



RHEL9

Session 12 – 4th Dec 2022 Summary

- The command to check “firewalld” installed “**rpm -q firewalld**”

```
[root@localhost ~]# rpm -q firewalld
firewalld-1.0.0-4.el9.noarch
```

- The command to check the status of “firewalld” “**systemctl status firewalld**”

```
[root@localhost ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled)
  Active: inactive (dead)
    Docs: man:firewalld(1)
lines 1-4/4 (END)
```

- The command to start the “firewalld” service “**systemctl start firewalld**”, the firewall decides whether to allow or block the network traffic based on the rules

```
[root@localhost ~]# systemctl start firewalld
[root@localhost ~]#
```

```
[root@localhost ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor>
  Active: active (running) since Sun 2022-12-04 14:20:08 IST; 1min 22s ago
    Docs: man:firewalld(1)
   Main PID: 2106 (firewalld)
      Tasks: 2 (limit: 50436)
     Memory: 27.1M
        CPU: 717ms
       CGroup: /system.slice/firewalld.service
                   └─2106 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nrepid

Dec 04 14:20:08 localhost.localdomain systemd[1]: Starting firewalld - dynamic >
Dec 04 14:20:08 localhost.localdomain systemd[1]: Started firewalld - dynamic f>
lines 1-13/13 (END)
```

- The command used to create rules is “**firewall-cmd**”, some of the rules are pre-created(zones) and some rules we create(custom rules)
- The command to check firewall is running “**firewall-cmd -- state**”

```
[root@localhost ~]# firewall-cmd --state
running
[root@localhost ~]#
```

- The command to see the pre-created zones given by RedHat “**firewall-cmd -- get - zones**”

```
[root@localhost ~]# firewall-cmd --get-zones
block dmz drop external home internal nm-shared public trusted work
[root@localhost ~]#
```

- The command to check the default zone “**firewall-cmd -- get-default - zone**”

```
[root@localhost ~]# firewall-cmd --get-default-zone
public
[root@localhost ~]#
```

- The “firewall” is implemented on the network card by the OS, in a system there may be multiple network cards. The command to check the active zone is “**firewall-cmd -- get - active - zones**”

```
[root@localhost ~]# firewall-cmd --get-active-zones
public
  interfaces: enp0s3
[root@localhost ~]#
```

- The command to list all the pre-created rules of the zone “**public**” is **firewall-cmd -- list - all -- zones=public**”, we see that they apply rules only on the **services**, anyone who tries to come to system on “ssh”, “cockpit” and “dhcpcv6-client” are allowed

```
[root@localhost ~]# firewall-cmd --list-all --zone=public
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpcv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@localhost ~]#
```

- The command to list all zones “**firewall-cmd -- list-all-zones**”. Every zone has a **target**, it will decide the final outcome. For example in the “**trusted**” zone the target is “**ACCEPT**”, even the services not listed, the final outcome is everybody allowed

```
[root@localhost ~]# firewall-cmd --list-all-zones
```

```
trusted
  target: ACCEPT
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

- The command to set the default zone “**firewall-cmd -- set – default – zone=trusted**

```
[root@localhost ~]# firewall-cmd --set-default-zone=trusted
success
[root@localhost ~]#
```

- The command to see the zone active “**firewall-cmd --get-active-zones**”

```
[root@localhost ~]# firewall-cmd --get-active-zones
trusted
interfaces: enp0s3
```

- The command to add the service “http” to the zone “public” is “**firewall-cmd --add-service=http --zone=public**”, we see that “http” service has been added. The target is “default”, all services specified is allowed and others are denied.

