



Follow – Krishan Bhatt

1. System Update Script

```
#!/bin/bash  
  
# Update the system  
sudo apt-get update -y && sudo apt-get upgrade -y
```

2. Disk Usage Report

```
#!/bin/bash  
  
# Display disk usage  
df -h > /tmp/disk_usage_report.txt  
cat /tmp/disk_usage_report.txt
```

3. Backup Script

```
#!/bin/bash  
  
# Backup home directory to /backup  
tar -czvf /backup/home_backup_$(date +%F).tar.gz /home/username
```

4. User Creation Script

```
#!/bin/bash  
  
# Create a new user  
if [ $# -eq 0 ]; then
```

```
    echo "Usage: $0 username"
    exit 1
fi
sudo useradd -m $1
sudo passwd $1
```

5. Service Status Check

```
#!/bin/bash
# Check if a service is running
if [ $# -eq 0 ]; then
    echo "Usage: $0 service_name"
    exit 1
fi
sudo systemctl status $1
```

6. Memory Usage Report

```
#!/bin/bash
# Display memory usage
free -m > /tmp/memory_usage_report.txt
cat /tmp/memory_usage_report.txt
```

7. Network Configuration Script

```
#!/bin/bash
# Display network configuration
ifconfig > /tmp/network_config.txt
```

```
cat /tmp/network_config.txt
```

8. Log Rotation Script

```
#!/bin/bash  
# Rotate logs  
logrotate /etc/logrotate.conf
```

9. Process Monitor Script

```
#!/bin/bash  
# Monitor processes and save to file  
ps aux > /tmp/process_list.txt  
cat /tmp/process_list.txt
```

10. Scheduled Task Script

```
#!/bin/bash  
# Add a cron job to backup home directory every day at midnight  
(crontab -l; echo "0 0 * * * /path/to/backup_script.sh") | crontab -
```

11. File Compression Script

```
#!/bin/bash  
# Compress a file  
if [ $# -eq 0 ]; then  
    echo "Usage: $0 filename"  
    exit 1
```

```
fi  
gzip $1
```

12. File Decompression Script

```
#!/bin/bash  
# Decompress a file  
if [ $# -eq 0 ]; then  
    echo "Usage: $0 filename"  
    exit 1  
fi  
gunzip $1
```

13. Check Disk Space Usage

```
#!/bin/bash  
# Check disk space usage for a specific directory  
if [ $# -eq 0 ]; then  
    echo "Usage: $0 directory"  
    exit 1  
fi  
du -sh $1
```

14. SSH Login Script

```
#!/bin/bash  
# SSH into a remote server  
if [ $# -lt 2 ]; then
```

```
    echo "Usage: $0 user@hostname"
    exit 1
fi
ssh $1@$2
```

15. Download File Script

```
#!/bin/bash
# Download a file using wget
if [ $# -eq 0 ]; then
    echo "Usage: $0 url"
    exit 1
fi
wget $1
```

16. Check Open Ports

```
#!/bin/bash
# Check open ports
sudo netstat -tuln
```

17. Check User Login History

```
#!/bin/bash
# Display user login history
last > /tmp/user_login_history.txt
cat /tmp/user_login_history.txt
```

18. Create Directory Script

```
#!/bin/bash  
  
# Create a new directory  
  
if [ $# -eq 0 ]; then  
    echo "Usage: $0 directory_name"  
    exit 1  
fi  
  
mkdir $1
```

19. Delete Directory Script

```
#!/bin/bash  
  
# Delete a directory  
  
if [ $# -eq 0 ]; then  
    echo "Usage: $0 directory_name"  
    exit 1  
fi  
  
rm -rf $1
```

20. System Reboot Script

```
#!/bin/bash  
  
# Reboot the system  
  
sudo reboot
```

21. System Shutdown Script

```
#!/bin/bash

# Shutdown the system

sudo shutdown -h now
```

22. Create Symbolic Link

```
#!/bin/bash

# Create a symbolic link

if [ $# -lt 2 ]; then

    echo "Usage: $0 target link_name"

    exit 1

fi

ln -s $1 $2
```

