**You said:**

shell script projects that are used in real word - show me examples

**ChatGPT said:**

Here are some examples of real-world shell script projects that are commonly used:

**1. Automated Backup Script**

**Purpose:** This script automates the process of backing up files or directories on a regular basis. It can be used to back up data from your local machine to a remote server or cloud storage.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define variables

backup\_source="/home/user/data"

backup\_dest="/home/user/backups"

timestamp=$(date +"%Y-%m-%d\_%H-%M-%S")

backup\_name="backup\_$timestamp.tar.gz"

# Create backup

tar -czf $backup\_dest/$backup\_name $backup\_source

# Optional: Delete backups older than 30 days

find $backup\_dest -type f -name "\*.tar.gz" -mtime +30 -exec rm {} \;

echo "Backup created: $backup\_name"

**2. Log File Monitoring Script**

**Purpose:** This script is used to monitor log files in real-time. It can trigger alerts if specific keywords are detected in logs (e.g., errors or critical warnings).

**Example:**

bash

CopyEdit

#!/bin/bash

log\_file="/var/log/syslog"

keyword="ERROR"

# Monitor the log file for errors

tail -F $log\_file | while read line

do

if echo "$line" | grep -i "$keyword"; then

echo "Error detected: $line"

# Send an email or alert

mail -s "Error Alert" admin@example.com <<< "$line"

fi

done

**3. Disk Usage Monitoring and Alerting**

**Purpose:** This script monitors disk usage and sends an email alert if disk usage exceeds a specified threshold.

**Example:**

bash

CopyEdit

#!/bin/bash

# Set the threshold (e.g., 90%)

threshold=90

# Check the disk usage

disk\_usage=$(df / | grep / | awk '{ print $5 }' | sed 's/%//g')

if [ $disk\_usage -gt $threshold ]; then

echo "Disk usage is over $threshold%. Current usage is $disk\_usage%" | mail -s "Disk Usage Alert" admin@example.com

fi

**4. User Management Script**

**Purpose:** Automates the creation, modification, or deletion of user accounts on a Linux server.

**Example:**

bash

CopyEdit

#!/bin/bash

# Check for argument

if [ $# -ne 2 ]; then

echo "Usage: $0 <add|del> <username>"

exit 1

fi

action=$1

username=$2

if [ "$action" == "add" ]; then

useradd $username

echo "User $username added."

elif [ "$action" == "del" ]; then

userdel $username

echo "User $username deleted."

else

echo "Invalid action. Use 'add' or 'del'."

exit 1

fi

**5. Server Health Check Script**

**Purpose:** This script runs various health checks on a server, such as CPU load, memory usage, disk space, and running services, and generates a report.

**Example:**

bash

CopyEdit

#!/bin/bash

# Get server uptime

uptime=$(uptime -p)

# Get CPU usage

cpu\_usage=$(top -bn1 | grep "Cpu(s)" | sed "s/.\*, \*\([0-9.]\*\)%\* id.\*/\1/" | awk '{print 100 - $1"%"}')

# Get Memory usage

mem\_usage=$(free | grep Mem | awk '{print $3/$2 \* 100.0}')

# Get Disk usage

disk\_usage=$(df -h | grep "/$" | awk '{ print $5 }')

# Get running services

running\_services=$(systemctl list-units --type=service --state=running)

# Generate report

echo "Server Health Check Report"

echo "==========================="

echo "Uptime: $uptime"

echo "CPU Usage: $cpu\_usage"

echo "Memory Usage: $mem\_usage%"

echo "Disk Usage: $disk\_usage"

echo "Running Services: $running\_services"

**6. Automated Deployment Script**

**Purpose:** This script automates the process of deploying an application to a server, including pulling the latest code from a repository, installing dependencies, and restarting services.

**Example:**

bash

CopyEdit

#!/bin/bash

repo\_path="/var/www/myapp"

git\_repo="https://github.com/user/myapp.git"

service\_name="myapp"

# Navigate to the project directory

cd $repo\_path

# Pull latest changes from git

git pull $git\_repo

# Install dependencies

npm install

# Restart the service

systemctl restart $service\_name

echo "Deployment completed successfully!"

**7. Scheduled Task Automation Script (Cron Jobs)**

**Purpose:** This script automates tasks on a scheduled basis (using cron), such as system updates, data syncing, or cleanup operations.

