Linux Commands for Professional Environment

Generic Linux Support Commands

Practical Examples (Basic & Advance)

CHOWN, CHMOD, SSH, GREP, TAIL, HEAD, TOP, DF, DU, KILL, SCP

DigiTalk

https://digitalksystems.com/

Reach us at digitalk.fmw@gmail.com
DigiTalk Channel: https://www.youtube.com/channel/UCCGTnI9vvF_ETMhGUXGdFWw
Playlists: https://www.youtube.com/@digitalk.middleware/playlists
Weblogic Server Architecture: https://youtu.be/gNqeIfLjUqw

DigiTalk Udemy Courses and Coupon Code

SOA Suite Administration
https://www.udemy.com/course/mastering-oracle-soa-suite-12c-administration/?couponCode=739A60915F86847014EB
Coupon Code: 739A60915F86847014EB

JBoss 8 Administration
https://www.udemy.com/course/mastering-jboss-eap-8-administration-from-intro-to-advanced/?couponCode=BF65EB008CFE16686BD2
Coupon Code:BF65EB008CFE16686BD2

OHS Administration
https://www.udemy.com/course/mastering-oracle-ohs-http-12c-web-server-administration/?couponCode=8E990556B21AF3E1A316
Coupon Code: 8E990556B21AF3E1A316

Weblogic Server Administration
https://www.udemy.com/course/oracle-weblogic-server-12c-and-14c-administration/?couponCode=87BC1314AC7690FD5294
Coupon Code:87BC1314AC7690FD5294

You can write us on digitalk.fmw@gmail.com if coupon code expired.

chown Command

The chown command is used to change the ownership of files and directories.

Syntax:

chown [options] OWNER[:GROUP] FILE

Common Options:

- -R : Recursively change ownership of directories and their contents.
- -v : Verbosely show the files whose ownership is being changed.

Examples:

Change the owner of a file:

chown user1 file.txt

Change the owner and group of a file:

chown user1:group1 file.txt

Recursively change the owner of a directory and its contents:

chown -R user1:group1 /path/to/directory

Advanced usage with find to change ownership of specific files:

find /path/to/directory -type f -name "*.txt" -exec chown user1:group1 {} \;

chmod Command

The chmod command is used to change the permissions of files and directories.

Syntax:

chmod [options] MODE FILE

Common Options:

- -R : Recursively change permissions of directories and their contents.
- -v : Verbosely show the files whose permissions are being changed.

Examples:

Change permissions to read, write, and execute for the owner:

chmod u+rwx file.txt

Change permissions to read and execute for everyone:

chmod a+rx file.txt

Recursively change permissions of a directory and its contents:

chmod -R 755 /path/to/directory

Advanced usage with find to change permissions of specific files:

find /path/to/directory -type f -name "*.sh" -exec chmod +x {} \;

ssh Command

The ssh command is used to securely connect to a remote machine.

Syntax:

ssh [options] USER@HOST

Common Options:

- -p PORT : Specifies the port to connect to on the remote host.
- -i IDENTITY_FILE : Specifies the file from which the identity (private key) is read.
- -L : Specifies that connections to the given port on the local (client) host are to be forwarded to the given host and port on the remote side.

Examples:

Connect to a remote machine:

ssh user@remotehost

Connect to a remote machine on a specific port:

ssh -p 2222 user@remotehost

Use a specific private key for authentication:

ssh -i /path/to/private_key user@remotehost

Usage to tunnel traffic:

ssh -L 8080:localhost:80 user@remotehost

Above command forwards local port 8080 to port 80 on remote_server.

Advance Commands

Execute a Command on a Remote Server

ssh user@remote_server "ls -lah /var/log"

Above command logs into remote_server as user and lists the files in the /var/log directory in long format, including hidden files.

Execute a Script on Multiple Servers

```
for server in server1 server2 server3; do
  ssh user@$server "bash -s" < /path/to/local_script.sh
done
```

This command runs local_script.sh on server1, server2, and server3.

