### Chapter 2

#### Recipe #1

[jsg@rhel9\_bpb ~]$ sudo dnf install mediawriter

[jsg@rhel9\_bpb ~]$ mediawriter

### Chapter 3

#### Recipe #9

[jsg@rhel9\_bpb ~]$ sudo insights-client –register

[jsg@rhel9\_bpb ~]$ sudo rhc connect -u <username> -p <password>

### Chapter 4

#### Recipe #11

[jsg@rhel9\_bpb ~]$ tar czf /mnt/backup/mybackup.tar.gz /home/

[jsg@rhel9\_bpb ~]$ sudo dnf update

[jsg@rhel9\_bpb ~]$ sudo subscription-manager repos --disable=\*

[jsg@rhel9\_bpb ~]$ sudo dnf install leapp

[jsg@rhel9\_bpb ~]$ sudo leapp preupgrade

[jsg@rhel9\_bpb ~]$ sudo leapp upgrade --target 9.0

[jsg@rhel9\_bpb ~]$ sudo cat /etc/redhat-release

#### Recipe #12

[jsg@rhel9\_bpb ~]$ sudo useradd jeromeg

[jsg@rhel9\_bpb ~]$ sudo adduser jeromeg

[jsg@rhel9\_bpb ~]$ sudo passwd jeromeg

[jsg@rhel9\_bpb ~]$ sudo usermod -aG marketing jeromeg

[jsg@rhel9\_bpb ~]$ sudo userdel jblack

[jsg@rhel9\_bpb ~]$ sudo chmod u+rwx myfile01

#### Recipe #13

[jsg@rhel9\_bpb ~]$ sudo timedatectl

[jsg@rhel9\_bpb ~]$ sudo timedatectl list-timezones

[jsg@rhel9\_bpb ~]$ sudo timedatectl set-timezone America/New\_York

[jsg@rhel9\_bpb ~]$ sudo timedatectl set-ntp yes

[jsg@rhel9\_bpb ~]$ sudo timedatectl set-time "2023-04-23 12:30:00"

[jsg@rhel9\_bpb ~]$ sudo timedatectl

#### Recipe #15

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd –state

[jsg@rhel9\_bpb ~]$ sudo systemctl start firewalld

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --add-service=ssh

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --add-service=dns

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --add-service=http

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --add-service=https

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="0.0.0.0/0" reject'

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="192.168.1.100" reject'

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --reload

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --list-all

#### Recipe #16

[jsg@rhel9\_bpb ~]$ sudo systemctl list-units --type service

[jsg@rhel9\_bpb ~]$ sudo systemctl list-unit-files --type service

[jsg@rhel9\_bpb ~]$ sudo systemctl enable <name>.service

[jsg@rhel9\_bpb ~]$ sudo systemctl start <name>.service

[jsg@rhel9\_bpb ~]$ sudo systemctl disable <name>.service

[jsg@rhel9\_bpb ~]$ sudo systemctl stop <name>.service

### Chapter 5

#### Recipe #19

[jsg@rhel9-bpb ~]$ sudo -i

[sudo] password for jsg:

[root@rhel9-bpb ~]# dnf install sos

[root@rhel9-bpb ~]# rpm -q sos

[jsg@rhel9-bpb ~]$ sudo sos report

[jsg@rhel9-bpb ~]$ sudo sos clean /var/tmp/sosreport-rhel9-bpb-2023-01-06-qppkokn.tar.xz

[jsg@rhel9-bpb ~]$ sudo ls -l /var/tmp/sosreport-host0-2023-01-06-qppkokn-private\_map /var/tmp/sosreport-host0-2023-01-06-qppkokn-obfuscated.tar.xz

[jsg@rhel9-bpb ~]$ sudo cat /var/tmp/sosreport-host0-2023-01-06-qppkokn-private\_map

#### Recipe #20

[jsg@rhel9-bpb ~]$ sudo sos report *--upload*

### Chapter 6

#### Recipe #22

[jsg@rhel9-bpb ~]$ sudo dnf check-update

[jsg@rhel9-bpb ~]$ sudo dnf upgrade <package>

[jsg@rhel9-bpb ~]$ sudo dnf upgrade

[jsg@rhel9-bpb ~]$ sudo dnf search emacs

[jsg@rhel9-bpb ~]$ sudo dnf install emacs.x86\_64

#### Recipe #23

[jsg@rhel9-bpb ~]$ sudo subscription-manager repos --enable codeready-builder-for-rhel-9-$(arch)-rpms

[jsg@rhel9-bpb ~]$ sudo dnf install \

https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm

[jsg@rhel9-bpb ~]$ sudo dnf update

[jsg@rhel9-bpb ~]$ sudo dnf install htop.x86\_64

#### Recipe #24

[jsg@rhel9-bpb ~]$ sudo dnf install --nogpgcheck https://mirrors.rpmfusion.org/free/el/rpmfusion-free-release-$(rpm -E %rhel).noarch.rpm https://mirrors.rpmfusion.org/nonfree/el/rpmfusion-nonfree-release-$(rpm -E %rhel).noarch.rpm

[jsg@rhel9-bpb ~]$ sudo dnf update

[jsg@rhel9-bpb ~]$ sudo dnf search kmod-nvidia

#### Recipe #25

[jsg@rhel9-bpb ~]$ sudo dnf repolist

[jsg@rhel9-bpb ~]$ sudo dnf repolist all

[jsg@rhel9-bpb ~]$ sudo dnf --enablerepo=repo-name install package

[jsg@rhel9-bpb ~]$ sudo dnf list installed

[jsg@rhel9-bpb ~]$ sudo dnf list recent

[jsg@rhel9-bpb ~]$ sudo dnf list available

[jsg@rhel9-bpb ~]$ sudo dnf autoremove emacs.x86\_64

[jsg@rhel9-bpb ~]$ sudo dnf autoremove

#### Recipe #26

[jsg@rhel9\_bpb ~]$ sudo dnf install dnf-automatic

[jsg@rhel9-bpb ~]$ sudo rpm -qi dnf-automatic

[jsg@rhel9-bpb ~]$ sudo systemctl enable --now dnf-automatic.timer

[jsg@rhel9-bpb ~]$ systemctl list-timers dnf-\*

### Chapter 7

#### Recipe #31

[jsg@rhel9\_bpb ~]$ sudo dnf install @input-methods

[jsg@rhel9\_bpb ~]$ sudo dnf install gnome-tweaks

### Chapter 8

#### Recipe #32

[jsg@rhel9\_bpb ~]$ sudo dnf update

[jsg@rhel9\_bpb ~]$ sudo dnf install httpd

[jsg@rhel9\_bpb ~]$ sudo systemctl start httpd

[jsg@rhel9\_bpb ~]$ sudo systemctl enable httpd

[jsg@rhel9\_bpb ~]$ sudo dnf install mariadb-server mariadb

[jsg@rhel9\_bpb ~]$ sudo systemctl start mariadb

[jsg@rhel9\_bpb ~]$ sudo systemctl enable mariadb

[jsg@rhel9\_bpb ~]$ sudo mysql\_secure\_installation

[jsg@rhel9\_bpb ~]$ sudo dnf install php php-mysqlnd

[jsg@rhel9\_bpb ~]$ sudo vi /var/www/html/info.php

[jsg@rhel9\_bpb ~]$  sudo rm /var/www/html/info.php

#### Recipe #33

[jsg@rhel9\_bpb ~]$ sudo dnf update

[jsg@rhel9\_bpb ~]$ sudo dnf install epel-release

[jsg@rhel9\_bpb ~]$ sudo dnf install nginx

[jsg@rhel9\_bpb ~]$ sudo systemctl start nginx

[jsg@rhel9\_bpb ~]$ sudo systemctl status nginx

[jsg@rhel9\_bpb ~]$ sudo systemctl enable nginx

#### Recipe #34

[jsg@rhel9\_bpb ~]$ sudo dnf install haproxy

[jsg@rhel9\_bpb ~]$ sudo systemctl start haproxy

[jsg@rhel9\_bpb ~]$ sudo systemctl enable haproxy

[jsg@rhel9\_bpb ~]$ sudo systemctl status haproxy

[jsg@rhel9\_bpb ~]$ sudo systemctl restart haproxy

#### Recipe #35

[jsg@rhel9\_bpb ~]$ sudo dnf update

[jsg@rhel9\_bpb ~]$ sudo dnf install varnish

[jsg@rhel9\_bpb ~]$ sudo systemctl start varnish

[jsg@rhel9\_bpb ~]$ sudo systemctl enable varnish

[jsg@rhel9\_bpb ~]$ sudo systemctl status varnish

[jsg@rhel9\_bpb ~]$ sudo nano /etc/varnish/default.vcl

[jsg@rhel9\_bpb ~]$ sudo nano /etc/systemd/system/varnish.service

[jsg@rhel9\_bpb ~]$ sudo systemctl daemon-reload

[jsg@rhel9\_bpb ~]$ sudo systemctl restart varnish

[jsg@rhel9\_bpb ~]$ sudo netstat -tunlp | grep varnish

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --zone=public --add-port=80/tcp --permanent