**Example:**

bash

CopyEdit

#!/bin/bash

# Update package lists

apt-get update

# Upgrade installed packages

apt-get upgrade -y

# Clean up old package files

apt-get autoremove -y

apt-get clean

# Log the operation

echo "System updated on $(date)" >> /var/log/system\_update.log

These shell script projects are commonly used for system administration, automation, and monitoring tasks in real-world applications. They can be easily adapted and customized for various environments or requirements.

4o

**You said:**

continue with more real world examples please most used in real world for diff type of tasks

**ChatGPT said:**

Here are more real-world shell script examples for different tasks commonly used in various environments:

**8. System Security Check Script**

**Purpose:** This script checks for common security vulnerabilities, like open ports, outdated packages, and user access configurations. It helps to ensure the server is secure.

**Example:**

bash

CopyEdit

#!/bin/bash

# Check for open ports

echo "Checking open ports..."

netstat -tuln

# Check for outdated packages

echo "Checking for outdated packages..."

apt-get update && apt-get -u upgrade

# Check for root login attempts in logs

echo "Checking for root login attempts..."

grep "root" /var/log/auth.log | grep "session opened"

# Check user accounts with no password

echo "Checking accounts with no password..."

awk -F: '($2 == "") {print $1}' /etc/shadow

echo "Security check completed."

**9. Automated File Search Script**

**Purpose:** This script automates the process of searching files by name, type, or content in a specific directory. It is useful for sysadmins or developers who need to find files across a large filesystem.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define search parameters

search\_dir="/home/user/documents"

search\_term="report"

# Search files by name

echo "Searching for files containing '$search\_term' in their name..."

find $search\_dir -type f -name "\*$search\_term\*"

# Search files by content

echo "Searching for files containing '$search\_term' in their content..."

grep -r "$search\_term" $search\_dir

**10. Email Notification Script**

**Purpose:** This script sends email notifications based on specific system events, like a critical error or a backup completion.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define email variables

recipient="admin@example.com"

subject="System Alert: Disk Usage High"

message="Disk usage on the server has exceeded 90%. Please check."

# Send an email notification

echo "$message" | mail -s "$subject" $recipient

**11. File Integrity Check Script**

**Purpose:** This script is used to verify that files or directories have not been altered, typically used for system monitoring and ensuring the integrity of important files.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define file paths to check

files\_to\_check="/etc/passwd /etc/ssh/sshd\_config"

# Check for file integrity using checksum (md5sum)

for file in $files\_to\_check; do

if [ -f "$file" ]; then

echo "Checking file integrity for $file..."

checksum=$(md5sum $file | awk '{ print $1 }')

echo "MD5 checksum for $file: $checksum"

else

echo "File $file not found!"

fi

done

**12. Database Backup Script**

**Purpose:** Automates the process of backing up databases, particularly useful for database administrators to ensure regular backups are performed.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define database credentials

db\_name="mydatabase"

db\_user="root"

db\_password="password"

backup\_dir="/backups"

timestamp=$(date +"%Y-%m-%d\_%H-%M-%S")

backup\_file="$backup\_dir/$db\_name\_$timestamp.sql"

# Backup the database

mysqldump -u $db\_user -p$db\_password $db\_name > $backup\_file

# Optional: Delete backups older than 7 days

find $backup\_dir -type f -name "\*.sql" -mtime +7 -exec rm {} \;

echo "Database backup completed: $backup\_file"

**13. System Update and Patch Management Script**

**Purpose:** This script automates the process of updating and patching a system, making sure that the server is up-to-date with the latest patches and security updates.

**Example:**

bash

CopyEdit

#!/bin/bash

# Update package lists

echo "Updating package lists..."

apt-get update

# Upgrade packages

echo "Upgrading packages..."

apt-get upgrade -y

# Clean up obsolete packages

echo "Removing obsolete packages..."

apt-get autoremove -y

echo "System update and patching complete."

**14. Automated Web Scraping Script**

**Purpose:** This script can be used for scraping data from websites (e.g., product prices, weather data, or news headlines) and saving it into a file for further analysis.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define the URL to scrape

url="https://example.com/products"

# Use curl to fetch the webpage and grep to extract specific data

curl -s $url | grep -oP 'product-name">.\*?</span>' | sed 's/<[^>]\*>//g' > scraped\_data.txt

echo "Data scraping completed. Results saved to scraped\_data.txt."

