

Task 6: Automation & Scripting

This task focuses on automating **server setup or log analysis** using a **Bash or Python script**. Automation helps in **reducing manual effort, improving efficiency, and ensuring consistency** in DevOps workflows.

Overview

Automation is essential for:

- ✓ **Consistent infrastructure provisioning** (server setup).
- ✓ **Faster troubleshooting** (log analysis).
- ✓ **Reducing human errors** (automating repetitive tasks).

I will provide **two scripts**:

1. **Bash script for server setup.**
 2. **Python script for log analysis.**
-

1. Automating Server Setup (Bash Script)

♦ Purpose

This Bash script **automates the installation of essential packages** (Nginx, Docker, Python) and **configures firewall settings** on an Ubuntu server.

♦ Bash Script (**server_setup.sh**)

```
sh
```

```
CopyEdit
```

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

```
echo "🚀 Starting Server Setup..."
```

```
# Update system
```

```
sudo apt update && sudo apt upgrade -y
```

```
# Install essential packages
sudo apt install -y nginx docker.io python3 python3-pip

# Enable & start services
sudo systemctl enable nginx
sudo systemctl start nginx

sudo systemctl enable docker
sudo systemctl start docker

# Configure Firewall
sudo ufw allow 'Nginx Full'
sudo ufw allow OpenSSH
sudo ufw enable

echo "✅ Server setup completed successfully!"
```

♦ How to Run the Script

1. **Save the script** as `server_setup.sh`.

Make it executable:

```
sh
CopyEdit
chmod +x server_setup.sh
```

- 2.

Run the script:

```
sh
CopyEdit
./server_setup.sh
```

- 3.

✅ **This script ensures the server is set up with necessary services.**

2. Automating Log Analysis (Python Script)

♦ Purpose

This Python script **analyzes log files** and **detects errors** in real-time.

♦ Python Script (**log_analysis.py**)

```
python
CopyEdit
import re

log_file = "/var/log/syslog" # Change this to your log file path

def analyze_logs():
    error_count = 0

    with open(log_file, "r") as file:
        for line in file:
            if "error" in line.lower():
                error_count += 1
                print(f"⚠ ERROR: {line.strip()}") # Print detected
error

    print(f"\n✅ Log Analysis Completed: {error_count} errors found.")

if __name__ == "__main__":
    analyze_logs()
```

♦ How to Run the Script

1. **Save the script** as **log_analysis.py**.

Run the script:

```
sh
CopyEdit
python3 log_analysis.py
```

- ✓ The script scans logs and highlights errors for debugging.

Automating Server Setup (Bash Script)

$$:q|$$

```
sadik@monitoring: ~  
sadik@monitoring:~$  
sadik@monitoring:~$ vim server_setup.sh  
sadik@monitoring:~$ sadik@monitoring:~$  
sadik@monitoring:~$  
sadik@monitoring:~$ chmod +x server_setup.sh  
sadik@monitoring:~$ ./server_setup.sh  
Starting Server Setup...  
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease  
Get:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Get:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]  
Get:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Get:5 https://packages.grafana.com/oss/deb stable InRelease [7660 B]  
Hit:6 https://artifacts.elastic.co/packages/8.x/apt stable InRelease  
Get:7 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]  
Get:8 http://azure.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [363 kB]  
Get:9 http://azure.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]  
Get:10 http://azure.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]  
Get:11 http://azure.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]  
Get:12 http://azure.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]  
Get:13 http://azure.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]  
Get:14 http://azure.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]  
Get:15 http://azure.archive.ubuntu.com/ubuntu noble-security/main amd64 Components [8988 B]  
Get:16 http://azure.archive.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.0 kB]  
Get:17 http://azure.archive.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]  
Get:18 http://azure.archive.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]  
Err:5 https://packages.grafana.com/oss/deb stable InRelease  
The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 963FA27710458545  
Reading package lists... Done  
W: GPG error: https://packages.grafana.com/oss/deb stable InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 963FA27710458545  
E: The repository 'https://packages.grafana.com/oss/deb stable InRelease' is not signed.  
N: Updating from such a repository can't be done securely, and is therefore disabled by default.  
N: See apt-secure(8) manpage for repository creation and user configuration details.  
Reading package lists... Done
```

```
sadik@monitoring: ~  
Setting up nginx (1.24.0-2ubuntu7.1) ...  
Setting up libheif1:amd64 (1.17.6-1ubuntu4.1) ...  
Setting up nginx-common (1.24.0-2ubuntu7.1) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.  
Setting up libgd3:amd64 (2.3.3-9ubuntu5) ...  
Setting up libc-devtools (2.39-0ubuntu8.3) ...  
Setting up libheif-plugin-aomdec:amd64 (1.17.6-1ubuntu4.1) ...  
Setting up libheif-plugin-libde265:amd64 (1.17.6-1ubuntu4.1) ...  
Setting up libheif-plugin-aomenc:amd64 (1.17.6-1ubuntu4.1) ...  
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...  
Processing triggers for ufw (0.36.2-6) ...  
Processing triggers for man-db (2.12.0-4build2) ...  
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...  
Scanning processes...  
Scanning linux images...  
  
Running kernel seems to be up-to-date.  
  
No services need to be restarted.  
  
No containers need to be restarted.  
  
No user sessions are running outdated binaries.  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
Synchronizing state of nginx.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx  
Rules updated  
Rules updated (v6)  
Rules updated  
Rules updated (v6)  
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y  
Firewall is active and enabled on system startup  
✔ Server setup completed successfully!  
sadik@monitoring:~$
```