```
[root@localhost ~]# firewall-cmd --add-service=http --zone=public
```

```
[root@localhost ~]# firewall-cmd --list-all --zone=public
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpcv6-client http ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@localhost ~]#
```

- To make a **rule permanent**, while adding service we use a keyword “**--permanent**”

```
[root@localhost ~]# firewall-cmd --add-service=http --zone=public --permanent
success
[root@localhost ~]#
```

- The command to remove the service “http” , this means the client will not be able to connect

```
[root@localhost ~]# firewall-cmd --remove-service=http --zone=public
success
[root@localhost ~]#
```

```
[root@localhost ~]# firewall-cmd --list-all --zone=public
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpcv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:      →
  source-ports:
  icmp-blocks:
  rich rules:
[root@localhost ~]# █
```

- The command to add the port number 80 is

```
[root@localhost ~]# firewall-cmd --add-port=80/tcp --zone=public
success
[root@localhost ~]# █
```

```
[root@localhost ~]# firewall-cmd --list-all --zone=public
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpcv6-client ssh
  ports: 80/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@localhost ~]# █
```

- The command to see “rich rules” added is “**firewall – cmd -- list – rich – rules**”, here we see that no rich rules added

```
[root@localhost ~]# firewall-cmd --list-rich-rules
```

- The command to add “rich rule” (advance rule) to a zone “**public**” is “**firewall – cmd -- zone=public -- add – rich - rule**”, here anybody with the source IP ‘192.168.1.12’ go to ‘port 80’ on protocol ‘tcp’ is allowed

```
[root@localhost ~]# firewall-cmd --zone=public --add-rich-rule 'rule family="ip
v4" port port=80 protocol=tcp source address=192.168.1.12 accept'
success
[root@localhost ~]#
```

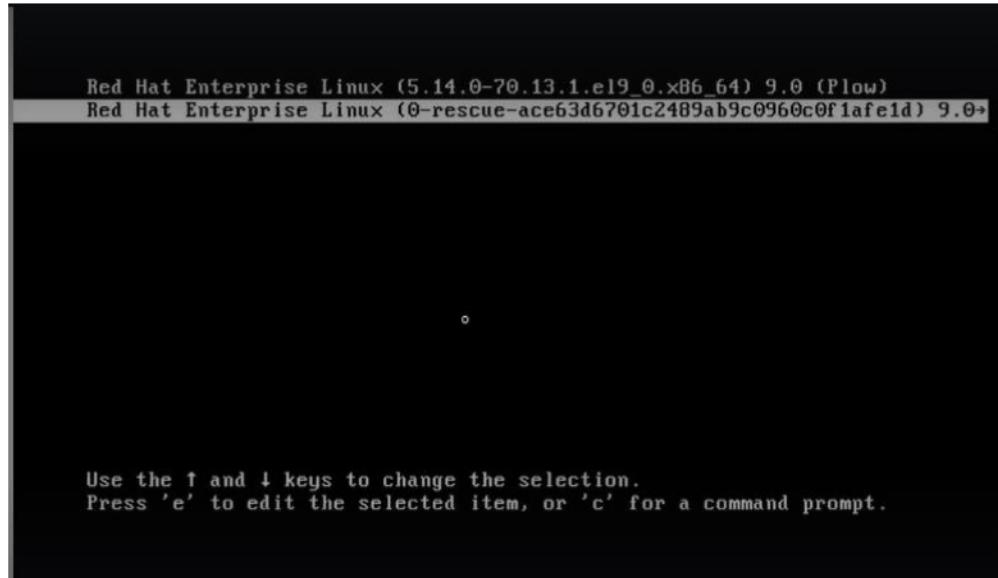
- Steps to reset the root password
 - Go to servera, reboot the system (alt+ctrl+del)

```
Red Hat Enterprise Linux 9.0 (Plow)
Kernel 5.14.0-70.13.1.el9_0.x86_64 on an x86_64

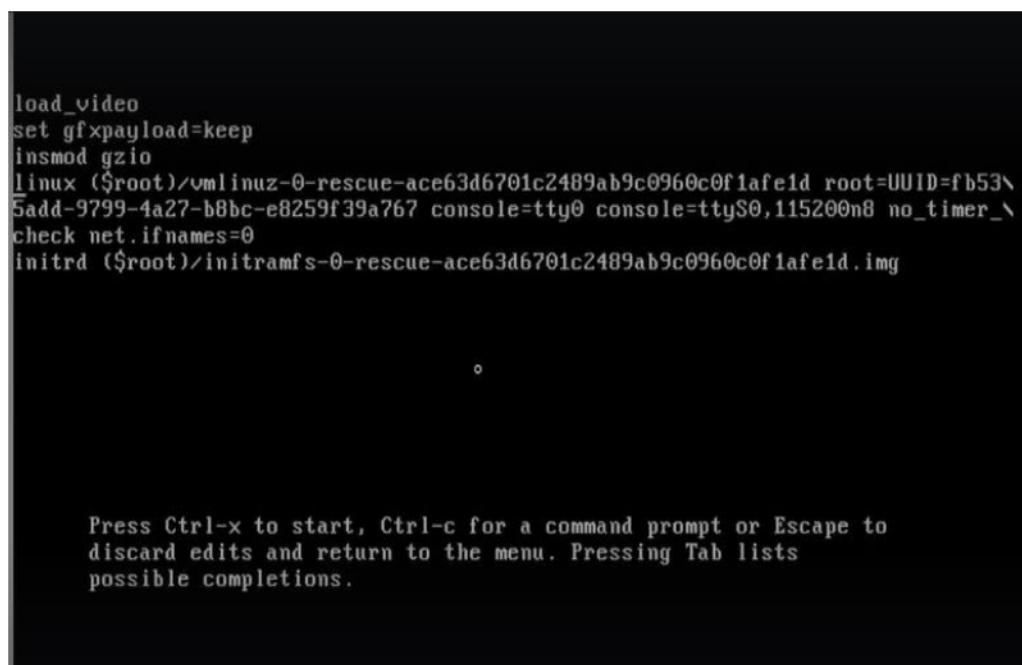
Activate the web console with: systemctl enable --now cockpit.socket

servera login:
```

- In the Grub screen- select “**second option**” and press “e” for edit option



- Go to line **number 4** and press “**end**” key, to go to the last of that line and write one keyword “**rd.break**”. During boot process, there is one stage in boot sequence, where system will login to root account without password. OS will break at this point in time, this “**rd.break**” keyword will help to login to root account without password. At the same time set enforcing=0, the SELinux is disabled.



```

load_video
set gfxpayload=keep
insmod gzio
linux ($root)/vmlinuz-0-rescue-ace63d6701c2489ab9c0960c0f1afe1d root=UUID=fb53\
5add-9799-4a27-b8bc-e8259f39a767 console=tty0 console=ttyS0,115200n8 no_timer_
check net.ifnames=0 rd.break enforcing=0
initrd ($root)/initramfs-0-rescue-ace63d6701c2489ab9c0960c0f1afe1d.img

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.

```

- Then press “**Ctrl – x**” to start the system, OS starts booting and system lands to break mode

```

sh-5.1#
sh-5.1#
sh-5.1# pwd
/root
sh-5.1#

```

- The main drive of linux OS is “/” drive, entire OS works on this drive, while booting actual content of this drive is mounted on /sysroot folder, on read only mode. If you want to change password, is cannot be done because it's read only. For this we have to remount the drive with read and write (rw) options.

```

sh-5.1# mount -o remount,rw /sysroot/
[ 147.262052] xfs filesystem being remounted at /sysroot supports timestamps until 2038 (0xffffffff)
sh-5.1# mount
none on / type rootfs (rw)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,size=961528k,nr_inodes=240382,mode=755,inode64)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,size=402796k,nr_inodes=819200,mode=755,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
/dev/vda4 on /sysroot type xfs (rw,relatime,attr2,inode64,logbufs=8,logbsize=32k,noquota)
rpc_pipefs on /sysroot/var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
sh-5.1# 

```

- The command to change the root to “/” drive is “**chroot /sysroot/**”