**15. Log Rotation and Archiving Script**

**Purpose:** This script helps in managing and rotating log files, archiving old logs, and ensuring that disk space is not consumed by log files.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define log directory and archive directory

log\_dir="/var/log"

archive\_dir="/var/log/archive"

timestamp=$(date +"%Y-%m-%d\_%H-%M-%S")

# Archive the old logs

for log\_file in $log\_dir/\*.log; do

if [ -f "$log\_file" ]; then

mv $log\_file $archive\_dir/$(basename $log\_file)\_$timestamp.log

fi

done

# Compress archived logs

tar -czf $archive\_dir/logs\_$timestamp.tar.gz $archive\_dir/\*.log\_$timestamp.log

# Clean up old archives (older than 30 days)

find $archive\_dir -type f -name "\*.tar.gz" -mtime +30 -exec rm {} \;

echo "Log rotation and archiving complete."

**16. Scheduled Server Reboot Script**

**Purpose:** This script schedules the reboot of a server during low-traffic hours to apply updates or maintenance.

**Example:**

bash

CopyEdit

#!/bin/bash

# Schedule a reboot at 2:00 AM every Sunday

echo "Scheduling a reboot at 2:00 AM every Sunday..."

echo "0 2 \* \* SUN /sbin/shutdown -r now" | crontab -

echo "Reboot scheduled successfully."

**17. Web Server Health Check Script**

**Purpose:** This script checks the health of a web server by verifying if the web service is running and if the server can respond to HTTP requests.

**Example:**

bash

CopyEdit

#!/bin/bash

# Check if Apache is running

service apache2 status > /dev/null

if [ $? -eq 0 ]; then

echo "Apache server is running."

else

echo "Apache server is not running. Attempting to start..."

service apache2 start

fi

# Check if the web server is responding

curl -s -o /dev/null -w "%{http\_code}" http://localhost

if [ $? -eq 200 ]; then

echo "Web server is responding."

else

echo "Web server is down. Please check the configuration."

fi

These are more real-world shell script examples that can automate various daily tasks in system administration, development, backup, monitoring, security, and web operations. They offer practical solutions for a variety of environments and are highly adaptable to specific needs.

4o

**You said:**

continue with more examples please that are used in real world

**ChatGPT said:**

Certainly! Here are more real-world shell script examples used for different tasks in various environments:

**18. File Synchronization Script**

**Purpose:** This script is used to synchronize files between two directories, which is useful for backup or file replication purposes.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define source and destination directories

source\_dir="/home/user/documents"

dest\_dir="/backup/documents"

# Sync files from source to destination using rsync

rsync -av --delete $source\_dir/ $dest\_dir/

echo "Files synchronized successfully."

**19. Server Reboot After Maintenance Script**

**Purpose:** This script is designed to reboot a server after maintenance tasks, such as updates or configuration changes, have been applied.

**Example:**

bash

CopyEdit

#!/bin/bash

# Apply system updates

apt-get update -y && apt-get upgrade -y

# Reboot the server

echo "Rebooting server after maintenance..."

reboot

**20. Automated SSL Certificate Renewal Script**

**Purpose:** Automates the process of renewing SSL certificates on a server, which is vital for maintaining secure connections to websites.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define the path to the SSL certificate

cert\_dir="/etc/ssl/certs"

cert\_name="mydomain.com.crt"

# Renew the SSL certificate using certbot

certbot renew --quiet

# Reload the web server to apply the new certificate

service apache2 reload

echo "SSL certificate renewed and server reloaded successfully."

**21. File Compression and Archiving Script**

**Purpose:** This script compresses files and directories into tarballs or zip files, often used for backup or transport purposes.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define the directory to compress and the output file name

source\_dir="/home/user/data"

archive\_name="/backup/data\_backup\_$(date +"%Y-%m-%d").tar.gz"

# Create a tar.gz archive

tar -czf $archive\_name $source\_dir

# Optionally delete files older than 30 days from the backup folder

find /backup -type f -name "\*.tar.gz" -mtime +30 -exec rm {} \;

echo "Backup compressed and archived: $archive\_name"

**22. User Account Cleanup Script**

**Purpose:** This script removes inactive user accounts and associated files after a specified period of inactivity, helping maintain a clean system.