23. Change File Permissions

```
#!/bin/bash

# Change file permissions

if [ $# -lt 2 ]; then

    echo "Usage: $0 permissions filename"

    exit 1

fi

chmod $1 $2
```

24. Change File Ownership

```
#!/bin/bash

# Change file ownership
```

```
if [ $# -lt 2 ]; then
    echo "Usage: $0 owner filename"
    exit 1
fi
chown $1 $2
```

25. Monitor System Uptime

```
#!/bin/bash
# Display system uptime
uptime
```

26. Archive Old Logs

```
#!/bin/bash
# Archive logs older than 7 days
find /var/log -type f -mtime +7 -exec gzip {} \;
```

27. Check System Load

```
#!/bin/bash
# Display system load averages
cat /proc/loadavg
```

28. Create a Swap File

```
#!/bin/bash
# Create a swap file
```



```
sudo fallocate -l 1G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile
sudo bash -c "echo '/swapfile none swap sw 0 0' >> /etc/fstab"
```

29. Disable Swap File

```
#!/bin/bash
# Disable swap file
sudo swapoff /swapfile
sudo rm /swapfile
sudo sed -i '/\swapfile/d' /etc/fstab
```

30. Clear Cache Memory

```
#!/bin/bash
# Clear cache memory
sudo sync; echo 1 > /proc/sys/vm/drop_caches
```

31. Install a Package

```
#!/bin/bash
# Install a package using apt-get
if [ $# -eq 0 ]; then
    echo "Usage: $0 package_name"
    exit 1
fi
```

```
sudo apt-get install -y $1
```

32. Remove a Package

```
#!/bin/bash  
# Remove a package using apt-get  
if [ $# -eq 0 ]; then  
    echo "Usage: $0 package_name"  
    exit 1  
fi  
sudo apt-get remove -y $1
```

33. List Installed Packages

```
#!/bin/bash  
# List all installed packages  
dpkg --get-selections
```

34. Check Listening Ports

```
#!/bin/bash  
# Check listening ports using lsof  
sudo lsof -i -P -n | grep LISTEN
```

35. Configure Firewall Rule

```
#!/bin/bash  
# Add a firewall rule to allow traffic on port 80
```

```
sudo ufw allow 80/tcp
```

36. Enable Firewall

```
```bash
#!/bin/bash
Enable the firewall
sudo ufw enable
```
```

37. **Disable Firewall**

```
```bash
#!/bin/bash
Disable the firewall
sudo ufw disable
```
```

38. **Check Firewall Status**

```
```bash
#!/bin/bash
Check the firewall status
sudo ufw status
```
```

39. **Generate SSH Key Pair**

```
```bash
#!/bin/bash
Generate an SSH key pair
```

