
# 100 Most Important Linux Commands For DevOps Engineers

## File System Management:

1. **ls**: List directory contents.

   ```
   ls
   ```

   Example Output:

   ```
   file1.txt  file2.txt  directory1  directory2
   ```

2. **cd**: Change directory.

   ```
   cd directory_name
   ```

   Example Output: (No Output, just changes directory)

3. **pwd**: Print working directory.

   ```
   pwd
   ```

   Example Output:

   ```
   /home/user/directory_name
   ```

4. **mkdir**: Make directory.

   ```
   mkdir directory_name
   ```

   Example Output: (No Output, creates directory)

5. **rm**: Remove files or directories.

   ```
   rm file_name
   ```

   Example Output: (No Output, removes file)

6. **cp**: Copy files or directories.

   ```
   cp source_file destination_file
   ```

   Example Output: (No Output, copies file)

7. **mv**: Move or rename files or directories.

   ```
   mv old_file_name new_file_name
   ```

   Example Output: (No Output, renames file)

8. **touch**: Create an empty file.

   ```
   touch file_name
   ```

   Example Output: (No Output, creates empty file)

9. **cat**: Concatenate and display files.

   ```
   cat file_name
   ```

   Example Output:

   ```
   Contents of file
   ```

10. **less/more**: View file contents page by page.

    ```
    less file_name
    ```

    Example Output:

    ```
    Contents of file
    ```

## File Manipulation:

11. **grep**: Search for a pattern in files.

    ```
    grep pattern file_name
    ```

    Example Output:

    ```
    Matching lines containing pattern
    ```

12. **find**: Search for files in a directory hierarchy.

    ```
    find directory_name -name file_name
    ```

    Example Output:

    ```
    /directory_name/file_name
    ```

13. **chmod**: Change file permissions.

    ```
    chmod permissions file_name
    ```

    Example Output: (No Output, changes file permissions)

14. **chown**: Change file owner and group.

    ```
    chown user:group file_name
    ```

    Example Output: (No Output, changes file owner and group)

## Archive Management:

15. **tar**: Manipulate archive files.

    ```
    tar -czvf archive_name.tar.gz directory_to_compress
    ```

    Example Output: (No Output, creates tar.gz archive)

16. **gzip/gunzip**: Compress or decompress files.

    ```
    gzip file_name
    ```

    Example Output: (No Output, compresses file)

17. **zip/unzip**: Compress or decompress zip files.

    ```
    zip -r archive_name.zip directory_to_compress
    ```

    Example Output: (No Output, creates zip archive)

## Process Management:

18. **ps**: Display information about running processes.

    ```
    ps
    ```

    Example Output:

    ```
    PID   TTY    TIME     CMD
    123   pts/1  00:00:00 bash
    ```

19. **kill**: Terminate processes.

    ```
    kill process_id
    ```

    Example Output: (No Output, terminates process)

20. **top**: Display and update sorted information about processes.

    ```
    top
    ```

    Example Output:

    ```
    Tasks: 100 total, 1 running, 99 sleeping
    ```

## Network Management:

21. **ifconfig/ip**: Display or configure network interface parameters.

    ```
    ifconfig
    ```

    Example Output:

    ```
    eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    ```

22. **ping**: Test a network connection.

    ```
    ping host_name
    ```

    Example Output:

    ```
    PING host_name (192.168.1.1) 56(84) bytes of data.
    ```

23. **ssh**: Securely connect to a remote server.

    ```
    ssh user@host
    ```

    Example Output: (No Output, connects to remote server)

24. **scp**: Securely copy files between hosts.

    ```
    scp file user@host:/path
    ```

    Example Output: (No Output, copies file to remote host)

## System Information:

25. **df**: Display disk space usage.

    ```
    df
    ```

    Example Output:

    ```
    Filesystem     1K-blocks  Used Available Use% Mounted on
    /dev/sda1      12345678   98765  11223344  10% /
    ```

26. **du**: Estimate file space usage.

    ```
    du
    ```

    Example Output:

    ```
    12345678  .
    ```

27. **uname**: Print system information.

    ```
    uname -a
    ```

    Example Output:

    ```
    Linux hostname 4.4.0-18362-Microsoft #1-Microsoft
    ```

28. **uptime**: Show how long the system has been running.

    ```
    uptime
    ```

    Example Output:

    ```
    12:34:56 up 10 days, 2:30, 1 user, load average: 0.00, 0.01, 0.05
    ```

## Users and Permissions:

29. **passwd**: Change user password.

    ```
    passwd
    ```

    Example Output: (No Output, changes user password)

30. **useradd**: Create a new user.

    ```
    useradd username
    ```

    Example Output: (No Output, creates new user)

31. **userdel**: Delete a user account.

    ```
    userdel username
    ```

    Example Output: (No Output, deletes user account)

32. **usermod**: Modify a user account.

    ```
    usermod -aG group_name username
    ```

    Example Output: (No Output, modifies user account)

33. **groupadd**: Create a new group.

    ```
    groupadd group_name
    ```

    Example Output: (No Output, creates new group)

34. **groupdel**: Delete a group.

    ```
    groupdel group_name
    ```

    Example Output: (No Output, deletes group)

## Package Management:

35. **yum/dnf**: Package manager for RPM-based Linux distributions.

    ```
    yum install package_name
    ```

    Example Output: (No Output, installs package)

36. **apt/apt-get**: Package manager for Debian-based Linux distributions.

    ```
    apt-get install package_name
    ```

    Example Output: (No Output, installs package)

37. **rpm**: Package manager for RPM-based

Linux distributions. `bash rpm -i package.rpm`

Example Output: (No Output, installs package)

38. **dpkg**: Package manager for Debian-based Linux distributions.

    ```
    dpkg -i package.deb
    ```

    Example Output: (No Output, installs package)

## System Maintenance:

39. **cron**: Schedule tasks to run periodically.

    ```
    crontab -e
    ```

    Example Output: (No Output, opens cron file for editing)

40. **at**: Schedule a one-time task.

    ```
    at now + 1 hour
    ```

    Example Output: (No Output, schedules task)

## System Control:

41. **systemctl**: Control the systemd system and service manager.

    ```
    systemctl start|stop|restart service_name
    ```

    Example Output: (No Output, starts/stops/restarts service)

42. **journalctl**: Query and display messages from the journal.

    ```
    journalctl
    ```

    Example Output: (No Output, displays system journal)

## Miscellaneous:

43. **hostname**: Display or set the system's hostname.

    ```
    hostname
    ```

    Example Output:

    ```
    Hostname
    ```

44. **whoami**: Print the current user.

    ```
    whoami
    ```

    Example Output:

    ```
    Username
    ```

## Monitoring and Performance:

45. **top**: Display and update sorted information about processes.

    ```
    top
    ```

    Example Output:

    ```
    top - 12:34:56 up 10 days, 2:30, 1 user, load average: 0.00, 0.01,
    0.05
    Tasks: 100 total, 1 running, 99 sleeping
    ```

46. **htop**: Interactive process viewer.

    ```
    htop
    ```

    Example Output:

    ```
    CPU%  MEM%
    10.2  5.0
    ```

47. **vmstat**: Report virtual memory statistics.

    ```
    vmstat
    ```

    Example Output:

    ```
    procs -----------memory---------- ---swap-- -----io---- -system-- ---
    ---cpu-----
     r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us
    sy id wa st
     1  0      0 123456  78901 2345678    0    0     0     0    0    0  0
    0  0  0  0
    ```

48. **sar**: Collect, report, or save system activity information.

    ```
    sar
    ```

    Example Output:

    ```
    12:00:00 PM   CPU  %user  %nice   %system   %iowait   %steal   %idle
    ```

49. **iostat**: Report CPU and I/O statistics.

    ```
    iostat
    ```

    Example Output:

    ```
    avg-cpu:  %user   %nice %system %iowait  %steal   %idle
              0.12    0.04    0.21    0.10    0.00   99.53
    ```

## Text Processing:

50. **sed**: Stream editor for filtering and transforming text.

    ```
    sed 's/old_string/new_string/g' file_name
    ```

    Example Output:

    ```
    Modified contents of file
    ```

51. **awk**: A versatile programming language for working on files.

    ```
    awk '{print $1}' file_name
    ```

    Example Output:

    ```
    Column_1_Data
    ```

52. **grep**: Search for a pattern in files.

    ```
    grep pattern file_name
    ```

    Example Output:

    ```
    Matching lines containing pattern
    ```

53. **sort**: Sort lines of text files.

    ```
    sort file_name
    ```

    Example Output:

    ```
    Sorted lines
    ```

## Input/Output:

54. **tee**: Redirect output to multiple files or displays.

    ```
    command | tee file1 file2
    ```

    Example Output: (No Output, redirects output to files)

55. **tail**: Output the last part of files.

    ```
    tail file_name
    ```

    Example Output:

    ```
    Last 10 lines of file
    ```

56. **head**: Output the first part of files.

    ```
    head file_name
    ```

    Example Output:

    ```
    First 10 lines of file
    ```

57. **less/more**: View file contents page by page.

    ```
    less file_name
    ```

    Example Output:

    ```
    Contents of file
    ```

## Compression and Archiving:

58. **gzip/gunzip**: Compress or decompress files.

    ```
    gzip file_name
    ```

    Example Output: (No Output, compresses file)

59. **zip/unzip**: Compress or decompress zip files.

    ```
    zip -r archive_name.zip directory_to_compress
    ```

    Example Output: (No Output, creates zip archive)

60. **tar**: Manipulate archive files.

    ```
    tar -czvf archive_name.tar.gz directory_to_compress
    ```

    Example Output: (No Output, creates tar.gz archive)

## System Information:

61. **df**: Display disk space usage.

    ```
    df
    ```

    Example Output:

    ```
    Filesystem     1K-blocks  Used Available Use% Mounted on
    /dev/sda1      12345678   98765  11223344  10% /
    ```

62. **free**: Display amount of free and used memory in the system.

```
free -m
```

Example Output:

```
total        used        free      shared    buffers     cached
Mem:          1234         567         890         123         456
789
```

63. **ps**: Display information about running processes.

```
ps
```

Example Output:

```
PID   TTY    TIME     CMD
123   pts/1  00:00:00 bash
```

## Networking:

64. **netstat**: Display network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.

```
netstat
```

Example Output:

```
Active Internet connections
```

65. **lsof**: List open files.

```
lsof
```

Example Output:

```
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
```

66. **ping**: Test a network connection.

```
ping host_name
```

Example Output:

```
PING host_name (192.168.1.1) 56(84) bytes of data.
```

67. **ssh**: Securely connect to a remote server.

```
ssh user@host
```

Example Output: (No Output, connects to remote server)

68. **scp**: Securely copy files between hosts.

    ```
    scp file user@host:/path
    ```

    Example Output: (No Output, copies file to remote host)

69. **wget**: Retrieve files from the internet via HTTP, HTTPS, or FTP.

    ```
    wget URL
    ```

    Example Output: (No Output, retrieves file)

70. **curl**: Transfer data from or to a server.

    ```
    curl URL
    ```

    Example Output: (No Output, transfers data)

## Process Management:

71. **kill**: Terminate processes.

    ```
    kill process_id
    ```

    Example Output: (No Output, terminates process)

## User and Group Management:

72. **groups**: Print group memberships for a user.

    ```
    groups username
    ```

    Example Output:

    ```
    username : group1 group2
    ```

## System Information:

73. **hostname**: Display or set the system's hostname.

    ```
    hostname
    ```

    Example Output:

    ```
    Hostname
    ```

74. **whoami**: Print the current user.

