Infrastructure Security Using Linux

Computer science / Cybersecurity

Monitoring & Performance Analysis

28 → 29

Monitoring

Monitoring

- Integrate every system into your monitoring platform before go-live
- Continuously track performance and health metrics
- Proactively detect and resolve issues early
- A monitoring-first mindset makes you a sysadmin superhero

Monitoring: Goals & Workflow

- **Goals:**
  - Ensure infrastructure operates as expected
  - Provide clear, actionable data for management & planning
- **Core Process:**
  - **Harvest:** collect raw metrics from systems & devices
  - **Analyze:** evaluate data & decide required actions
  - **Execute & Report:** forward data and actions to back-end systems for remediation, alerting, and record-keeping

Monitoring: Instrumentation & Data Collection

- **Actionable Data:** performance metrics, availability stats, capacity measures, state changes, logs, business KPIs
- **Data Types:**
  - **Real-time Metrics** (current state)
  - **Events** (logs & notifications)
  - **Historic Trends** (aggregated time-series)
- **Time-Series Storage:**
  - High-res recent data → summarized over time
  - e.g., 1 h @1 s; 1 week @1 min; 1 year @1 h
- **Notifications:** deliver targeted alerts to administrators & developers

The monitoring culture

- When you embark on a monitoring journey, embrace the following tenets:
- **Universal Coverage**
  - Monitor everything users depend on
  - Track all available system metrics
  - Include high-availability components and backups
- **Organizational Integration**
  - Make monitoring a required part of all technical roles' work
  - Share monitoring data widely through accessible dashboards
  - Distribute alert response across all technical teams

The monitoring culture

- **Best Practices**
  - Fix root causes rather than suppressing alerts
  - Tune alerts to eliminate false positives
  - Use monitoring to improve quality of life by reducing uncertainty
  - Treat monitoring as essential infrastructure, not an optional extra

Monitoring Platforms

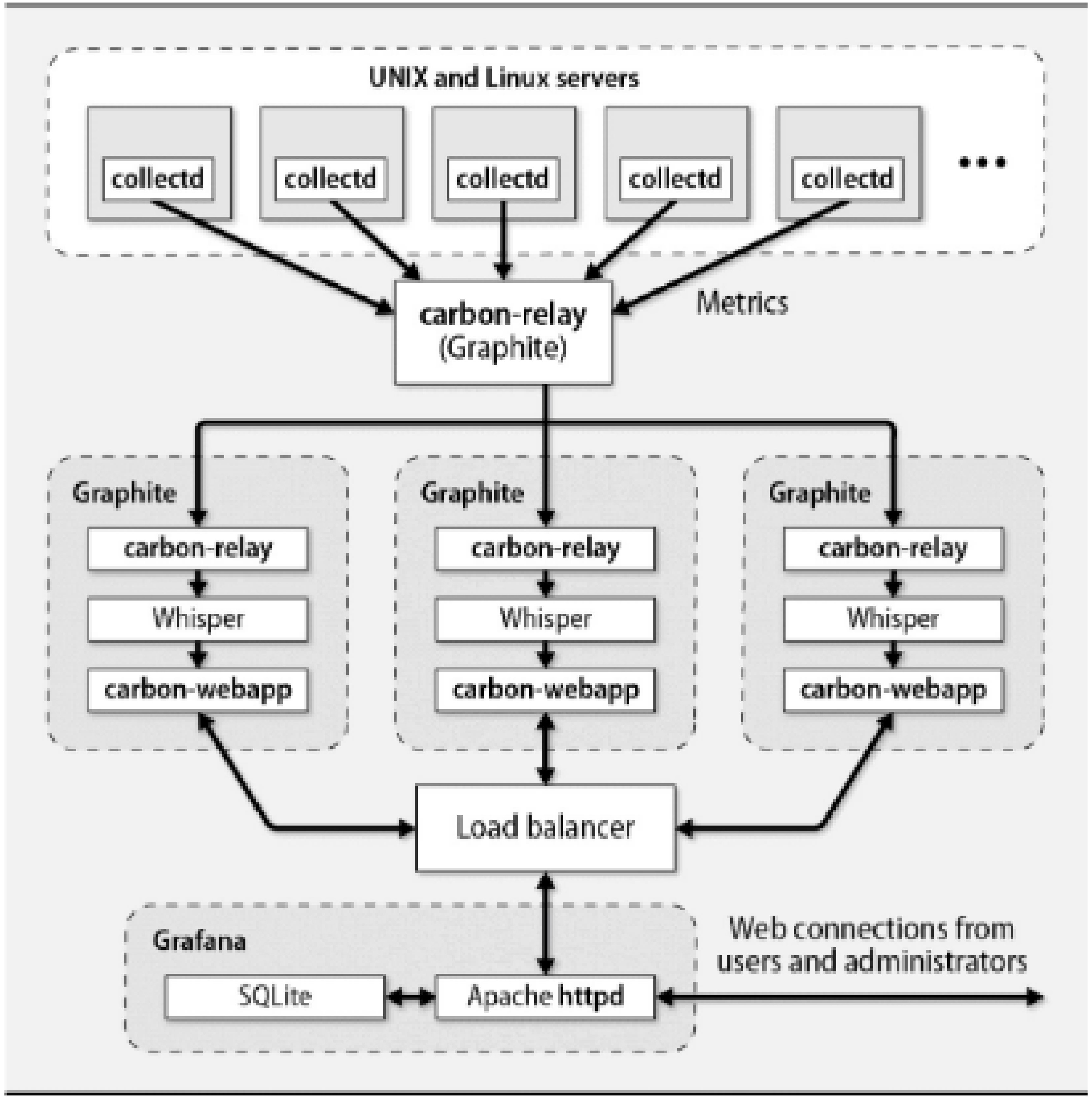
- **Data-Gathering Flexibility**
  SQL, DNS, HTTP, SNMP, custom APIs
- **User Interface**
  Customizable dashboards for different teams
- **Cost & Licensing**
  Fit within your budget and scale needs
- **Automated Discovery**
  Network scans (ping, ARP, SNMP) to map devices
- **Reporting & Alerts**
  Email/SMS, ChatOps integration, auto-ticketing

