**1 .Title:- Bash Script for Server Setup**

**Script File: server\_setup.sh**

#!/bin/bash

# Automate Basic Server Setup

# This script will:

# 1. Update the system

# 2. Install essential packages

# 3. Create a new user

# 4. Configure the firewall

# Step 1: Update the system

echo "Updating system packages..."

sudo apt-get update -y

sudo apt-get upgrade -y

# Step 2: Install essential packages

echo "Installing essential packages..."

sudo apt-get install -y git curl wget ufw

# Step 3: Create a new user

echo "Creating a new user..."

read -p "Enter the username: " username

sudo adduser $username

sudo usermod -aG sudo $username # Add user to sudo group

# Step 4: Configure the firewall

echo "Configuring the firewall..."

sudo ufw allow ssh

sudo ufw allow http

sudo ufw allow https

sudo ufw enable

echo "Server setup complete!"

**Explanation of the Script**

1. **System Update:**
   * **The script starts by updating the system packages using apt-get update and apt-get upgrade.**
2. **Install Essential Packages:**
   * **It installs commonly used packages like git, curl, wget, and ufw (Uncomplicated Firewall).**
3. **Create a New User:**
   * **The script prompts the user to enter a username and creates a new user with adduser.**
   * **The user is added to the sudo group to grant administrative privileges.**
4. **Configure Firewall:**
   * **The script enables the firewall (ufw) and allows traffic for ssh, http, and https.**
5. **Completion Message:**
   * **Once all steps are completed, the script prints a message indicating that the server setup is done.**

**How to Run the Script**

1. **Save the script as server\_setup.sh.**
2. **Make the script executable:**

chmod +x server\_setup.sh

1. **Run the script:**

./server\_setup.sh

**2.  Python Script for Log Analysis**

**Script File: log\_analysis.py**

# Log Analysis Script

# This script will:

# 1. Read a log file

# 2. Count occurrences of specific keywords (e.g., "ERROR", "WARNING")

# 3. Generate a summary report

import re

def analyze\_log(file\_path, keywords):

# Dictionary to store keyword counts

keyword\_counts = {keyword: 0 for keyword in keywords}

# Read the log file

with open(file\_path, 'r') as file:

for line in file:

for keyword in keywords:

if re.search(keyword, line, re.IGNORECASE):

keyword\_counts[keyword] += 1

# Print the summary report

print("Log Analysis Report:")

print("====================")

for keyword, count in keyword\_counts.items():

print(f"{keyword}: {count} occurrences")

# Define the log file path and keywords to search for

log\_file = "server.log"

keywords = ["ERROR", "WARNING", "INFO"]

# Run the analysis

analyze\_log(log\_file, keywords)

**Explanation of the Script**

1. **Read the Log File:**
   * **The script reads a log file (server.log) line by line.**
2. **Count Keyword Occurrences:**
   * **It searches for specific keywords (e.g., ERROR, WARNING, INFO) in each line of the log file.**
   * **The script uses regular expressions (re.search) to perform case-insensitive searches.**
3. **Generate a Summary Report:**
   * **The script maintains a count of how many times each keyword appears in the log file.**
   * **It prints a summary report showing the number of occurrences for each keyword.**

**How to Run the Script**

1. **Save the script as log\_analysis.py.**
2. **Ensure you have a log file named server.log in the same directory (or update the log\_file variable with the correct path).**
3. **Run the script:**

python3 log\_analysis.py

**Sample Output**

**If the log file contains the following lines:**

**[INFO] System started**

**[ERROR] Failed to connect to database**

**[WARNING] Disk usage is high**

**[INFO] Backup completed**

**[ERROR] Invalid user input**

The script will output:

Log Analysis Report:

====================

**ERROR: 2 occurrences**

**WARNING: 1 occurrences**

**INFO: 2 occurrences**