

Minion



Distributed Monitoring with OpenNMS Minion

A Minion is a lightweight stateless service that enables OpenNMS to communicate with devices and services in hard-to-reach locations. Minions are best suited for deployment at remote network locations that the core OpenNMS server may not have direct access to.

As long as the Minion can communicate back to your OpenNMS server, it can act as both a proxy for processing polling tasks and a receiver for SNMP traps, syslog messages, and network flow data to send back to your main data center.

Features

Distributed Monitoring

Deploy one (or more) Minion at remote sites to proxy data collection tasks back to your core data center for centralized alerting and reporting.

Easy to Configure

With a few simple configuration settings, your Minions will start talking back to your OpenNMS server and can be up and running in minutes. No need to replicate all your custom polling settings – those stay managed on your core server.

Help Scale Flow Processing

Minion can work with our Sentinel component to provide horizontal scalability for data processing of sites with a lot of flow data.

Deal with Overlapping Address Space

Use a Minion in each address space to ensure services are monitored properly: an IP address in one location will not collide with the same IP in another location.

Application Perspective Monitoring

Monitor a service's availability from different perspectives to help identify if nodes are down for everyone or if an outage affects only certain sites. Pinpoint not only where an issue occurs, but its impact on a user's or machine's digital experience.



Technical Requirements

OpenNMS instance	Identical version numbers for OpenNMS Horizon/ Meridian and Minion package
Communication endpoints	OpenNMS communicates with REST (8980/tcp) and ActiveMQ (616161/tcp) endpoints or Kafka (9092/tcp)