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --reload

[jsg@rhel9\_bpb ~]$ sudo curl -I http:*//localhost*

[jsg@rhel9\_bpb ~]$ varnishstat

#### Recipe #36

[jsg@rhel9\_bpb ~]$ sudo dnf update

[jsg@rhel9\_bpb ~]$ sudo dnf install squid

[jsg@rhel9\_bpb ~]$ sudo nano /etc/squid/squid.conf

[jsg@rhel9\_bpb ~]$ sudo systemctl enable squid

[jsg@rhel9\_bpb ~]$ sudo systemctl start squid

[jsg@rhel9\_bpb ~]$ sudo systemctl status squid

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --add-port=3128/tcp

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --reload

#### Recipe #37

[jsg@rhel9\_bpb ~]$ sudo dnf install mysql-server

[jsg@rhel9\_bpb ~]$ sudo systemctl start mysqld.service

[jsg@rhel9\_bpb ~]$ sudo systemctl enable mysqld.service

[jsg@rhel9\_bpb ~]$ sudo mysql\_secure\_installation

#### Recipe #38

[jsg@rhel9\_bpb ~]$ sudo dnf install postgresql-server

[jsg@rhel9\_bpb ~]$ sudo postgresql-setup --initdb

[jsg@rhel9\_bpb ~]$ sudo systemctl start postgresql.service

[jsg@rhel9\_bpb ~]$ sudo systemctl enable postgresql.service

[jsg@rhel9\_bpb ~]$ sudo postgres=*# CREATE USER myuser WITH PASSWORD 'password' CREATEROLE CREATEDB;*

#### Recipe #39

[jsg@rhel9\_bpb ~]$ sudo dnf update

[jsg@rhel9\_bpb ~]$ sudo dnf install mariadb-server

[jsg@rhel9\_bpb ~]$ sudo systemctl start mariadb.service

[jsg@rhel9\_bpb ~]$ sudo systemctl enable mariadb.service

#### Recipe #40

[jsg@rhel9\_bpb ~]$ sudo yum update -y

[jsg@rhel9\_bpb ~]$ sudo yum install -y mongodb-org

[jsg@rhel9\_bpb ~]$ sudo systemctl start mongod

[jsg@rhel9\_bpb ~]$ sudo systemctl enable mongod

[jsg@rhel9\_bpb ~]$ sudo systemctl status mongod

[jsg@rhel9\_bpb ~]$ sudo mongosh

#### Recipe #41

[jsg@rhel-bpb ~]$ sudo cp -i Downloads/cockroach-v24.2.4.linux-arm64/cockroach /usr/local/bin/

[jsg@rhel-bpb ~]$ cockroach version

[jsg@rhel-bpb ~]$ sudo mkdir -p /usr/local/lib/cockroach

[jsg@rhel-bpb cockroach-v24.2.4.linux-arm64]$ sudo cp -i lib/libgeos.so /usr/local/lib/cockroach/

[jsg@rhel-bpb cockroach-v24.2.4.linux-arm64]$ sudo cp -i lib/libgeos\_c.so /usr/local/lib/cockroach/

[jsg@rhel-bpb ~]$ which cockroach

[jsg@rhel-bpb ~]$ cockroach demo

#### Recipe #42

[jsg@rhel9\_bpb ~]$ sudo yum install java-17-openjdk

[jsg@rhel9\_bpb ~]$ sudo yum install java-17-openjdk-devel

[jsg@rhel9\_bpb ~]$ javac -version

[jsg@rhel9\_bpb ~]$ rpm --import https:*//debian.neo4j.com/neotechnology.gpg.key*

[jsg@rhel9\_bpb ~]$ cat <  /etc/yum.repos.d/neo4j.repo

[jsg@rhel9\_bpb ~]$ sudo yum install neo4j-5.9.0

[jsg@rhel9\_bpb ~]$ sudo systemctl enable neo4j

#### Recipe #43

[jsg@rhel9\_bpb ~]$ sudo yum update

[jsg@rhel9\_bpb ~]$ sudo yum install cassandra

[jsg@rhel9\_bpb ~]$ sudo service cassandra start

[jsg@rhel9\_bpb ~]$ sudo tail -f logs/system.log

#### Recipe #44

jsg@rhel9\_bpb ~]$ sudo curl -o /etc/yum.repos.d/mssql-server.repo https://packages.microsoft.com/config/rhel/8/mssql-server-2022.repo  
jsg@rhel9\_bpb ~]$ sudo yum install -y mssql-server

jsg@rhel9\_bpb ~]$ sudo /opt/mssql/bin/mssql-conf setup

jsg@rhel9\_bpb ~]$ sudo systemctl status mssql-server

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --zone=public --add-port=1433/tcp --permanent

[jsg@rhel9\_bpb ~]$ sudo firewall-cmd --reload

[jsg@rhel9\_bpb ~]$ sudo rpm --import https:*//packages.microsoft.com/keys/microsoft.asc*

[jsg@rhel9\_bpb ~]$ sudo curl -o /etc/yum.repos.d/msprod.repo https://packages.microsoft.com/config/rhel/9/prod.repo

[jsg@rhel9\_bpb ~]$ sudo dnf install -y mssql-tools unixODBC-devel

[jsg@rhel9\_bpb ~]$ echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash\_profile

[jsg@rhel9\_bpb ~]$ source ~/.bash\_profile

[jsg@rhel9\_bpb ~]$ sqlcmd -S localhost -U sa -P '<Password>'

### Chapter 9

#### Recipe #45

[jsg@rhel9\_bpb ~]$ sudo dnf update

[jsg@rhel9\_bpb ~]$ sudo dnf install qemu-kvm libvirt virt-install virt-viewer

[jsg@rhel9\_bpb ~]$ sudo systemctl start libvirtd

[jsg@rhel9\_bpb ~]$ sudo systemctl enable libvirtd

[jsg@rhel9\_bpb ~]$ sudo systemctl status libvirtd

#### Recipe #46

[jsg@rhel9\_bpb ~]$ sudo virt-install \

*--name rhel9-guest01 --memory 2048 --vcpus 4 \*

*--disk size=80 --os-variant rhel9.0 \*

*--cdrom /home/jsg/Downloads/rhel9.iso*

[jsg@rhel9\_bpb ~]$ sudo virsh console rhel9-guest01

#### Recipe #47

[jsg@rhel9\_bpb ~]$ sudo virt-install --name=win-guest02 --vcpus=2 --memory=4096 --cdrom=/home/jsg/Downloads/windows.iso --disk size=50

[jsg@rhel9\_bpb ~]$ sudo virsh attach-disk win-guest02 --source /home/jsg/Downloads/virtio-win.iso --target hdc --persistent --subdriver raw[jsg@rhel9\_bpb ~]$ sudo virsh start win-guest02

