**Extracted Code and Scripts from Chapter 16**

**1. Standard Server Deployments**

**Firewall Configuration**

sudo firewall-cmd --permanent --add-service=http

sudo firewall-cmd --permanent --add-service=https

sudo firewall-cmd –reload

**Apache Installation and Management**

sudo systemctl stop nginx

sudo systemctl disable nginx

sudo dnf install -y httpd

sudo systemctl enable httpd

sudo systemctl start httpd

sudo systemctl status httpd

sudo systemctl stop httpd

**2. Multi-Server Deployment**

**Nginx Reverse Proxy Setup**

sudo dnf install nginx -y

sudo systemctl start nginx

sudo systemctl enable --now nginx

sudo nano /etc/nginx/nginx.conf

sudo systemctl restart nginx

**Reload Nginx Without Downtime**

sudo systemctl reload nginx

**3. Virtualized Deployments with KVM**

**KVM Installation**

sudo dnf install virt-install -y

sudo dnf install libvirt-daemon -y

**Start and Enable Libvirt Service**

sudo systemctl start libvirtd

sudo systemctl enable --now libvirtd

**Additional Configurations**

export LIBVIRT\_DEFAULT\_URI=qemu:///system

echo 'export LIBVIRT\_DEFAULT\_URI=qemu:///system' >> ~/.bashrc

source ~/.bashrc

sudo dnf install libvirt-daemon-kvm libvirt-client virt-manager -y

**Verify System Support for KVM**

egrep -c '(vmx|svm)' /proc/cpuinfo

**List Available Network Bridges**

sudo virsh net-list --all

**4. Container-Based Deployment with Podman**

**Install Podman**

sudo dnf install podman -y

**Pull and Run Rocky Linux Container**

podman pull docker.io/rockylinux/rockylinux:9

podman run -it rockylinux/rockylinux:9 /bin/bash

**List Downloaded Images**

podman images

**List Running Containers**

podman ps

**5. Automated Deployment with Ansible**

**Install Ansible**

sudo dnf install ansible -y

**Create Apache Playbook (apache\_setup.yml)**

- name: Install Apache on Rocky Linux

hosts: rocky\_servers

become: yes

tasks:

- name: Install httpd

dnf:

name: httpd

state: present

**Run Ansible Playbook**

sudo -E ansible-playbook apache\_setup.yml

**6. Load Balancing with HAProxy**

**Install and Configure HAProxy**

sudo dnf install haproxy -y

sudo nano /etc/haproxy/haproxy.cfg

sudo systemctl enable --now haproxy

**Sample HAProxy Configuration**

frontend http\_front

bind \*:80

default\_backend web\_servers

backend web\_servers

balance roundrobin

server web1 192.168.1.10:80 check

server web2 192.168.1.11:80 check

**Check HAProxy Configuration**

sudo haproxy -c -f /etc/haproxy/haproxy.cfg

**Restart HAProxy**

sudo systemctl restart haproxy

**7. Rocky Linux Deployment in AWS**

**Auto Scaling with AWS CLI**

aws autoscaling create-auto-scaling-group \

--auto-scaling-group-name rocky-asg \

--launch-template LaunchTemplateName=my-template \

--min-size 1 --max-size 5

**Create AWS ELB**

aws elb create-load-balancer \

--load-balancer-name my-load-balancer \

--listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"

**Register EC2 Instances with ELB**

aws elb register-instances-with-load-balancer --load-balancer-name my-load-balancer --instances i-0123456789abcdef0

**Enable Health Checks**

aws elb configure-health-check --load-balancer-name my-load-balancer --health-check Target=HTTP:80/index.html,Interval=30,UnhealthyThreshold=2,HealthyThreshold=2,Timeout=5

This document extracts all the critical deployment scripts and commands from Chapter 16, structuring them into an easy-to-use reference for configuring and managing Rocky Linux environments.