Exercise 2:

1. Write Script using php
   1. Telnet to server
      1. <?PHP $output = **shell\_exec**(“ telnet localhost  ”); echo $output; ?>
   2. Ssh to server
      1. <?PHP $output = **shell\_exec**(“ssh -i private\_key.txt demo@<serverIP>”); ?>
   3. Check the disk usage
      1. <?PHP $output = **shell\_exec**(“df -kh”); ?>
   4. Inode usage
      1. <?PHP $output = **shell\_exec**(“df -i”); ?>
   5. Get the list of files from path
      1. <?PHP $output = **shell\_exec**(“ls –ltrh /dirname/”); ?>
   6. Copy files to the remote server
      1. <?PHP $output = **shell\_exec**(“scp –rp <filename> user@<serverip>:<dir location>”) ?>
2. Telnet to server, create n number of files (accept user input), zip all created files , then download the files & extract it
   1. Command: telnet <serverIp>
   2. Create n number files, zip all  
        
      #!/bin/bash

read -p "How many files you are willing to create : " fileInput

a=0

# -lt is less than operator

#create directory based on time

backup\_dir=$(date +'%s')

mkdir "dir\_${backup\_dir}"

#Iterate the loop until a less than fileInput

while [ $a -lt $fileInput ]

do

# Print the values

touch "dir\_${backup\_dir}"/randomTxtFile\_$a.txt

tar -czvf "dir\_${backup\_dir}".tar.gz "dir\_${backup\_dir}"

# increment the value

a=`expr $a + 1`

done

1. Restart apache in case of high load
   1. Scrip to check high spee

for I in 0 1 2 3 4 5; do

check=$(uptime | tr -d ',.' | awk '{print $10}')

if [ "$check" -gt 5 ]; then

/usr/bin/systemctl restart httpd.service

fi

sleep 10

done

* 1. Cron job

\* \* \* \* \* /PATH/TO/YOUR/SCRIPT

1. Monitor server performance
   1. Command to observer or troubleshoot
      1. # top
   2. Identify which process are consuming more time
      1. # ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head
   3. Minimize the server load
      1. There are too many ways to do this we don’t have any specific command
2. Create view from query
   1. Don’t have query