```
ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa
```

```
```
```

40. ****Sync Time with NTP Server****

```
```bash
```

```
#!/bin/bash
```

```
Sync time with an NTP server
```

```
sudo ntpdate pool.ntp.org
```

```
```
```

41. ****Check Kernel Version****

```
```bash
```

```
#!/bin/bash
```

```
Display the kernel version
```

```
uname -r
```

```
```
```

42. ****List Loaded Kernel Modules****

```
```bash
```

```
#!/bin/bash
```

```
List all loaded kernel modules
```

```
lsmod
```

```
```
```

43. ****Display System Information****

```
```bash
```

```
#!/bin/bash
```

```
Display system information
```

```
uname -a
```

```
```
```

44. ****Check CPU Info****

```
```bash
```

```
#!/bin/bash
```

```
Display CPU information
```

```
lscpu
```

```
```
```

45. ****Check PCI Devices****

```
```bash
```

```
#!/bin/bash
```

```
List all PCI devices
```

```
lspci
```

```
```
```

46. ****Check USB Devices****

```
```bash
```

```
#!/bin/bash
```

```
List all USB devices
```

```
lsusb
```

```
```
```

47. ****Check Block Devices****

```
```bash
```

```
#!/bin/bash
```

```
List all block devices
```

```
lsblk
```

```
```
```

48. ****Monitor Disk I/O****

```
```bash
```

```
#!/bin/bash
```

```
Monitor disk I/O
```

```
iostat -x
```

```
```
```

49. ****Monitor Network Traffic****

```
```bash
```

```
#!/bin/bash
```

```
Monitor network traffic
```

```
iftop
```

```
```
```

50. ****Monitor System Performance****

```
```bash
```

```
#!/bin/bash
```

```
Monitor system performance using top
```

```
top
```

```
```
```

51. ****Check Open Files****

```
```bash
```

```
#!/bin/bash
```

```
List open files
```

lsuf

```

52. ****Clear System Logs****

```bash

#!/bin/bash

# Clear system logs

sudo truncate -s 0 /var/log/\*.log

```

53. ****Install Docker****

```bash

#!/bin/bash

# Install Docker

sudo apt-get update

sudo apt-get install -y docker.io

```

54. ****Start Docker Service****

```bash

#!/bin/bash

# Start Docker service

sudo systemctl start docker

```

55. ****Enable Docker Service****

```bash

#!/bin/bash

```
Enable Docker service to start on boot
```

```
sudo systemctl enable docker
```

```
```
```

```
56. **Pull Docker Image**
```

```
```bash
```

```
#!/bin/bash
```

```
Pull a Docker image
```

```
if [$# -eq 0]; then
```

```
 echo "Usage: $0 image_name"
```

```
 exit 1
```

```
fi
```

```
sudo docker pull $1
```

```
```
```

```
57. **Run Docker Container**
```

```
```bash
```

```
#!/bin/bash
```

```
Run a Docker container
```

```
if [$# -eq 0]; then
```

```
 echo "Usage: $0 image_name"
```

```
 exit 1
```

```
fi
```

```
sudo docker run -d $1
```

```
```
```

```
58. **List Docker Containers**
```

```
```bash
```



```
#!/bin/bash
```

```
List all Docker containers
```

```
sudo docker ps -a
```

```
```\n
```

59. ****Stop Docker Container****

```
```bash
```

```
#!/bin/bash
```

```
Stop a running Docker container
```

```
if [$# -eq 0]; then
```

```
 echo "Usage: $0 container_id"
```

```
 exit 1
```

```
fi
```

```
sudo docker stop $1
```

```
```\n
```

60. ****Remove Docker Container****

```
```bash
```

```
#!/bin/bash
```

```
Remove a Docker container
```

```
if [$# -eq 0]; then
```

```
 echo "Usage: $0 container_id"
```

```
 exit 1
```

```
fi
```

```
sudo docker rm $1
```

```
```\n
```

61. ****List Docker Images****

```
```bash

#!/bin/bash

List all Docker images

sudo docker images

```
```

62. **Remove Docker Image**

```
```bash

#!/bin/bash

Remove a Docker image

if [$# -eq 0]; then

 echo "Usage: $0 image_id"

 exit 1

fi

sudo docker rmi $1

```
```

63. **Create a MySQL Database Backup**

```
```bash

#!/bin/bash

Create a backup of a MySQL database

if [$# -lt 3]; then

 echo "Usage: $0 db_user db_password db_name"

 exit 1

fi

mysqldump -u $1 -p$2 $3 > /tmp/$3_backup.sql

```
```

64. ****Restore a MySQL Database****

```
```bash
#!/bin/bash
Restore a MySQL database
if [$# -lt 3]; then
 echo "Usage: $0 db_user db_password db_name"
 exit 1
fi
mysql -u $1 -p$2 $3 < /tmp/$3_backup.sql
```
```

65. ****Monitor Apache Access Logs****

```
```bash
#!/bin/bash
Monitor Apache access logs in real-time
tail -f /var/log/apache2/access.log
```
```

66. ****Monitor Apache Error Logs****

```
```bash
#!/bin/bash
Monitor Apache error logs in real-time
tail -f /var/log/apache2/error.log
```
```

67. ****Restart Apache Service****

```
```bash
#!/bin/bash
```

```
Restart Apache service
sudo systemctl restart apache2
` ``
```

#### 68. **\*\*Start Apache Service\*\***

```
` `` bash
#!/bin/bash
Start Apache service
sudo systemctl start apache2
` ``
```

#### 69. **\*\*Stop Apache Service\*\***