```
sh-5.1# chroot    /sysroot/
sh-5.1# pwd
/
sh-5.1#
```

- The command to change the root password “**passwd root**”

```
sh-5.1# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
sh-5.1# -
```

- Now login to root account with the reset password

```
Red Hat Enterprise Linux 9.0 (Plow)
Kernel 5.14.0-70.13.1.el9_0.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

servera login: root
Password:
Last failed login: Sun Dec  4 05:10:58 EST 2022 on tty1
There were 3 failed login attempts since the last successful login.
[root@servera ~]#
[root@servera ~]#
[root@servera ~]#
[root@servera ~]# _
```

- We see that SELinux is **disabled**, but it will again be **enable** if system is rebooted again

```
[root@servera ~]#  
[root@servera ~]# getenforce  
Permissive  
[root@servera ~]# _
```

- For this whatever permissions that SELinux has changed in the /etc/shadow file, we have to reset it

```
[root@servera ~]#  
[root@servera ~]#  
[root@servera ~]# ls -lZ /etc/shadow  
-----. 1 root root system_u:object_r:unlabeled_t:s0 1077 Dec 4 05:12 /etc/shadow  
[root@servera ~]# _
```

- The command to restore the by default SELinux context is “**restorecon /etc/shadow**”

```
[root@servera ~]# ls -lZ /etc/shadow  
-----. 1 root root system_u:object_r:unlabeled_t:s0 1077 Dec 4 05:12 /etc/shadow  
[root@servera ~]# restorecon /etc/shadow  
[root@servera ~]# ls -lZ /etc/shadow  
-----. 1 root root system_u:object_r:shadow_t:s0 1077 Dec 4 05:12 /etc/shadow  
[root@servera ~]#
```

- Now if you reboot the system again, you need not go to rescue program, you can login to servera with the password

- Refer the link to get to know the new RHEL9:-
<https://www.tecmint.com/rhel-9-download/>
- To configure the system with website(WebUI), **Cockpit** is used

```
[root@localhost ~]# yum install cockpit
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Repository 'dvd1' is missing name in configuration, using id.
Repository 'dvd2' is missing name in configuration, using id.
Last metadata expiration check: 1:15:10 ago on Sun 04 Dec 2022 02:43:59 PM IST.
Package cockpit-264.1-1.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

- The command to start the services “`systemctl start cockpit`”

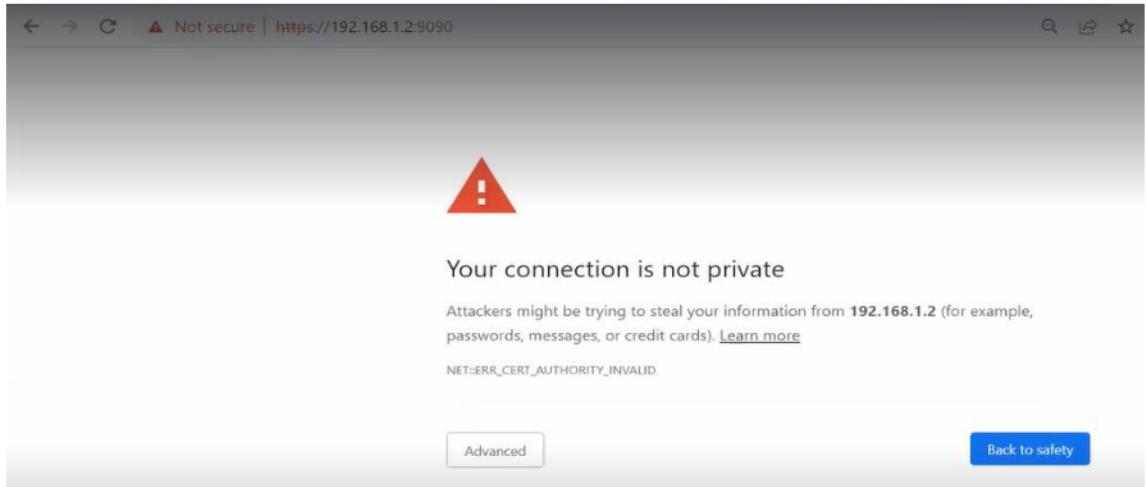
```
[root@localhost ~]# systemctl start cockpit
[root@localhost ~]# systemctl status cockpit
● cockpit.service - Cockpit Web Service
  Loaded: loaded (/usr/lib/systemd/system/cockpit.service; static)
  Active: active (running) since Sun 2022-12-04 15:59:24 IST; 3s ago
    TriggeredBy: ● cockpit.socket
    Docs: man:cockpit-ws(8)
   Process: 6046 ExecStartPre=/usr/libexec/cockpit-certificate-ensure --for-co>
 Main PID: 6068 (cockpit-tls)
    Tasks: 1 (limit: 50436)
   Memory: 2.1M
      CPU: 2.414s
     CGroup: /system.slice/cockpit.service
             └─6068 /usr/libexec/cockpit-tls

Dec 04 15:59:22 localhost.localdomain systemd[1]: Starting Cockpit Web Service.>
Dec 04 15:59:24 localhost.localdomain cockpit-certificate-ensure[6059]: .+....>
Dec 04 15:59:24 localhost.localdomain cockpit-certificate-ensure[6059]: .....>
Dec 04 15:59:24 localhost.localdomain cockpit-certificate-ensure[6059]: ----->
Dec 04 15:59:24 localhost.localdomain systemd[1]: Started Cockpit Web Service.
lines 1-18/18 (END)
```

