

# 17: Netdata - Real-time Performance Monitoring

Netdata is a distributed, real-time performance monitoring solution built with C and Node.js. It provides sub-second granularity metrics collection, zero-configuration operation, and comprehensive system monitoring including CPU, memory, disk, network, processes, and Docker containers. For API documentation, advanced configuration, and development guides, see the [official Netdata documentation](#).

## Prerequisites

- **Docker installed** (Chapter 3)
- **Docker Compose** (Chapter 3)
- **Optional: Traefik installed** (Chapter 4) for HTTPS with Let's Encrypt
- **Optional: Domain configured** (Chapter 4.5), e.g., `monitor.example.com`
- **Optional: Apprise installed** (Chapter 5) for alert notifications

## Installation via Infinity Tools

### Menu Installation

```
▣▣APPLICATIONS → Netdata → Install
```

### CLI Installation

```
sudo bash /opt/InfinityTools/Solutions/setup-netdata.sh --install
```

## Deployment Modes

### Traefik Mode (Default)

Uses Traefik for SSL termination and domain routing:

- Automatic Let's Encrypt certificate provisioning
- Domain-based access: `https://monitor.example.com`
- Security headers configured
- Requires: Traefik running, DNS A record configured

## Standalone Mode

Direct access with HTTP or HTTPS (self-signed):

- HTTP: `http://SERVER_IP:19999`
- HTTPS: `https://SERVER_IP:19999` (self-signed cert via nginx proxy)
- Default port: 19999 (configurable)
- No domain required

## Architecture

### Container

- **netdata** - Main application (netdata/netdata:latest)
- **netdata-ssl-proxy** - Nginx SSL proxy (standalone HTTPS mode only)

### Data Persistence

- **Config:** `/opt/speedbits/netdata-client/netdata/` (configuration files)
- **Lib:** `/opt/speedbits/netdata-client/netdata/lib/` (database, metrics)
- **Cache:** `/opt/speedbits/netdata-client/netdata/cache/` (temporary data)
- **SSL:** `/opt/speedbits/netdata-client/ssl/` (standalone mode certificates)

### Host Access

Netdata requires access to host system for monitoring:

- `/proc` - Process and system information (read-only)
- `/sys` - System information (read-only)
- `/var/run/docker.sock` - Docker API (read-only, for container monitoring)
- `/etc/passwd`, `/etc/group` - User information (read-only)
- `/etc/os-release` - OS information (read-only)

### Networks

- **Traefik network:** Joins Traefik's proxy network (Traefik mode)
- **netdata-internal:** Isolated bridge network (standalone mode)
- **borgmatic-db:** Network for Apprise integration (if enabled)

# Installation Process

## Configuration Steps

1. **SSL Mode Selection:** Choose Traefik (default) or Standalone
2. **If Traefik:** Provide domain name
3. **If Standalone:** Specify port (default: 19999) and SSL mode
4. **Streaming:** Optional parent-child streaming configuration
5. **Apprise Integration:** Optional alert notification setup

## What Gets Created

- **Directory:** `/opt/speedbits/netdata-client`
- **Container:** `netdata`
- **Docker Compose:** `/opt/speedbits/netdata-client/docker-compose.yml`
- **Config Files:** Health alerts, streaming config, Apprise integration

## Access Methods

### Traefik Mode

```
https://monitor.example.com
```

Direct web access after DNS propagation and SSL certificate generation (30-60 seconds).

### Standalone Mode

#### HTTP:

```
http://SERVER_IP:19999
```

#### HTTPS:

```
https://SERVER_IP:19999
```

Accept self-signed certificate warning (Advanced → Proceed).

# Security Configuration

## Access Security

- Traefik mode uses Let's Encrypt SSL (production-ready)
- Standalone HTTPS uses self-signed certificates (acceptable for internal use)
- Security headers configured (X-Frame-Options, CSP, etc.)
- **NO default authentication** - Dashboard is publicly accessible

## Container Security

- Runs with `SYS_PTRACE` and `SYS_ADMIN` capabilities (required for monitoring)
- Security option: `apparmor:unconfined`
- Read-only mounts for host filesystem access
- Docker socket mounted read-only

## Authentication

⚠ **CRITICAL:** Netdata has NO username/password protection by default!