#### Recipe #48

[jsg@rhel9\_bpb ~]$ sudo virsh start rhel9-guest01

[jsg@rhel9\_bpb ~]$ sudo virsh autostart rhel9-guest01

#### Recipe #49

[jsg@rhel9\_bpb ~]$ sudo virsh list *--all*

[jsg@rhel9\_bpb ~]$ sudo virsh net-list --all

#### Recipe #50

[jsg@rhel9\_bpb ~]$ sudo virsh shutdown demo-guest1

[jsg@rhel9\_bpb ~]$ sudo virsh destroy demo-guest1

[jsg@rhel9\_bpb ~]$ sudo virsh undefine rhel9-guest01 --remove-all-storage

### Chapter 10

#### Recipe #51

[jsg@rhel9\_bpb ~]$ sudo dnf install container-tools

[jsg@rhel9\_bpb ~]$ sudo dnf install podman-docker

#### Recipe #52

[jsg@rhel9\_bpb ~]$ podman pull registry.access.redhat.com/ubi9/ubi

[jsg@rhel9\_bpb ~]$ podman images

[jsg@rhel9\_bpb ~]$ podman run --rm --name=bpb\_ubi registry.access.redhat.com/ubi9/ubi cat /etc/os-release

#### Recipe #53

[jsg@rhel9\_bpb ~]$ podman pull registry.access.redhat.com/ubi9/ubi

[jsg@rhel9-bpb ~]$ podman search registry.access.redhat.com

[jsg@rhel9-bpb ~]$ podman search openjdk-11

[jsg@rhel9-bpb ~]$ podman search openjdk-11

[jsg@rhel9-bpb ~]$ podman info -f json

#### Recipe #54

[jsg@rhel9-bpb ~]$ podman help  
[jsg@rhel9-bpb ~]$ man podman  
[jsg@rhel9-bpb ~]$ podman search httpd

[jsg@rhel9-bpb ~]$ podman run -dt -p 8080:80/tcp registry.access.redhat.com/ubi9/httpd-24

[jsg@rhel9-bpb ~]$ podman ps

[jsg@rhel9-bpb ~]$ podman stop 2028382fcd05

[jsg@rhel9-bpb ~]$ podman start 2028382fcd05

[jsg@rhel9-bpb ~]$ podman images

[jsg@rhel9-bpb ~]$ podman rmi 52fef27b0f78

#### Recipe #55

[jsg@rhel9\_bpb ~]$ podman run -dt --name=mycontainer -p 8282:8282 --health-cmd='curl http://localhost:8282 || exit 1' --health-interval=0 registry.access.redhat.com/ubi9/httpd-24

b6a6a113371140ea2b84de27cb641dca4dd826a58677787c5c44d83d33944bdf

[jsg@rhel9\_bpb ~]$ podman healthcheck run mycontainer

#### Recipe #56

[jsg@rhel9-bpb ~]$ podman network ls

[jsg@rhel9-bpb ~]$ podman network inspect podman

[jsg@rhel9-bpb ~]$ podman network create bpbnetwork

[jsg@rhel9-bpb ~]$ podman network ls

[jsg@rhel9-bpb ~]$ podman network connect bpbnetwork name\_of\_container

[jsg@rhel9-bpb ~]$ podman network disconnect bpbnetwork name\_of\_container  
[jsg@rhel9-bpb ~]$ podman network rm bpbnetwork

### Chapter 11

#### Recipe #57

[lb@rhel9\_bpb ~]$ sudo nmcli connection add type ethernet con-name "Office-LAN" ifname eth0 ip4 192.168.1.100/24 gw4 192.168.1.1 [lb@rhel9\_bpb ~]$ sudo nmcli device wifi connect "MyHomeWiFi" password "mysecurepassword"

[lb@rhel9\_bpb ~]$ sudo dnf install network-scripts

#### Recipe #58

[lb@rhel9\_bpb ~]$ sudo nmcli connection add type vlan con-name "Finance-VLAN" ifname vlan10 vlan.parent eth0 vlan.id 10 ip4 192.168.10.50/24 gw4 192.168.10.1

[lb@rhel9\_bpb ~]$ sudo nmcli connection up "Finance-VLAN"

[lb@rhel9\_bpb ~]$ ip addr show vlan10

[lb@rhel9\_bpb ~]$ ping -c 4 -I vlan10 192.168.10.10

[lb@rhel9\_bpb ~]$ ping -c 4 -I vlan10 192.168.1.100

[lb@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --new-zone=finance

[lb@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --zone=finance --add-interface=vlan10

[lb@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --zone=finance --add-service=mysql --add-service=https

[lb@rhel9\_bpb ~]$ sudo firewall-cmd --reload

#### Recipe #59

[lb@rhel9\_bpb ~]$ sudo nmcli connection add type bond con-name bond0 ifname bond0 mode 802.3ad ip4 192.168.1.100/24 gw4 192.168.1.1

[lb@rhel9\_bpb ~]$ sudo nmcli connection add type ethernet slave-type bond con-name bond0-port1 ifname eth1 master bond0

[lb@rhel9\_bpb ~]$ sudo nmcli connection add type ethernet slave-type bond con-name bond0-port2 ifname eth2 master bond0

[lb@rhel9\_bpb ~]$ sudo nmcli connection up bond0

[lb@rhel9\_bpb ~]$ sudo nmcli connection up bond0-port1

[lb@rhel9\_bpb ~]$ sudo nmcli connection up bond0-port2

[lb@rhel9\_bpb ~]$ cat /proc/net/bonding/bond0

[lb@rhel9\_bpb ~]$ iperf3 -c 192.168.1.200 -t 30 -P 2 --bind bond0

[lb@rhel9\_bpb ~]$ sudo nmcli connection down bond0 bond0-port1 bond0-port2

[lb@rhel9\_bpb ~]$ sudo nmcli connection up "Ethernet 1"

[lb@rhel9\_bpb ~]$ iperf3 -c 192.168.1.200 -t 30 --bind eth1

[lb@rhel9\_bpb ~]$ sudo ip link set eth1 down

[lb@rhel9\_bpb ~]$ ping -c 4 192.168.1.200

#### Recipe #60

lb@rhel9 ~]$ sudo dnf install NetworkManager NetworkManager-libreswan-gnome

[lb@rhel9 ~]$ sudo ipsec checknss  
[lb@rhel9 ~]$ ipsec import company.p12

[lb@rhel9 ~]$ ping 10.8.0.1

[lb@rhel9 ~]$ curl http://intranet.company.com

[lb@rhel9 ~]$ sudo nmcli con mod "VPN connection" +vpn.ip4.routes "192.168.0.0/16 10.0.0.0/8" vpn.ip4.never-default true

#### Recipe #61

[lb@rhel9\_bpb ~]$ sudo nmcli connection add type ip-tunnel

con-name "IPv6-over-IPv4"

ifname gre1

ip-tunnel.mode gre

ip-tunnel.local 203.0.113.5

ip-tunnel.remote 198.51.100.7

ip6 2001:db8:1234::2/64

[lb@rhel9\_bpb ~]$ sudo nmcli connection up "IPv6-over-IPv4"

[lb@rhel9\_bpb ~]$ ip -d link show gre1

[lb@rhel9\_bpb ~]$ nmcli connection add type ip-tunnel \

con-name "IPv6-over-IPv4" \

ifname gre1 \

ip-tunnel.mode gre \

ip-tunnel.local 198.51.100.7 \

ip-tunnel.remote 203.0.113.5 \

ip6 2001:db8:1234::1/64

[lb@rhel9\_bpb ~]$ ping6 -c 4 2001:db8:1234::1

[lb@rhel9\_bpb ~]$ sudo tcpdump -i eth0 -n proto 47

[lb@rhel9\_bpb ~]$ sudo dnf install libreswan

[lb@rhel9\_bpb ~]$ sudo tee /etc/ipsec.d/gre-tunnel.conf << EOF

conn gre-tunnel

left=203.0.113.5

right=198.51.100.7

authby=secret

type=transport

auto=start

phase2=esp

esp=aes256-sha2

EOF

[lb@rhel9\_bpb ~]$ sudo tee /etc/ipsec.d/gre-tunnel.secrets << EOF

203.0.113.5 198.51.100.7: PSK "v3ryS3cur3Sh@r3dK3y!"