```
` `` bash
#!/bin/bash
Stop Apache service
sudo systemctl stop apache2
` ``
```

#### 70. **\*\*Enable Apache Service\*\***

```
` `` bash
#!/bin/bash
Enable Apache service to start on boot
sudo systemctl enable apache2
` ``
```

#### 71. **\*\*Check Apache Service Status\*\***

```
` `` bash
#!/bin/bash
```

```
Check Apache service status
sudo systemctl status apache2
` ``
```

## 72. \*\*Check Active Users\*\*

```
` `` bash
#!/bin/bash
Display currently active users
w
` ``
```

## 73. \*\*Check Last System Reboot\*\*

```
` `` bash
#!/bin/bash
Display the last system reboot time
last reboot
` ``
```

## 74. \*\*Create a Tar Archive\*\*

```
` `` bash
#!/bin/bash
Create a tar archive of a directory
if [$# -lt 2]; then
 echo "Usage: $0 directory archive_name"
 exit 1
fi
tar -cvf $2.tar $1
` ``
```

#### 75. **\*\*Extract a Tar Archive\*\***

```
```bash
#!/bin/bash
# Extract a tar archive
if [ $# -eq 0 ]; then
    echo "Usage: $0 archive_name"
    exit 1
fi
tar -xvf $1
```
```

#### 76. **\*\*Find Files by Extension\*\***

```
```bash
#!/bin/bash
# Find files by extension
if [ $# -lt 2 ]; then
    echo "Usage: $0 directory extension"
    exit 1
fi
find $1 -type f -name ".*$2"
```
```

#### 77. **\*\*Find Large Files\*\***

```
```bash
#!/bin/bash
# Find files larger than 100MB
find / -type f -size +100M
```

```

#### 78. **\*\*Find Recently Modified Files\*\***

```
```bash
#!/bin/bash
# Find files modified in the last 7 days
find / -type f -mtime -7
```

```

#### 79. **\*\*Backup MySQL Databases\*\***

```
```bash
#!/bin/bash
# Backup all MySQL databases
mysqldump -u root -p --all-databases > /tmp/all_databases_backup.sql
```

```

#### 80. **\*\*Monitor Disk I/O in Real-time\*\***

```
```bash
#!/bin/bash
# Monitor disk I/O in real-time using iotop
sudo iotop -o
```

```

#### 81. **\*\*Display User Groups\*\***

```
```bash
#!/bin/bash
# Display groups of a user
if [ $# -eq 0 ]; then
```

```
    echo "Usage: $0 username"
    exit 1
fi
groups $1
` ``
```

82. ****Change Hostname****

```
` `` bash
#!/bin/bash
# Change the system hostname
if [ $# -eq 0 ]; then
    echo "Usage: $0 new_hostname"
    exit 1
fi
sudo hostnamectl set-hostname $1
` ``
```

83. ****Display Network Interfaces****

```
` `` bash
#!/bin/bash
# Display network interfaces
ip addr show
` ``
```

84. ****Ping a Host****

```
` `` bash
#!/bin/bash
# Ping a host
```



```
if [ $# -eq 0 ]; then
    echo "Usage: $0 hostname"
    exit 1
fi
ping -c 4 $1
` ``
```

85. ****Traceroute to a Host****

```
` `` bash
#!/bin/bash
# Traceroute to a host
if [ $# -eq 0 ]; then
    echo "Usage: $0 hostname"
    exit 1
fi
traceroute $1
` ``
```

86. ****Display Routing Table****

```
` `` bash
#!/bin/bash
# Display routing table
netstat -rn
` ``
```

87. ****Add Static Route****

```
` `` bash
#!/bin/bash
```

```
# Add a static route

if [ $# -lt 2 ]; then
    echo "Usage: $0 destination gateway"
    exit 1
fi

sudo ip route add $1 via $2
` ``
```

88. ****Delete Static Route****

```
` `` bash

#!/bin/bash

# Delete a static route

if [ $# -lt 2 ]; then
    echo "Usage: $0 destination gateway"
    exit 1
fi

sudo ip route
```