Open-Source Real-Time Monitoring Platforms

- **Nagios & Icinga**
  - First-generation tools with extensive SNMP and service-monitoring scripts
  - Highly modular configuration—write custom checks for any metric
- **Sensu Core**
  - Similar real-time focus with a pluggable, event-driven architecture
- **Trend:** transitioning from these platforms to specialized time-series systems for long-term data analysis

Open-Source Time-Series Monitoring Platforms

- **Key Players:** Graphite, Prometheus, InfluxDB, Munin
- **Graphite Highlights:**
  - **Sub-second collection:** high-resolution metrics
  - **Whisper DB + Carbon:** simple TS database with modular collector
  - **Query language:** easy retrieval & aggregation
  - **Powerful summarization:** roll-ups for long-term trends
  - **Visualization:** often paired with **Grafana** for dashboards


UNIX and Linux servers
collectd
collectd
collectd
collectd
collectd
• • •
carbon-relay
(Graphite)
Metrics
Graphite
carbon-relay
Whisper
carbon-webapp
Graphite
carbon-relay
Whisper
carbon-webapp
Graphite
carbon-relay
Whisper
carbon-webapp
Load balancer
Grafana
SQLite
Apache httpd
Web connections from
users and administrators

Open source time-series platforms

- **Prometheus**: Amazing but does not allow for clustering, however, which is a big cons!
- **InfluxDB** is an extraordinarily developer-friendly time-series monitoring platform that supports a broad array of programming languages. Much like Graphite, InfluxDB is **really just a time-series database engine**. You'll need to complete the package with external components such as Grafana to form a complete monitoring system that includes features like alerting.