EOF

[lb@rhel9\_bpb ~]$ sudo systemctl enable --now ipsec

#### Recipe #62

[lb@rhel9\_bpb ~]$ lsblk

[lb@rhel9\_bpb ~]$ sudo fdisk /dev/sdb

[lb@rhel9\_bpb ~]$ sudo parted /dev/sdc

sudo mkfs.ext4 /dev/sdc1

sudo mkdir /mnt/data

sudo mount /dev/sdc1 /mnt/data

sudo chown lb:lb /mnt/data

sudo mkswap /dev/sdc3

sudo swapon /dev/sdc3

#### Recipe #63

[lb@rhel9\_bpb ~]$ sudo dnf install lvm2

[lb@rhel9\_bpb ~]$ sudo pvcreate /dev/sdb2 /dev/sdc2

[lb@rhel9\_bpb ~]$ sudo pvs

[lb@rhel9\_bpb ~]$ sudo vgcreate data-vg /dev/sdb2

sudo vgs

[lb@rhel9\_bpb ~]$ sudo lvcreate -n var-lv -L 20G data-vg

[lb@rhel9\_bpb ~]$ sudo lvcreate -n home-lv -L 40G data-vg

[lb@rhel9\_bpb ~]$ sudo vgextend data-vg /dev/sdc2

[lb@rhel9\_bpb ~]$ sudo lvconvert --type striped -i 2 /dev/data-vg/home-lv

[lb@rhel9\_bpb ~]$ sudo lvs

[lb@rhel9\_bpb ~]$ sudo mkfs.xfs /dev/data-vg/var-lv

[lb@rhel9\_bpb ~]$ sudo mkfs.ext4 /dev/data-vg/home-lv

[lb@rhel9\_bpb ~]$ sudo mkdir /mnt/newvar /mnt/newhome

[lb@rhel9\_bpb ~]$ sudo mount /dev/data-vg/var-lv /mnt/newvar

[lb@rhel9\_bpb ~]$ sudo mount /dev/data-vg/home-lv /mnt/newhome

[lb@rhel9\_bpb ~]$ sudo rsync -avHSX /var/ /mnt/newvar/

[lb@rhel9\_bpb ~]$ sudo rsync -avHSX /home/ /mnt/newhome/

diff -r /var /mnt/newvar

diff -r /home /mnt/newhome

/dev/data-vg/var-lv /var xfs defaults 0 0

/dev/data-vg/home-lv /home ext4 defaults 0 0

[lb@rhel9\_bpb ~]$ sudo umount /var /home /mnt/newvar /mnt/newhome

sudo mount -a

lb@rhel9\_bpb ~]$ sudo lvcreate -s -n var-snapshot -L 5G /dev/data-vg/var-lv

[lb@rhel9\_bpb ~]$ sudo lvconvert --merge /dev/data-vg/var-snapshot

[lb@rhel9\_bpb ~]$ sudo lvremove /dev/data-vg/var-snapshot

[lb@rhel9\_bpb ~]$ sudo lvextend -L +10G /dev/data-vg/var-lv

[lb@rhel9\_bpb ~]$ sudo xfs\_growfs /var

[lb@rhel9\_bpb ~]$ sudo lvextend -l +100%FREE /dev/data-vg/var-lv

[lb@rhel9\_bpb ~]$ sudo xfs\_growfs /var

[lb@rhel9\_bpb ~]$ sudo vgdisplay

[lb@rhel9\_bpb ~]$ sudo lvdisplay

[lb@rhel9\_bpb ~]$ sudo pvdisplay

[lb@rhel9\_bpb ~]$ sudo lvs -o +devices

#### Recipe #64

[lb@rhel9\_bpb ~]$ sudo dnf install iscsi-initiator-utils

[lb@rhel9\_bpb ~]$ sudo iscsiadm --mode discovery --type sendtargets --portal storage.example.com

[lb@rhel9\_bpb ~]$ sudo iscsiadm --mode node --targetname iqn.2024-06.com.example.storage:array1 --portal storage.example.com --login

[lb@rhel9\_bpb ~]$ sudo lsblk

[lb@rhel9\_bpb ~]$ sudo parted /dev/sdd mklabel gpt

[lb@rhel9\_bpb ~]$ sudo parted /dev/sdd mkpart primary 0% 100%

[lb@rhel9\_bpb ~]$ sudo mkfs.xfs -K /dev/sdd1

[lb@rhel9\_bpb ~]$ sudo mkdir -p /mnt/iscsi/database

[lb@rhel9\_bpb ~]$ echo "/dev/sdd1 /mnt/iscsi/database xfs \_netdev,x-

[lb@rhel9\_bpb ~]$ systemd.requires=iscsi.service 0 0" | sudo tee -a /etc/fstab

[lb@rhel9\_bpb ~]$ sudo mount /mnt/iscsi/database

[lb@rhel9\_bpb ~]$ sudo pvcreate /dev/sde

[lb@rhel9\_bpb ~]$ sudo vgcreate shares-vg /dev/sde

[lb@rhel9\_bpb ~]$ sudo lvcreate -n engineering-lv -L 500G shares-vg

[lb@rhel9\_bpb ~]$ sudo mkfs.xfs /dev/shares-vg/engineering-lv

[lb@rhel9\_bpb ~]$ sudo mkdir -p /mnt/iscsi/shares/engineering

[lb@rhel9\_bpb ~]$ echo "/dev/shares-vg/engineering -lv /mnt/iscsi/shares/engineering xfs \_netdev,x-systemd.requires=iscsi.service 0 0" | sudo tee -a /etc/fstab

[lb@rhel9\_bpb ~]$ sudo mount /mnt/iscsi/shares/engineering

[lb@rhel9\_bpb ~]$ df -h | grep iscsi

[lb@rhel9\_bpb ~]$ sudo systemctl enable --now iscsid

[lb@rhel9\_bpb ~]$ sudo iscsiadm --mode node --targetname iqn.2024-06.com.example.storage:array1 --portal storage.example.com --op update -n node.startup -v automatic

[lb@rhel9\_bpb ~]$ sudo dnf install device-mapper-multipath

[lb@rhel9\_bpb ~]$ sudo mpathconf --enable --with\_multipathd y

[lb@rhel9\_bpb ~]$ echo 'WWID\_OF\_SDD\_FROM\_MULTIPATH\_-L\_OUTPUT' | sudo tee -a /etc/multipath/wwids

[lb@rhel9\_bpb ~]$ sudo systemctl restart multipathd

#### Recipe #65

[lb@rhel9\_bpb ~]$ sudo dnf install samba samba-client samba-common

[lb@rhel9\_bpb ~]$ sudo systemctl enable smb nmb

[lb@rhel9\_bpb ~]$ sudo systemctl start smb nmb

[lb@rhel9\_bpb ~]$ sudo smbpasswd -a username

[lb@rhel9\_bpb ~]$ sudo systemctl restart smb nmb

[lb@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --add-service=samba

[lb@rhel9\_bpb ~]$ sudo firewall-cmd --reload

#### Recipe #66

[lb@rhel9\_bpb ~]$ sudo dnf install nfs-utils

[lb@rhel9\_bpb ~]$ sudo systemctl enable nfs-server rpcbind

[lb@rhel9\_bpb ~]$ sudo systemctl start nfs-server rpcbind

[lb@rhel9\_bpb ~]$ sudo mkdir /sharedfiles

[lb@rhel9\_bpb ~]$ sudo nano /etc/exports

[lb@rhel9\_bpb ~]$ sudo exportfs -ra

[lb@rhel9\_bpb ~]$ sudo systemctl restart nfs-server

[lb@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --add-service=nfs

[lb@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --add-service=mountd

[lb@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --add-service=rpc-bind

[lb@rhel9\_bpb ~]$ sudo firewall-cmd --reload

[lb@rhel9\_bpb ~]$ sudo dnf install nfs-utils

[lb@rhel9\_bpb ~]$ sudo mkdir /mnt/nfs

[lb@rhel9\_bpb ~]$ sudo mount -t nfs nfs\_server\_ip:/sharedfiles /mnt/nfs

### Chapter 12

#### Recipe #67

[lb@rhel9\_bpb ~]$ sudo dnf install -y java-1.8.0-openjdk

[lb@rhel9\_bpb ~]$ sudo dnf install -y java-1.8.0-openjdk-headless

[lb@rhel9\_bpb ~]$ sudo dnf install -y java-11-openjdk

[lb@rhel9\_bpb ~]$ sudo dnf install -y java-11-openjdk-headless

[lb@rhel9\_bpb ~]$ sudo dnf install -y java-17-openjdk

[lb@rhel9\_bpb ~]$ sudo dnf install -y java-17-openjdk-headless

[lb@rhel9\_bpb ~]$ java -version

[lb@rhel9\_bpb ~]$ export JAVA\_HOME=$(dirname $(dirname $(readlink -f $(which java))))

