

BIGRED – Datacenter in a box

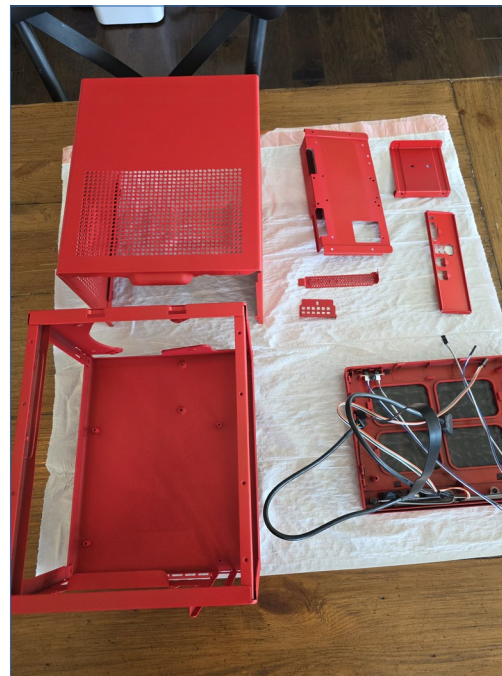
Marc Skinner
Principal RHEL Specialist Solutions Architect

Hardware

- ASRock Rack – EPYC3251D4I-2T Motherboard
 - SoC - AMD EPYC 3251 8 Core / 16 Threads @ 2.5Ghz
- 128 GB ECC RAM @ 2666 MT/s
- Intel Dual port 2.5GB NIC
- ASRock PCIe x16 quad NVMe adaptor running bifurcation 4x4x4x4
- Four 1TB Samsung NVMe disks (VM IMAGES)
- Dual 60GB SSD (OS)
- Lian-Li 750 Watt SFX Power Supply
- Silverston SG13 Case
- Noctua 140mm PWM Case Fan

Case Mods

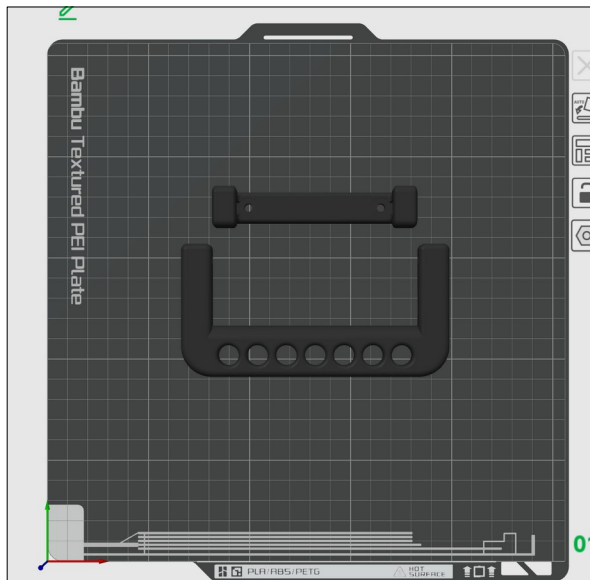
- BIGRED needs to be red!



