Linux Guide for Aspiring DevOps Engineers

Introduction

Linux is the backbone of most modern IT infrastructure, making it essential for DevOps engineers to master its commands, tools, and best practices. This guide provides a comprehensive overview of Linux concepts and practical examples to help you achieve DevOps expertise.

---

1. Basics of Linux

1.1 Common Commands

- **File and Directory Management**:
- `ls` `# List directory contents`
- `cd` `# Change directory`
- `pwd` `# Print working directory`
- `mkdir` `# Create a directory`
- `rm` `# Remove files or directories`
- `cp` `# Copy files or directories`
- `mv` `# Move or rename files or directories`
- **File Viewing**:
- `cat` `# View file content`
- `less` `# View file content with scrolling`
- `head` `# Display first lines of a file`
- `tail` `# Display last lines of a file`
- **Permissions and Ownership**:
- `chmod` `# Change permissions`
- `chown` `# Change ownership`

1.2 File Permissions

Linux uses a three-tiered permission system for user, group, and others:

- **Example**: `-rwxr-xr--`
  - `r` = read, `w` = write, `x` = execute
  - Use `chmod 755 filename` to set permissions.

---

2. Advanced Linux Concepts

2.1 Process Management

- View running processes:
- `ps aux     # View all processes`
- `top        # Interactive process viewer`
- `htop       # Enhanced top (requires installation)`
- Kill a process:
- `kill PID   # Terminate a process by PID`
- `kill -9 PID  # Force terminate a process`

2.2 Networking Commands

- **Check IP Address**:
- `ip addr show`
- **Ping and Connectivity**:
- `ping google.com`
- **Check Open Ports**:
- `netstat -tuln  # Deprecated`
- `ss -tuln       # Preferred alternative`

2.3 Disk Management

- Check disk usage:
- `df -h   # Disk space usage`
- `du -sh  # Directory size`
- Mount a disk:
- `mount /dev/sdb1 /mnt`
- Unmount a disk:
- `umount /mnt`

---

3. Essential DevOps Tools

3.1 Version Control with Git

- Clone a repository:
- `git clone https://github.com/user/repo.git`
- Create a branch:
- `git checkout -b feature-branch`
- Commit changes:
- `git add .`
- `git commit -m "Commit message"`
- `git push origin branch-name`

3.2 Configuration Management with Ansible

- Install Ansible:
- `sudo apt update`
- `sudo apt install ansible`

- Run a playbook:
- `ansible-playbook playbook.yml`

3.3 Containerization with Docker

- Install Docker:
- `sudo apt update`
- `sudo apt install docker.io`
- Run a container:
- `docker run -d -p 80:80 nginx`
- List running containers:
- `docker ps`

3.4 Orchestration with Kubernetes

- Install kubectl:
- `sudo apt install kubectl`
- View cluster nodes:
- `kubectl get nodes`
- Deploy an application:
- `kubectl apply -f deployment.yaml`

---

4. Automation and Scripting

4.1 Bash Scripting

- Example: Backup script
- `#!/bin/bash`
- `tar -czf backup.tar.gz /path/to/important/data`
- Make the script executable:
- `chmod +x backup.sh`
- `./backup.sh`

4.2 Python Scripting

- Example: Monitor disk space
- `import shutil`
- `total, used, free = shutil.disk_usage("/")`
- `print(f"Free space: {free // (2**30)} GB")`

---

5. Troubleshooting and Monitoring

5.1 Logs

- View system logs:

- `tail -f /var/log/syslog`
- View application logs:
- `journalctl -u service-name`

5.2 Monitoring Tools

- **htop**: Monitor processes interactively.
- **iostat**: Analyze CPU and disk I/O usage.
- **ping**: Test network connectivity.

---

6. Practical Exercises

1. **Set up a web server**:
    - Install Nginx:
    - `sudo apt update`
    - `sudo apt install nginx`
    - Start the server:
    - `sudo systemctl start nginx`
    - Configure the firewall:
    - `sudo ufw allow 'Nginx HTTP'`
2. **Automate deployment with Ansible**:
    - Write a playbook to install and configure Apache.
3. **Deploy a containerized application**:
    - Use Docker to deploy a Python web app.

Conclusion

This guide covers the foundational and advanced Linux concepts necessary for a DevOps engineer. Practice regularly, automate repetitive tasks, and explore new tools to stay updated in this dynamic field.