[lb@rhel9\_bpb ~]$ export PATH=$JAVA\_HOME/bin:$PATH

[lb@rhel9\_bpb ~]$ source ~/.bashrc # For user-specific changes

[lb@rhel9\_bpb ~]$ source /etc/profile # For system-wide changes

#### Recipe #68

[lb@rhel9\_bpb ~]$ sudo dnf install -y dotnet-sdk-6.0

[lb@rhel9\_bpb ~]$ dotnet --info

[lb@rhel9\_bpb ~]$ dotnet new console --output my-app

[lb@rhel9\_bpb ~]$ dotnet run --project my-app

[lb@rhel9\_bpb ~]$ dotnet publish my-app -f net6.0 -c Release

[lb@rhel9\_bpb ~]$ dotnet publish my-app -f net6.0 -c Release -r rhel.9-x64 --self-contained false

[lb@rhel9\_bpb ~]$ dotnet new mvc --output mvc\_runtime\_example

[lb@rhel9\_bpb ~]$ dotnet publish mvc\_runtime\_example -f net6.0 -c Release

[lb@rhel9\_bpb ~]$ cat > Dockerfile <<EOF

FROM registry.access.redhat.com/ubi8/dotnet-60-runtime

ADD bin/Release/net6.0/publish/ .

CMD ["dotnet", "mvc\_runtime\_example.dll"]

EOF

[lb@rhel9\_bpb ~]$ podman build -t dotnet-60-runtime-example .

[lb@rhel9\_bpb ~]$ podman run -d -p 8080:8080 dotnet-60-runtime-example

#### Recipe #69

[lb@rhel9\_bpb ~]$ sudo yum install -y git

[lb@rhel9\_bpb ~]$ git –version  
[lb@rhel9\_bpb ~]$ git config --global user.name "Your Name"

[lb@rhel9\_bpb ~]$ git config --global user.email "your.email@example.com"

[lb@rhel9\_bpb ~]$ git config --list

[lb@rhel9\_bpb ~]$ mkdir ~/my-git-repo

[lb@rhel9\_bpb ~]$ cd ~/my-git-repo

[lb@rhel9\_bpb ~]$ git init

[lb@rhel9\_bpb ~]$ ls -a

[lb@rhel9\_bpb ~]$ echo "# My Git Repository" > README.md

[lb@rhel9\_bpb ~]$ git add README.md

[lb@rhel9\_bpb ~]$ git commit -m "Initial commit"

[lb@rhel9\_bpb ~]$ git log

[lb@rhel9\_bpb ~]$ git remote add origin https://github.com/yourusername/your-repo.git

[lb@rhel9\_bpb ~]$ git push -u origin master

#### Recipe #70

[lb@rhel9\_bpb ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

[lb@rhel9\_bpb ~]$ sudo yum install -y jenkins

[lb@rhel9\_bpb ~]$ sudo systemctl start jenkins

[lb@rhel9\_bpb ~]$ sudo systemctl enable jenkins

[lb@rhel9\_bpb ~]$ sudo firewall-cmd --permanent --zone=public --add-port=8080/tcp

[lb@rhel9\_bpb ~]$ sudo firewall-cmd --reload

[lb@rhel9\_bpb ~]$ sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

[lb@rhel9\_bpb ~]$ sudo yum install -y docker-ce docker-ce-cli containerd.io

[lb@rhel9\_bpb ~]$ sudo systemctl start docker

[lb@rhel9\_bpb ~]$ sudo systemctl enable docker

[lb@rhel9\_bpb ~]$ sudo usermod -aG docker $USER

newgrp docker

[lb@rhel9\_bpb ~]$ docker –version  
[lb@rhel9\_bpb ~]$ sudo yum install -y ansible-core

[lb@rhel9\_bpb ~]$ ansible --version

[lb@rhel9\_bpb ~]$ sudo yum install -y conntrack

[lb@rhel9\_bpb ~]$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

sudo install minikube-linux-amd64 /usr/local/bin/minikube

[lb@rhel9\_bpb ~]$ minikube start --driver=none

[lb@rhel9\_bpb ~]$ minikube status

[lb@rhel9\_bpb ~]$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

[lb@rhel9\_bpb ~]$ sudo install kubectl /usr/local/bin/kubectl

[lb@rhel9\_bpb ~]$ kubectl version --client

#### Recipe #71

[lb@rhel9\_bpb ~]$ sudo dnf install -y ansible-core

[lb@rhel9\_bpb ~]$ ansible --version

[lb@rhel9\_bpb ~]$ mkdir ~/ansible

[lb@rhel9\_bpb ~]$ cd ~/ansible

[lb@rhel9\_bpb ~]$ vi hosts

[lb@rhel9\_bpb ~]$ vi ansible.cfg

[lb@rhel9\_bpb ~]$ vi site.yml

[lb@rhel9\_bpb ~]$ vi index.html

[lb@rhel9\_bpb ~]$ ansible-playbook site.yml

### Chapter 13

#### Recipe #72

[lb@rhel9-bpb ~]$ sudo dnf install -y pcs pacemaker fence-agents-all

[lb@rhel9-bpb ~]$ sudo systemctl start pcsd.service

[lb@rhel9-bpb ~]$ sudo systemctl enable pcsd.service

[lb@rhel9-bpb ~]$ sudo firewall-cmd –permanent –add-service=high-availability

[lb@rhel9-bpb ~]$ sudo firewall-cmd –reload

[lb@rhel9-bpb ~]$ sudo passwd hacluster

[lb@rhel9-bpb ~]$ sudo pcs host auth <node\_hostname>

[lb@rhel9-bpb ~]$ sudo pcs cluster setup my\_cluster –start <node\_hostname>

[lb@rhel9-bpb ~]$ sudo pcs cluster status

[lb@rhel9-bpb ~]$ sudo pcs property set stonith-enabled=false

[lb@rhel9-bpb ~]$ sudo dnf install -y httpd wget

[lb@rhel9-bpb ~]$ sudo firewall-cmd –permanent –add-service=http

[lb@rhel9-bpb ~]$ sudo firewall-cmd –reload

[lb@rhel9-bpb ~]$ echo “<html><body>My Test Site - $(hostname)</body></html>” | sudo tee /var/www/html/index.html

[lb@rhel9-bpb ~]$ cat <<EOF | sudo tee /etc/httpd/conf.d/status.conf

[lb@rhel9-bpb ~]$ sudo pcs resource create ClusterIP ocf:heartbeat:Ipaddr2 ip=<floating\_ip> --group apachegroup

sudo pcs resource create WebSite ocf:heartbeat:apache configfile=/etc/httpd/conf/httpd.conf statusurl=”http://localhost/server-status” –group apachegroup

