DevOps Shack

LINUX Best Practices

Click Here To Enrol To Batch-5 | DevOps & Cloud DevOps

1. Keep System Updated

- Regularly update your system to ensure you have the latest security patches and software versions.

```
sudo apt update && sudo apt upgrade
```

2. Use Package Managers Efficiently

- Use `apt`, `yum`, `dnf`, `pacman`, or other package managers to install, update, and remove software.

```
sudo apt install package_name
```

3. Manage Services with Systemd

- Control services using `systemd` to start, stop, and manage services.
- `sudo systemctl start service_name`
- `sudo systemctl enable service_name`
  `sudo systemctl status service_name`

4. User and Group Management

- Add, modify, and delete users and groups to manage permissions and access control.
- `sudo adduser username`
- `sudo usermod -aG groupname username`
  `sudo deluser username`

5. File Permissions and Ownership

- Use `chmod`, `chown`, and `chgrp` to set appropriate file permissions and ownership.

-
  ```
  sudo chown user:group filename
  sudo chmod 755 filename
  ```

6. Use SSH for Remote Management

- Securely manage servers and remote systems using SSH.

```
ssh user@remote_host
```

7. Set Up SSH Key-Based Authentication

- Enhance security by using SSH keys instead of passwords.

-
  ```
  ssh-keygen
  ssh-copy-id user@remote_host
  ```

8. Monitor System Performance

- Use `top`, `htop`, and `glances` to monitor system performance and resource usage.

```
top
```

9. Automate Tasks with Cron Jobs

- Schedule and automate recurring tasks using cron.
- `crontab -e`

-
  ```
  # Add a job, e.g., to run a script every day at midnight
  0 0 * * * /path/to/script.sh
  ```

10. Use Aliases for Efficiency

- Create aliases to simplify and speed up command execution.

```
alias ll='ls -la'
```

11. Backup and Restore Data

- Regularly backup important data using tools like `rsync` or `tar`.

```
rsync -avh /source/directory /backup/directory
```

12. Use Scripting to Automate Tasks

- Write Bash scripts to automate repetitive tasks and processes.

-
  ```
  #!/bin/bash
  echo "Hello, World!"
  ```

13. Understand and Utilize Redirection

- Use `>`, `>>`, `2>`, and `|` to redirect output and errors.
- `ls > output.txt`
- `ls >> output.txt`

-
  ```
  ls 2> error.txt
  ls | grep pattern
  ```

14. Use Text Processing Tools

- Utilize `grep`, `awk`, `sed`, and `cut` for text processing and manipulation.
- `grep "search_term" file.txt`
- `awk '{print $1}' file.txt`

-
  ```
  sed 's/old/new/g' file.txt
  cut -d',' -f1 file.txt
  ```

15. Manage Disk Usage

- Use `df`, `du`, and `ncdu` to monitor and manage disk space.
- `df -h`

-
  ```
  du -sh /directory
  ncdu
  ```

16. Secure Your System with Firewalls

- Use `ufw`, `iptables`, or `firewalld` to configure firewalls.

-
  ```
  sudo ufw allow 22
  sudo ufw enable
  ```

17. Use Version Control

- Manage code and configurations using Git.
- `git init`

-
  ```
  git add .
  git commit -m "Initial commit"
  ```

18. Use Environment Variables

- Set and use environment variables for configuration and scripts.
- `export VAR_NAME=value`

```
echo $VAR_NAME
```

19. Secure Sensitive Information

- Store sensitive information in environment variables or use secret management tools.

```
export DB_PASSWORD='securepassword'
```

20. Set Up a Firewall

- Configure a firewall to protect your system from unauthorized access.

-
  ```
  sudo ufw enable
  sudo ufw allow ssh
  ```

21. Install Software from Source

- Compile and install software from source when necessary.
- `./configure`

-
  ```
  make
  sudo make install
  ```

22. Create and Manage Virtual Environments

- Use tools like `virtualenv` for Python projects to manage dependencies.

-
  ```
  virtualenv venv
  source venv/bin/activate
  ```

23. Use Containers for Isolation

- Utilize Docker or Podman for containerizing applications.

```
docker run -it ubuntu
```

24. Monitor Logs

- Use `journalctl` and log files in `/var/log` to troubleshoot issues.

-
  ```
  journalctl -xe
  tail -f /var/log/syslog
  ```

25. Set Up Network Configurations

- Configure network interfaces and settings using `ip`, `ifconfig`, and `nmcli`.

-
  ```
  ip addr show
  sudo ifconfig eth0 up
  ```