Using SSH Key Authentication

```
ssh -i /path/to/private_key user@remote_server
```

This command logs into remote_server using the specified private key for authentication.

Compress and Transfer Files in One Command

```
tar czf - /path/to/local_directory | ssh user@remote_server "tar xzf - -C /path/to/remote_directory"
```

This command compresses local_directory and extracts it on remote_server in a single step.

Running Background Commands on Remote Server

```
ssh user@remote_server "nohup /path/to/command > /dev/null 2>&1 &"
```

This command runs command in the background on remote_server and redirects its output to /dev/null.

Monitoring Remote Logs in Real-Time

```
ssh user@remote_server "tail -f /var/log/syslog"
```

This command continuously monitors the syslog on remote_server.

SSH with Specific Port

```
ssh -p 2222 user@remote_server
```

This command connects to remote_server using port 2222 instead of the default port 22.

Running Commands with sudo on a Remote Server

```
ssh user@remote_server "sudo systemctl restart apache2"
```

This command restarts the Apache service on remote_server using sudo.

Tunneling a Remote Desktop Session

```
ssh -L 5901:localhost:5901 user@remote_server
```

This command creates a secure tunnel for a VNC session running on port 5901 on remote_server.

4. grep Command

The grep command is used to search for patterns in files.

Syntax:

grep [options] PATTERN [FILE...]

Common Options:

- -i : Ignore case distinctions.
- -r : Recursively search directories.
- -n : Show line numbers with output.
- -v : Invert the match, showing lines that do not match the pattern.

Examples:

Search for a pattern in a file:

grep "pattern" file.txt

Search for a pattern, ignoring case:

grep -i "pattern" file.txt

Recursively search for a pattern in a directory:

grep -r "pattern" /path/to/directory

Advanced usage with ps to filter processes:

ps aux | grep "ssh"

Advance Commands

Search and Replace String in All Files in a Directory

find /path/to/dir -type f -exec sed -i 's/old_string/new_string/g' {} +

Above command finds all files in /path/to/dir and replaces old_string with new_string in each file.

Search for a String in All Files and List Matching Files

grep -rl "search_string" /path/to/dir

This command recursively searches for search_string in all files under /path/to/dir and lists the matching files.

Count Occurrences of a String in All Files in a Directory

```
grep -ro "search_string" /path/to/dir | wc –l
```

This command counts how many times search_string appears in all files under /path/to/dir.

Find and Compress Files Modified in the Last 7 Days

```
find /path/to/dir -type f -mtime -7 -exec tar -rvf recent_files.tar {} +
```

This command finds all files modified in the last 7 days under /path/to/dir and adds them to a tar archive.

Find and Delete Empty Files and Directories

```
find /path/to/dir -type f -empty -delete
```

```
find /path/to/dir -type d -empty -delete
```

These commands find and delete empty files and directories under /path/to/dir.

Search for Files with Specific Extension and Count Lines

```
find /path/to/dir -name "*.txt" -exec wc -l {} +
```

This command finds all .txt files under /path/to/dir and counts the lines in each file.

Find and Change Permissions of All Files with Specific Extension

```
find /path/to/dir -name "*.sh" -exec chmod +x {} +
```

This command finds all .sh files under /path/to/dir and makes them executable.

Search for a String in Files and Display Surrounding Lines

```
grep -rnC 3 "search_string" /path/to/dir
```

This command searches for search_string in all files under /path/to/dir and displays 3 lines of context around each match.

Monitor Log Files in Real-Time for Specific Strings

```
tail -f /var/log/syslog | grep --line-buffered "error_string"
```

This command monitors the /var/log/syslog file in real-time and filters lines containing error_string.

Backup Configuration Files Before Editing

```
find /etc -name "*.conf" -exec cp {} {}.bak \;
```

This command finds all .conf files in /etc and creates a backup of each file with a .bak extension.

Find and Replace Strings in Files with Confirmation

```
find /path/to/dir -type f -exec sed -i 's/old_string/new_string/g' {} +
```

This command finds all files in /path/to/dir and replaces old_string with new_string in each file with confirmation for each replacement.