Automating Log Analysis (Python Script)

```
sadik@monitoring: ~  
import re  
  
log_file = "/var/log/syslog" # Change this to your log file path  
  
def analyze_logs():  
    error_count = 0  
  
    with open(log_file, "r") as file:  
        for line in file:  
            if "error" in line.lower():  
                error_count += 1  
                print(f"▲ERROR: {line.strip()}") # Print detected error  
  
    print(f"\n✅ Log Analysis Completed: {error_count} errors found.")  
  
if __name__ == "__main__":  
    analyze_logs()  
  
"log_analysis.py" [New] 18L, 456B written 18,0-1 All
```

```
sadik@monitoring: ~  
sadik@monitoring:~$ vim log_analysis.py  
sadik@monitoring:~$ python3 log_analysis.py  
▲ ERROR: 2025-02-06T09:42:03.420664+00:00 monitoring kernel: GPT: Use GNU Parted to correct GPT errors.  
▲ ERROR: 2025-02-06T09:42:03.420699+00:00 monitoring kernel: RAS: Correctable Errors collector initialized.  
▲ ERROR: 2025-02-06T09:42:03.428772+00:00 monitoring systemd[1]: apport-autoreport.path - Process error reports when automatic reporting is enabled (file watch) was skipped because of an unmet condition check (ConditionPathExists=/var/lib/apport/autoreport).  
▲ ERROR: 2025-02-06T09:42:03.428783+00:00 monitoring systemd[1]: apport-autoreport.timer - Process error reports when automatic reporting is enabled (timer based) was skipped because of an unmet condition check (ConditionPathExists=/var/lib/apport/autoreport).  
▲ ERROR: 2025-02-06T09:42:03.437099+00:00 monitoring kernel: I/O error, dev sr0, sector 0 op 0x0:(READ) flags 0x80700 phys_seg 1 prio class 0  
▲ ERROR: 2025-02-06T09:42:03.437106+00:00 monitoring kernel: I/O error, dev sr0, sector 0 op 0x0:(READ) flags 0x0 phys_seg 1 prio class 0  
▲ ERROR: 2025-02-06T09:42:03.437107+00:00 monitoring kernel: Buffer I/O error on dev sr0, logical block 0, async page read  
▲ ERROR: 2025-02-06T09:42:03.893019+00:00 monitoring snapd[947]: helpers.go:160: error trying to compare the snap system key: system-key missing on disk  
▲ ERROR: 2025-02-06T09:42:12.365174+00:00 monitoring python3[1446]: 2025-02-06T09:42:12.365082Z INFO ExtHandler ExtHandler [HEARTBEAT] Agent WALinuxAgent-2.12.0.2 is running as the goal state agent [DEBUG HeartbeatCounter: 0;HeartbeatId: 22452E8A-723E-4792-A3A6-B56557B9A961;DroppedPackets: 0;UpdateGSErrors: 0;AutoUpdate: 1;UpdateMode: SelfUpdate;]  
▲ ERROR: 2025-02-06T09:49:09.098476+00:00 monitoring prometheus[3225]: ts=2025-02-06T09:49:09.079Z caller=query_logger.go:91 level=error component=activeQueryTracker msg="Error opening query log file" file=/var/lib/prometheus/metrics2/queries.active err="open /var/lib/prometheus/metrics2/queries.active: permission denied"  
▲ ERROR: 2025-02-06T10:12:13.535359+00:00 monitoring python3[1446]: 2025-02-06T10:12:13.534900Z INFO ExtHandler ExtHandler [HEARTBEAT] Agent WALinuxAgent-2.12.0.2 is running as the goal state agent [DEBUG HeartbeatCounter: 1;HeartbeatId: 22452E8A-723E-4792-A3A6-B56557B9A961;DroppedPackets: 0;UpdateGSErrors: 0;AutoUpdate: 1;UpdateMode: SelfUpdate;]  