26. Optimize System Performance

- Use `sysctl` to configure kernel parameters for better performance.

  ```
  sudo sysctl -w net.ipv4.ip_forward=1
  ```

27. Use Disk Partitioning Tools

- Manage disk partitions with `fdisk`, `parted`, and `lsblk`.
- `sudo fdisk /dev/sda`

-
  ```
  sudo parted /dev/sda
  lsblk
  ```

28. Implement RAID for Redundancy

- Set up RAID using `mdadm` for data redundancy and performance.

  ```
  sudo mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2
  /dev/sd[ab]
  ```

29. Encrypt Sensitive Data

- Use tools like `gpg` and `openssl` to encrypt data.

-
  ```
  gpg -c file.txt
  openssl enc -aes-256-cbc -salt -in file.txt -out file.enc
  ```

30. Configure System Backups

- Schedule regular backups using tools like `rsnapshot` or `duplicity`.

-
  ```
  rsnapshot configtest
  rsnapshot hourly
  ```

31. Use Process Management Tools

- Manage running processes with `ps`, `kill`, `pkill`, and `nice`.
- `ps aux`
- `kill -9 PID`

-
  ```
  pkill process_name
  nice -n 10 command
  ```

32. Set Up Swap Space

- Configure swap space to improve system stability.
- `sudo fallocate -l 4G /swapfile`
- `sudo chmod 600 /swapfile`
- `sudo mkswap /swapfile`

```
sudo swapon /swapfile
```

33. Implement Security Best Practices

- Follow security guidelines and practices to harden your system.

-
  ```
  sudo ufw enable
  sudo fail2ban-client status
  ```

34. Use Monitoring and Alerting Tools

- Implement tools like `Nagios`, `Zabbix`, or `Prometheus` for monitoring.

  ```
  sudo apt install nagios
  ```

35. Set Up and Manage Databases

- Install, configure, and manage databases like MySQL, PostgreSQL, or MongoDB.

-
  ```
  sudo systemctl start mysql
  sudo -u postgres psql
  ```

36. Optimize Network Performance

- Use tools like `iperf` and `netstat` to optimize network performance.

-
  ```
  iperf -s
  netstat -tuln
  ```

37. Use Virtualization Tools

- Utilize `KVM`, `VirtualBox`, or `VMware` for virtualization.

-
  ```
  sudo apt install qemu-kvm libvirt-bin
  sudo virt-manager
  ```

38. Manage Configuration Files

- Use version control for configuration files to keep track of changes.
- `git init`

-
  ```
  git add /etc/config_file
  git commit -m "Initial config file"
  ```

39. Use Network File Systems

- Set up and use NFS, SMB, or CIFS for network file sharing.
- `sudo apt install n`

fs-kernel-server sudo exportfs -a ```

40. Implement Logging and Auditing

- Use `auditd` and logging tools to track system activity.

-
  ```
  sudo apt install auditd
  sudo auditctl -e 1
  ```

41. Use Screen and Tmux for Terminal Management

- Manage multiple terminal sessions using `screen` or `tmux`.

-
  ```
  screen
  tmux
  ```

42. Optimize Boot Time

- Reduce boot time by disabling unnecessary services.

```
sudo systemctl disable service_name
```

43. Use Disk Quotas

- Implement disk quotas to limit user disk usage.

-
  ```
  sudo apt install quota
  sudo edquota username
  ```

44. Set Up DNS

- Configure DNS settings using `bind` or other DNS servers.

-
  ```
  sudo apt install bind9
  sudo systemctl start bind9
  ```

45. Use Tools for Disk Recovery

- Utilize `fsck` and `testdisk` for disk recovery and repair.

-
  ```
  sudo fsck /dev/sda1
  sudo testdisk
  ```

46. Implement High Availability

- Set up high availability with tools like `keepalived` or `HAProxy`.

-
  ```
  sudo apt install keepalived
  sudo systemctl start keepalived
  ```

47. Use Load Balancing

- Distribute load using tools like `Nginx`, `HAProxy`, or `LoadBalancer`.

-
  ```
  sudo apt install nginx
  sudo systemctl start nginx
  ```

48. Use Caching Mechanisms

- Improve performance with caching tools like `Memcached` or `Redis`.

-
  ```
  sudo apt install redis-server
  sudo systemctl start redis
  ```

49. Use Ansible for Configuration Management

- Automate configuration management using Ansible.

  ```
  ansible-playbook -i inventory playbook.yml
  ```

50. Implement Continuous Integration/Deployment

- Use CI/CD tools like Jenkins, GitLab CI, or Travis CI.

-
  ```
  sudo apt install jenkins
  sudo systemctl start jenkins
  ```