[lb@rhel9-bpb ~]$ sudo pcs status

[lb@rhel9-bpb ~]$ sudo killall -9 httpd

[lb@rhel9-bpb ~]$ sudo pcs status

[lb@rhel9-bpb ~]$ sudo pcs resource cleanup WebSite

[lb@rhel9-bpb ~]$ sudo pcs cluster stop –all

#### Recipe #73

dnf install pcs pacemaker fence-agents-all

firewall-cmd --permanent --add-service=high-availability

firewall-cmd --reload

systemctl start pcsd.service

systemctl enable pcsd.service

passwd hacluster

pcs cluster setup my\_cluster --start node1.example.com node2.example.com

pcs cluster enable --all

pcs stonith create myapc fence\_apc\_snmp ipaddr=192.168.1.200 \

pcmk\_host\_map="node1.example.com:1;node2.example.com:2" \

login=apc passwd=apc

pvcreate /dev/sdb1

vgcreate --setautoactivation n my\_vg /dev/sdb1

lvcreate -L 1G -n my\_lv my\_vg

mkfs.xfs /dev/my\_vg/my\_lv

dnf install httpd

pcs resource create my\_lvm ocf:heartbeat:LVM-activate vgname=my\_vg vg\_access\_mode=system\_id --group apachegroup

pcs resource create my\_fs Filesystem device="/dev/my\_vg/my\_lv" directory="/var/www" fstype="xfs" --group apachegroup

pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.1.100 cidr\_netmask=24 --group apachegroup

pcs resource create Website apache configfile="/etc/httpd/conf/httpd.conf" statusurl="http://localhost/server-status" --group apachegroup

#### Recipe #74

dnf install pcs pacemaker fence-agents-all httpd

firewall-cmd --permanent --add-service=high-availability

firewall-cmd --reload

systemctl start pcsd.service

systemctl enable pcsd.service

pcs host auth node1 node2

pcs cluster setup my\_cluster --start node1 node2

pcs cluster enable --all

pcs stonith create myfence fence\_apc\_snmp ipaddr=FENCING\_IP \

pcmk\_host\_map="node1:1;node2:2" \

login=FENCING\_USERNAME passwd=FENCING\_PASSWORD

pcs resource create web\_lvm ocf:heartbeat:LVM-activate vgname=vg\_web vg\_access\_mode=system\_id --group webserver

pcs resource create web\_fs Filesystem device="/dev/vg\_web/lv\_web" directory="/var/www" fstype="xfs" --group webserver

pcs resource create web\_ip ocf:heartbeat:IPaddr2 ip=FLOATING\_IP cidr\_netmask=24 --group webserver

pcs resource create web\_apache apache configfile="/etc/httpd/conf/httpd.conf" statusurl="http://localhost/server-status" --group webserver

#### Recipe #75

dnf install pcs pacemaker fence-agents-all nfs-utils lvm2-lockd

firewall-cmd --permanent --add-service=high-availability

firewall-cmd --permanent --add-service=nfs

firewall-cmd –reload

systemctl start pcsd.service

systemctl enable pcsd.service

pcs host auth node1 node2

pcs cluster setup my\_cluster --start node1 node2

pcs cluster enable --all

pcs stonith create myfence fence\_apc\_snmp ipaddr=FENCING\_IP \

pcmk\_host\_map="node1:1;node2:2" \

login=FENCING\_USERNAME passwd=FENCING\_PASSWORD

pcs resource create nfs\_lvm ocf:heartbeat:LVM-activate vgname=vg\_nfs vg\_access\_mode=lvmlockd --group nfs\_server

pcs resource create nfs\_fs Filesystem device="/dev/vg\_nfs/lv\_nfs" directory="/nfsshare" fstype="xfs" --group nfs\_server

pcs resource create nfs\_ip ocf:heartbeat:IPaddr2 ip=FLOATING\_IP cidr\_netmask=24 --group nfs\_server

pcs resource create nfs\_daemon nfsserver nfs\_shared\_infodir=/nfsshare/nfsinfo nfs\_no\_notify=true --group nfs\_server

#### Recipe #76

pcs resource relocate run resource\_name

pcs resource relocate clear resource\_name

pcs resource move resource\_name target\_node --wait

pcs resource disable resource\_name

pcs resource enable resource\_name

pcs resource ban resource\_name node\_name

pcs resource clear resource\_name node\_name

pcs node standby node\_name

pcs node unstandby node\_name

pcs cluster node add new\_node\_name

pcs cluster node remove node\_name

### Chapter 14

#### Recipe #77

dnf updateinfo list security --installed

#### Recipe #78

dnf updateinfo list updates security

#### Recipe #79

dnf updateinfo info RHSA-2019:0997

dnf update --advisory=RHSA-2019:0997

dnf upgrade-minimal --advisory=RHSA-2019:0997

dnf needs-restarting

#### Recipe #80

sudo vi /etc/dnf/automatic.conf

sudo systemctl enable --now dnf-automatic-install.timer

sudo systemctl status dnf-automatic-install.timer

man dnf-automatic

#### Recipe #81

lsblk

sudo cryptsetup luksFormat /dev/nvme0n1p1

sudo cryptsetup open /dev/nvme0n1p1 nvme0n1p1\_encrypted

sudo mkfs.ext4 /dev/mapper/nvme0n1p1\_encrypted

sudo mount /dev/mapper/nvme0n1p1\_encrypted /mnt/encrypted

sudo umount /dev/sdx1

sudo cryptsetup reencrypt --encrypt --reduce-device-size 32M /dev/sdx1

#### Recipe #82

sudo dnf install firewalld

sudo systemctl start firewalld

sudo systemctl enable firewalld

sudo systemctl status firewalld

sudo firewall-cmd --get-active-zones

sudo firewall-cmd --zone=home --change-interface=eth0 --permanent

sudo firewall-cmd --reload

sudo firewall-cmd --zone=public --add-service=ssh --permanent

sudo firewall-cmd --reload

sudo firewall-cmd --zone=public --list-services

sudo firewall-cmd --zone=public --add-port=443/tcp --permanent

sudo firewall-cmd --reload

sudo firewall-cmd --zone=public --list-ports

sudo firewall-cmd --zone=public --remove-port=443/tcp --permanent

sudo firewall-cmd --reload

ansible-playbook open\_port\_443.yml

#### Recipe #83

iptables-save > /root/iptables.dump

ip6tables-save > /root/ip6tables.dump

iptables-restore-translate -f /root/iptables.dump > /etc/nftables/ruleset-migrated-from-iptables.nft

ip6tables-restore-translate -f /root/ip6tables.dump > /etc/nftables/ruleset-migrated-from-ip6tables.nft

include "/etc/nftables/ruleset-migrated-from-iptables.nft"

include "/etc/nftables/ruleset-migrated-from-ip6tables.nft"

systemctl disable --now iptables

systemctl enable --now nftables

nft list ruleset

/etc/nftables/example\_script.nft

nft -f /etc/nftables/example\_script.nft

nft add table ip nat

nft add chain ip nat postrouting { type nat hook postrouting priority 100 \; }

nft add rule ip nat postrouting oifname "ens3" masquerade

nft add table ip nat

nft add chain ip nat prerouting { type nat hook prerouting priority -100 \; }

nft add rule ip nat prerouting tcp dport 80 dnat to 192.0.2.1:8080

nft add set inet example\_table example\_set { type ipv4\_addr \; }

nft add element inet example\_table example\_set { 192.0.2.1, 192.0.2.2 }

nft add rule inet example\_table example\_chain ip saddr @example\_set drop

nft add map inet example\_table example\_map { type ipv4\_addr : verdict \; }

nft add rule inet example\_table example\_chain ip saddr vmap @example\_map

nft add element inet example\_table example\_map { 192.0.2.1 : accept, 192.0.2.2 : drop }

nft add rule inet example\_table example\_chain tcp dport 22 counter accept

nft replace rule inet example\_table example\_chain handle 4 tcp dport 22 meta nftrace set 1 accept

nft monitor

#### Recipe #84

linux /vmlinuz ... fips=1

fips-mode-setup --check

fips-mode-setup --enable

fips-mode-setup --check

#### Recipe #85

$ sestatus

$ sudo setenforce 1

$ sudo setenforce 0

$ sudo semanage user -a -R "SELinux\_Role" new\_user

$ sudo semanage login -a -s SELinux\_User login\_name

$ sudo semanage fcontext -a -t httpd\_sys\_content\_t "/custom/directory(/.\*)?"

