1. **Managing Users and Groups:**

# Add a new user

useradd -m newuser

# Change user password

echo "newuser:password" | chpasswd

# Add a user to a group

usermod -aG sudo newuser

1. **File and Directory Operations**:

# Create a directory

mkdir /var/www/myapp

# Change directory ownership

chown -R newuser:newgroup /var/www/myapp

# Set directory permissions

chmod -R 755 /var/www/myapp

1. **Process Management**:

# List all running processes

ps aux

# Kill a process by name

pkill -f myprocess

# Start a process in the background

nohup ./myprocess &

1. **File Transfers**:

# Secure copy (SCP) files between servers

scp localfile user@remote:/path/to/destination

1. **Package Management Using Red hat**

# Update package list

yum update -y

# Install a package

yum install -y nginx

# Remove a package

yum remove -y nginx

1. **Git Commands for Automation**

# Clone a repository

git clone https://github.com/user/repo.git

# Check status

cd repo

git status

# Pull latest changes

git pull origin main

# Commit and push changes

git add .

git commit -m "Automated commit message"

git push origin main

1. **Docker Commands**

# Build a Docker image

docker build -t myapp .

# Run a Docker container

docker run -d -p 80:80 myapp

# List running containers

docker ps

# Stop and remove a container

docker stop myapp

docker rm myapp

1. **Log Management**:

# View system logs

tail -f /var/log/syslog

# Search logs for specific keywords

grep "error" /var/log/syslog

1. **Shell Script Examples for DevOps Tasks**
   1. **Automated Backup Script**:

#!/bin/bash

# Script to backup a directory

SOURCE\_DIR="/path/to/source"

BACKUP\_DIR="/path/to/backup"

TIMESTAMP=$(date +%Y%m%d%H%M%S)

BACKUP\_FILE="$BACKUP\_DIR/backup-$TIMESTAMP.tar.gz"

tar -czvf $BACKUP\_FILE $SOURCE\_DIR

echo "Backup completed: $BACKUP\_FILE"

* 1. **Deployment Script:**

#!/bin/bash

# Script to deploy a web application

# Pull the latest changes from the repository

git pull origin main

# Build the Docker image

docker build -t myapp .

# Stop the existing container

docker stop myapp

docker rm myapp

# Run the new container

docker run -d -p 80:80 myapp

echo "Deployment completed."