51. Understand and Use SELinux

- Enhance security using SELinux policies and tools.

-
  ```
  sudo setenforce 1
  sudo getenforce
  ```

52. Use AppArmor for Security

- Implement security profiles using AppArmor.

-
  ```
  sudo apt install apparmor
  sudo aa-status
  ```

53. Set Up and Use LDAP

- Configure LDAP for centralized authentication.

-
  ```
  sudo apt install slapd
  sudo dpkg-reconfigure slapd
  ```

54. Use Nginx or Apache for Web Serving

- Set up web servers using Nginx or Apache.

-
  ```
  sudo apt install nginx
  sudo systemctl start nginx
  ```

55. Use Fail2Ban to Protect Against Brute Force Attacks

- Install and configure Fail2Ban to protect your system.

-
  ```
  sudo apt install fail2ban
  sudo systemctl start fail2ban
  ```

56. Use Snort for Intrusion Detection

- Set up Snort for network intrusion detection.

-
  ```
  sudo apt install snort
  sudo systemctl start snort
  ```

57. Use ClamAV for Antivirus Protection

- Install and use ClamAV for virus scanning.

-
  ```
  sudo apt install clamav
  sudo clamscan -r /
  ```

58. Set Up a Mail Server

- Configure a mail server using Postfix, Sendmail, or Exim.

-
  ```
  sudo apt install postfix
  sudo systemctl start postfix
  ```

59. Use Rsync for Efficient File Transfers

- Synchronize files and directories efficiently using Rsync.

  ```
  rsync -avh source/ destination/
  ```

60. Configure and Use Proxy Servers

- Set up and manage proxy servers using Squid or HAProxy.

-
  ```
  sudo apt install squid
  sudo systemctl start squid
  ```

61. Implement Two-Factor Authentication

- Enhance security with two-factor authentication.

-
  ```
  sudo apt install libpam-google-authenticator
  google-authenticator
  ```

62. Use Tools for Packet Analysis

- Analyze network packets using tools like Wireshark or tcpdump.

-
  ```
  sudo apt install wireshark
  sudo tcpdump -i eth0
  ```

63. Set Up and Use VPNs

- Configure VPNs using OpenVPN or WireGuard.

-
  ```
  sudo apt install openvpn
  sudo systemctl start openvpn
  ```

64. Use Configuration Management Tools

- Use tools like Puppet, Chef, or Salt for configuration management.

-
  ```
  sudo apt install puppet
  sudo systemctl start puppet
  ```

65. Use Load Testing Tools

- Test and optimize system performance with tools like `ab` or `JMeter`.

  ```
  ab -n 100 -c 10 http://example.com/
  ```

66. Set Up DNS Caching

- Configure DNS caching with tools like `dnsmasq`.

-
  ```
  sudo apt install dnsmasq
  sudo systemctl start dnsmasq
  ```

67. Use Centralized Logging

- Implement centralized logging using tools like `Logstash` or `Fluentd`.

-
  ```
  sudo apt install logstash
  sudo systemctl start logstash
  ```

68. Implement Security Audits

- Regularly perform security audits using tools like `Lynis`.

-
  ```
  sudo apt install lynis
  sudo lynis audit system
  ```

69. Use Certificate Management Tools

- Manage SSL/TLS certificates using tools like `certbot`.

-
  ```
  sudo apt install certbot
  sudo certbot --nginx
  ```

70. Set Up Remote Desktop Access

- Configure remote desktop access using `xrdp` or `VNC`.

-
  ```
  sudo apt install xrdp
  sudo systemctl start xrdp
  ```

71. Use Docker for Containerization

- Simplify application deployment and management using Docker.

-
  ```
  sudo apt install docker.io
  sudo systemctl start docker
  ```

72. Implement File Integrity Monitoring

- Use tools like `AIDE` or `Tripwire` for file integrity monitoring.

-
  ```
  sudo apt install aide
  sudo aideinit
  ```

73. Use GPG for Secure Communication

- Encrypt and sign communications using GPG.

-
  ```
  gpg --gen-key
  gpg --encrypt --recipient user@example.com file.txt
  ```

74. Set Up and Manage Caches

- Use caching mechanisms like `varnish` to improve performance.

-
  ```
  sudo apt install varnish
  sudo systemctl start varnish
  ```

75. Use Python Virtual Environments

- Isolate Python environments using `virtualenv` or `venv`.

-
  ```
  python3 -m venv myenv
  source myenv/bin/activate
  ```