▲ ERROR: 2025-02-06T10:42:13.609627+00:00 monitoring python3[1446]: 2025-02-06T10:42:13.609408Z INFO ExtHandler ExtHandler [HEARTBEAT] Agent WALinuxAgent-2.12.0.2 is running as the goal state agent [DEBUG HeartbeatCounter: 2;HeartbeatId: 22452E8A-723E-4792-A3A6-B56557B9A961;DroppedPackets: 0;UpdateGSErrors: 0;AutoUpdate: 1;UpdateMode: SelfUpdate;]  
▲ ERROR: 2025-02-06T11:06:28.189598+00:00 monitoring prometheus-node-exporter[12424]: ts=2025-02-06T11:06:28.048Z caller=collector.go:169 level=error msg="collector failed" name=systemd duration_seconds=89.301690049 err="couldn't get dbus connection: write unix @->/run/dbus/system_bus_socket: write: broken pipe"  
▲ ERROR: 2025-02-06T11:11:43.497075+00:00 monitoring prometheus-node-exporter[12424]: ts=2025-02-06T11:11:34.337Z caller=collector.go:169 level=error msg="collector failed" name=systemd duration_seconds=125.393241303 err="couldn't get dbus connection: write unix @->/run/dbus/system_bus_socket: write: broken pipe"  
▲ ERROR: 2025-02-06T11:12:18.242357+00:00 monitoring python3[1446]: 2025-02-06T11:12:18.189828Z INFO ExtHandler ExtHandler [HEARTBEAT] Agent WALinuxAgent-2.12.0.2 is running as the goal state agent [DEBUG HeartbeatCounter: 3;HeartbeatId: 22452E8A-723E-4792-A3A6-B56557B9A961;DroppedPackets: 0;UpdateGSErrors: 0;AutoUpdate: 1;UpdateMode: SelfUpdate;]
```

```
sadik@monitoring: ~  
loading plugin "/io.containerd.snapshotter.v1.zfs" error="path /var/lib/containerd/io.containerd.snapshotter.v1.zfs must be a zfs  
filesystem to be used with the zfs snapshotter: skip plugin" type=io.containerd.snapshotter.v1  
ERROR: 2025-02-06T13:39:15.773599+00:00 monitoring containerd[20403]: time="2025-02-06T13:39:15.773046719Z" level=info msg="skip l  
loading plugin "/io.containerd.tracing.processor.v1.otlp" error="skip plugin: tracing endpoint not configured" type=io.containerd  
.tracing.processor.v1  
ERROR: 2025-02-06T13:39:15.773914+00:00 monitoring containerd[20403]: time="2025-02-06T13:39:15.773087419Z" level=info msg="skip l  
loading plugin "/io.containerd.internal.v1.tracing" error="skip plugin: tracing endpoint not configured" type=io.containerd.inter  
nal.v1  
ERROR: 2025-02-06T13:39:15.775447+00:00 monitoring containerd[20403]: time="2025-02-06T13:39:15.773292320Z" level=info msg="Start  
cri plugin with config {PluginConfig:{ContainerdConfig:{Snapshotter:overlayfs DefaultRuntimeName:runc DefaultRuntime:{Type: Path: Eng  
ine: PodAnnotations:[] ContainerAnnotations:[] Root: Options:map[] PrivilegedWithoutHostDevices:false PrivilegedWithoutHostDevicesAll  
DevicesAllowed:false BaseRuntimeSpec: NetworkPluginConfDir: NetworkPluginMaxConfNum:0 Snapshotter: SandboxMode:} UntrustedWorkloadRun  