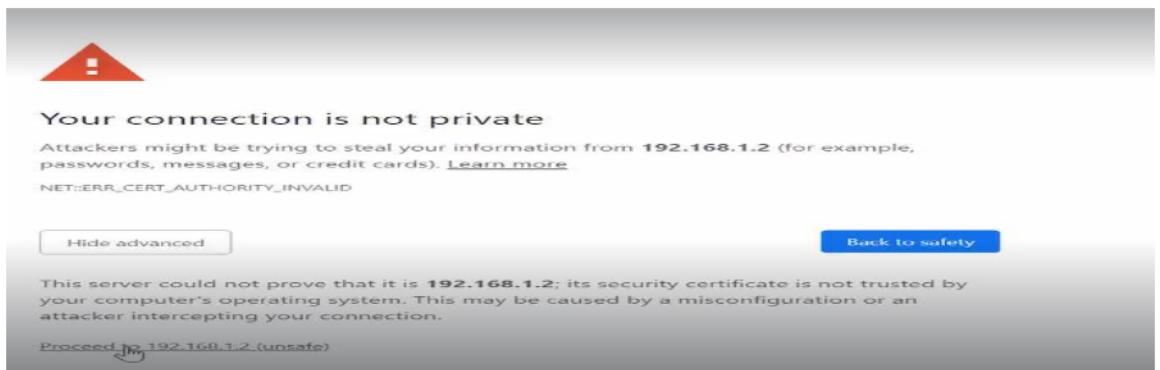
- The command to check the port number on which the cockpit works
“`netstat -tnlp`”

```
[root@localhost ~]# netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:111              0.0.0.0:*
1/systemd
tcp        0      0 0.0.0.0:8080             0.0.0.0:*
3455/httpd
tcp        0      0 0.0.0.0:80              0.0.0.0:*
3455/httpd
tcp        0      0 0.0.0.0:22              0.0.0.0:*
813/sshd: /usr/sbin
tcp        0      0 127.0.0.1:631            0.0.0.0:*
809/cupsd
tcp6       0      0 :::9090                :::*
1/systemd
tcp6       0      0 :::111                 :::*
1/systemd
tcp6       0      0 :::22                  :::*
813/sshd: /usr/sbin
tcp6       0      0 :::1:631               :::*
809/cupsd
```

