

ChatGPT   Free offer 

This is a copy of a conversation between ChatGPT & Anonymous.

Report conversation

Tell me what you know about the company Red Hat

Here's a comprehensive overview of **Red Hat, Inc.**, one of the most influential companies in open-source and enterprise software:



What Red Hat Is

Red Hat, Inc. is an American software company and a *leader in open-source technologies*, especially for enterprise IT infrastructure and cloud computing. It develops and supports open-source software solutions that businesses use to build, deploy, and manage applications across hybrid cloud environments.

- **Founded:** 1993 by Bob Young and Marc Ewing.
- **Headquarters:** Raleigh, North Carolina, USA.
- **Employees:** ~19,000.
- **Parent Company:** **IBM** acquired Red Hat in 2019 for about **US \$34 billion**, one of the largest software acquisitions ever.



Open Source and Business Model

Red Hat's business model is based on *open source software* — software whose source code is publicly accessible and can be freely modified and shared. Rather than selling proprietary software licenses, Red Hat **charges for subscriptions** that include technical support, training, certification, and integration services.

The company **actively contributes to many open-source communities** and projects around the world (e.g., Kubernetes, Fedora, Linux kernel development), and often takes community-built projects and “enterprise-hardens” them for reliability, security, and long-term support.



Key Products and Technologies

Red Hat offers a broad range of software and services for modern IT environments:

1. Red Hat Enterprise Linux (RHEL)

- The company's flagship enterprise operating system — widely used by businesses and governments globally.

2. Hybrid Cloud & Platform Tools

- **Red Hat OpenShift:** A Kubernetes-based container platform for building cloud-native applications.
- **Ansible Automation Platform:** Tools for enterprise automation of IT tasks.

3. Middleware, Virtualization, and Management

- Middleware and integration tools (historically including JBoss).
- Virtualization solutions (Red Hat Virtualization).

4. Edge computing, AI, and developer services

- Capabilities for edge deployments, AI-driven workloads, and comprehensive enterprise support.



Market Presence

Red Hat products and subscriptions are used **worldwide**, and the company is especially popular with large organizations:

- Trusted by **over 90% of Fortune 500 companies**.
- Offices in **more than 40 countries**.

Because of its open-source foundations and enterprise support model, Red Hat plays a central role in many companies' **IT modernization, cloud transformation, and automation strategies**.



Culture and Community

Red Hat is known for:

- A strong **commitment to open-source community participation** — contributing code back to upstream projects.
- Supporting initiatives like **Fedora** (a community distribution that serves as an upstream innovation source for RHEL).
- Offering training and certification programs widely recognized in the tech industry.

Relationship with IBM

Since the IBM acquisition in 2019, Red Hat has operated as an independent subsidiary with a focus on preserving its open-source culture while integrating with IBM's broader enterprise strategy.

If you want, I can also explain **specific Red Hat products in more detail** (like RHEL, OpenShift, or Ansible) or how the company *fits into current trends like AI and cloud computing*.

   Sources

How does the company's offering of technical support for subscription work?

Red Hat's **subscription-based technical support** is central to how the company makes money from open-source software. Here's how it works in practice, step by step.

1. What You're Actually Buying

When a company buys a **Red Hat subscription**, they are **not buying the software license** (the software is open source). They are buying:

- **Access to tested, enterprise-ready builds**
- **Security updates and patches**
- **Technical support from Red Hat engineers**
- **Long-term lifecycle guarantees**
- **Tools, documentation, and management portals**
- **Legal assurances** (e.g., IP indemnification)

Think of it as *insurance + expertise + stability* for running critical systems.

2. Subscription Structure

Subscriptions are usually priced **per system, per core, or per cluster**, depending on the product.

For example:

- **RHEL** → per server or virtual machine
- **OpenShift** → per CPU core
- **Ansible Automation Platform** → per managed node

Subscriptions are **time-based** (typically 1–3 years) and must be renewed to keep receiving updates and support.

3. Support Levels (SLAs)

Red Hat offers different **support tiers**, each with defined service-level agreements (SLAs):

Standard Support

- Business-hours support
- Web and phone access
- Slower response times
- Suitable for non-mission-critical workloads

Premium Support

- **24/7 support**
- Faster response times
- Phone + web
- Designed for production, mission-critical systems

Each tier defines:

- **Response time** (how fast Red Hat acknowledges an issue)
- **Severity levels** (impact on business)
- **Availability** (business hours vs. 24/7)

4. How a Support Case Works

1. Customer opens a support case

- Via Red Hat Customer Portal (web)
- Or by phone (for high-severity issues)

2. Issue is classified by severity

- Severity 1: System down / production outage
- Severity 2–4: Decreasing levels of impact

3. Red Hat support engineers respond

- They help with troubleshooting, configuration, performance, bugs, and failures
- They may request logs, diagnostics, or system data

4. Engineering escalation if needed

- If it's a product bug, the case is escalated to Red Hat's engineering teams
- Fixes may come as patches, workarounds, or future updates

5. Updates, Patches, and Lifecycle

Subscriptions include:

- **Security updates** (including critical CVEs)
- **Bug fixes**
- **Minor and major version updates**

Red Hat is known for **long support lifecycles**:

- RHEL versions are supported for **10+ years**
- This is crucial for enterprises that can't upgrade frequently

Without an active subscription:

- You can still legally run the software
- But you **lose access to updates, patches, and support**

6. Certified Ecosystem & Stability

Subscriptions also give access to:

- **Certified hardware** (servers, storage, cloud platforms)
- **Certified software** (databases, applications, drivers)

This reduces risk for enterprises by ensuring:

- Vendors will support the full stack
- Problems won't be blamed on "unsupported configurations"

7. Legal and Business Value

An often-overlooked part:

- Red Hat provides **IP indemnification**
- If there's a legal issue around open-source code, Red Hat stands behind the customer

This is extremely important for large enterprises and governments.