76. Implement Data Encryption

- Use `LUKS` or other encryption tools to secure data.

-
  ```
  sudo cryptsetup luksFormat /dev/sda1
  sudo cryptsetup open /dev/sda1 encrypted
  ```

77. Use Load Balancing Techniques

- Distribute load using `Nginx` or `HAProxy`.

-
  ```
  sudo apt install haproxy
  sudo systemctl start haproxy
  ```

78. Implement High Availability Clustering

- Use tools like `Pacemaker` for high availability clustering.

-
  ```
  sudo apt install pacemaker
  sudo systemctl start pacemaker
  ```

79. Use Terraform for Infrastructure as Code

- Manage infrastructure using Terraform.

-
  ```
  terraform init
  terraform apply
  ```

80. Implement Continuous Monitoring

- Use tools like `Zabbix` or `Prometheus` for continuous monitoring.

-
  ```
  sudo apt install zabbix-server-mysql
  sudo systemctl start zabbix-server
  ```

81. Configure Multi-Factor Authentication

- Set up multi-factor authentication for enhanced security.

-
  ```
  sudo apt install google-authenticator
  google-authenticator
  ```

82. Use Kubernetes for Orchestration

- Manage containerized applications with Kubernetes.

-
  ```
  sudo apt install kubectl
  kubectl cluster-info
  ```

83. Set Up Logging with ELK Stack

- Use Elasticsearch, Logstash, and Kibana for centralized logging.
-

bash sudo apt install elasticsearch logstash kibana sudo systemctl start elasticsearch logstash kibana ```

84. Use Let's Encrypt for SSL Certificates

- Obtain free SSL certificates using Let's Encrypt.

-
  ```
  sudo apt install certbot
  sudo certbot --nginx
  ```

85. Implement Rate Limiting

- Protect against abuse by implementing rate limiting.

-
  ```
  sudo apt install nginx
  sudo vim /etc/nginx/nginx.conf
  ```

86. Use Fail2Ban for Security

- Protect your server from brute force attacks using Fail2Ban.

-
  ```
  sudo apt install fail2ban
  sudo systemctl start fail2ban
  ```

87. Optimize Database Performance

- Tune database settings for optimal performance.

  ```
  sudo vim /etc/mysql/my.cnf
  ```

88. Use Network Monitoring Tools

- Monitor network traffic using tools like `iftop` or `nload`.

-
  ```
  sudo apt install iftop
  sudo iftop
  ```

89. Implement Disk Encryption

- Encrypt disks using tools like `LUKS` for added security.

-
  ```
  sudo cryptsetup luksFormat /dev/sda1
  sudo cryptsetup open /dev/sda1 encrypted
  ```

90. Use Python for Automation

- Write Python scripts to automate tasks.

-
  ```
  #!/usr/bin/env python3
  print("Hello, World!")
  ```

91. Set Up NTP for Time Synchronization

- Ensure accurate system time using NTP.

-
  ```
  sudo apt install ntp
  sudo systemctl start ntp
  ```

92. Use Log Rotation

- Manage log file sizes using `logrotate`.

```
sudo vim /etc/logrotate.conf
```

93. Implement IDS/IPS Systems

- Use tools like `Snort` for intrusion detection and prevention.

-
  ```
  sudo apt install snort
  sudo systemctl start snort
  ```

94. Use Cloud Services

- Integrate with cloud services like AWS, Azure, or GCP.

  ```
  aws configure
  ```

95. Set Up Web Application Firewalls

- Protect web applications using WAFs like `ModSecurity`.

-
  ```
  sudo apt install libapache2-mod-security2
  sudo systemctl start apache2
  ```

96. Use Centralized Configuration Management

- Manage configurations centrally using tools like `Puppet` or `Chef`.

-
  ```
  sudo apt install puppet
  sudo systemctl start puppet
  ```

97. Implement SSL/TLS Encryption

- Secure communications using SSL/TLS.

-
  ```
  sudo apt install openssl
  openssl req -new -x509 -days 365 -keyout /etc/ssl/private/server.key
  -out /etc/ssl/certs/server.crt
  ```

98. Use LXC for Lightweight Containers

- Create and manage lightweight containers using LXC.

-
  ```
  sudo apt install lxc
  sudo lxc-create -t download -n mycontainer
  ```

99. Implement Disaster Recovery Plans

- Prepare for disasters with comprehensive recovery plans.

  ```
  rsync -avh /data /backup
  ```

100. Regularly Review and Update Security Policies

- Keep security policies up to date and review them regularly.

  ```
  sudo vim /etc/security/policies.conf
  ```