Network Monitoring Tools Cheat Sheet

A comprehensive guide to essential network monitoring tools, covering their functionalities, usage, and key metrics for effective network administration.

Basic Monitoring Tools

Ping

Description:	Tests the reachability of a host on an IP network. Measures the round-trip time for messages sent from the originating host to a destination computer.
Usage:	`ping <hostname or IP address>`
Key Metrics:	Round-trip time (RTT), packet loss.
Example:	`ping google.com`
Troubleshooting:	Used to identify basic connectivity issues.

Traceroute/Tracepath

Description:	Traces the route packets take to a destination. Lists all the routers the packet passes through.

Usage:	`traceroute <hostname or IP address>`
	`tracepath <hostname or IP address>`

Key Metrics:	Path taken, latency at each hop.
Example:	`traceroute google.com`
Troubleshooting:	Identifies network bottlenecks and routing issues.

Netstat

Description:	Displays network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.
Usage:	`netstat -a`
Key Metrics:	Active network connections, listening ports.
Example:	`netstat -ant`
Troubleshooting:	Detects unauthorized connections and port usage.

Advanced Monitoring Tools

Tcpdump

Description:	A powerful command-line packet analyzer. It allows you to intercept and display TCP/IP and other network traffic that is being transmitted or received over a network.
Usage:	`tcpdump -i <interface> <filter>`
Key Metrics:	Captured packets, source/destination addresses, protocols.
Example:	`tcpdump -i eth0 port 80`
Troubleshooting:	Diagnoses network traffic issues, analyzes packet content.

Wireshark

Description:	A network protocol analyzer that captures and analyzes network traffic. It has a GUI and provides detailed information about network packets.
Usage:	GUI-based application; select interface and start capturing.
Key Metrics:	Packet details, protocol analysis, traffic patterns.
Example:	Capturing HTTP traffic and analyzing request/response headers.
Troubleshooting:	In-depth packet analysis for troubleshooting complex network issues.

Nmap

Description:	A network scanner used to discover hosts and services on a computer network by sending packets and analyzing the responses. It is also used for security auditing.
Usage:	`nmap <target>`
Key Metrics:	Open ports, services running, OS detection.
Example:	`nmap scanme.nmap.org`
Troubleshooting:	Detects open ports and identifies potential vulnerabilities.

System Monitoring Tools

Iperf/Iperf3

Description:	A tool for active measurements of the maximum achievable bandwidth on IP networks. It supports tuning of various parameters related to timing, buffers and protocols (TCP, UDP, SCTP)
Usage:	Server: `iperf3 -s`, Client: `iperf3 -c <server_ip>`
Key Metrics:	Bandwidth, jitter, packet loss.
Example:	`iperf3 -c 192.168.1.100`
Troubleshooting:	Tests network throughput and identifies bandwidth limitations.

NetHogs

Description:	A network top-like tool. Instead of breaking the traffic down per protocol or per subnet, like most such tools do, it groups bandwidth by process.
Usage:	`nethogs <interface>`
Key Metrics:	Bandwidth usage per process.
Example:	`nethogs eth0`
Troubleshooting:	Identifies processes consuming excessive bandwidth.

IfTop

Description:	Displays a real-time console display of network usage on an interface. It shows a list of network connections and the bandwidth they are using.
Usage:	`iftop -i <interface>`
Key Metrics:	Real-time bandwidth usage per connection.
Example:	`iftop -i eth0`
Troubleshooting:	Monitors real-time network traffic and identifies heavy connections.

Comprehensive Monitoring Solutions

Nagios

Description:	A powerful monitoring system that enables organizations to identify and resolve IT infrastructure problems before they affect critical business processes.
Usage:	Web-based interface; configure hosts, services, and checks.
Key Metrics:	Uptime, downtime, service status, resource utilization.
Example:	Monitoring CPU load, disk space, and network services on multiple servers.
Troubleshooting:	Proactive monitoring and alerting for critical infrastructure components.

Zabbix

Description:	An enterprise-class open source distributed monitoring solution. It monitors numerous network parameters and the health and integrity of servers.
Usage:	Web-based interface; configure hosts, items, triggers, and actions.
Key Metrics:	CPU utilization, memory usage, disk I/O, network traffic.
Example:	Collecting performance data, visualizing trends, and triggering alerts based on thresholds.
Troubleshooting:	Comprehensive monitoring and alerting for diverse IT environments.

Prometheus

Description:	An open-source systems monitoring and alerting toolkit. It collects metrics from configured targets at given intervals, evaluates rule expressions, displays the results, and can trigger alerts if some condition is observed to be true.
Usage:	Configure exporters, scrape targets, define alerting rules.
Key Metrics:	Time-series data, resource utilization, application performance.
Example:	Monitoring containerized applications using Kubernetes and visualizing metrics with Grafana.
Troubleshooting:	Effective monitoring and alerting for dynamic and scalable environments.