2-user/CompanyA/HR/Managers.csv
/home/ec2-user/CompanyA/HR/Assessments.csv
/home/ec2-user/CompanyA/IA/
/home/ec2-user/CompanyA/SharedFolders/
[ec2-user@ip-10-0-10-106 ~]$
```

Step 4: Review backup creation

To verify that the archive is created in the **backups** folder, enter the ls backups/ command.

```
[ec2-user@ip-10-0-10-106 ~]$ ls backups/
2024_04_10-backup-CompanyA.tar.gz
[ec2-user@ip-10-0-10-106 ~]$
```



Conclusions

Bash Shell Scripting

Bash scripting offers a powerful and versatile way to automate tasks on Unix-like systems. By writing scripts in Bash, users can streamline repetitive tasks, improve efficiency, and ensure consistent execution of commands and processes.

Automating tasks

Automating tasks using Bash scripting allows for the automation of complex processes, such as file manipulation, data processing, system administration tasks, and more. This automation reduces manual errors, saves time, and enables tasks to be executed consistently and reliably.

Scheduling tasks

Scheduling tasks with Bash scripting involves using tools like cron to automate the execution of scripts at specified times or intervals. This capability is invaluable for scheduling routine maintenance, backups, report generation, and other recurring tasks, freeing up human resources and ensuring timely execution of critical operations.

aws re/start

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru