



Placement Empowerment Program
Cloud Computing and DevOps Centre

60 Days of DevOps Challenge

Day 2: Linux Shell Scripting & Automation

Name: MONISHA J R

Department: CSE



This Proof of Concept (PoC) demonstrates the Linux Basics Shell Scripting, Managing Processes, Using Package Managers, Basic Networking Commands and Editing Configuration Files.

Challenge 1: Write a simple Bash script that prints “Hello DevOps” along with the current date and time.

```
monisha@linux-mint:~$ nano hello_devops.sh
monisha@linux-mint:~$ chmod +x hello_devops.sh
monisha@linux-mint:~$ ./hello_devops.sh
Hello DevOps! Today's date and time is: Monday 07 July 2025 07:15:40 PM IST
monisha@linux-mint:~$
```

Challenge 2: Create a script that checks if a website (e.g., <https://www.learnxops.com>) is reachable using curl or ping. Print a success or failure message.

```
monisha@linux-mint:~$ nano check_web.sh
monisha@linux-mint:~$ chmod +x check_web.sh
monisha@linux-mint:~$ ./check_web.sh
✓ Success: https://www.learnxops.com is reachable via curl!
monisha@linux-mint:~$
```

Challenge 3: Write a script that takes a filename as an argument, checks if it exists, and prints the content of the file accordingly.

```
monisha@linux-mint:~$ nano check_file.sh
monisha@linux-mint:~$ chmod +x check_file.sh
monisha@linux-mint:~$ ./check_file.sh
✗ Error: File 'devops_challenge_day_1' does not exist.
monisha@linux-mint:~$
```

Challenge 4: Create a script that lists all running processes and writes the output to a file named process_list.txt.

```
monisha@linux-mint:~$ nano list_processes.sh
monisha@linux-mint:~$ chmod +x list_processes.sh
monisha@linux-mint:~$ ./list_processes.sh
✓ Process list saved to process_list.txt
monisha@linux-mint:~$ cat process_list.txt
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  22792 13100 ?        Ss   18:41   0:02 /sbin/init splash
root         2  0.0  0.0      0     0 ?        S    18:41   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    18:41   0:00 [pool_workqueue_release]
root         4  0.0  0.0      0     0 ?        I<   18:41   0:00 [kworker/R-rcu_g]
root         5  0.0  0.0      0     0 ?        I<   18:41   0:00 [kworker/R-rcu_p]
root         6  0.0  0.0      0     0 ?        I<   18:41   0:00 [kworker/R-slub_]
root         7  0.0  0.0      0     0 ?        I<   18:41   0:00 [kworker/R-netns]
root         9  0.0  0.0      0     0 ?        I<   18:41   0:00 [kworker/0:0H-events_highpri]
root        10  0.0  0.0      0     0 ?        I    18:41   0:00 [kworker/0:1-events]
root        12  0.0  0.0      0     0 ?        I<   18:41   0:00 [kworker/R-mm_pe]
root        13  0.0  0.0      0     0 ?        I    18:41   0:00 [rcu_tasks_kthread]
root        14  0.0  0.0      0     0 ?        I    18:41   0:00 [rcu_tasks_rude_kthread]
root        15  0.0  0.0      0     0 ?        I    18:41   0:00 [rcu_tasks_trace_kthread]
root        16  0.0  0.0      0     0 ?        S    18:41   0:00 [ksoftirqd/0]
root        17  0.2  0.0      0     0 ?        I    18:41   0:07 [rcu_preempt]
root        18  0.0  0.0      0     0 ?        S    18:41   0:01 [migration/0]
root        19  0.0  0.0      0     0 ?        S    18:41   0:00 [idle_inject/0]
root        20  0.0  0.0      0     0 ?        S    18:41   0:00 [cpuhp/0]
root        21  0.0  0.0      0     0 ?        S    18:41   0:00 [cpuhp/2]
root        22  0.0  0.0      0     0 ?        S    18:41   0:00 [idle_inject/2]
root        23  0.0  0.0      0     0 ?        S    18:41   0:00 [migration/2]
root        24  0.0  0.0      0     0 ?        S    18:41   0:00 [ksoftirqd/2]
root        26  0.0  0.0      0     0 ?        I<   18:41   0:00 [kworker/2:0H-events_highpri]
root        27  0.0  0.0      0     0 ?        S    18:41   0:00 [cpuhp/4]
root        28  0.0  0.0      0     0 ?        S    18:41   0:00 [idle_inject/4]
root        29  0.0  0.0      0     0 ?        S    18:41   0:01 [migration/4]
root        30  0.0  0.0      0     0 ?        S    18:41   0:00 [ksoftirqd/4]
root        32  0.0  0.0      0     0 ?        I<   18:41   0:00 [kworker/4:0H-events_highpri]
root        33  0.0  0.0      0     0 ?        S    18:41   0:00 [cpuhp/5]
root        34  0.0  0.0      0     0 ?        S    18:41   0:00 [idle_inject/5]
root        35  0.0  0.0      0     0 ?        S    18:41   0:00 [migration/5]
root        36  0.0  0.0      0     0 ?        S    18:41   0:00 [ksoftirqd/5]
root        38  0.0  0.0      0     0 ?        I<   18:41   0:00 [kworker/5:0H-events_highpri]
root        39  0.0  0.0      0     0 ?        S    18:41   0:00 [cpuhp/6]
root        40  0.0  0.0      0     0 ?        S    18:41   0:00 [idle_inject/6]
root        41  0.0  0.0      0     0 ?        S    18:41   0:00 [migration/6]
```

Challenge 5: Write a script that installs multiple packages at once (e.g., git, vim, curl). The script should check if each package is already installed before attempting installation.

```
monisha@linux-mint:~$ nano install_packages.sh
monisha@linux-mint:~$ chmod +x install_packages.sh
monisha@linux-mint:~$ ./install_packages.sh
✓ git is already installed.
✓ vim is already installed.
✓ curl is already installed.
monisha@linux-mint:~$
```

Challenge 6: Create a script that monitors CPU and memory usage every 5 seconds and logs the results to a file.

```
monisha@linux-mint:~$ nano monitor_resources.sh
monisha@linux-mint:~$ chmod +x monitor_resources.sh
monisha@linux-mint:~$ ./monitor_resources.sh
Monitoring CPU and Memory usage... Logs will be saved in resource_usage.log
^Z
[1]+  Stopped                  ./monitor_resources.sh
monisha@linux-mint:~$ cat resource_usage.log
Timestamp | CPU (%) | Memory (%)
2025-07-07 19:41:02 | 3.6 | 24.93
2025-07-07 19:41:07 | 4.5 | 24.84
2025-07-07 19:41:13 | 3.8 | 24.82
2025-07-07 19:41:18 | 3.1 | 25.32
2025-07-07 19:41:23 | 3.8 | 24.93
monisha@linux-mint:~$
```

Challenge 7: Write a script that automatically deletes log files older than 7 days from /var/log.

```
monisha@linux-mint:~$ nano clean_old_logs.sh
monisha@linux-mint:~$ chmod +x clean_old_logs.sh
monisha@linux-mint:~$ ./clean_old_logs.sh
rm: cannot remove '/var/log/bootstrap.log': Permission denied
rm: cannot remove '/var/log/lightdm/seat0-greeter.log': Permission denied
find: '/var/log/speech-dispatcher': Permission denied
find: '/var/log/private': Permission denied
rm: cannot remove '/var/log/installer/casper.log': Permission denied
rm: cannot remove '/var/log/alternatives.log': Permission denied
✓ Deleted log files older than 7 days from /var/log.
monisha@linux-mint:~$
```

Challenge 8 : Automate user account creation – Write a script that takes the username as an argument, checks, if the user exists, gives the message “user already exists” else creates a new user, adds it to a “devops” group, and sets up a default home directory.

```
monisha@linux-mint:~$ nano create_user.sh
monisha@linux-mint:~$ chmod +x create_user.sh
monisha@linux-mint:~$ ./create_user.sh devops_user
✓ User 'devops user' already exists.
monisha@linux-mint:~$
```

Challenge 9 : Use awk or sed in a script to process a log file and extract only error messages.

```
monisha@linux-mint:~$ nano extract_errors.sh
monisha@linux-mint:~$ chmod +x extract_errors.sh
monisha@linux-mint:~$ ./extract_errors.sh
✓ Extracted error messages saved to 'error_messages.log'.
monisha@linux-mint:~$ cat error_messages.log
2025-06-27T17:47:53.220760+05:30 linux-mint systemd-coredump[1048]: Process 1026 (Xorg) of user 0 dumped core.#012#012Module libzstd.so.1 from deb libzstd-1.5
.5+dfsg2-2build1.1.amd64#012Module libsystemd.so.0 from deb systemd-255.4-1ubuntu8.5.amd64#012Module libudev.so.1 from deb systemd-255.4-1ubuntu8.5.amd64#012S
tack trace of thread 1026:#012#0 0x000074e42469eb1c pthread_kill implementation (libc.so.6 + 0x9eb1c)#012#1 0x000074e42464526e GI raise (libc.so.6 + 0x4
526e)#012#2 0x000074e4246288ff GI abort (libc.so.6 + 0x288ff)#012#3 0x0000636394ee7c9e OsAbort (Xorg + 0x1e4c9e)#012#4 0x0000636394eed7e7 n/a (Xorg + 0x1
ea7e7)#012#5 0x0000636394eee867 FatalError (Xorg + 0x1eb867)#012#6 0x0000636394d6a176 n/a (Xorg + 0x67176)#012#7 0x000074e42462a1ca libc_start_call_main
(libc.so.6 + 0x2a1ca)#012#8 0x000074e42462a28b __libc_start_main_impl (libc.so.6 + 0x2a28b)#012#9 0x0000636394d52395 _start (Xorg + 0x4f395)#012ELF object b
inary architecture: AMD X86-64
2025-06-27T17:47:53.316331+05:30 linux-mint systemd-coredump[1070]: Process 1068 (Xorg) of user 0 dumped core.#012#012Module libzstd.so.1 from deb libzstd-1.5
.5+dfsg2-2build1.1.amd64#012Module libsystemd.so.0 from deb systemd-255.4-1ubuntu8.5.amd64#012Module libudev.so.1 from deb systemd-255.4-1ubuntu8.5.amd64#012S
tack trace of thread 1068:#012#0 0x000074a6369eb1c pthread_kill implementation (libc.so.6 + 0x9eb1c)#012#1 0x000074a6364526e GI raise (libc.so.6 + 0x4
526e)#012#2 0x000074a636288ff GI abort (libc.so.6 + 0x288ff)#012#3 0x00005c2ffc067c9e OsAbort (Xorg + 0x1e4c9e)#012#4 0x00005c2ffc06d7e7 n/a (Xorg + 0x1
ea7e7)#012#5 0x00005c2ffc06e867 FatalError (Xorg + 0x1eb867)#012#6 0x00005c2ffbeaa176 n/a (Xorg + 0x67176)#012#7 0x000074a6362a1ca libc_start_call_main
(libc.so.6 + 0x2a1ca)#012#8 0x000074a6362a28b __libc_start_main_impl (libc.so.6 + 0x2a28b)#012#9 0x00005c2ffbed2395 _start (Xorg + 0x4f395)#012ELF object b
inary architecture: AMD X86-64
2025-06-27T17:47:54.691744+05:30 linux-mint pipewire[1212]: mod.jackdbus-detect: Failed to receive jackdbus reply: org.freedesktop.DBus.Error.ServiceUnknown:
The name org.jackaudio.service was not provided by any .service files
2025-06-27T17:47:55.112316+05:30 linux-mint wireplumber[1214]: <WpPortalPermissionStorePlugin:0x604e747b0f00> Failed to call Lookup: GDBus.Error:org.freedesk
op.portal.Error.NotFound: No entry for camera
2025-06-27T17:49:35.513864+05:30 linux-mint gvfsd-network[2488]: Couldn't create directory monitor on wsd:///. Error: Automount failed: Failed to spawn the u
nderlying wsd daemon.
2025-06-27T17:48:35.388801+05:30 linux-mint gvfsd-network[2488]: Couldn't create directory monitor on wsd:///. Error: Automount failed: mountpoint for org.gt
k.vfs.mountpoint wsd already running
2025-07-03T09:38:35.851755+05:30 linux-mint kernel: RAS: Correctable Errors collector initialized.
2025-07-03T09:38:35.852506+05:30 linux-mint kernel: EDAC igen6 MC1: HANDLING IBEC MEMORY ERROR
2025-07-03T09:38:35.852508+05:30 linux-mint kernel: EDAC igen6 MC0: HANDLING IBEC MEMORY ERROR
2025-07-03T09:38:35.296949+05:30 linux-mint pipewire[1127]: mod.jackdbus-detect: Failed to receive jackdbus reply: org.freedesktop.DBus.Error.ServiceUnknown:
The name org.jackaudio.service was not provided by any .service files
2025-07-03T09:38:38.898790+05:30 linux-mint wireplumber[1131]: <WpPortalPermissionStorePlugin:0x56df4c26e010> Failed to call Lookup: GDBus.Error:org.freedesk
op.portal.Error.NotFound: No entry for camera
2025-07-03T09:49:32.324671+05:30 linux-mint NetworkManager[788]: <warn> [1751516372.3245] device (wlp1s0): Deactivation failed: GDBus.Error:fi.w1.wpa_supplic
ant1.NotConnected: This interface is not connected
2025-07-03T10:29:11.963506+05:30 linux-mint cinnamon-screensaver-pam-helper: pam.ecryptfs: seteuid error
2025-07-03T10:29:42.057482+05:30 linux-mint org.cinnamon.ScreenSaver[3664]: Error in sys.excepthook:
2025-07-03T10:29:42.057678+05:30 linux-mint org.cinnamon.ScreenSaver[3664]: Error in sys.excepthook:
2025-07-03T10:29:42.057737+05:30 linux-mint org.cinnamon.ScreenSaver[3664]: Error in sys.excepthook:
2025-07-05T10:00:42.190442+05:30 linux-mint kernel: RAS: Correctable Errors collector initialized.
2025-07-05T10:00:42.191521+05:30 linux-mint kernel: EDAC igen6 MC1: HANDLING IBEC MEMORY ERROR
```

Challenge 10 : Set up a cron job that runs a script to back up (zip/tar) a directory daily.

```
monisha@linux-mint:~$ nano backup.sh
monisha@linux-mint:~$ chmod +x backup.sh
monisha@linux-mint:~$ crontab -e
crontab: installing new crontab
monisha@linux-mint:~$ crontab -l

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/5 * * * * /home/monisha/sys_monitor.sh
0 2 * * * /path/to/backup.sh >> /var/log/backup.log 2>&1
monisha@linux-mint:~$
```