- **Traefik mode:** Add Basic Auth via `websiteprotection.sh`
- **Standalone mode:** Keep on private network or use SSH tunnel
- **VPN:** Access via WireGuard VPN for secure remote access
- **SSH Tunnel:** `ssh -L 19999:localhost:19999 user@server`

# Metrics Collection

## System Metrics

- **CPU** - Per-core usage, load averages, interrupts
- **Memory** - RAM, swap, buffers, cache
- **Disk** - I/O, space usage, per-filesystem metrics
- **Network** - Per-interface traffic, connections, errors
- **Processes** - Per-process CPU, memory, I/O
- **System Load** - Load averages, context switches

# Docker Metrics

- Auto-discovery of all containers
- Per-container CPU, memory, disk, network
- Container health status
- Real-time metrics (1-second granularity)

# Data Retention

- Configurable retention period
- Default: 1 hour of 1-second data
- Can extend for longer historical data
- Low storage footprint (~50MB RAM)

# Alert Configuration

## Default Alerts

Pre-configured alerts in `health.d/`:

- **CPU Usage:** Warning > 80%, Critical > 95%
- **RAM Usage:** Warning > 80%, Critical > 95%
- **Disk Space:** Warning > 80%, Critical > 90%

## Apprise Integration

If Apprise is enabled:

- Alerts sent to Apprise via HTTP POST
- Apprise forwards to configured channels
- Supports all Apprise notification providers
- Config file: `health_alarm_notify.conf`

## Custom Alerts

Create custom alerts in `health.d/`:

```
# Example: Custom alert
alarm: custom_metric
```

```
on: system.cpu
lookup: average -3m unaligned of user,system
units: %
every: 1m
warn: $this > 75
crit: $this > 90
delay: down 5m multiplier 1.5 max 1h
info: Custom CPU alert
to: sysadmin
```

# Parent-Child Streaming

## Configuration

Stream metrics to a Netdata Director (parent server):

- Configured in `stream.conf`
- Requires director hostname/IP and API key
- Enables centralized monitoring of multiple servers
- Local dashboard remains available

## Use Cases

- Multi-server monitoring
- Centralized dashboards
- Unified alerting
- Historical data aggregation

# Environment Variables

## Netdata Container

- `NETDATA_CLAIM_TOKEN` - Optional Netdata Cloud claim token
- `NETDATA_CLAIM_ROOMS` - Optional Netdata Cloud rooms
- `NETDATA_CLAIM_URL` - Netdata Cloud URL (default: `https://app.netdata.cloud`)
- `DOCKER_HOST` - Docker socket path (default: `/var/run/docker.sock`)

# Configuration Files

## Main Configuration

- `netdata.conf` - Main Netdata configuration
- `stream.conf` - Parent-child streaming configuration
- `health_alarm_notify.conf` - Alert notification configuration
- `health.d/*.conf` - Individual alert definitions

## Customization

```
# Edit main config
nano /opt/speedbits/netdata-client/netdata/netdata.conf

# Edit alerts
nano /opt/speedbits/netdata-client/netdata/health.d/cpu_usage.conf

# Edit streaming
nano /opt/speedbits/netdata-client/netdata/stream.conf
```

# Troubleshooting

## Container Not Starting

```
docker logs netdata
docker ps -a | grep netdata
```

## Missing Metrics

- Verify host mounts: `docker exec netdata ls /host/proc`
- Check Docker socket: `docker exec netdata ls /var/run/docker.sock`
- Verify container capabilities

## Docker Containers Not Showing

- Verify Docker socket access

- Check Docker daemon is running
- Restart Netdata container

## Alerts Not Working

- Verify Apprise is running and accessible
- Check `health_alarm_notify.conf` configuration
- Test alert thresholds
- Check Netdata logs for errors

## Production Considerations

- **Access Method:** Use Traefik mode for production (trusted SSL)
- **Authentication:** Add Basic Auth protection (critical!)
- **Network Security:** Restrict access via firewall or VPN
- **Resource Usage:** ~50MB RAM, minimal CPU impact
- **Data Retention:** Configure appropriate retention period
- **Alerting:** Configure multiple notification channels for redundancy
- **Streaming:** Use director for multi-server environments

## Integration with Infinity Tools

Netdata complements Infinity Tools by providing:

- Real-time monitoring of all Infinity Tools applications
- Docker container monitoring for infrastructure
- System resource monitoring
- Alert integration with Apprise

### Recommended Monitoring:

- All Infinity Tools application containers
- System resources (CPU, RAM, disk)
- Network traffic
- Docker daemon health

## API & Automation

### REST API

- Netdata provides REST API for metrics access
- API documentation available in web interface
- Useful for automation and integration
- Export metrics to external systems

## Exporting Data

- Export graphs as images
- Export metrics as JSON/CSV
- Integrate with external monitoring systems

## Next Steps

Netdata is now operational. Use it to:

- Monitor all Infinity Tools applications
- Track Docker container health
- Monitor system resources in real-time
- Set up alert notifications
- Analyze performance trends

For advanced features, API usage, custom collectors, and development guides, refer to the [official Netdata documentation](#).

---

Revision #2

Created 17 November 2025 17:16:22 by bjoern

Updated 17 November 2025 17:20:23 by bjoern