Commercial platforms

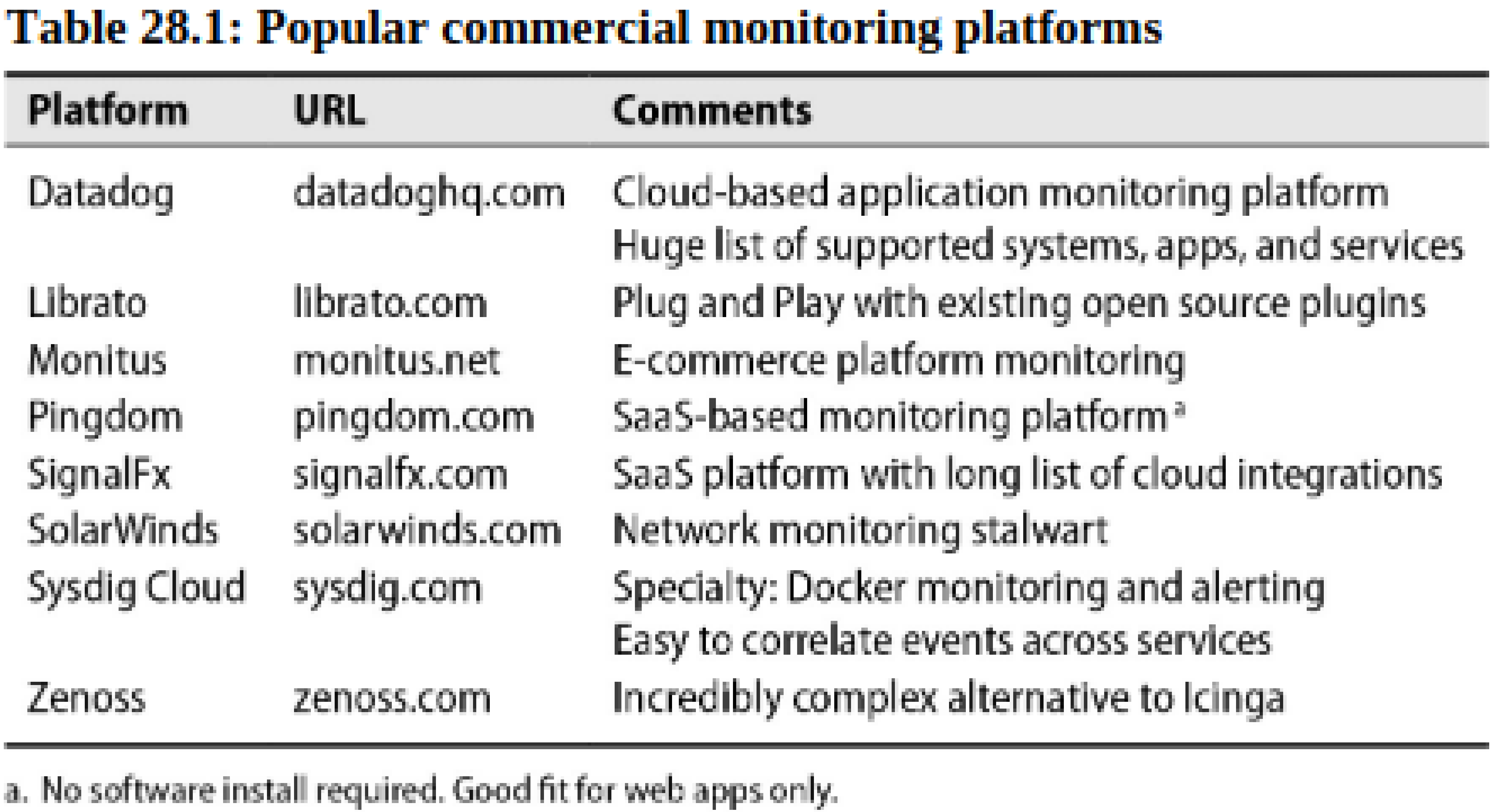


Table 28.1: Popular commercial monitoring platforms

Platform	URL	Comments
Datadog	datadoghq.com	Cloud-based application monitoring platform Huge list of supported systems, apps, and services
Librato	librato.com	Plug and Play with existing open source plugins
Monitus	monitus.net	E-commerce platform monitoring
Pingdom	pingdom.com	SaaS-based monitoring platform[a]
SignalFx	signalfx.com	SaaS platform with long list of cloud integrations
SolarWinds	solarwinds.com	Network monitoring stalwart
Sysdig Cloud	sysdig.com	Specialty: Docker monitoring and alerting Easy to correlate events across services
Zenoss	zenoss.com	Incredibly complex alternative to Icinga

[a] No software install required. Good fit for web apps only.

Network Monitoring

- **Unit:** ICMP Echo Request (ping)
- **How it works:**
  - Send echo request → receive echo reply
  - Confirms network path & host availability
- **Verifies:**
  - Gateways & intermediate devices are up
  - Target host powered on & kernel running
- **Caveat:**
  - Gateways may drop pings
  - Occasional packet loss is normal

System Monitoring

- Monitor kernel-managed metrics: CPU, memory, I/O, devices
- Retrieve values via:

  /proc/loadavg (Linux)

  sysctl -n vm.loadavg (FreeBSD)
- Common commands reporting load averages:

  uptime, w, sar

Command	**Description**
df	Show disk space usage
du	Directory sizes
free	Free, used, and swap (virtual) memory
iostat	Disk performance and throughput
mpstat	Per-processor utilization on multi-processor systems
lsof	List open files and network connections
netstat	Network connection tracking
vmstat	Process, CPU, and memory statistics
w	List of logged-in users and their activity
uptime	System uptime and load average
top	Display a list of running processes
sysctl	Access to system configuration data
sar	System Activity Report

Application & Log Monitoring

- **Application Monitoring:** track performance and health of servers, databases & web services to ensure they remain responsive and performant
- **Log Monitoring:** parse free-form log entries (e.g., via grep or dedicated agents), extract key events & metrics, then feed into dashboards and alerts (pipeline complexity ranges from simple scripts to full log-management systems)

Security Monitoring (SecOps)

- Leverage open-source, commercial tools & managed security service providers (MSSPs)
- Detect breaches early—most go unnoticed for months
- Balance outsourcing vs. in-house vigilance
- Ask yourself: would you trust someone to watch your wallet among 10,000?

Monitoring Tips & Tricks

- **Prevent Burnout**: rotate on-call teams; enforce regular breaks
- **Define Critical Alerts**: specify 24×7 issues vs. business-hour fixes
- **Cut Noise**: eliminate false positives and noncritical alerts
- **Create Runbooks**: document restart/reset procedures for all systems
- **Monitor the Monitors**: ensure your monitoring platform itself is supervised
- **Enforce Coverage**: no server or service goes live unmonitored

Performance Analysis

Performance Analysis & Tuning

- **Science & Art:**
  - Science: quantitative measurement & scientific method
  - Art: pragmatic resource balancing & trade-offs
- **Enduring Principles:**
  - Core performance determinants remain despite growing complexity
- **Cloud Abstraction:**
  - Multiple layers hide underlying hardware details
- **Virtualization ≠ No Tuning:**
  - Efficiency impacts cloud costs via billing models
- **Key Practice:**
  - Accurate measurement & continuous evaluation

Ways to Improve Performance

- Provision sufficient memory
- Upgrade to SSD storage
- Distribute load with load balancers
- Optimize code and job scheduling
- Use RAID for higher I/O throughput
- Monitor network traffic continuously

Factors Affecting Performance

- **CPU Utilization:** high % usage → CPU-bound workloads
- **Memory:** insufficient RAM causes swapping and delays
- **Storage I/O:** disk latency stalls processes; SSDs/RAID improve throughput
- **Network I/O:** impacted by NICs, switches, packet size/volume
- **Resource Contention:** waiting time for scarce resources degrades performance