**Example:**

bash

CopyEdit

#!/bin/bash

# Set the number of days of inactivity before removal

inactive\_days=90

# Find users who have been inactive for more than the specified days

for user in $(awk -F: '{if ($5 ~ /inactive/ && $7 != "/sbin/nologin") print $1}' /etc/passwd); do

last\_login=$(lastlog -u $user | awk 'NR==2 {print $4}')

days\_inactive=$(( $(date +%s) - $(date -d $last\_login +%s) ))

if (( days\_inactive / 86400 >= inactive\_days )); then

echo "Removing inactive user: $user"

userdel $user

# Optionally delete user files

rm -rf /home/$user

fi

done

**23. Automated Data Import/Export Script**

**Purpose:** This script automates the process of importing data into a database and exporting it to a file, often used in data migration or backup scenarios.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define database and file paths

db\_name="mydatabase"

db\_user="root"

db\_password="password"

export\_file="/backups/database\_backup\_$(date +"%Y-%m-%d").sql"

# Export data from the database

mysqldump -u $db\_user -p$db\_password $db\_name > $export\_file

# Import data into the database

mysql -u $db\_user -p$db\_password $db\_name < $export\_file

echo "Data export and import completed successfully."

**24. Disk Space Alert Script**

**Purpose:** This script monitors disk space usage and sends alerts if any partition exceeds a specified threshold, which helps to prevent disk full errors.

**Example:**

bash

CopyEdit

#!/bin/bash

# Set the threshold (e.g., 80% disk usage)

threshold=80

# Get disk usage percentage for the root partition

disk\_usage=$(df / | grep / | awk '{ print $5 }' | sed 's/%//g')

if [ $disk\_usage -gt $threshold ]; then

echo "Warning: Disk usage is above $threshold%. Current usage is $disk\_usage%" | mail -s "Disk Space Alert" admin@example.com

fi

**25. Cron Job Scheduler for Regular Tasks**

**Purpose:** This script schedules regular system tasks like backups, updates, or data syncing using cron jobs, allowing for automation without manual intervention.

**Example:**

bash

CopyEdit

#!/bin/bash

# Schedule a task to run a backup every day at 2 AM

echo "Scheduling daily backup at 2 AM..."

echo "0 2 \* \* \* /home/user/backup.sh" | crontab -

# Schedule a task to check server health every hour

echo "Scheduling server health check every hour..."

echo "0 \* \* \* \* /home/user/health\_check.sh" | crontab -

echo "Cron jobs scheduled successfully."

**26. Automated Website Deployment Script**

**Purpose:** This script automates the deployment of a website, ensuring that the latest code changes are reflected on the server.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define web directory and repository

web\_dir="/var/www/mywebsite"

git\_repo="https://github.com/user/mywebsite.git"

# Pull the latest code from the repository

cd $web\_dir

git pull $git\_repo

# Restart the web server to apply changes

systemctl restart apache2

echo "Website deployed successfully!"

**27. Network Interface Check Script**

**Purpose:** This script checks the status of a network interface and can notify the system administrator if the interface is down.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define the network interface to check

iface="eth0"

# Check the status of the network interface

iface\_status=$(cat /sys/class/net/$iface/operstate)

if [ "$iface\_status" != "up" ]; then

echo "Network interface $iface is down. Please check!" | mail -s "Network Alert" admin@example.com

else

echo "Network interface $iface is up and running."

fi

**28. Automated System Clean-Up Script**

**Purpose:** This script helps to clean up the system by removing unnecessary files, such as cached files or old logs, to free up disk space.

**Example:**

bash

CopyEdit

#!/bin/bash

# Clean up system logs older than 30 days

find /var/log -type f -name "\*.log" -mtime +30 -exec rm {} \;

# Clean up apt cache

apt-get clean

# Remove orphaned packages

apt-get autoremove -y

echo "System cleanup completed."

**29. Automated System Monitoring with Alerts**

**Purpose:** This script monitors system metrics like CPU usage, memory usage, and disk space, and sends an alert if they exceed a threshold.

**Example:**

bash

CopyEdit

#!/bin/bash

# Set thresholds for CPU and memory usage

cpu\_threshold=90

mem\_threshold=80

# Check CPU usage

cpu\_usage=$(top -bn1 | grep "Cpu(s)" | sed "s/.\*, \*\([0-9.]\*\)%\* id.\*/\1/" | awk '{print 100 - $1}')

if [ $(echo "$cpu\_usage > $cpu\_threshold" | bc) -eq 1 ]; then

echo "CPU usage is above threshold: $cpu\_usage%" | mail -s "CPU Usage Alert" admin@example.com

fi

# Check memory usage

mem\_usage=$(free | grep Mem | awk '{print $3/$2 \* 100.0}')

if [ $(echo "$mem\_usage > $mem\_threshold" | bc) -eq 1 ]; then

echo "Memory usage is above threshold: $mem\_usage%" | mail -s "Memory Usage Alert" admin@example.com

fi

**30. SSH Key Management Script**

**Purpose:** This script automates the management of SSH keys, including adding, removing, or verifying authorized keys on a server.

**Example:**

bash

CopyEdit

#!/bin/bash

# Add an SSH key to the authorized\_keys file

echo "Adding SSH key..."

cat /path/to/public\_key >> ~/.ssh/authorized\_keys

# Remove an SSH key from the authorized\_keys file

echo "Removing SSH key..."

sed -i '/<key-to-remove>/d' ~/.ssh/authorized\_keys

echo "SSH key management completed."

These additional examples represent a wide range of real-world use cases for shell scripts, from system maintenance and monitoring to web deployment and network management. These scripts can help automate repetitive tasks, enhance security, and ensure system stability.

4o

**You said:**

continue please with more use that are used in real world please

**ChatGPT said:**

Here are more real-world shell script examples that are commonly used in various industries:

**31. Automated Cloud Backup Script**

**Purpose:** This script automates the backup of important files or directories to a cloud storage service like AWS S3, Google Cloud Storage, or Azure Blob Storage.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define local directory and cloud storage

local\_dir="/home/user/data"

cloud\_bucket="s3://my-backup-bucket/data"

# Sync the local directory with the cloud storage

aws s3 sync $local\_dir $cloud\_bucket --delete

echo "Cloud backup completed successfully."

**32. Website Availability Monitoring Script**

**Purpose:** This script monitors the availability of a website by periodically checking if the site is up and sends an alert if it goes down.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define the URL to monitor

url="https://example.com"

# Check if the website is up

http\_status=$(curl -o /dev/null -s -w "%{http\_code}" $url)

if [ "$http\_status" -ne 200 ]; then

echo "Website is down. HTTP Status: $http\_status" | mail -s "Website Down Alert" admin@example.com

else

echo "Website is up and running."

fi

**33. Automated Server Hardening Script**

**Purpose:** This script performs a series of security tasks to harden a server, such as disabling root login, setting up firewalls, and applying necessary security patches.

**Example:**

bash

CopyEdit

#!/bin/bash

# Disable root login via SSH

sed -i 's/^PermitRootLogin yes/PermitRootLogin no/' /etc/ssh/sshd\_config

systemctl restart sshd

# Set up a basic firewall

ufw allow OpenSSH

ufw enable

ufw default deny incoming

ufw default allow outgoing

# Apply security updates

apt-get update && apt-get upgrade -y

echo "Server hardening completed."

**34. File Synchronization with Rsync Script**

**Purpose:** This script automates file synchronization between a local server and a remote server or between two remote servers, ensuring that the most recent versions of files are available in both locations.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define the local directory and remote server details

local\_dir="/home/user/data"

remote\_server="user@remote\_server:/home/user/backup"

# Synchronize files using rsync

rsync -avz --delete $local\_dir $remote\_server

echo "File synchronization completed successfully."

**35. Disk Cleanup and Log Rotation Script**

**Purpose:** This script deletes old log files and temporary files, helping to prevent the system from running out of disk space due to excessive logs.

**Example:**

bash

CopyEdit

#!/bin/bash

# Clean up old log files older than 30 days

find /var/log -type f -name "\*.log" -mtime +30 -exec rm -f {} \;

# Clean up temporary files

rm -rf /tmp/\*

echo "Disk cleanup and log rotation completed."

**36. Automated Database Optimization Script**

**Purpose:** This script helps optimize the performance of a MySQL database by running maintenance tasks like optimizing tables, updating statistics, and repairing corrupted tables.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define database credentials

db\_user="root"

db\_password="password"

db\_name="mydatabase"

# Optimize all tables in the database

mysqlcheck -u $db\_user -p$db\_password --optimize $db\_name

# Repair any corrupted tables

mysqlcheck -u $db\_user -p$db\_password --repair $db\_name

echo "Database optimization completed."

**37. Disk Space Usage Notification Script**

**Purpose:** This script monitors disk space and sends a notification when the disk usage exceeds a specified threshold, helping to prevent server downtime due to a full disk.

**Example:**

bash

CopyEdit

#!/bin/bash

# Set the threshold (e.g., 90% disk usage)

threshold=90

# Get the disk usage of the root partition

disk\_usage=$(df / | grep / | awk '{ print $5 }' | sed 's/%//g')

if [ $disk\_usage -gt $threshold ]; then

echo "Disk usage on root partition is above $threshold%. Current usage is $disk\_usage%" | mail -s "Disk Space Alert" admin@example.com

fi

**38. Server Load Monitoring and Alerting Script**

**Purpose:** This script monitors the load average on a server and sends an alert if the load average exceeds a set threshold.

**Example:**

bash

CopyEdit

#!/bin/bash

# Set the threshold (e.g., 2.0)

threshold=2.0

# Get the load average (1-minute load)

load\_avg=$(uptime | awk -F'load average: ' '{ print $2 }' | cut -d',' -f1)

# Check if the load average exceeds the threshold

if (( $(echo "$load\_avg > $threshold" | bc -l) )); then

echo "Server load is above the threshold: $load\_avg" | mail -s "High Load Alert" admin@example.com

fi

**39. Automated SSL Certificate Expiry Check Script**

**Purpose:** This script checks the expiration date of SSL certificates for a website and sends an alert if the certificate is about to expire.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define the domain to check

domain="example.com"

# Get the SSL certificate expiration date

expiry\_date=$(echo | openssl s\_client -connect $domain:443 2>/dev/null | openssl x509 -noout -dates | grep 'notAfter=' | cut -d= -f2)

# Convert the expiration date to Unix timestamp

expiry\_timestamp=$(date -d "$expiry\_date" +%s)

current\_timestamp=$(date +%s)

days\_left=$(( (expiry\_timestamp - current\_timestamp) / 86400 ))

# Send an alert if the certificate expires in less than 30 days

if [ $days\_left -lt 30 ]; then

echo "SSL certificate for $domain is expiring in $days\_left days." | mail -s "SSL Certificate Expiry Alert" admin@example.com

fi

**40. Automated Disk Partitioning Script**

**Purpose:** This script automates the process of creating new partitions and file systems, useful during server setup or expansion.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define the disk and partition details

disk="/dev/sdb"

partition="${disk}1"

mount\_point="/mnt/data"

# Create a new partition

parted $disk mklabel gpt

parted -a optimal $disk mkpart primary ext4 0% 100%

# Create a new file system on the partition

mkfs.ext4 $partition

# Mount the new partition

mount $partition $mount\_point

echo "New partition created and mounted at $mount\_point."

**41. Email Backup and Restore Script**

**Purpose:** This script automates the backup and restore of email accounts for an email server (e.g., using Dovecot or Postfix).

**Example:**

bash

CopyEdit

#!/bin/bash

# Define email directory and backup location

email\_dir="/var/mail"

backup\_dir="/backup/email"

# Backup emails

tar -czf $backup\_dir/email\_backup\_$(date +"%Y-%m-%d").tar.gz $email\_dir

# Restore emails (example for restoring a specific backup)

# tar -xzf $backup\_dir/email\_backup\_2025-01-01.tar.gz -C $email\_dir

echo "Email backup completed successfully."

**42. Automated System Shutdown Script**

**Purpose:** This script schedules an automatic shutdown of the system at a specific time, useful for performing maintenance or power-saving tasks.

**Example:**

bash

CopyEdit

#!/bin/bash

# Schedule system shutdown at 11:00 PM

echo "Scheduling system shutdown at 11:00 PM..."

echo "0 23 \* \* \* /sbin/shutdown -h now" | crontab -

echo "Shutdown scheduled successfully."

**43. File Integrity Monitoring Script**

**Purpose:** This script checks if specific critical files have been modified or tampered with, typically used in a security context to ensure file integrity.

**Example:**

bash

CopyEdit

#!/bin/bash

# Define the critical files to monitor

files\_to\_monitor=("/etc/passwd" "/etc/ssh/sshd\_config")

# Check the integrity of the files

for file in "${files\_to\_monitor[@]}"; do

# Generate checksum and compare with the saved checksum

checksum=$(sha256sum $file)

if [ "$checksum" != "$(cat /var/log/$(basename $file)\_checksum)" ]; then

echo "Warning: File $file has been modified!"

# Optional: Send an alert

mail -s "File Integrity Alert" admin@example.com <<< "The file $file has been altered."

fi

done

These shell script examples represent a variety of real-world scenarios where automation is essential for system administration, backup, security, deployment, and monitoring. Shell scripts are highly adaptable and can save time, reduce errors, and improve system efficiency in a wide range of use cases.