Day 5 Task: Advanced Linux Shell Scripting for DevOps Engineers with User management

Tasks

1. Write a bash script createDirectories.sh that when the script is executed with three given arguments (one is directory name and second is start number of directories and third is the end number of directories) it creates specified number of directories with a dynamic directory name.

Example 1: When the script is executed as

./createDirectories.sh day 1 90
then it creates 90 directories as day1 day2 day3 .... day90

```
echo "Hello I am creating 90 directories with name as Day1,Day2...Day90"

#Input arguments passed
Directoryname=$1
Startday=$2
Endday=$3

for ((i =$Startday; i <=$Endday; i++))
do
        mkdir "$Directoryname-$i"
done

echo "Thanks for creating directories"
#sleep 2

echo " Bye"
```

```
[ec2-user@ip-172-31-80-226 chanllege]$ nano 90days_file.sh
[ec2-user@ip-172-31-80-226 chanllege]$ chmod +x 90days_file.sh
[ec2-user@ip-172-31-80-226 chanllege]$ ls
90days_file.sh  day-14  day-2   day-25  day-30  day-36  day-41  day-47  day-52  day-58  day-63  day-69  day-74  day-8   day-85  day-90
day-1           day-15  day-20  day-26  day-31  day-37  day-42  day-48  day-53  day-59  day-64  day-7   day-75  day-80  day-86
day-10          day-16  day-21  day-27  day-32  day-38  day-43  day-49  day-54  day-6   day-65  day-70  day-76  day-81  day-87
day-11          day-17  day-22  day-28  day-33  day-39  day-44  day-5   day-55  day-60  day-66  day-71  day-77  day-82  day-88
day-12          day-18  day-23  day-29  day-34  day-4   day-45  day-50  day-56  day-61  day-67  day-72  day-78  day-83  day-89
day-13          day-19  day-24  day-3   day-35  day-40  day-46  day-51  day-57  day-62  day-68  day-73  day-79  day-84  day-9
```

Example 2: When the script is executed as ./createDirectories.sh Movie 20 50 then it creates 50 directories as Movie20 Movie21 Movie23 ...Movie50

```
echo "Hello I am creating 50  directories with name as Movie1,Movie2...Moive50"

#Input arguments passed
Directoryname=$1
Startday=$2
Endday=$3

for ((i =$Startday; i <=$Endday; i++))
do
        mkdir "$Directoryname-$i"
done

echo "Thanks for creating directories"
#sleep 2

echo " Bye"
```

```
[ec2-user@ip-172-31-80-226 movies]$ nano movie50_.sh
[ec2-user@ip-172-31-80-226 movies]$ chmod +x movie50_.sh
[ec2-user@ip-172-31-80-226 movies]$ ./movie50_.sh movie 20 50
Hello I am creating 50  directories with name as Movie1,Movie2...Moive50
Thanks for creating directories
 Bye
```

```
[ec2-user@ip-172-31-80-226 movies]$ ls
movie-20  movie-25  movie-30  movie-35  movie-40  movie-45  movie-50
movie-21  movie-26  movie-31  movie-36  movie-41  movie-46  movie50_.sh
movie-22  movie-27  movie-32  movie-37  movie-42  movie-47
movie-23  movie-28  movie-33  movie-38  movie-43  movie-48
movie-24  movie-29  movie-34  movie-39  movie-44  movie-49
[ec2-user@ip-172-31-80-226 movies]$
```

2. Create a Script to backup all your work done till now.

   Step 1 : Create a file with .sh using nano command and put below inputs.

```
#!/bin/bash

#we are setting current date and time as the backuo suffix

suffix=$(date +%Y%m%d%H%M%S)

#setting a backup directory

backup_dir="backup_$suffix"

# creating a backup directory

mkdir $backup_dir

#to copy all the files in the current directory to the backup directory

cp * $backup_dir

# print success full message

echo " Done , successfully created backup in $backup_dir"
```

Step 2: Give execute permission to the file using chmod command

Step 3 : Now, run the file using ./ command

```
[ec2-user@ip-172-31-80-226 chanllege]$ nano backup.sh
[ec2-user@ip-172-31-80-226 chanllege]$ chmod +x backup.sh
[ec2-user@ip-172-31-80-226 chanllege]$ ./backup.sh
cp: omitting directory 'backup_20230109153526'
cp: omitting directory 'day-1'
cp: omitting directory 'day-10'
cp: omitting directory 'day-11'
cp: omitting directory 'day-12'
cp: omitting directory 'day-13'
```

```
cp: omitting directory 'day-85'
cp: omitting directory 'day-86'
cp: omitting directory 'day-87'
cp: omitting directory 'day-88'
cp: omitting directory 'day-89'
cp: omitting directory 'day-9'
cp: omitting directory 'day-90'
cp: omitting directory 'movies'
 Done , successfully created backup in backup_20230109153526
[ec2-user@ip-172-31-80-226 chanllege]$
```

Done, Successfully created backup in backup_20230109153526 directory.

3. **Read About Cron and Crontab, to automate the backup Script**

Cron is the system's main scheduler for running jobs or tasks unattended. Cron is a time-based scheduling daemon that runs on Unix and Linux systems. It is used to schedule commands or scripts to run automatically at a certain time or date.

Crontabe : To edit, submit or delete in cron a command is used called as crontab which allows user to do. A crontab file is a user file that holds the scheduling information. You can use crontab to run any of your scripts or one-liners every hour, or maybe just three days a week, it’s up to you. Each user can have their own crontab file if they wish, but on large systems root usually disallows this and just uses one main crontab file for the whole system. Root is able to do this because of a file called ‘cron.deny’ and ‘cron.allow’ where root can specify who can and cannot have their own crontab.

How to use crontab to schedule a script at a certain time or date.

1. Open the crontab configuration file by running a command crontab -e
2. Add a new line to the file with the schedule and command you want to run
3. Save and exit the file Cron will automatically run the command or script at the specified time according to the configuration in the crontab file.
4. To check list of current crontab configuration use command crontab -l.

4. **Read about User Management and Let me know on Linkedin if you're ready for Day 6.**

As Linux is a multi-user operating system, there is a high need of an administrator, who can manage user accounts, their rights, and the overall system security for User management in Linux. You should know the basics of Linux admin so that you can handle the user accounts and usergroups for user management in Linux such as command line tools like useradd, usermod, gpasswd, passwd and others. Server administrators are likely to use these tools, since they are familiar and very similar across many different distributions. This chapter will focus on these command line

Statements	Commands
To list out all the users in Linux	awk -F':' '{print $1}' /etc/passwd
To get the ID of any username	id username
To add a new user	sudo useradd username
To assign a password to a user	passwd username
To change the user ID for a user	usermod -u new_id username
To modify the group id of a user	usermod -g new_group_id username
To change the user login name	sudo usermod -1 new_login_name old_login_name
To change the home directory	usermod -d new_home_directory_path username
To delete a user name	userdel -r username

5. **Create 2 users and just display their Username**

   Step 1 : Use sudo useradd command to create a 2 usernames

   Step 2 : Use awk command to check username list

```
[ec2-user@ip-172-31-80-226 ~]$ sudo useradd cloudengineer
[ec2-user@ip-172-31-80-226 ~]$ sudo useradd devopsengineer
[ec2-user@ip-172-31-80-226 ~]$ awk -F: '{print $1}' /etc/passwd
```

```
cloudengineer
devopsengineer
[ec2-user@ip-172-31-80-226 ~]$
```