Find Files Larger Than a Certain Size and List Details

```
find /path/to/dir -type f -size +100M -exec ls -lh {} +
```

This command finds all files larger than 100MB under /path/to/dir and lists their details.

Combining find with xargs for Performance Improvement

```
find /path/to/dir -type f -print0 | xargs -0 grep -l "search_string"
```

This command finds all files in /path/to/dir and uses xargs to search for search_string more efficiently.

Automatically Delete Files Older Than 30 Days

```
find /path/to/dir -type f -mtime +30 -exec rm {} +
```

This command finds and deletes files older than 30 days under /path/to/dir.

Find and Replace Strings in Specific File Types Only

```
find /path/to/dir -name "*.html" -exec sed -i 's/old_string/new_string/g' {} +
```

This command finds all .html files under /path/to/dir and replaces old_string with new_string.

Using find with exec for Complex Tasks

```
find /path/to/dir -type f -exec sh -c 'grep -q "search_string" "{}" && echo "{}"' \;
```

This command finds all files under /path/to/dir and prints the file names if they contain search_string.

5. tail Command

The tail command is used to display the end of a file.

Syntax:

```
tail [options] [FILE]
```

Common Options:

- -n NUM : Show the last NUM lines.
- -f : Follow the file as it grows.

Examples:

Show the last 10 lines of a file:

```
tail file.txt
```

Show the last 20 lines of a file:

tail -n 20 file.txt

Follow a file as it grows (useful for logs):

tail -f /var/log/syslog

Advanced usage with grep to monitor specific logs:

tail -f /var/log/syslog | grep "error"

6. head Command

The head command is used to display the beginning of a file.

Syntax:

head [options] [FILE]

Common Options:

- -n NUM : Show the first NUM lines.

Examples:

Show the first 10 lines of a file:

head file.txt

Show the first 20 lines of a file:

head -n 20 file.txt

Advanced usage with cat to display specific sections:

cat file.txt | head -n 50 | tail -n 10

7. top Command

The top command is used to display a dynamic view of system processes.

Syntax:

top [options]

Common Options:

- -u USER : Show processes for a specific user.
- -p PID : Show a specific process by PID.
- -n NUM : Update the display NUM times and then exit.

Examples:

Display system processes:

top

Show processes for a specific user:

top -u user1

Advanced usage with grep to filter output:

top -n 1 -b | grep "python"

8. df Command

The df command is used to display disk space usage.

Syntax:

df [options]

Common Options:

- -h : Show sizes in human-readable format.
- -T : Show file system type.
- -i : Show inode usage.

Examples:

Display disk space usage:

df

Show sizes in human-readable format:

df -h

Show file system type:

df -T

Advanced usage to monitor specific mounts:

df -h | grep "/dev/sda1"

Advance Commands

df -h | awk '{print $1, $5}'

This command shows only the file system and the percentage of disk space used.

```
df -h | sort -k 5 –h
```

Above command sorts the file systems by their disk usage percentage in ascending order.

```
df -h | sed 's/%/ percent/'
```

Above command replaces the % sign with the word "percent" in the output.

```
while true; do df -h | grep '/dev/sda'; sleep 10; done
```

Above command continuously monitors the disk usage of /dev/sda every 10 seconds.

```
*/30 * * * * df -h | mail -s "Disk Usage Report" user@example.com
```

Above cron job sends an email with the disk usage report every 30 minutes.

```
df -h | grep '/dev/sda' | xargs -I {} echo "Disk Usage: {}"
```

Above command adds a custom message to the disk usage output for /dev/sda.

```
find /home -type d -exec df -h {} \;
```

Above command finds all directories under /home and shows their disk usage.

```
df -h | tee -a /var/log/disk_usage.log
```

Above command appends the disk usage output to a log file.

```
df -h | cut -d ' ' -f 1,5
```

Above command shows only the file system and the usage percentage.

```
df -h | perl -lane 'print "$F[0]: $F[4]" if $F[4] =~ /%/'
```

