### 1. \*System Health Monitoring Script\*

This script checks CPU usage, memory usage, disk space, and running processes. If any metrics exceed predefined thresholds, it logs an alert.

python

import psutil

import logging

import datetime

# Configure logging

logging.basicConfig(filename='system\_health.log', level=logging.INFO)

# Define thresholds

CPU\_THRESHOLD = 80.0

MEMORY\_THRESHOLD = 80.0

DISK\_THRESHOLD = 80.0

def log\_alert(message):

logging.error(f"{datetime.datetime.now()}: {message}")

# Check CPU usage

cpu\_usage = psutil.cpu\_percent(interval=1)

if cpu\_usage > CPU\_THRESHOLD:

log\_alert(f"High CPU Usage: {cpu\_usage}%")

# Check Memory usage

memory\_info = psutil.virtual\_memory()

if memory\_info.percent > MEMORY\_THRESHOLD:

log\_alert(f"High Memory Usage: {memory\_info.percent}%")

# Check Disk usage

disk\_info = psutil.disk\_usage('/')

if disk\_info.percent > DISK\_THRESHOLD:

log\_alert(f"High Disk Usage: {disk\_info.percent}%")

# Check running processes

processes = psutil.pids()

logging.info(f"Currently running processes: {len(processes)}")

### 2. \*Automated Backup Solution\*

This script automates the backup of a specified directory to a remote server using rsync. It reports on the success or failure of the operation.

python

import os

import subprocess

import logging

import datetime

# Configure logging

logging.basicConfig(filename='backup\_report.log', level=logging.INFO)

# Backup configurations

SOURCE\_DIR = '/path/to/source\_directory' # Replace with your source directory

REMOTE\_DIR = 'user@remote\_host:/path/to/backup\_directory' # Replace with your remote directory

def backup\_directory(source, remote):

try:

# Run rsync command to backup

result = subprocess.run(['rsync', '-avz', source, remote], check=True, capture\_output=True, text=True)

logging.info(f"{datetime.datetime.now()}: Backup successful. Output:\n{result.stdout}")

except subprocess.CalledProcessError as e:

logging.error(f"{datetime.datetime.now()}: Backup failed. Error:\n{e.stderr}")

backup\_directory(SOURCE\_DIR, REMOTE\_DIR)

### Instructions to Run the Scripts

1. \*System Health Monitoring Script\*:

- Ensure you have the psutil library installed. You can install it via pip:

bash

pip install psutil

- Save the script to a file, e.g., health\_monitor.py, and run it:

bash

python health\_monitor.py

2. \*Automated Backup Solution\*:

- Ensure rsync is installed on your system and that you have SSH access to the remote server.

- Save the script to a file, e.g., backup.py, and update the SOURCE\_DIR and REMOTE\_DIR variables.

- Run the script:

bash

python backup.py

### Conclusion

These scripts provide a basic framework for system monitoring and backup automation. You can expand them further based on specific requirements or additional functionality.