8. Why Customers Pay for This

Organizations pay for Red Hat subscriptions because they get:

- Predictable costs
- Enterprise-grade reliability
- Expert help when things break at 3 a.m.
- Long-term stability without vendor lock-in

In short:

Red Hat sells confidence and continuity, not code.

If you want, I can:

- Compare Red Hat's model to **Ubuntu/Canonical**
- Explain **why companies don't just use free Linux**
- Walk through a **real-world support case example**
- Explain how this model works specifically for **OpenShift or Ansible**

Just tell me.

Let's walk through a real-world support case example

Great—let's walk through a **concrete, realistic Red Hat support case** from start to finish.

I'll use **RHEL** as the example, but the flow is very similar for OpenShift and Ansible.

Scenario Overview

Company: Mid-size financial services firm

Environment:

- Red Hat Enterprise Linux 8
- Running a customer-facing payment application
- 24/7 production system
- Premium Support subscription

Problem:

After a routine kernel update and reboot, one production server **fails to come back online**.

The application is down.

This is a **Severity 1 (critical)** issue.

Step 1: Issue Occurs (Customer Side)

- Monitoring alerts trigger
- Operations team attempts:
 - Reboot → fails
 - Roll back application → impossible (OS not booting)
- Business impact:
 - Payments failing
 - Revenue loss per minute

Decision: **Call Red Hat Support immediately**

Step 2: Opening the Support Case

Because it's Severity 1:

- Customer calls Red Hat's **24/7 support phone line**
- Provides:
 - Subscription number
 - System details (RHEL version, kernel version)
 - Short description: "System fails to boot after kernel update"

Red Hat:

- Opens a **Severity 1 case**
- Assigns a **Senior Support Engineer**
- SLA: Response in minutes, not hours

Step 3: Initial Triage (Red Hat Support)

The support engineer:

- Confirms severity and impact
- Asks for:
 - Boot error messages
 - Console output (via iLO / IPMI / cloud console)
 - Whether this occurred on multiple systems

They identify:

- Kernel panic related to a **storage driver**

Step 4: Deep Troubleshooting

Red Hat support walks the customer through:

1. **Booting into an older kernel**
 - Using GRUB menu
 - Confirms system boots successfully
2. **Collecting diagnostics**
 - `sosreport` from the working kernel
 - Kernel version comparison
3. **Root cause identification**
 - A known regression affecting a specific RAID controller
 - Triggered only on certain firmware versions

At this point:

- Red Hat confirms: **this is a product bug**

Step 5: Escalation to Engineering

Because it's a confirmed bug:

- Support escalates the case to **Red Hat Engineering**
- Engineering:
 - Confirms issue matches an internal bug report
 - Provides a **temporary workaround**

Workaround:

- Blacklist the problematic driver module
- Use the previous kernel until a fix is available

Support engineer:

- Stays on the call while workaround is applied
- Confirms system boots and application is restored

 **Downtime reduced from hours to ~45 minutes**

Step 6: Permanent Fix Delivered

A few days later:

- Red Hat releases a **patched kernel update**
- Support notifies the customer via the case
- Provides:
 - Patch advisory
 - Safe upgrade instructions
 - Recommendation for rolling deployment

Customer:

- Applies fix first in staging
- Then production
- Closes the case

Step 7: Post-Incident Value

What the subscription delivered:

- ✓ Immediate access to experts
- ✓ Rapid root cause identification
- ✓ Direct engineering escalation
- ✓ Temporary workaround
- ✓ Permanent fix
- ✓ Confidence the issue won't recur

Without a subscription:

- The company would have relied on forums
- No guaranteed response time
- No engineering escalation
- Higher downtime and business risk

How This Differs for OpenShift (Quick Note)

For **OpenShift**, a similar case might involve:

- Cluster not upgrading
- Nodes stuck in `NotReady`
- Operators failing health checks

Red Hat support would:

- Analyze cluster state
- Review operator logs
- Provide rollback or upgrade path
- Coordinate fixes across Kubernetes, OS, and platform layers

Key Takeaway

A Red Hat support case is not just:

“Here’s a workaround, good luck.”

It’s:

A guided, accountable process backed by engineering and long-term fixes.

If you’d like, next we can:

- Walk through an **OpenShift outage example**
- Show what happens in a **non-critical (Severity 3)** case
- Compare this with **community-only support**

- Explain what Red Hat support *will not* help with (important nuance)

Just say the word.

In Step 4: Depp Troubleshooting part 2. Collecting diagnostics it talked about retrieving an sosreport from the working kernel? What is an sosreport?

An `sosreport` is a **standardized diagnostic snapshot** of a Linux system that Red Hat support uses to quickly understand what's going on with a machine—without having to manually ask you for dozens of files.

Think of it as a **“black box flight recorder”** for a RHEL system.

What `sosreport` Is

`sosreport` is a tool (part of the `sos` package) that:

- Collects **system configuration**
- Gathers **logs and runtime information**
- Captures **hardware, storage, and networking details**
- Packages everything into **one compressed archive**

It does **not** modify the system. It only reads data.

Why Red Hat Uses It

Instead of asking:

- “Send `/var/log/messages`”
- “What kernel modules are loaded?”
- “What’s your storage configuration?”
- “What kernel parameters are set?”

Red Hat support can say:

"Please upload an sosreport."

This gives them a **complete, consistent view** of the system in minutes.

What an sosreport Contains (High Level)

An sosreport typically includes:

System Basics

- OS version and release
- Kernel version(s)
- Boot parameters
- Installed RPMs

Logs

- /var/log/messages
- journalctl output
- Boot logs
- dmesg (kernel ring buffer)

Hardware & Drivers

- PCI devices (`lspci`)
- Loaded kernel modules
- Storage controllers
- CPU and memory info

Storage & Filesystems

- Block devices
- RAID/LVM configuration
- Filesystem mount info
- /etc/fstab

Networking

- Interface configuration
- Routing tables

- Firewall rules

Services & Processes

- systemd unit states
- Running processes
- Resource usage snapshots

Red Hat–Specific Metadata

- Subscription status
- Installed Red Hat packages
- SELinux status and logs

Why “From the Working Kernel” Matters

In the earlier example:

- The **new kernel** fails to boot
- The **previous kernel** boots successfully

Red Hat support asks for an sosreport from the *working kernel* because:

- It lets them inspect:
 - Hardware drivers
 - Kernel modules
 - Storage configuration
- They can compare:
 - Working vs. failing kernel versions
- The data is still relevant because:
 - Hardware and configuration haven’t changed
 - Only the kernel version differs

This is often enough to identify **regressions** or incompatible drivers.

How You Generate One (Typical Flow)

```
bash
```

 Copy code

```
sudo sosreport
```

During execution:

- You'll be asked to:
 - Enter a **case number** (so Red Hat can associate it)
 - Optionally enable data redaction

Output:

- A file like:

 Copy code




```
sosreport-myhost-2026-01-29-abc123.tar.xz
```

You then:

- Upload it via the **Red Hat Customer Portal**
- Or provide it via secure upload if instructed

Security & Privacy Considerations

Important points enterprises care about:

-  **Read-only** (no system changes)
-  Can **mask sensitive data** (passwords, keys, tokens)
-  Red Hat support handles it under contractual confidentiality

That said:

- It may still include:
 - IP addresses
 - Hostnames
 - Internal topology details
- Many orgs review or redact before uploading (Red Hat supports this)

Why It's So Central to Red Hat Support

For Red Hat engineers, an sosreport:

- Reduces back-and-forth questions
- Speeds up root cause analysis
- Makes escalation to engineering faster
- Creates a repeatable support workflow

In practice:

No serious Red Hat support case proceeds without an `sosreport`.

One Important Limitation

`sosreport` is a **snapshot**, not a live trace:

- It won't capture transient issues unless they're logged
- For performance problems, support may also ask for:
 - `perf`
 - `strace`
 - `tcpdump`
 - or OpenShift-specific diagnostics

If you want, I can next:

- Walk through an **actual `sosreport` directory structure**
- Explain how **plugins** work inside `sos`
- Show what data Red Hat support typically checks first
- Compare `sosreport` with tools like `supportconfig` (SUSE) or `apport` (Ubuntu)

Just tell me where to go next.

Lets walk through an actual `sosreport` directory structure

Great idea. Walking the structure makes `sosreport` click very quickly.

Below is a **realistic, simplified `sosreport` directory tree**, followed by **what Red Hat engineers actually look at first** and **why each area exists**.

1. Top-Level Structure

When you extract an sosreport:

```
bash
```

[Copy code](#)

```
tar -xf sosreport-myhost-2026-01-29-abc123.tar.xz
cd sosreport-myhost-2026-01-29-abc123/
```

You'll see something like:

```
text
```

[Copy code](#)

```
.
├─ sosreport.log
├─ sos_commands/
├─ etc/
├─ proc/
├─ sys/
├─ var/
├─ usr/
├─ root/
├─ boot/
├─ lib/
├─ run/
└─ version.txt
```

2. version.txt (Start Here)

```
text
```

[Copy code](#)

```
Red Hat Enterprise Linux release 8.9 (Ootpa)
sos version 4.6.0
```

Why it matters:

- Confirms OS version

- Confirms `sos` tool version
- Helps support understand *what collectors ran*

📌 This is usually the first file a support engineer opens.

3. `sosreport.log`

text

 Copy code

```
sosreport starting...
Running plugin kernel
Running plugin storage
Running plugin networking
...
```

Purpose:

- Shows which plugins ran
- Shows failures or skipped collectors
- Explains *missing data*

If something is missing, this file explains why.

4. `sos_commands/` (Command Output Goldmine)

This is the most-used directory.

text

 Copy code

```
sos_commands/
├─ kernel/
│   ├─ uname_-a
│   ├─ lsmod
│   ├─ sysctl_-a
│   └─ dmesg
├─ hardware/
│   ├─ lspci_-vv
│   ├─ lsusb
│   └─ free_-m
```

```
├─ storage/
│   ├── lsblk
│   ├── blkid
│   ├── pvs
│   ├── vgs
│   └── lvs
├─ networking/
│   ├── ip_addr
│   ├── ip_route
│   └── ss_-lntup
└─ process/
    ├── ps_auxww
    └── top_-b_-n_1
```

Why this is critical

Each file is the **exact output of a command** run at collection time.

Example:

```
bash
```

 Copy code

```
cat sos_commands/kernel/uname_-a
```

Output is literally what `uname -a` returned on that system.

 Support engineers **trust this more than screenshots or copy-paste.**

5. etc/ (Configuration Snapshot)

```
text
```

 Copy code

```
etc/
├─ fstab
├─ hosts
├─ passwd
├─ group
├─ sysctl.conf
└─ default/
```

```
├─ systemd/  
|   └─ system.conf  
└─ NetworkManager/
```

This mirrors `/etc` (with sensitive data masked).

Used to:

- Verify filesystem mounts
- Check `sysctl` tuning
- Review service configuration
- Spot misconfigurations

6. `var/log/` (Historical Evidence)


text

 Copy code

```
var/log/  
├─ messages  
├─ secure  
├─ dmesg  
├─ boot.log  
└─ journal/
```

Why it matters:

- Shows what happened *before* failure
- Kernel panics often appear here
- Authentication or SELinux issues surface here

 If the system crashed earlier, this is where clues live.

7. `boot/` (Kernel & Bootloader)

text

 Copy code

```
boot/  
├─ grub2/
```

```
|   └─ grub.cfg
└─ config-4.18.0-513.el8.x86_64
└─ vmlinuz-4.18.0-513.el8.x86_64
```

Used to:

- Confirm installed kernels
- Inspect boot parameters
- Check default kernel selection

This is **key** in kernel regression cases.

8. proc/ and sys/ (Runtime State)

text

 Copy code

```
proc/
└─ cmdline
└─ cpuinfo
└─ meminfo
```

```
sys/
└─ block/
```

These capture kernel-exposed runtime information.

Useful for:

- CPU topology
- Memory layout
- Block device details

9. Plugin-Specific Directories

Some products add extra folders, for example:

SELinux

text

```
sos_commands/selinux/  
├─ sestatus  
└─ sealert_-l_*
```

Containers / OpenShift

text

 Copy code

```
sos_commands/containers/  
├─ podman_ps  
├─ crictl_ps  
└─ oc_get_nodes
```

Support engineers **jump straight here** for product-specific cases.

10. What Support Looks At First (Real Order)

In a kernel boot failure case, Red Hat engineers typically:

1. `version.txt`
2. `sosreport.log`
3. `sos_commands/kernel/`
4. `boot/grub2/grub.cfg`
5. `sos_commands/storage/`
6. `var/log/messages` or `journal`

They are hunting for:

- Driver mismatches
- Unsupported hardware
- Regression signatures
- Known bug patterns

11. Why This Structure Matters

This layout:


- Is consistent across customers
- Is consistent across cases
- Allows **fast pattern recognition**

That's why Red Hat can escalate cases efficiently:

Engineering teams **already know where to look**.

Final Mental Model

Think of an sosreport as:

 **A frozen, forensically useful filesystem + command output snapshot**, organized for support engineers—not humans.

If you want next, I can:

- Walk through **a real kernel regression example using these files**
- Show how **plugins decide what to collect**
- Explain **how to redact or customize sosreports**
- Compare this with **OpenShift must-gather**

Just say where you want to dig deeper.

Explain how to redact or customize sosreports

Redacting and customizing `sosreport` is a **very common enterprise requirement**, and Red Hat designed the tool with that in mind. I'll break this into **how redaction works**, **how to customize collection**, and **best-practice workflows** support teams actually use.

1. Built-in Redaction (Automatic & Safe)

Default Behavior

By default, `sosreport` already **masks obvious secrets**, including:

- Password fields in config files
- Authentication tokens
- Some private keys

You don't need to do anything extra for basic safety.

Enabling Extra Redaction

When running:

```
bash
```

 Copy code

```
sudo sosreport
```

You'll be prompted:


```
text
```

 Copy code

```
Do you want to enable automatic redaction of sensitive data? (y/N)
```

Answer `y` to:

- Obfuscate usernames
- Mask IP addresses
- Mask hostnames
- Scrub MAC addresses

 This is commonly used in regulated environments.

How Redaction Actually Works

Redaction is handled by **regex-based filters** inside sos plugins.

Example (conceptual):

```
text
```

 Copy code

```
192.168.10.45 → XXX.XXX.XX.XX  
db-prod-01 → hostname-REDACTED
```

Important nuance:

- Redaction preserves **structure**, not values
- This allows troubleshooting without leaking details

2. Manual Redaction (Post-Collection)

Many organizations do **manual review before upload**.

Typical Workflow

```
bash
```

 Copy code

```
tar -xf sosreport-*.tar.xz  
cd sosreport-*/
```

Then:


- Open sensitive files:
 - etc/fstab
 - etc/hosts
 - sos_commands/networking/ip_addr
- Replace values manually
- Repack the archive

```
bash
```

 Copy code

```
tar -cJf sosreport-redacted.tar.xz sosreport-*/
```

 Red Hat explicitly allows this.

 **Warning:** Over-redaction can slow support if critical context is removed.

3. Selective Plugin Control (Customize What's Collected)

`sosreport` is **plugin-based**. You can include or exclude entire data domains.

List Available Plugins

```
bash
```

 Copy code

```
sosreport -l
```

Example output:

```
text
```

 Copy code

```
kernel
storage
networking
selinux
security
containers
```

Disable Specific Plugins

Example: exclude networking data

```
bash
```

 Copy code

```
sudo sosreport -k networking.skip=true
```

Or exclude SELinux info:

```
bash
```

 Copy code

```
sudo sosreport -k selinux.skip=true
```

 Useful if:

- Security policies forbid sharing certain configs
- The issue is unrelated (e.g., pure storage problem)

Enable Only Certain Plugins

Example: kernel + storage only

```
bash
```

[Copy code](#)

```
sudo sosreport -o kernel,storage
```

This:

- Reduces collection size
- Limits exposure
- Speeds up upload

4. File-Level Exclusions

You can exclude specific paths:

```
bash
```

[Copy code](#)

```
sudo sosreport --exclude=/etc/ssh
```

Or multiple:

```
bash
```

[Copy code](#)

```
sudo sosreport \  
  --exclude=/etc/ssh \  
  --exclude=/etc/pki
```

 Common for:

- SSH host keys
- Internal PKI infrastructure

5. Case-Aware Collection (Best Practice)

Always associate the sosreport with a case:

```
bash
```

[Copy code](#)

```
sudo sosreport --case-id=01234567
```

Benefits:

- Red Hat support sees it immediately
- Engineers know the context
- Faster triage

6. Performance & Safety Considerations

- Collection is **read-only**
- Usually takes **2–10 minutes**
- Safe on production systems
- Disk usage: typically 50–300 MB compressed

For very busy systems:

```
bash
```

 Copy code

```
sudo sosreport --batch
```

(skips prompts)

7. What *Not* to Remove (Important)

Avoid redacting:

- Kernel version info
- Hardware identifiers (model, vendor)
- Driver/module names
- Storage topology

These are often **required to identify known bugs**.

8. Red Hat Support Reality Check

If something essential is missing:

- Support will ask for a **follow-up sosreport**
- Or a **targeted command output**

They won't reject a redacted report—but they *will* need enough signal to help.

9. Comparison: sosreport vs OpenShift must-gather

Tool	Purpose	Redaction
sosreport	OS-level diagnostics	Built-in + manual
must-gather	OpenShift cluster state	Namespace filtering

They're often used **together** in OpenShift cases.

Final Takeaway

The goal is:

Maximize diagnostic value while minimizing sensitive exposure.

sosreport gives you:

- Automated redaction
- Plugin-level control
- File-level exclusion
- Manual review flexibility

Which is why it's trusted in finance, healthcare, and government environments.

If you want next, I can:

- Show **example redaction before/after**
- Explain **writing custom sos plugins**
- Walk through **OpenShift must-gather redaction**
- Share **enterprise sosreport policies/templates**

Just tell me where to go next.

What is the OpenShift must-gather?

OpenShift must-gather is the **cluster-level diagnostic tool** used by Red Hat support for **OpenShift** in the same way `sosreport` is used for **RHEL**—but scoped to **Kubernetes and the OpenShift platform**, not the underlying OS.

If `sosreport` is a *server snapshot*, **must-gather is a full OpenShift cluster snapshot**.

1. What must-gather Is (In One Sentence)

`must-gather` **collects the complete state of an OpenShift cluster—API objects, logs, and configuration—so Red Hat can diagnose cluster and platform issues.**

It is:

- Read-only
- Non-disruptive
- The **first thing** Red Hat support asks for in OpenShift cases

2. Why It Exists

OpenShift problems often involve:

- Kubernetes control plane
- Operators
- CRDs
- Etcd
- Networking overlays
- Nodes + containers + OS

You **cannot debug this reliably with ad-hoc commands**.

must-gather :

- Standardizes data collection
- Ensures nothing critical is missed
- Matches Red Hat's internal troubleshooting playbooks

3. What must-gather Collects

At a high level, it gathers:

Kubernetes / OpenShift Objects

- All namespaces (or selected ones)
- Pods, deployments, daemonsets
- Nodes and node conditions
- Events (very important)

OpenShift-Specific Components

- Cluster Operators (COs)
- Operator logs
- OpenShift APIs
- Cluster configuration resources

Logs

- Control plane logs
- Operator logs
- Node-related logs (via pods)

Platform Metadata

- OpenShift version
- Upgrade history
- Feature gates
- Cluster networking config

 **This is logical state**, not filesystem state (unlike sosreport).

4. How must-gather Is Run

From a machine with `oc` access:

```
bash
```

[Copy code](#)

```
oc adm must-gather
```

This:

- Creates a temporary **must-gather pod**
- Runs collectors inside the cluster
- Writes output locally

Result:

```
text
```

[Copy code](#)

```
must-gather/  
└─ local/  
    ├── cluster-scoped-resources/  
    ├── namespaces/  
    ├── operators/  
    ├── nodes/  
    └─ version
```

You then:

- Compress the directory
- Upload it to the Red Hat case

5. Directory Structure (Conceptual)

```
text
```

[Copy code](#)

```
must-gather/  
└─ version  
└─ cluster-scoped-resources/  
    ├── config.openshift.io/  
    ├── operator.openshift.io/  
    └─ machineconfiguration.openshift.io/
```

```
├─ namespaces/
│   ├── openshift-kube-apiserver/
│   ├── openshift-etcd/
│   └── default/
├─ operators/
│   ├── cluster-version-operator/
│   └── network-operator/
└─ nodes/
    ├── node-1/
    └── node-2/
```

This structure is **intentionally predictable**, just like `sosreport` .

6. Relationship to sosreport (Very Important)

Tool	Scope	Layer
sosreport	Single OS	Linux / RHEL
must-gather	Entire cluster	Kubernetes / OpenShift
Both together	Full picture	Platform + OS

In real support cases:

Red Hat often asks for both

must-gather → cluster logic

sosreport → node/OS issues

7. Custom must-gather Images (Advanced but Common)

OpenShift supports **specialized must-gather collectors**.

Examples:

- Storage (ODF, Ceph)
- Networking (OVN, SDN)
- Service Mesh

- Logging
- Virtualization

Example:

```
bash
```

 Copy code

```
oc adm must-gather \
  --image=registry.redhat.io/openshift4/ose-must-gather
```

or vendor-specific collectors.


 This is how Red Hat gets **deep subsystem data** without manual steps.

8. Redaction & Security Model

- No secrets are intentionally collected
- Kubernetes secrets are **metadata only**, not values
- Output is plain text and YAML
- Customers can review and redact before upload

However:

- Internal IPs
- Hostnames
- Namespace names
- Application names

 These are usually required for troubleshooting.

9. When Red Hat Will Ask for must-gather

Almost always for:

- Cluster upgrades failing
- Operators degraded
- Nodes stuck NotReady
- Control plane instability
- Networking or storage issues

- Performance or scaling problems

If the case mentions **OpenShift**:

Assume must-gather is coming.

10. Mental Model (Best Way to Remember)

- **sosreport** = *"What does this Linux system look like?"*
- **must-gather** = *"What does this OpenShift cluster believe its state is?"*

Both are snapshots.

Both are structured.

Both are required for serious support.

Final Takeaway

OpenShift must-gather is the authoritative diagnostic artifact for OpenShift support.

If you don't provide it:

- Triage slows dramatically
- Engineers can't escalate to product teams
- Root cause analysis stalls

If you want next, I can:

- Walk through a **real OpenShift outage using must-gather**
- Compare **default vs custom must-gather**
- Explain **how operators log and surface failures**
- Show **how must-gather changes during upgrades**

Just tell me where you want to go.

Are there any ChatGPT apps available to help with Red Hat

Support?

Yes — there *are* AI-powered tools and integrations that can help with Red Hat support work, ranging from official Red Hat assistants to general ChatGPT-based helpers you might use as part of troubleshooting workflows:

◆ 1. Ask Red Hat — Official AI Assistant (Integrated with Red Hat Support)

Red Hat has launched an **AI-powered assistant** called “**Ask Red Hat**” directly in the *Red Hat Customer Portal*. It’s designed specifically for Red Hat customers and subscribers to:

- Answer **natural-language questions** about Red Hat products.
- Surface relevant **knowledgebase articles**, documentation, and advisories.
- Help you find the right tools or support resources faster.

It’s built on an OpenAI/IBM-class model and is tailored to Red Hat content, not general web search, so the answers are aligned with official Red Hat guidance.

Where to find it:

You access it when logged into the **Red Hat Customer Portal** — it’s not a standalone app but an integrated support tool.

◆ 2. Red Hat Developer Lightspeed / AI Assistance Tools

Red Hat is also rolling out **AI-powered productivity assistants** under the *Lightspeed* umbrella. These aren’t exactly “Red Hat support chat apps” in the traditional sense, but they provide *context-aware AI help* for developers working with Red Hat tools, documentation, and workflows (including troubleshooting advice and code/documentation assistance).

While these are more developer-oriented than purely support-oriented, they can be extremely helpful when diagnosing issues in app stacks running on Red Hat platforms.

◆ 3. Home-Brew or Community ChatGPT Bots

There are third-party ChatGPT-based bots built around RHEL or Linux support concepts (e.g., “Red Hat Enterprise Linux Assistant” bots found on some bot directories or platforms), but these are **not official Red Hat products** — they use general GPT tech to answer Linux questions (and might help with basic troubleshooting).

⚠ *Caution:* These unofficial tools aren’t tied to Red Hat’s support knowledgebase and do not replace official support or portal access. They’re more like automated helpers with general Linux knowledge.

◆ 4. General ChatGPT Apps (via ChatGPT Apps & Extensions)

OpenAI’s **ChatGPT apps/connector ecosystem** lets you bring external tools or data into conversations (for example, connecting internal docs, logs, or ticket systems). While there isn’t yet an *official Red Hat support app* in that ecosystem, you *can*:

- Build or integrate custom apps that inject your internal Red Hat knowledge, logs, or documentation into ChatGPT.
- Use interactive apps in ChatGPT that help with workflows, research, or leveraging your data.

This is more of a *developer-customized* option than a plug-and-play support bot.

🧠 Summary: What’s Available Now

Tool	Purpose	Official?
Ask Red Hat (Portal Chatbot)	Natural-language access to support content and docs	✅ Yes
Red Hat Developer Lightspeed	AI assistance for developer workflows	✅ Yes
Third-party ChatGPT Linux Bots	Community AI helpers for troubleshooting	❌ No
ChatGPT Apps /	Custom integrations with your own	🟡 Possible with

Tool	Purpose	Official?
Connectors	system data	development

How These Help in Real Support Cases

- **Ask Red Hat** can quickly surface relevant KB articles, advisories, and support tickets related to error messages or specific symptoms you describe — saving time in triage.
- **Lightspeed tools** can help with understanding logs, config issues, or debugging output in context.
- **Custom ChatGPT integrations** can bring your own internal scripts, logs, or case history to bear in a conversational troubleshooting session — if you build or configure them.

If you'd like, I can show:

- **How to enable or access Ask Red Hat** step-by-step in your Red Hat Customer Portal,
- Or **examples of prompts** that get the most accurate support guidance from these tools.

Just let me know!



Is the sosreport tool installed on red hat enterprise linux 9 and 10 by default?

Short answer:

On **Red Hat Enterprise Linux 9 and 10**, the `sosreport -style tool (sos report)` is **not guaranteed to be installed by default** in all installation cases — especially *minimal* installations.

Here's what that means in practice:

RHEL 9

- A **minimal installation** of RHEL 9 **does not** include the `sos` package that provides the `sosreport` / `sos report` command by default.
- On more complete server installations (e.g., "Server with GUI" or "Standard System"), it **may** already be present, but you should verify.
- If it's not available, you must install it manually:

```
bash
```

[Copy code](#)

```
sudo dnf install sos
```

- After installation, the command you'll use is:

```
bash
```

[Copy code](#)

```
sudo sos report
```

(On RHEL 8+ this replaced the older `sosreport` command, though `sosreport` may still redirect for compatibility.)

RHEL 10

- RHEL 10 behaves similarly: it **does not guarantee** that the `sos` package is installed by default on all installation profiles.
- So you may need to install it yourself using:

```
bash
```

[Copy code](#)

```
sudo dnf install sos
```

- Once installed, generate a report with:

```
bash
```

[Copy code](#)

```
sudo sos report
```

🧠 Why This Matters

- **Minimal installs** are common for production systems where administrators want only essential packages installed — this reduces attack surface and disk footprint.
- Red Hat documentation notes that for RHEL 9 the `sos` utility may need to be installed separately if absent.

✅ Quick Verification

Before running a report, check if the command is available:

```
bash
```

[Copy code](#)

```
rpm -q sos
```

- If it prints a version → `sos` is installed.
- If it says “package `sos` is not installed” → install it with `dnf install sos`.

⚡ Running the Report

Once installed:

```
bash
```

[Copy code](#)

```
sudo sos report
```

This produces a diagnostic archive typically in `/var/tmp/` that you can upload to Red Hat Support when opening a case.

If you'd like, I can show you **how to install and run** `sos report` **step by step**, including how to verify the package and handle scenarios where it's missing.

👑 Sources

By default in red hat enterprise linux 9 and 10 is there a container runtime available by default?

Here's how **container runtimes** are handled by default on **Red Hat Enterprise Linux (RHEL) 9 and 10**:

Default Container Tools and Runtimes

Podman and Related Tools

- **Podman**, along with container tooling like **Buildah** and **Skopeo**, are the supported tools for managing containers on RHEL 9 and RHEL 10 — and replace the older Docker engine entirely. Red Hat does **not ship the Docker engine** or the traditional `docker` package in the standard RHEL repos for RHEL 8 and later releases.
- These tools are **part of the “Container Tools” content** Red Hat provides and are intended to be used for pulling, building, running, and managing OCI containers on the host.

In other words: **Podman is the supported container runtime toolset on RHEL 9 and 10.**

What RHEL Includes by Default

Whether Podman *is installed by default* depends on the **installation profile** you choose:

- **Minimal or custom lightweight installations** may *not* include container tooling by default.
- On more complete systems (like standard server installs with development or container tooling groups), Podman *often is available without manual installation*.
- In many RHEL images, **Podman and its dependencies are part of the default package set** — but you should verify on your specific install.

If Podman is not present, you can install it easily:

```
bash
```

 Copy code

```
sudo dnf install podman
```

And for the broader toolset:

```
bash
```

 Copy code

```
sudo dnf install buildah skopeo
```

(Or install the full container tools meta-package where applicable.)

What Underlies Podman

Podman itself uses an **OCI (Open Container Initiative) runtime** to actually run containers.

On RHEL:

- **RHEL 9:** Uses **crun** as the default container runtime by default (with `runc` also available).
- **RHEL 10:** Also uses **crun** as the default runtime; the older `runc` is removed or optional per documentation.





This runtime (`crun`) is what actually implements the container process isolation — Podman is the **tooling layer** above it.

Docker Engine and Traditional Runtime

- The **Docker engine (daemon)** and `docker` package are **not included or supported by default** in RHEL 9 and RHEL 10.
- You *can* install Docker from external repositories if you want, but:
 - It won't be supported by Red Hat's official repos,
 - It might require custom configuration, and
 - Red Hat *recommends Podman* instead.

Podman can even provide a **compatibility layer** so commands like `docker run` work via Podman by installing `podman-docker`, which makes Podman respond to the Docker CLI.


Summary

Feature / Version	Default Present?	Notes
Podman (container engine)	 Usually / Yes	Supported container runtime toolset, central in RHEL 9/10
Buildah / Skopeo	 Often	Complementary container tooling
Docker engine	 No	Not included or supported by default
OCI runtime (crun)	 Yes (used by Podman)	Underlying container runtime execution
OCI runtime (runc)	Optional/older	Default in RHEL 9 alongside crun; RHEL 10 favors crun

Practical Check

On your RHEL system, you can confirm whether Podman is installed:

```
bash
```

 Copy code

```
podman --version
```

If it's not installed, the package can be added via `dnf install podman`.

Bottom Line

- **RHEL 9 and RHEL 10 do include container tooling**, with **Podman as the standard container engine** once installed.
- Traditional Docker is **not provided by default**, and Red Hat **encourages using Podman** and other OCI-compliant tools instead.

If you want, I can explain **the differences between Podman and Docker**, or show how the container runtime is used with Kubernetes/OpenShift on RHEL hosts. Just let me know!

 Sources

Does ChatGPT have the ability to call the `sos` tool on red hat enterprise linux 9 and 10 if it is installed and available?

Short answer: **No. ChatGPT cannot directly call or run the `sos / sosreport` tool on RHEL 9 or 10**, even if it is installed and available on the system.

Here's the clear breakdown, because this is an important boundary to understand.

Why ChatGPT Cannot Run `sos report`

1. ChatGPT Has No Direct Access to Your Systems

ChatGPT:

- Does **not** have SSH access
- Does **not** run inside your RHEL host
- Does **not** have permission to execute OS commands
- Cannot interact with your filesystem, kernel, or processes

So it cannot do this:

```
bash
```

 Copy code

```
sudo sos report
```

on your behalf.

This is true even if:

- The `sos` package is installed
- The system is RHEL 9 or 10
- You are logged in as root on your machine

What ChatGPT *Can* Do Instead (Very Effectively)

While ChatGPT can't *run* `sosreport`, it **can assist before, during, and after** its use.

✓ Before Running `sosreport`

ChatGPT can:

- Tell you **whether** `sos` **should be installed**
- Help you choose **plugins to include/exclude**
- Recommend **redaction options**
- Help you decide **which collectors are relevant** to your issue

Example:

"I'm troubleshooting a kernel panic on RHEL 9—what `sos` plugins should I include?"

✓ While Running `sosreport`

ChatGPT can:

- Provide the **exact command** to run
- Explain each flag:

```
bash
```

 Copy code

```
sudo sos report --case-id=01234567 -o kernel,storage
```

- Help you handle:
 - Minimal installs
 - Offline systems
 - Air-gapped environments

✓ After Generating the `sosreport`

ChatGPT can:

- Explain the **directory structure**

- Help you understand **what files Red Hat support will inspect**
- Help you **manually review or redact** sensitive data
- Explain error messages if collection failed

You can also paste:

- Selected file contents
- Error output
- Logs from inside the sosreport

...and ChatGPT can help **interpret them**.

Important Security Boundary (By Design)

This limitation is **intentional and desirable**:

- Allowing an AI to execute system diagnostics would be a **major security risk**
- Enterprises require:
 - Explicit operator control
 - Auditable actions
 - Human approval for data collection

Red Hat support follows the same rule:

Support engineers **never run sosreport directly on customer systems** either.

You (or your automation) always run it.

Can ChatGPT Be Integrated Indirectly?

Yes — but **only with tooling you control**.

Common Patterns (Advanced / Enterprise)

1. ChatGPT + Runbooks

- ChatGPT suggests commands
- Humans execute them

2. ChatGPT + Automation

- You build a wrapper script that:

- Runs `sos report`
- Uploads artifacts
- ChatGPT helps design or reason about that workflow

3. ChatGPT + Log Analysis

- You paste outputs from:
 - `sosreport` files
 - `must-gather` results
- ChatGPT helps analyze patterns

But even here:

ChatGPT never initiates execution.

Comparison With Red Hat Tools

Tool	Can run <code>sosreport</code> ?	Notes
ChatGPT	 No	Advisory / analytical only
Human admin	 Yes	Required
Script / Ansible	 Yes	You control execution
Red Hat Support	 No	Customer always runs it
OpenShift <code>must-gather</code>	 (indirect)	Still user-initiated

Bottom Line

- **ChatGPT cannot call or execute** `sos report`
- It **can guide, explain, review, and analyze**
- Execution always remains under **your control**

That separation is exactly what makes ChatGPT safe to use in enterprise support workflows.

If you want, next I can:

- Show a **safe ChatGPT-assisted sosreport runbook**
- Explain how teams **standardize sos collection**
- Compare **human vs automated sosreport workflows**
- Show how ChatGPT can help analyze **sosreport + must-gather together**

Just tell me where you want to go.