Above command uses Perl to extract and print the file system and usage percentage.

```
df -h | grep '/dev/sda' | awk '{if($5+0 > 80) system("notify-send \"Disk Usage Alert: "$5" on "$1"\"")}'
```

Above command sends a desktop notification if the disk usage exceeds 80%.

```
watch -n 30 'df -h | grep "/dev/sda"'
```

Above command displays the disk usage of /dev/sda in real-time, updating every 30 seconds.

```
df -h / | tail -1 | awk '{print $5}' | xargs -I {} ps aux --sort=-%mem | head -10
```

Above command finds the disk usage of the root file system and shows the top 10 memory-consuming processes.

```
#!/bin/bash
trap 'echo "Disk usage script interrupted"; exit' INT
while true; do
  df -h | grep '/dev/sda' | awk '{if($5+0 > 90) print "Alert: Disk usage at "$5" on "$1}'
  sleep 60
done
```

Above script continuously monitors the disk usage of /dev/sda and prints an alert if usage exceeds 90%.

9. du Command

The du command is used to display disk usage of files and directories.

Syntax:

du [options] [FILE...]

Common Options:

- -h : Show sizes in human-readable format.
- -s : Display a summary for each argument.
- -c : Produce a grand total.

Examples:

Display disk usage of a directory:

```
du /path/to/directory
```

Show sizes in human-readable format:

```
du -h /path/to/directory
```

Display a summary for each argument:

```
du -sh /path/to/directory
```

Advanced usage with sort to find largest directories:

```
du -h /path/to/directory | sort -hr | head -n 10
```

Advance Commands

Display Disk Usage of Specific File Types

```
find /home/user/documents -name '*.pdf' -exec du -ch {} + | grep total$
```

Above command finds all .pdf files within the /home/user/documents directory and displays their total disk usage.

Check Disk Usage of Multiple Directories

```
du -ch /home/user/documents /home/user/downloads
```

Above command shows the disk usage of both the /home/user/documents and /home/user/downloads directories and provides a cumulative total.

Generate a Disk Usage Report and Save to a File

```
du -ah /home/user/documents > disk_usage_report.txt
```

Above command generates a detailed disk usage report for the /home/user/documents directory and saves it to disk_usage_report.txt.

Combining du with sort to Find Largest Directories

```
du -ah /home/user/documents | sort -rh | head -n 10
```

This command shows the top 10 largest files and directories within the /home/user/documents directory.

Check Disk Usage of a Specific Directory and Its Subdirectories

```
du -h /home/user/documents/*
```

Above command shows the disk usage of each subdirectory and file within the /home/user/documents directory.

Display Disk Usage with Grand Total

```
du -c /home/user/documents
```

This command shows the disk usage of the /home/user/documents directory and provides a grand total.

Exclude Files Based on Size

```
find /home/user/documents -size +1M -exec du -ch {} + | grep total$
```

This command finds all files larger than 1 MB within the /home/user/documents directory and displays their total disk usage.

10. kill Command

The kill command is used to terminate processes.

Syntax:

kill [options] PID

Common Options:

- -9 : Forcefully kill the process.

Examples:

Terminate a process by PID:

```
kill 1234
```

Forcefully terminate a process by PID:

```
kill -9 1234
```

Kill processes using more than 50% CPU for more than 30 seconds.

```
ps -eo pid,pcpu | awk '$2 > 50 {print $1}' | xargs kill
```

This uses ps to list processes with their CPU usage, awk to filter those above 50%, and xargs to pass their PIDs to kill.

Kill the most recently backgrounded process.

```
kill $!
```

Find and Kill a Process by Name:

```
ps aux | grep exampleprocess | grep -v grep | awk '{print $2}' | xargs kill
```

This pipeline lists all processes, filters for exampleprocess, excludes the grep process itself, extracts the PID, and sends the kill command.