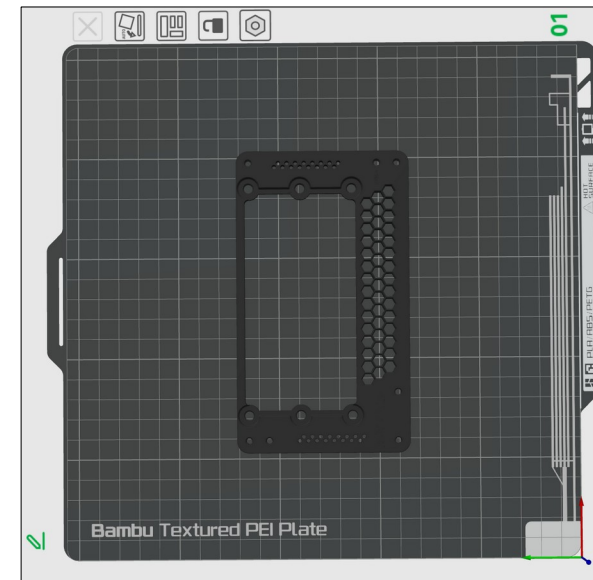
Case Mods

● 3D Printing

- Handle



- SFX Power Supply Mount



Features

● ASRock IPMI

ASRockRack

Username

Password

Remember Username

Sign me in

I forgot my password

Language

English

ASRockRack

Heat OnlineUSB Off

Dashboard

Sensor

System Information

Logs & Reports

Settings

Remote Control

Image Redirection

Power Control

Miscellaneous

Maintenance

Sign out

DashboardControl Panel

Product Information

MB Model NameEPYC3251D4I-2T

Firmware Information

BMC Firmware Version1.30.00

BIOS Firmware VersionP1.20

PSP Firmware Version0.7.0.74

Microcode Version08001230

CPLD Version

Network InformationDetails

MAC AddressD0:50:99:E6:F3:94

V4 Network ModeStatic

IPv4 Address192.168.33.177

V6 Network ModeNot Available

IPv6 AddressNot Available

Sensor Monitoring

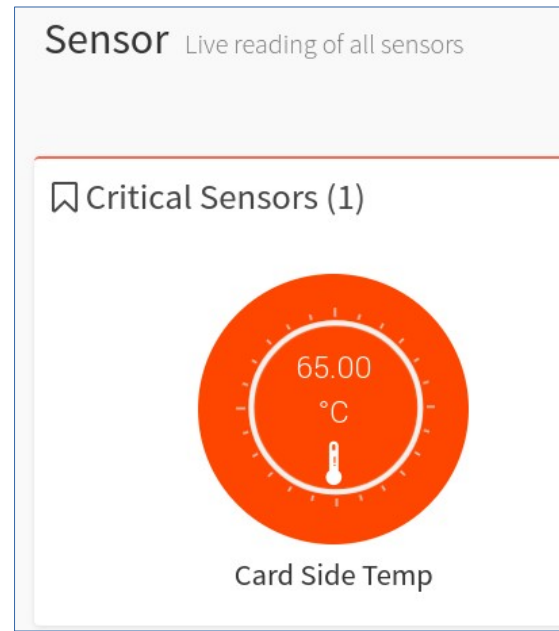
Status	Sensor Name	Reading
●	3VSB	3.30 V
●	5VSB	5.07 V
●	CPU1_Vcore	1.07 V
●	V50C	0.84 V
●	VCCM_ABCD	1.21 V
●	BAT	2.82 V
●	3V	3.30 V
●	5V	5.07 V
●	12V	11.80 V
●	PSU1_VIN	Not Available
●	PSU2_VIN	Not Available
●	PSU1_IOUT	Not Available
●	PSU2_IOUT	Not Available
●	CPU1 Temp	40.00 °C
●	MB Temp	40.00 °C
●	Card Side Temp	54.00 °C
●	Onboard LAN Temp	59.00 °C
●	DDR4_A1 Temp	45.00 °C
●	DDR4_A2 Temp	45.00 °C
●	DDR4_B1 Temp	42.00 °C
●	DDR4_B2 Temp	45.00 °C
●	PSU1 Temp	Not Available
●	PSU2 Temp	Not Available
●	FAN1	900.00 RPM
●	FAN2	Not Available
●	FAN3	1400.00 RPM

Event Logs (0)Details

No event log

Overheating Issue

- Critical Sensors
- Fan Speeds – Stuck at 400!



System Inventory			
Processor Memory Controller PCIE Device PCIE Function Storage Fan			
Fan Info			
Name	Fan Speed	Fan Speed Duty	Fan Health
FAN1	900 RPM	30	OK
FAN2	0 RPM	0	Critical
FAN3	400 RPM	30	OK

Fan Control

- lm_sensors?

```
#sensors | grep -i fan
```

```
fan1:          0 RPM (min =  0 RPM)
```

```
fan2:          0 RPM (min =  0 RPM)
```

```
fan3:          0 RPM (min =  0 RPM)
```

Fan Control

● IPMI Command Line Tool

- 0x3a = IPMI Network Function Code
- 0x01 = Command ID for Thermal Fan Control
- 0x64 = 100 in Hexadecimal for 100% fan speed

```
# ipmitool -I open raw 0x3a 0x01 0x00 0x00 0x64 0x00 0x00  
0x00 0x00 0x00
```

```
# ipmitool sensor reading FAN3
```

```
FAN3      | 1400
```

Fan Control

● Quick SystemD OneShot Service

```
#cat /etc/systemd/system/fan3-set-1400rpm.service
```

[Unit]

Description=Run Custom IPMItool Command

After=network.target

[Service]

Type=oneshot

ExecStart=/usr/bin/ipmitool -l open raw 0x3a 0x01 0x00 0x00 0x64 0x00 0x00
0x00 0x00 0x00

RemainAfterExit=yes

[Install]

WantedBy=multi-user.target

Fan Control

- More air flow removes the thermal issue
- Fan Speeds – 1,400 much better

System Inventory			
Processor Memory Controller PCIE Device PCIE Function Storage Fan			
Fan Info			
Name	Fan Speed	Fan Speed Duty	Fan Health
FAN1	900 RPM	30	OK
FAN2	0 RPM	0	Critical
FAN3	1400 RPM	30	OK



Check on NVMe Temperatures

- NVMe thermal throttle over **70C**
- nvme command line

```
# nvme smart-log /dev/nvme0 | grep "^temperature"
```

```
nvme0: temperature                : 41 °C (314 K, 105 °F)
```

Check on CPU Temperatures

- EPYC 3251 CPU thermal throttles over **105C**
- Check with sensors

```
#sensors | grep Tctl
```

```
Tctl:    +41.2°C
```

Disk Configuration

- Dual SATA 60 GB SSD
 - mdadm RAID-1 for **OS**
- Quad NVMe 1 TB drives in software RAID-5
 - mdadm RAID-5 for **VM IMAGES**
 - mounted at `/var/lib/libvirt/images`

Check SSD Drive Firmware

- Use smartctl to check SSD information

```
# smartctl -a /dev/sda
```

```
=== START OF INFORMATION SECTION ===
```

```
Model Family:   Samsung based SSDs
```

```
Device Model:   SAMSUNG SSD 830 Series
```

```
Serial Number:  S0VTNYAC100177
```

```
LU WWN Device Id: 5 002538 043584d30
```

```
Firmware Version: CXM03B1Q
```



Check NVMe Drive Firmware

- Use nvme to check NVMe information

```
# nvme list
```

```
# nvme id-ctrl /dev/ng0n1
```

NVME Identify Controller:

```
vid    : 0x144d
```

```
ssvid   : 0x144d
```

```
sn      : S4S2NA1N932005
```

```
mn      : SAMSUNG MZ1LB960HAJQ-000FB
```

```
fr      : EDA73F2Q
```



Software Stack

- Base OS – RHEL 10.latest
- Using KVM as hypervisor
- Using libvirt as hypervisor API

Networking

- Dual port 2.5GB Intel NIC
- MTU 9000
- Configured as LACP bond10 with VLANs
 - VLAN (default untagged) - KVM
 - VLAN (tagged 40) - RHLAB
 - VLAN (tagged 60) - OPENSIFT
- Commands to view network settings
 - # cat /proc/net/bonding/bond10
 - # nmcli connection

```
root@bigred:~# cat /proc/net/bonding/bond10
Ethernet Channel Bonding Driver: v6.12.0-211.49.1.el10_2.x86_64

Bonding Mode: IEEE 802.3ad Dynamic link aggregation
Transmit Hash Policy: layer3+4 (1)
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0
Peer Notification Delay (ms): 0

802.3ad info
LACP active: on
LACP rate: fast
Min links: 0
Aggregator selection policy (ad_select): stable
System priority: 65535
System MAC address: a8:a1:59:c2:5d:ec
Active Aggregator Info:
  Aggregator ID: 2
  Number of ports: 2
  Actor Key: 11
  Partner Key: 1007
  Partner Mac Address: 84:78:48:72:cf:b9

Slave Interface: enp3s0f0
MII Status: up
Speed: 2500 Mbps
Duplex: full
Link Failure Count: 1
Permanent HW addr: a8:a1:59:c2:5d:ec
Slave queue ID: 0
Aggregator ID: 2
Actor Churn State: none
Partner Churn State: none
Actor Churned Count: 0
Partner Churned Count: 0
details actor lacp pdu:
  system priority: 65535
  system mac address: a8:a1:59:c2:5d:ec
  port key: 11
  port priority: 255
  port number: 1
  port state: 63
details partner lacp pdu:
  system priority: 1
  system mac address: 84:78:48:72:cf:b9
  oper key: 1007
  port priority: 1
  port number: 5
  port state: 61

Slave Interface: enp3s0f1
MII Status: up
Speed: 2500 Mbps
Duplex: full
Link Failure Count: 1
Permanent HW addr: a8:a1:59:c2:5d:ed
Slave queue ID: 0
Aggregator ID: 2
Actor Churn State: none
Partner Churn State: none
Actor Churned Count: 0
Partner Churned Count: 0
details actor lacp pdu:
  system priority: 65535
  system mac address: a8:a1:59:c2:5d:ec
  port key: 11
  port priority: 255
  port number: 2
  port state: 63
details partner lacp pdu:
  system priority: 1
  system mac address: 84:78:48:72:cf:b9
  oper key: 1007
  port priority: 1
  port number: 6
  port state: 61
root@bigred:~#
```

Management

- VM management via light weight client virt-manager
- Cockpit for management of entire system
 - Demo after after

Containers and VMs

- Containers

- Red Hat Offline Knowledge Portal

- VMs

- Satellite
 - RHEL 7, 8, 9, 10
 - ELS
 - Image Builder
 - Image Mode
 - OpenSCAP with CIS

Red Hat Offline Knowledge Portal with Podman Quadlets

What is Podman?

- Podman is short for Pod Manager
 - Pods are one or more containers running a process utilizing one or more resources
- Podman is a fully open source daemonless container manager
- Podman is OCI (Open Container Initiative) compliant
- Podman by default uses runC (the OCI runtime) to launch containers
- Podman can leverage containerfiles (OCI) and dockerfiles for builds
- Podman supports both root and rootless containers
- Podman (modular) is a drop in replacement for Docker (monolithic)
- Podman plugs into SystemD
- Additional tooling
 - Buildah – builds OCI container images
 - Skopeo – manages images, copy, inspect, sign
 - Quadlet – run containers as SystemD services

Root vs Rootless Containers

- Root container launch with root privileges
 - Full access
 - Easy, more flexible, more risk
- Rootless container launch without root privileges
 - Limited access
 - More secure / isolated
 - Operate within a users namespace

What are Quadlets?

- Declarative way to manage containers with SystemD unit files
- Quadlet types
 - .container – defines a Podman container
 - .kube – used to define pods and containers based on Kubernetes YAML
 - .volume – creates a Podman volume, allowing persistent storage
 - .network – defines a Podman container network device
 - .build – handles the building of images
 - .image – manages container image pulls or configuration
 - .artifact – defines artifacts for Podman
- Learn Quadlets now – “podman generate systemd” is DEPRECATED!

What are Quadlets?

- Once unit files created, reload SystemD to automatically generate service and start

```
# systemctl --user daemon-reload
```

What are Quadlets?

Podman rootful locations (in order of precedence)

- Temporary and testing quadlets
 - `/run/containers/systemd/`
- System Administrator's defined quadlets
 - `/etc/containers/systemd/`
- Distribution defined quadlets:
 - `/usr/share/containers/systemd/`

What are Quadlets?

Podman rootless locations

- `$XDG_RUNTIME_DIR/containers/systemd/`
- `$XDG_CONFIG_HOME/containers/systemd/` or `~/.config/containers/systemd/`
- `/etc/containers/systemd/users/${UID}`
- `/etc/containers/systemd/users/`
- `/usr/share/containers/systemd/users/${UID}`
- `/usr/share/containers/systemd/users/`

What is the Red Hat Offline Knowledge Portal?

- Containerized Red Hat Knowledge Portal, available to Satellite customers
- Download, deploy on any OCI container runtime
- Can be used in 100% offline or hybrid, where majority is offline, but additional links send you to Red Hat Portal
- Updated monthly
- Currently, about 10GB uncompressed

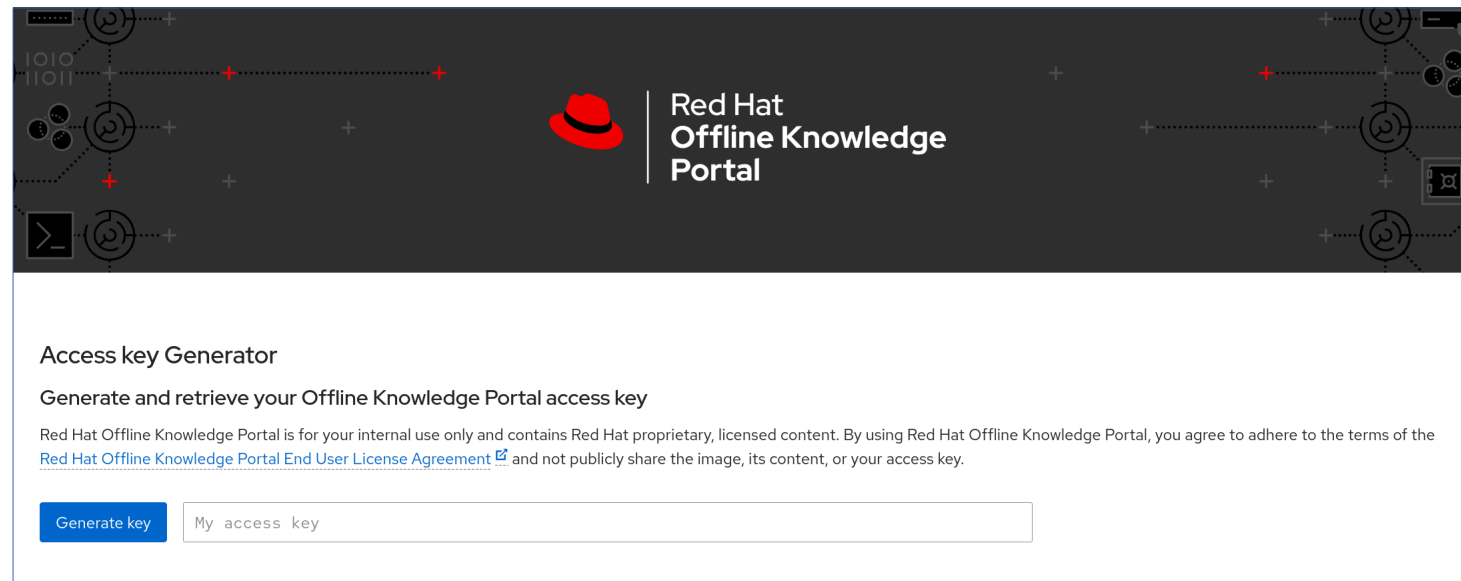
What is the Red Hat Offline Knowledge Portal?

- Requirements to run
 - Red Hat Portal account
 - RHOKP container image download
 - Internet access to registry.redhat.io
 - Access Key
 - <https://access.redhat.com/offline/access/>
 - Registry Service Account
 - <https://access.redhat.com/terms-based-registry/>
 - Resources
 - 2 vCPU
 - 2 GB RAM
 - 75 GB Storage

How to install RHOKP using Rootless Podman Quadlets

● Obtain Access Key

- <https://access.redhat.com/offline/access/>



The screenshot shows the 'Red Hat Offline Knowledge Portal' interface. At the top, there is a dark header with the Red Hat logo and the text 'Red Hat Offline Knowledge Portal'. Below the header, the main content area is white and contains the 'Access key Generator' section. This section includes a heading 'Access key Generator', a subheading 'Generate and retrieve your Offline Knowledge Portal access key', and a paragraph of text stating that the portal is for internal use only and contains Red Hat proprietary content. It also includes a link to the 'Red Hat Offline Knowledge Portal End User License Agreement'. At the bottom of the section, there is a blue 'Generate key' button and a text input field labeled 'My access key'.

Access key Generator

Generate and retrieve your Offline Knowledge Portal access key

Red Hat Offline Knowledge Portal is for your internal use only and contains Red Hat proprietary, licensed content. By using Red Hat Offline Knowledge Portal, you agree to adhere to the terms of the [Red Hat Offline Knowledge Portal End User License Agreement](#) and not publicly share the image, its content, or your access key.

[Generate key](#)

How to install RHOKP using Rootless Podman Quadlets

- Create Registry Service Account – I called my “quadlet”
 - <https://access.redhat.com/terms-based-registry/>

Registry Service Accounts

Registry service accounts are named tokens that can be used in environments where credentials will be shared, such as deployment systems.

Note: Registry service accounts can only be deleted or regenerated by an organization administrator or the person that created it.

New Service Account

Showing accounts 1 to 3

Account name	Description	Created	Created by	Token Last Generated	
aap	AAP Registry Account	Apr 13, 2026	Marc Skinner	Apr 13, 2026	⋮
podman-desktop	Service account to use from Podman Desktop	Jul 15, 2025	Marc Skinner	Jul 15, 2025	⋮
quadlet	Quadlet Registry Account	Apr 21, 2026	Marc Skinner	Apr 21, 2026	⋮

1-10 of 3 << < 1 of 1 > >>

How to install RHOKP using Rootless Podman Quadlets

- Create a user to run the container – rootless requires a user
`# useradd rhokp`
- **Must be logged in as user rhokp – not “su – rhokp” / logged in directly**
- Create directory structure for Quadlet files
`# mkdir .config/containers/systemd -p`
`# mkdir .config/systemd/user -p`
- Create two Quadlet files and one SystemD timer

How to install RHOKP using Rootless Podman Quadlets

● Create rhokp.container Quadlet file

[Unit]

Wants=podman-auto-update.timer

[Container]

ContainerName=rhokp

Environment=ACCESS_KEY=mskinner@redhat.com_RANDOM-NUMBERS

Environment=ONLINE_VIEW=true

Image=rhokp.image

PublishPort=8080:8080

AutoUpdate=local

[Service]

Restart=always

[Install]

WantedBy=default.target

How to install RHOKP using Rootless Podman Quadlets

- Use Creds from Service Account [username:password]

- Create rhokp.image Quadlet file

[Image]

Policy=newer

Image=registry.redhat.io/offline-knowledge-portal/rhokp-rhel9:latest

Creds= username:password

[Service]

RemainAfterExit=no

How to install RHOKP using Rootless Podman Quadlets

● Create rhokp-image.timer SystemD unit file

[Unit]

Description=timer for rhokp-image.service

[Timer]

OnCalendar=daily

Persistent=true

[Install]

WantedBy=timers.target

How to install RHOKP using Rootless Podman Quadlets

- Force SystemD to generate a service unit file from Quadlets

```
# systemctl --user daemon-reload
```

- Start and Enable SystemD Timer

```
# systemctl --user enable --now rhokp-image.timer
```

- **DISK SPACE ALERT – USES APPROX 14GB - /home/rhokp**

- Start container for first time – forcing image download

```
# systemctl --user start rhokp.service
```

How to install RHOKP using Rootless Podman Quadlets

- List “rhokp” unit files

```
[rhokp@z1 .config]$ systemctl --user list-unit-files | grep rhokp
rhokp-image.service          generated -
rhokp.service                generated -
rhokp-image.timer            enabled   disabled
```

How to install RHOKP using Rootless Podman Quadlets

- Tree view of Quadlets and Unit files in user file structure

```
[rhokp@z1 .config]$ tree
.
├── cni
│   └── net.d
│       └── cni.lock
├── containers
│   ├── podman
│   │   └── machine
│   └── systemd
│       ├── rhokp.container
│       └── rhokp.image
└── systemd
    └── user
        ├── rhokp-image.timer
        ├── timers.target.wants
        │   └── rhokp-image.timer -> /home/rhokp/.config/systemd/user/rhokp-image.timer
9 directories, 5 files
```

How to install RHOKP using Rootless Podman Quadlets

● Podman process and image view

```
[rhokp@z1 ~]$ podman ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
092a6f5b33b5	registry.redhat.io/offline-knowledge-portal/rhokp-rhel9:latest	/bin/sh -c mel	18 minutes ago	Up 18 minutes	0.0.0.0:8080->8080/tcp, 8443/tcp	rhokp

```
[rhokp@z1 ~]$ podman images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
registry.redhat.io/offline-knowledge-portal/rhokp-rhel9	latest	f436a4591bbf	12 days ago	10.3 GB

How to install RHOKP using Rootless Podman Quadlets

● Podman quadlet listing

```
[rhokp@z1 ~]$ podman quadlet list
```

NAME	UNIT NAME	PATH ON DISK	STATUS	APPLICATION
rhokp.container	rhokp.service	/home/rhokp/.config/containers/systemd/rhokp.container	active/running	
rhokp.image	rhokp-image.service	/home/rhokp/.config/containers/systemd/rhokp.image	inactive/dead	

How to install RHOKP using Rootless Podman Quadlets

- View user SystemD Timer information

```
[rhokp@z1 ~]$ systemctl --user list-timers
```

NEXT	LEFT	LAST	PASSED	UNIT	ACTIVATES
Thu 2026-04-23 08:40:39 CDT	7s left	-	-	systemd-tmpfiles-clean.timer	systemd-tmpfiles-clean.service
Thu 2026-04-23 08:46:49 CDT	6min left	Wed 2026-04-22 09:03:21 CDT	23h ago	podman-auto-update.timer	podman-auto-update.service
Fri 2026-04-24 00:00:00 CDT	15h left	Thu 2026-04-23 08:35:40 CDT	4min 52s ago	rhokp-image.timer	rhokp-image.service
-	-	Thu 2026-04-23 08:38:15 CDT	2min 16s ago	grub-boot-success.timer	grub-boot-success.service

4 timers listed.
Pass --all to see loaded but inactive timers, too.

How to install RHOKP using Rootless Podman Quadlets

- What happens when the user “rhokp” logs out?
- **All services will shutdown ... unless, root enables Linger!**

```
# loginctl enable-linger rhokp  
# loginctl show-user rhokp
```

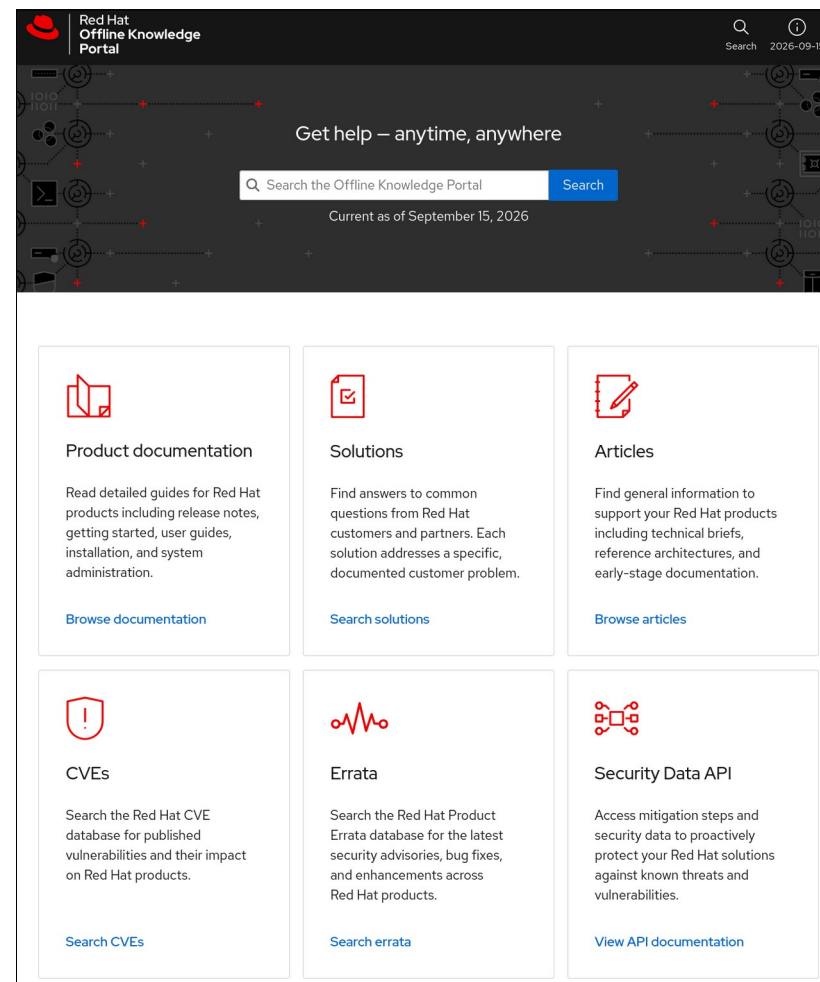
- This will also, ensure the user container restarts on reboot.



```
[root@z2 ~]# loginctl show-user rhokp  
UID=1002  
GID=1002  
Name=rhokp  
Timestamp=Thu 2026-04-23 12:05:25 CDT  
TimestampMonotonic=81939203949  
RuntimePath=/run/user/1002  
Service=user@1002.service  
Slice=user-1002.slice  
Display=7  
State=active  
Sessions=7  
IdleHint=no  
IdleSinceHint=1776963941960449  
IdleSinceHintMonotonic=81956146161  
Linger=yes
```

How to install RHOKP using Rootless Podman Quadlets

- Connect to URL:
`http://rhokp.i.skinnerlabs.com:8080`
- Top right corner – updated as of 9/15/2026



Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



linkedin.com/company/red-hat



youtube.com/user/RedHatVideos



facebook.com/redhatinc



twitter.com/RedHat