    ```
    whoami
    ```

    Example Output:

    ```
    Username
    ```

75. **uptime**: Show how long the system has been running.

    ```
    uptime
    ```

    Example Output:

    ```
    12:34:56 up 10 days, 2:30, 1 user, load average: 0.00, 0.01, 0.05
    ```

76. **date**: Display or set the system date and time.

    ```
    date
    ```

    Example Output:

    ```
    Mon Apr  4 12:34:56 UTC 2024
    ```

77. **uname**: Print system information.

    ```
    uname -a
    ```

    Example Output:

    ```
    Linux hostname 4.4.0-18362-Microsoft #1-Microsoft
    ```

78. **lsblk**: List block devices.

    ```
    lsblk
    ```

    Example Output:

    ```
    NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
    sda      8:0    0   20G  0 disk
    └─sda1   8:1    0   20G  0 part /
    ```

79. **df**: Display disk space usage.

```
df -h
```

Example Output:

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        20G  9.8G  9.2G  52% /
```

80. **free**: Display amount of free and used memory in the system.

```
free -m
```

Example Output:

```
              total        used        free      shared  buff/cache
available
Mem:           2002        1054         346         199         602
648
```

81. **ps**: Display information about running processes.

```
ps
```

Example Output:

```
PID   TTY    TIME     CMD
123   pts/1  00:00:00 bash
```

82. **lsof**: List open files.

```
lsof
```

Example Output:

```
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
```

83. **netstat**: Display network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.

```
netstat
```

Example Output:

```
Active Internet connections
```

## System Control:

84. **systemctl**: Control the systemd system and service manager.

    ```
    systemctl start|stop|restart service_name
    ```

    Example Output: (No Output, starts/stops/restarts service)

85. **journalctl**: Query and display messages from the journal.

    ```
    journalctl
    ```

    Example Output: (No Output, displays system journal)

## Package Management:

86. **yum/dnf**: Package manager for RPM-based Linux distributions.

    ```
    yum install package_name
    ```

    Example Output: (No Output, installs package)

87. **apt/apt-get**: Package manager for Debian-based Linux distributions.

    ```
    apt-get install package_name
    ```

    Example Output: (No Output, installs package)

88. **rpm**: Package manager for RPM-based Linux distributions.

    ```
    rpm -i package.rpm
    ```

    Example Output: (No Output, installs package)

89. **dpkg**: Package manager for Debian-based Linux distributions.

    ```
    dpkg -i package.deb
    ```

    Example Output: (No Output, installs package)

## System Maintenance:

90. **cron**: Schedule tasks to run periodically.

    ```
    crontab -e
    ```

    Example Output: (No Output, opens cron file for editing)

91. **at**: Schedule a one-time task.

    ```
    at now + 1 hour
    ```

    Example Output: (No Output, schedules task)

## Miscellaneous:

92. **su**: Run a command with substitute user and group ID.

    ```
    su - username
    ```

    Example Output: (No Output, switches user)

93. **sudo**: Execute a command as another user.

    ```
    sudo command
    ```

    Example Output: (No Output, executes command as another user)

94. **chmod**: Change file mode bits.

    ```
    chmod 755 file_name
    ```

    Example Output: (No Output, changes file permissions)

95. **chown**: Change file owner and group.

    ```
    chown user:group file_name
    ```

    Example Output: (No Output, changes file owner and group)

96. **passwd**: Change user password.

    ```
    passwd
    ```

    Example Output: (No Output, changes user password)

97. **useradd**: Create a new user or update default new user information.

    ```
    useradd username
    ```

    Example Output: (No Output, creates new user)

98. **userdel**: Delete a user account and related files.

    ```
    userdel username
    ```

    Example Output: (No Output, deletes user account)

99. **usermod**: Modify a user account.

    ```
    usermod -aG group_name username
    ```

    Example Output: (No Output, modifies user account)

100. **groupadd**: Create a new group. `bash groupadd group_name` Example Output: (No Output, creates new group)