Kill Processes Consuming High Memory:

```
ps aux --sort=-%mem | awk 'NR>1 && $4>80 {print $2}' | xargs kill
```

This lists processes sorted by memory usage, filters those using more than 80%, and kills them.

11. scp Command

The scp command is used to securely copy files between hosts.

Syntax:

scp [options] SOURCE DEST

Common Options:

- -r : Recursively copy entire directories.
- -P PORT : Specifies the port to use when connecting to the remote host.

Examples:

Copy a file to a remote host:

scp file.txt user@remotehost:/path/to/destination

Copy a directory to a remote host:

scp -r /path/to/directory user@remotehost:/path/to/destination

Copy a file from a remote host:

scp user@remotehost:/path/to/source/file.txt /path/to/destination

Advanced usage with tar to copy and extract files:

tar czf - /path/to/directory | ssh user@remotehost 'tar xzf - -C /path/to/destination'

More Examples:

scp /home/user/file.txt user@192.168.1.2:/home/user/backup/

Above command copies file.txt from the local machine to the /home/user/backup/ directory on the remote machine with IP 192.168.1.2.

scp user@192.168.1.2:/home/user/file.txt /home/user/downloads/

Above command copies file.txt from the remote machine to the /home/user/downloads/ directory on the local machine.

scp -r /home/user/my_project user@192.168.1.2:/home/user/projects/

Above command recursively copies the my_project directory from the local machine to the /home/user/projects/ directory on the remote machine.

scp -r user@192.168.1.2:/home/user/projects/my_project /home/user/backup/

Above command recursively copies the my_project directory from the remote machine to the /home/user/backup/ directory on the local machine.

Advance Commands

Specifying a Non-Standard SSH Port

scp -P 2222 /home/user/file.txt user@192.168.1.2:/home/user/backup/

This command uses port 2222 to copy file.txt to the remote machine.

Copy Multiple Files

scp /home/user/file1.txt /home/user/file2.txt user@192.168.1.2:/home/user/backup/

Above command copies both file1.txt and file2.txt to the remote machine.

Using Wildcards

scp /home/user/*.txt user@192.168.1.2:/home/user/backup/

This command copies all .txt files from the local machine to the remote machine.

Limiting Bandwidth

scp -l 1000 /home/user/file.txt user@192.168.1.2:/home/user/backup/

Above command limits the transfer speed to 1000 Kbit/s.

Using Compression

scp -C /home/user/file.txt user@192.168.1.2:/home/user/backup/

Above command compresses the file during transfer.

Verbose Mode

scp -v /home/user/file.txt user@192.168.1.2:/home/user/backup/

Above command provides detailed information about the transfer process.

Combining scp with Other Commands

tar czf - /home/user/my_project | ssh user@192.168.1.2 'cat > /home/user/backup/my_project.tar.gz'

Above command archives and compresses the my_project directory on the local machine, then transfers it to the remote machine, saving it as my_project.tar.gz.

DISCLAIMER AND CONSENT

This document is being provided by DigiTalk as part of its effort to assist users in understanding and working with Linux. While every effort has been made to ensure the accuracy and reliability of the information presented in this document, there is a possibility of typographical errors or inaccuracies. DigiTalk does not guarantee the correctness or completeness of the content provided in this document.

Users of this document are encouraged to cross-reference the information presented here with official documentation available on their website or other authoritative sources. Any discrepancies or inaccuracies found in this document should be reported to us at digitalk.fmw@gmail.com.

By using this document, you acknowledge and consent to the following:

This document is not officially endorsed or verified by any other third party organization..

The Company makes no claims or guarantees about the accuracy or suitability of the information contained in this document.

Users are responsible for verifying and validating any information presented here for their specific use case.

DigiTalk disclaims any liability for any errors, omissions, or damages that may result from the use of this document.

If you discover any inaccuracies or errors in this document, please report them to digitalk.fmw@gmail.com, and the Company will endeavor to correct them as necessary.

This consent statement is provided to ensure transparency and understanding of the limitations of the information contained in this document. By using this document, you agree to abide by the terms and conditions outlined herein.