**1. Linux Basics**

**File System**

* /home – User home directories
* /var – Logs and variable data
* /etc – Configuration files
* /opt – Third-party software
* /tmp – Temporary files
* Common commands: ls -l, cd, pwd, rm, cp, mv, find

**File Permissions**

* chmod 755 file → Read/write/execute for owner, read/execute for group & others
* chown user:group file → Change file ownership
* umask → Default permission settings

**Process Management**

* ps aux – List running processes
* top / htop – Monitor system performance
* kill -9 <PID> – Force kill process

**Networking**

* ip a or ifconfig – Check IP address
* netstat -tulnp – View listening ports
* ss -tulnp – Alternative to netstat
* curl -I https://example.com – Check HTTP response

**2. User & Group Management**

* Create user: useradd -m devopsuser
* Set password: passwd devopsuser
* Add user to sudo group: usermod -aG sudo devopsuser
* Create group: groupadd devops
* Add user to group: usermod -aG devops devopsuser

**3. Package Management**

**Debian-based (Ubuntu)**

* Update: sudo apt update && sudo apt upgrade -y
* Install: sudo apt install nginx -y
* Remove: sudo apt remove nginx -y

**RHEL-based (CentOS, Fedora, Amazon Linux)**

* Update: sudo yum update -y
* Install: sudo yum install httpd -y
* Remove: sudo yum remove httpd -y

**4. Systemd & Service Management**

* systemctl start nginx – Start service
* systemctl stop nginx – Stop service
* systemctl restart nginx – Restart service
* systemctl status nginx – Check service status
* systemctl enable nginx – Enable auto-start
* systemctl disable nginx – Disable auto-start

**5. Log Management**

* /var/log/syslog – System logs (Debian)
* /var/log/messages – System logs (RHEL)
* /var/log/auth.log – Authentication logs
* /var/log/nginx/access.log – Web server logs
* journalctl -u nginx --since today – View logs of a service

**6. Disk Management**

* Check disk usage: df -h
* Check available inodes: df -i
* Check directory size: du -sh /var/log/
* List block devices: lsblk
* Mount disk: mount /dev/sdb1 /mnt

**7. SSH & Remote Access**

* Connect to server: ssh user@server-ip
* Key-based login:
  + ssh-keygen -t rsa -b 4096
  + ssh-copy-id user@server-ip
* Disable root login: Edit /etc/ssh/sshd\_config → PermitRootLogin no

**8. Firewall & Security**

**UFW (Ubuntu Firewall)**

* Allow port: ufw allow 22/tcp
* Enable firewall: ufw enable
* Status: ufw status

**FirewallD (CentOS/RHEL)**

* Start: systemctl start firewalld
* Allow port: firewall-cmd --add-port=80/tcp --permanent
* Reload: firewall-cmd --reload

**SELinux (Security-Enhanced Linux)**

* Check status: sestatus
* Disable temporarily: setenforce 0
* Disable permanently: Edit /etc/selinux/config → SELINUX=disabled

**9. Cron Jobs & Automation**

* Edit crontab: crontab -e
* List jobs: crontab -l
* Schedule example: 0 2 \* \* \* /home/user/backup.sh # Run script at 2 AM daily

**10. Container & Kubernetes Basics**

**Docker**

* Start service: systemctl start docker
* Run container: docker run -d -p 80:80 nginx
* List running: docker ps
* Stop container: docker stop <container\_id>
* Remove container: docker rm <container\_id>

**Kubernetes (kubectl)**

* List pods: kubectl get pods
* View logs: kubectl logs <pod\_name>
* Scale deployment: kubectl scale deployment nginx --replicas=3
* Apply YAML: kubectl apply -f deployment.yaml

**11. Azure DevOps-Specific Linux Tasks**

**Azure CLI**

* Install: curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
* Login: az login
* Set subscription: az account set --subscription <sub\_id>

**Azure VM Management**

* List VMs: az vm list -o table
* Start VM: az vm start --name myVM --resource-group myRG
* Stop VM: az vm stop --name myVM --resource-group myRG

**Azure DevOps Agent on Linux**

* Download agent:
* mkdir myagent && cd myagent
* curl -O https://vstsagentpackage.azureedge.net/agent/3.220.2/vsts-agent-linux-x64-3.220.2.tar.gz
* tar zxvf vsts-agent-linux-x64-3.220.2.tar.gz
* ./config.sh
* Start agent: ./svc.sh install && ./svc.sh start

**12. Terraform & Infrastructure as Code (IaC)**

**Terraform on Linux**

* Install:
* wget https://releases.hashicorp.com/terraform/1.6.0/terraform\_1.6.0\_linux\_amd64.zip
* unzip terraform\_1.6.0\_linux\_amd64.zip
* sudo mv terraform /usr/local/bin/
* Initialize: terraform init
* Plan: terraform plan
* Apply: terraform apply -auto-approve

**Conclusion**

These are the essential Linux skills for an Azure DevOps Engineer. If you're focusing on CI/CD, Azure CLI, Terraform, and Docker/Kubernetes, having a strong Linux foundation will make automation and troubleshooting much easier.