# #DAY 5: Advanced Linux Shell Scripting for DevOps Engineers with User management

Topics will cover: -

1. Loops in shell scripting
2. Cron and Crontab
3. Automate the backup script
4. User management

1. LOOPS IN SHELL SCRIPTING: -

Depending on the use case and the problem it is trying to automate, there are a couple of ways to use loops.

* Simple for loop:-

#!/bin/bash  
  
for n in a b c;  
do  
 echo $n  
done

* Range-based for loop:-

#!/bin/bash  
  
for n in {1..5};   
do  
 echo $n  
done

#!/bin/bash  
  
for n in {1..5..2};   
do  
 echo $n  
done

* Array iteration for loops

#!/bin/bash  
  
s=("football" "cricket" "hockey")   
for n in ${s[@]};   
do  
 echo $n  
done

* C-Styled for loops

#!/bin/bash  
  
n=7  
for (( i=1 ; i<=$n ; i++ ));   
do  
 echo $i  
done

* Infinite for loop

#!/bin/bash  
  
n=4  
for (( ; ; ));   
do  
 if [ $n -eq 9 ];then  
 break  
 fi  
 echo $n  
 ((n=n+1))  
done

TASKS: -

1. You have to do the same using Shell Script i.e using either Loops or command with start day and end day variables using arguments -So Write a bash script create directories.sh that when the script is executed with three given arguments (one is the directory name and second is start number of directories and third is the end number of directories ) it creates a specified number of directories with a dynamic directory name.

ANS. ./createDirectories.sh day 1 90

#!/bin/bash

day="$1"

for ((i=$2 ; i<=$3; i++))

do

mkdir "${day}${i}"

done

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

ans. #!/bin/bash

src\_dir="/home/vagrant/scripts"

des\_dir="/home/vagrant/backups"

curr\_time=$(date "+%Y-%m-%d-%H-%M-%S")

backup\_file="$des\_dir/$curr\_time.tgz"

tar czf "$backup\_file" "$src\_dir"

echo "Backup complete"

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

Ans. Cron is the system's main scheduler for running jobs or tasks unattended. A command called crontab allows the user to submit, edit or delete entries to cron. A crontab file is a user file that holds the scheduling information.

1. Read about User Management

Ans. A user is an entity, in a Linux operating system, that can manipulate files and perform several other operations. Each user is assigned an ID that is unique for each user in the operating system. In this post, we will learn about users and commands which are used to get information about the users. After installation of the operating system, the ID 0 is assigned to the root user and the IDs 1 to 999 (both inclusive) are assigned to the system users and hence the ids for local user begins from 1000 onwards.

# SHELL SCRIPTING 3: -

1. Free: - check free space
2. Top: -process using
3. Df –H : - disk files with space
4. Df-h | awk ‘{print $1}’:- will print 1 column
5. Df –h | awk ‘{print $1 “ ” $5}’ :- will show column 1 and column 5
6. Cut: to cut something from content
7. #!/bin/bash

df -H | awk '{print $5 " " $1}' | while read output;

do

echo "disk detail: output"

done

1. Script to get alert for space utilization:-

#!/bin/bash

df -H | awk '{print $5 " " $1}' | while read output;

do

#echo "disk detail: output"

usage=$(echo $output | awk '{print $1}' | cut -d'%' -f1)

file\_sys=$(echo $output | awk '{print $2}')

# echo $usage

if [ $usage -le 90 ]

then

echo "CRITICAL for $file\_sys"

fi

done

1. Crontab: - crontab –l
2. Crontab –e :- to edit or create crontab
3. Crontab to run the check\_disk.sh

\* \* \* \* \* bash /home/vagrant/check\_disk.sh >> /home/vagrant/log.txt

#!/bin/bash

alert=90

backup\_date=$(date +'%m/%d/%Y %H:%M:%S')

df -H | awk '{print $5 " " $1}' | while read output;

do

#echo "disk detail: output"

usage=$(echo $output | awk '{print $1}' | cut -d'%' -f1)

file\_sys=$(echo $output | awk '{print $2}')

# echo $usage

if [ $usage -le $alert ]

then

echo "CRITICAL for $file\_sys on $backup\_date"

fi

done

1. Add\_user.sh

#!/bin/bash

add\_user()

{

USER=$1

PASS=$2

useradd -m -p $PASS $USER && echo "succesfullly added user"

}

#MAIN

add\_user raviraj 1234