time:{Type: Path: Engine: PodAnnotations:[] ContainerAnnotations:[] Root: Options:map[] PrivilegedWithoutHostDevices:false Privileged  
WithoutHostDevicesAllDevicesAllowed:false BaseRuntimeSpec: NetworkPluginConfDir: NetworkPluginMaxConfNum:0 Snapshotter: SandboxMode:}  
Runtimes:map[runc:{Type:io.containerd.runc.v2 Path: Engine: PodAnnotations:[] ContainerAnnotations:[] Root: Options:map[BinaryName:  
CriuImagePath: CriuPath: CriuWorkPath: IoGid:0 IoUid:0 NoNewKeyring:false NoPivotRoot:false Root: ShimCgroup: SystemdCgroup:false] Pr  
ivilegedWithoutHostDevices:false PrivilegedWithoutHostDevicesAllDevicesAllowed:false BaseRuntimeSpec: NetworkPluginConfDir: NetworkPl  
uginMaxConfNum:0 Snapshotter: SandboxMode:podssandbox}] NoPivot:false DisableSnapshotAnnotations:true DiscardUnpackedLayers:false Ign  
reBlockIONotEnabledErrors:false IgnoreRdtNotEnabledErrors:false} CniConfig:{NetworkPluginBinDir:/opt/cni/bin NetworkPluginConfDir:/et  
c/cni/net.d NetworkPluginMaxConfNum:1 NetworkPluginSetupSerially:false NetworkPluginConfTemplate: IPPreference:} Registry:{ConfigPath  
: Mirrors:map[] Configs:map[] Auths:map[] Headers:map[]}} ImageDecryption:{KeyModel:node} DisableTCPService:true StreamServerAddress:1  
27.0.0.1 StreamServerPort:0 StreamIdleTimeout:4h0m0s EnableSelinux:false SelinuxCategoryRange:1024 SandboxImage:registry.k8s.io/pause  
:3.8 StatsCollectPeriod:10 SystemdCgroup:false EnableTLSStreaming:false X509KeyPairStreaming:{TLSCertFile: TLSKeyFile:} MaxContainerL  
ogLineSize:16384 DisableCgroup:false DisableApparmor:false RestrictOOMScoreAdj:false MaxConcurrentDownloads:3 DisableProcMount:false  
UnsetSeccompProfile: TolerateMissingHugetlbController:true DisableHugetlbController:true DeviceOwnershipFromSecurityContext:false Ign  
oreImageDefinedVolumes:false NetNSMountsUnderStateDir:false EnableUnprivilegedPorts:false EnableUnprivilegedICMP:false EnableCDI:fa  
lse CDI SpecDirs:[/etc/cdi /var/run/cdi] ImagePullProgressTimeout:5m0s DrainExecSyncIOTimeout:0s ImagePullWithSyncFs:false IgnoreDepreca  
tionWarnings:[] ContainerdRootDir:/var/lib/containerd ContainerdEndpoint:/run/containerd/containerd.sock RootDir:/var/lib/containerd  
/io.containerd.grpc.v1.cri StateDir:/run/containerd/io.containerd.grpc.v1.cri}"  
ERROR: 2025-02-06T13:39:15.779482+00:00 monitoring containerd[20403]: time="2025-02-06T13:39:15.778816342Z" level=error msg="faile  
d to load cni during init, please check CRI plugin status before setting up network for pods" error="cni config load failed: no netwo  
rk config found in /etc/cni/net.d: cni plugin not initialized: failed to load cni config"  
  
Log Analysis Completed: 31 errors found.  
sadik@monitoring:~$
```

4. Conclusion

- ✓ Server setup is fully automated with a Bash script.
- ✓ Log analysis is streamlined using a Python script.
- ✓ Both scripts improve DevOps efficiency and reduce manual effort.

 Now, automation is successfully implemented! 🎉