$ sudo restorecon -Rv /custom/directory

$ sudo semanage boolean -l

$ sudo setsebool boolean\_name on

$ sudo ausearch -m avc -ts recent\_date

$ sudo sealert -a /var/log/audit/audit.log

$ sudo audit2allow -M mymodule < /path/to/log/file

$ sudo semodule -i mymodule.pp

#### Recipe #86

$ sudo dnf install openssh-server openssh-clients

$ sudo systemctl enable sshd

$ sudo systemctl start sshd

$ sudo systemctl status sshd

$ sudo firewall-cmd --permanent --add-service=ssh

$ sudo firewall-cmd --reload

$ sudo systemctl restart sshd

$ ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"

$ ssh-copy-id user@server\_ip

$ sudo setsebool -P ssh\_home\_tty 1

$ sudo ausearch -m avc -c sshd

$ sudo audit2allow -M mypol < /path/to/audit.log

$ sudo semodule -i mypol.pp

$ sudo grep sshd /var/log/secure

$ sudo dnf install fail2ban

#### Recipe #87

$ sudo dnf install ansible rhel-system-roles

$ ansible-playbook configure\_ssh\_clients.yml

$ cat ~/.ssh/config

#### Recipe #88

sudo dnf install httpd mod\_ssl openssl

sudo dnf install nginx openssl

sudo openssl genpkey -algorithm RSA -out /etc/ssl/private/server.key -aes256

sudo openssl req -new -key /etc/ssl/private/server.key -out /etc/ssl/certs/server.csr

sudo openssl x509 -req -days 365 -in /etc/ssl/certs/server.csr -signkey /etc/ssl/private/server.key -out /etc/ssl/certs/server.crt

sudo nano /etc/httpd/conf.d/ssl.conf

sudo systemctl restart httpd

sudo systemctl restart nginx

openssl s\_client -connect your\_domain.com:443

sudo dnf install certbot python3-certbot-apache

sudo dnf install certbot python3-certbot-nginx

sudo certbot --apache

sudo certbot --nginx

sudo certbot renew --dry-run

#### Recipe #89

sudo dnf install libreswan

sudo nano /etc/ipsec.conf

sudo nano /etc/ipsec.d/vpn.conf

sudo nano /etc/ipsec.secrets

sudo systemctl start ipsec

sudo systemctl enable ipsec

sudo systemctl status ipsec

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

sudo firewall-cmd --reload

sudo ipsec status

#### Recipe #90

sudo dnf install nfs-utils

sudo nano /etc/exports

sudo exportfs -rav

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

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

sudo firewall-cmd --permanent --add-service=rpc-bind

sudo firewall-cmd --reload

sudo setsebool -P nfs\_export\_all\_rw 1

sudo semanage fcontext -a -t nfs\_t "/data/nfs\_share(/.\*)?"

sudo restorecon -Rv /data/nfs\_share

sudo dnf install krb5-server krb5-workstation

### Chapter 15

#### Recipe #92

sudo dnf install sysstat

sudo systemctl enable sysstat

sudo systemctl start sysstat

sudo vi /etc/sysconfig/sysstat

sudo systemctl restart sysstat

sar -u 1 5

iostat -dx 5 3

free -m

sar -n DEV 1 3

sar -A > /tmp/system\_performance\_report.txt

sudo crontab -e

#### Recipe #93

sudo dnf update -y

sudo dnf install wget -y

wget https://github.com/prometheus/prometheus/releases/download/v2.54.1/prometheus-2.54.1.linux-amd64.tar.gz

tar xvf prometheus-2.54.1.linux-amd64.tar.gz

sudo mv prometheus-2.54.1.linux-amd64/prometheus /usr/local/bin/

sudo mv prometheus-2.54.1.linux-amd64/promtool /usr/local/bin/

sudo mv prometheus-2.54.1.linux-amd64/prometheus.yml /etc/prometheus/

sudo mv prometheus-2.54.1.linux-amd64/consoles /etc/prometheus/

sudo mv prometheus-2.54.1.linux-amd64/console\_libraries /etc/prometheus/

sudo useradd --no-create-home --shell /bin/false prometheus

sudo chown -R prometheus:prometheus /etc/prometheus

sudo chown prometheus:prometheus /usr/local/bin/prometheus

sudo chown prometheus:prometheus /usr/local/bin/promtool

sudo vi /etc/systemd/system/prometheus.service

sudo systemctl daemon-reload

sudo systemctl enable prometheus

sudo systemctl start prometheus

sudo systemctl status prometheus

sudo systemctl reload Prometheus

wget https://github.com/prometheus/node\_exporter/releases/download/v1.8.2/node\_exporter-1.8.2.linux-amd64.tar.gz

tar xvf node\_exporter-1.8.2.linux-amd64.tar.gz

sudo mv node\_exporter-1.8.2.linux-amd64/node\_exporter /usr/local/bin/

sudo vi /etc/systemd/system/node\_exporter.service

sudo systemctl daemon-reload

sudo systemctl enable node\_exporter

sudo systemctl start node\_exporter

sudo systemctl reload prometheus

#### Recipe #94

pip install orca

import pandas as pd

# Load the CSV file into a DataFrame

df = pd.read\_csv('cpu\_usage.csv')

# Display the first few rows of the DataFrame

print(df.head())

df['time'] = pd.to\_datetime(df['time'])

plt.figure(figsize=(10, 6))

plt.plot(df['time'], df['value'], label='CPU Usage')

plt.xlabel('Time')

plt.ylabel('CPU Usage (%)')

plt.title('CPU Usage Over Time')

plt.legend()

plt.grid(True)

plt.show()

avg\_cpu\_usage = df['value'].mean()

print(f'Average CPU Usage: {avg\_cpu\_usage:.2f}%')

# Detecting spikes in CPU usage

spikes = df[df['value'] > avg\_cpu\_usage + 2 \* df['value'].std()]

print(f'Number of spikes detected: {len(spikes)}')

import plotly.express as px

fig = px.line(df, x='time', y='value', title='CPU Usage Over Time')

fig.update\_xaxes(title\_text='Time')

fig.update\_yaxes(title\_text='CPU Usage (%)')

fig.show()

#### Recipe #95

sudo vi /etc/prometheus/alert.rules.yml

sudo systemctl reload prometheus

wget https://github.com/prometheus/alertmanager/releases/download/v0.27.0/alertmanager-0.27.0.linux-amd64.tar.gz

tar xvf alertmanager-0.27.0.linux-amd64.tar.gz

sudo mv alertmanager-0.27.0.linux-amd64/alertmanager /usr/local/bin/

sudo mv alertmanager-0.27.0.linux-amd64/amtool /usr/local/bin/

sudo vi /etc/alertmanager/alertmanager.yml

sudo vi /etc/systemd/system/alertmanager.service

sudo systemctl daemon-reload

sudo systemctl start alertmanager

sudo systemctl enable alertmanager

sudo systemctl status alertmanager

sudo systemctl reload prometheus

#### Recipe #96

sudo dnf install audit

sudo systemctl enable auditd

sudo systemctl start auditd

sudo vi /etc/audit/auditd.conf

sudo vi /etc/audit/rules.d/audit.rules

sudo augenrules --load

sudo systemctl restart auditd

sudo systemctl status auditd

sudo less /var/log/audit/audit.log

sudo ausearch -k passwd\_changes

sudo aureport --summary

sudo chattr +i /var/log/audit/audit.log

#### Recipe #97

auditctl -w /etc/passwd -p wa -k passwd\_changes

auditctl -a always,exit -F arch=b64 -S chmod -S fchmod -S fchmodat -k permission\_change

auditctl -a always,exit -F arch=b64 -F auid=$(id -u user1) -S execve -k user\_commands

auditctl -w /var/log/secure -p rwxa -k secure\_log\_access

auditctl -l

auditctl -D

auditctl -e 1

auditctl -e 0

auditctl -s

auditctl -r 100

auditctl -b 8192

#### Recipe #98

sudo dnf install scap-security-guide

oscap info /usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml

sudo oscap xccdf eval --profile xccdf\_org.ssgproject.content\_profile\_cis --remediate /usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml

sudo cp /usr/share/audit/sample-rules/30-pci-dss-v31.rules /etc/audit/rules.d/

sudo augenrules –load

sudo auditctl -l

sudo vi /etc/audit/rules.d/30-pci-dss-v31.rules

sudo augenrules --load

sudo aureport --summary

echo "0 0 \* \* \* root /usr/sbin/aureport --summary > /var/log/audit/daily\_summary\_report.txt" | sudo tee -a /etc/crontab

sudo vi /etc/rsyslog.d/audit.conf

sudo systemctl restart rsyslog

#### Recipe #99

auditctl -w /var/log/secure -p wa -k logins

auditctl -a always,exit -F arch=b64 -S execve -C uid!=euid -F euid=0 -k pam

auditctl -w /usr/sbin/useradd -p x -k user\_mgmt

auditctl -w /usr/sbin/usermod -p x -k user\_mgmt

auditctl -w /usr/sbin/userdel -p x -k user\_mgmt

echo "-w /var/log/secure -p wa -k logins" | sudo tee -a /etc/audit/rules.d/audit.rules

echo "-a always,exit -F arch=b64 -S execve -C uid!=euid -F euid=0 -k pam" | sudo tee -a /etc/audit/rules.d/audit.rules

echo "-w /usr/sbin/useradd -p x -k user\_mgmt" | sudo tee -a /etc/audit/rules.d/audit.rules

echo "-w /usr/sbin/usermod -p x -k user\_mgmt" | sudo tee -a /etc/audit/rules.d/audit.rules

echo "-w /usr/sbin/userdel -p x -k user\_mgmt" | sudo tee -a /etc/audit/rules.d/audit.rules

sudo augenrules --load

sudo ausearch -k logins

sudo ausearch -ua <username>

sudo aureport -l

$ last

$ w

echo "0 0 \* \* \* root /usr/sbin/aureport -l > /var/log/audit/login\_report\_$(date +\%Y\%m\%d).txt" | sudo tee -a /etc/crontab

sudo vi /etc/rsyslog.d/audit.conf

sudo systemctl restart rsyslog

#### Recipe #100

auditctl -w /usr/bin/dnf -p x -k package\_mgmt

auditctl -w /var/lib/rpm/ -p wa -k rpm\_db\_changes

echo "-w /usr/bin/dnf -p x -k package\_mgmt" | sudo tee -a /etc/audit/rules.d/audit.rules

echo "-w /usr/bin/yum -p x -k package\_mgmt" | sudo tee -a /etc/audit/rules.d/audit.rules

echo "-w /var/lib/rpm/ -p wa -k rpm\_db\_changes" | sudo tee -a /etc/audit/rules.d/audit.rules

sudo augenrules --load

sudo ausearch -k package\_mgmt

sudo ausearch -k rpm\_db\_changes

sudo aureport -k --summary | grep -E "package\_mgmt|rpm\_db\_changes"

sudo cat /var/log/dnf.log

sudo tail -f /var/log/dnf.log

echo "0 0 \* \* \* root /usr/sbin/aureport -k --summary > /var/log/audit/package\_mgmt\_report\_$(date +\%Y\%m\%d).txt" | sudo tee -a /etc/crontab

sudo vi /etc/rsyslog.d/audit.conf

sudo systemctl restart rsyslog

### Chapter 16

#### Recipe #101

sudo dnf update -y

sudo dnf groupinstall "Development Tools" -y

sudo dnf install epel-release -y

sudo dnf install python3 python3-devel gcc gcc-c++ make git -y

sudo subscription-manager repos --enable=rhel-9-server-optional-rpms

sudo subscription-manager repos --enable=rhel-9-server-extras-rpms

sudo dnf install rhel-ai -y

rhel-ai --version

python3 -m venv rhel-ai-env

source rhel-ai-env/bin/activate

sudo dnf install nvidia-driver nvidia-cuda-toolkit -y

export CUDA\_HOME=/usr/local/cuda

export PATH=$CUDA\_HOME/bin:$PATH

export LD\_LIBRARY\_PATH=$CUDA\_HOME/lib64:$LD\_LIBRARY\_PATH

sudo dnf install rocm-dkms -y

export ROCM\_HOME=/opt/rocm

export PATH=$ROCM\_HOME/bin:$PATH

export LD\_LIBRARY\_PATH=$ROCM\_HOME/lib64:$LD\_LIBRARY\_PATH

rhel-ai init

rhel-ai model download --model-name <model\_name>

rhel-ai run --test

#### Recipe #102

python3 -m venv rhel-ai-env

source rhel-ai-env/bin/activate

pip install rhel-ai

pip install tensorflow

rhel-ai init

rhel-ai model download --model-name <model\_name>

rhel-ai run --test

import pandas as pd

df = pd.read\_csv('data.csv')

df\_clean = df.dropna() # Remove missing values

df\_processed = df\_clean.apply(lambda x: x\*2) # Example transformation

df\_processed.to\_csv('processed\_data.csv')

rhel-ai data ingest --input data.csv --output clean\_data.csv

rhel-ai model train --data train\_data.csv --model random\_forest

rhel-ai model deploy --model random\_forest --environment production

rhel-ai model serve --model random\_forest --optimize

rhel-ai model test --endpoint http://localhost:8000/predict

#### Recipe #103

sudo dnf install cuda

sudo dnf install rocm

rhel-ai config set gpu=true

rhel-ai model train --batch-size 64

rhel-ai model train --precision half

rhel-ai model deploy --cloud aws --instance-type p3.2xlarge

aws autoscaling set-desired-capacity --auto-scaling-group-name myScalingGroup --desired-capacity 5

aws s3 sync /local/data s3://my-bucket/data

rhel-ai model train --data-source hybrid --on-prem-gpus 2 --cloud-gpus 4

rhel-ai model deploy --hybrid --on-prem-server myServer --cloud-instance p3.8xlarge

rhel-ai encrypt --data /path/to/data --key myEncryptionKey

rhel-ai access control --role AIEngineer --allow model.train,model.deploy

rhel-ai audit enable --log /var/log/rhel-ai/audit.log

rhel-ai data anonymize --input raw\_data.csv --output anon\_data.csv

rhel-ai compliance audit --standard GDPR --output audit\_report.txt

rhel-ai report generate --compliance --output compliance\_report.pdf

#### Recipe #104

python3 -m venv ai-env

source ai-env/bin/activate

pip install tensorflow keras scikit-learn

rhel-ai model train --dataset path/to/dataset --epochs 50 --gpus 2

rhel-ai model evaluate --model path/to/model --dataset path/to/validation\_dataset

rhel-ai model deploy --model path/to/custom\_model --endpoint /ai/model/predict

pip install spacy matplotlib seaborn

import spacy

nlp = spacy.load("en\_core\_web\_sm")

doc = nlp("RHEL AI makes AI development easy.")

print([(token.text, token.pos\_) for token in doc])

import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'])

plt.title('Model Accuracy')

plt.ylabel('Accuracy')

plt.xlabel('Epoch')

plt.show()

rhel-ai data preprocess --input raw\_data.csv --output preprocessed\_data.csv

rhel-ai model train --dataset preprocessed\_data.csv --epochs 50

rhel-ai model deploy --model trained\_model

rhel-ai monitor --pipeline my\_pipeline --log /var/log/ai\_pipeline.log

#### Recipe #105

sudo apt-get install prometheus

sudo apt-get install grafana

sudo systemctl start prometheus

sudo systemctl start grafana-server

ilab system info

ilab model serve --monitor

tail -f /var/log/instructlab/instructlab.log

grep "ERROR" /var/log/instructlab/instructlab.log

ilab model serve --verbose

ilab system info

ilab model chat --debug

pip install instructlab --upgrade

ilab --version

pip list --outdated

pip install --upgrade <package\_name>

python3 -m venv instructlab\_env

source instructlab\_env/bin/activate

ilab model test