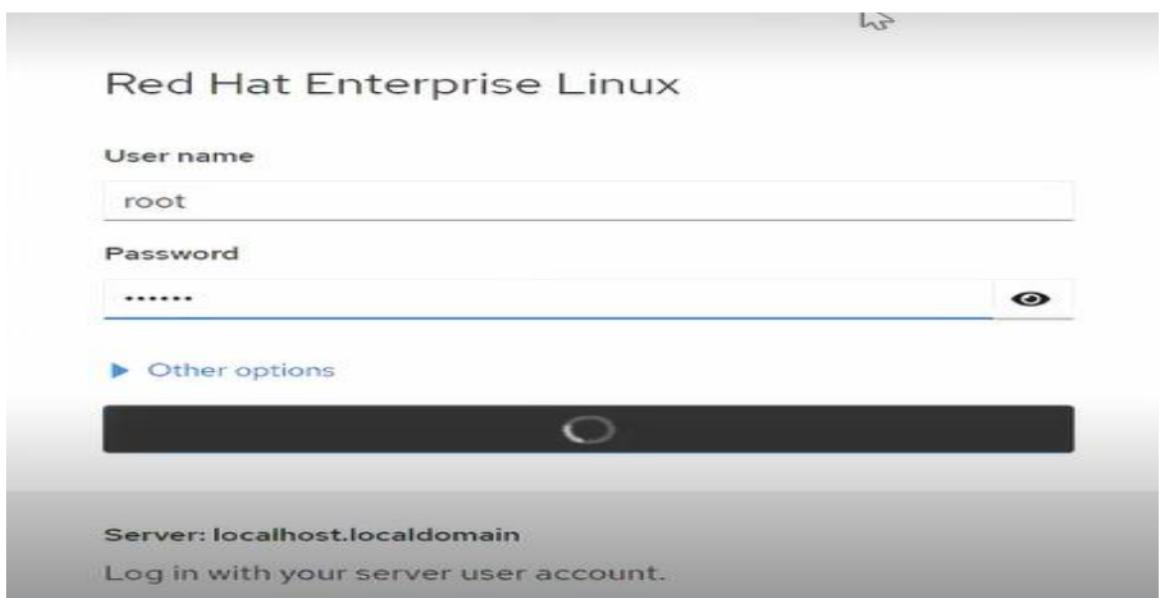
- From windows browser connect to the linux OS on port 9090



- Click on **Advance** options and then on **Proceed**



- Give the username and password, to login as root to manage the system by WebUI



Not secure | https://192.168.1.2:9090/system

root@localhost

Search

System

Overview

Logs

Storage

Networking

Accounts

Services

Tools

Applications

Diagnostic Reports

localhost running Red Hat Enterprise Linux 9.0 (Plow)

Reboot

Help Session

Welcome Back from diwali festival

now focus on study.....

Health

Checking for package updates...

Not connected to Insights

Last successful login: Dec 04, 02:19 PM on tty2

View login history

Usage

CPU 87% of 2 CPUs

Memory 14 / 7.8 GB

View details and history

System information

Model	innotek GmbH VirtualBox
Asset tag	0
Machine ID	d3540f72c0b14cf28dbd95181029652d
Uptime	about 2 hours

View hardware details

This screenshot shows the Red Hat Enterprise Linux 9.0 system dashboard. The left sidebar includes links for Overview, Logs, Storage, Networking, Accounts, Services, Tools, Applications, and Diagnostic Reports. The main content area displays the following sections: 'localhost running Red Hat Enterprise Linux 9.0 (Plow)' with a 'Reboot' button; a message box with a welcome message and a note to focus on study; 'Health' status with a note about package updates and being disconnected from Insights; 'Usage' metrics for CPU (87% of 2 CPUs) and Memory (14 / 7.8 GB); and 'System information' with details like Model (innotek GmbH VirtualBox), Asset tag (0), Machine ID (d3540f72c0b14cf28dbd95181029652d), and Uptime (about 2 hours). A 'View hardware details' link is also present.