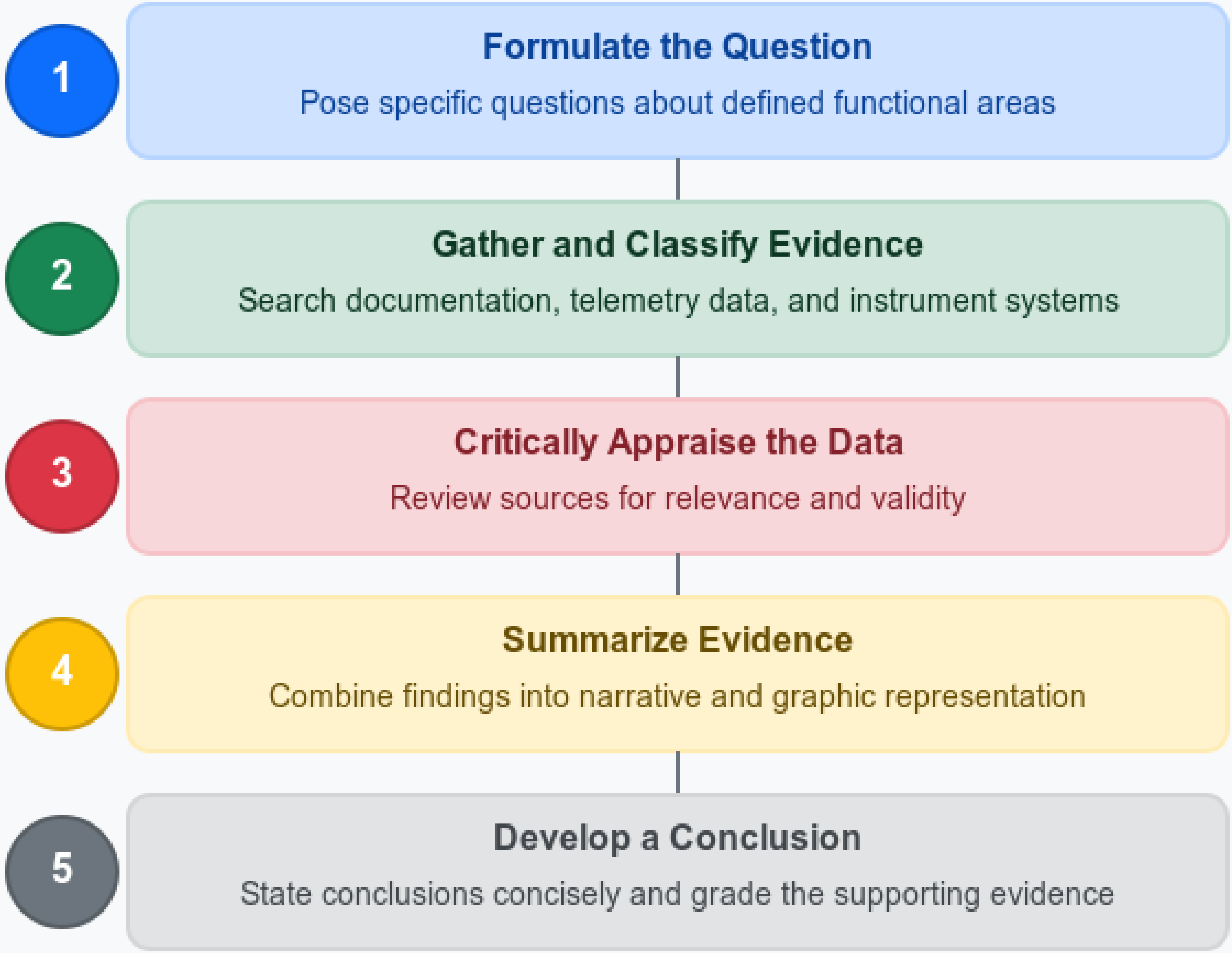
Stolen CPU Cycles

- **Hypervisor CPU Quotas:** VMs limited to allocated CPU share
- **Physical Oversubscription:** host hardware can be over-committed
- **Mitigations:**
  - Increase VM quota or upgrade instance size
  - Restart VM to land on less-busy host
  - Use tools like VMware DRS for dynamic balancing
- **Detection**: monitor the st (“stolen”) metric in **top**, **vmstat**, or **mpstat**

Analysis of Performance Problems: 5-Step Method

⚠ Don't Assume Anecdotal Reports Are Accurate

Apply scientific method to reach reliable, transparent conclusions



1 **Formulate the Question**

Pose specific questions about defined functional areas

2 **Gather and Classify Evidence**

Search documentation, telemetry data, and instrument systems

3 **Critically Appraise the Data**

Review sources for relevance and validity

4 **Summarize Evidence**

Combine findings into narrative and graphic representation

5 **Develop a Conclusion**

State conclusions concisely and grade the supporting evidence

Performance Analysis Workflow

1. Question: Why did page load jump from <2 s to >8 s during 9 am–5 pm?

2. Data Collection: CPU/memory/I/O metrics, network logs, browser timings, change logs

3. Analysis: Business-hour CPU spikes correlate with recent per-request image processing

4. Summary: Inefficient thumbnail processing (no caching) is CPU-bound bottleneck

5. Action: Implement result caching for images to restore performance without new hardware

System performance checkup

- On Linux OS, the /**proc** filesystem is the place to find an overview of the hardware your OS thinks you have.

File	**Contents**
/proc/cpuinfo	Information about the CPU(s)
/proc/meminfo	Memory size and usage
/proc/diskstats	Disk devices and usage statistics

DMI (Desktop Management Interface)

- Run for information on both FreeBSD and Linux is dmicdecode. It dumps the system's DMI (Desktop Management Interface) data.

- Ex : **dmidecode -t 4** will give you information about the processor.

Value	**Description**
1	System information
2	Baseboard (or motherboard)
3	Chassis information
4	Processor information
7	Cache information
8	Port connection information
9	System slot information
11	OEM strings
12	System configuration options
13	BIOS language information
16	Physical memory array
17	Memory device
19	Memory array mapped address
32	System boot information
38	IPMI device information