

18: Netdata Director - Multi-Server Monitoring Hub

Netdata Director is a parent-child streaming architecture that enables centralized monitoring of multiple servers. The Director (parent) receives metrics streams from child nodes, providing a unified dashboard, centralized alerting, and long-term historical data retention for all monitored infrastructure.

⚠ **LICENSE REQUIREMENT:** Netdata Director is a Pro+ feature requiring a license. Community Netdata provides single-server monitoring only.

For advanced features, API documentation, and streaming configuration, see the [official Netdata documentation](#).

Prerequisites

- **Pro+ License** - Required for Director functionality
- **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., `monitoring.example.com`
- **Optional: Apprise installed** (Chapter 5) for alert notifications
- **Multiple servers** - Director is designed for 2+ server environments

Installation via Infinity Tools

Menu Installation

```
❏❏APPLICATIONS → Netdata Director → Install
```

CLI Installation

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

Architecture

Parent-Child Streaming

- **Director (Parent):** Central dashboard server receiving streams
- **Child Nodes:** Regular Netdata installations streaming to Director
- **Stream API Key:** Authentication token for child-to-parent connection
- **Unidirectional:** Metrics flow child → parent only

Container

- **netdata-director** - Director instance (netdata/netdata:latest)
- **netdata-director-ssl-proxy** - Nginx SSL proxy (standalone HTTPS mode only)

Data Persistence

- **Config:** `/opt/speedbits/netdata-director/netdata/`
- **Lib:** `/opt/speedbits/netdata-director/netdata/lib/` (metrics database)
- **Cache:** `/opt/speedbits/netdata-director/netdata/cache/`
- **API Key:** `/opt/speedbits/netdata-director/stream-api-key.txt`

Deployment Modes

Traefik Mode (Default)

Uses Traefik for SSL termination and domain routing:

- Automatic Let's Encrypt certificate provisioning
- Domain-based access: `https://monitoring.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)

Stream API Key

Generation

- 32-character random key generated during installation
- Saved in `stream-api-key.txt`
- Used for child node authentication
- Must be kept secret

Usage

Child nodes use this key to authenticate when streaming metrics:

- Configured in child node's `stream.conf`
- Director validates key before accepting streams
- Multiple children can use same key (or separate keys per child)

Streaming Configuration

Director Configuration

File: `/opt/speedbits/netdata-director/netdata/stream.conf`

```
[stream]
    enabled = no # Director doesn't stream to anyone

[$STREAM_API_KEY]
    enabled = yes
    default memory mode = dbengine
    health enabled by default = auto
    default postpone alarms on connect seconds = 60
    default history = 3600
    allow from = *
```

Child Node Configuration

Configured during child node installation:

- Director hostname/IP
- Director port (default: 19999)
- Stream API key

Access Methods

Traefik Mode

```
https://monitoring.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
```

Security Configuration

Access Security

- Traefik mode uses Let's Encrypt SSL (production-ready)
- Standalone HTTPS uses self-signed certificates
- Security headers configured
- **NO default authentication** - Dashboard is publicly accessible
- **Basic Auth incompatible** - Blocks child node streaming

Authentication Limitations

CRITICAL: Basic Auth cannot be used with Director because:

- Child nodes use HTTP API to stream metrics
- Basic Auth blocks unauthenticated API requests

- Child nodes cannot authenticate via Basic Auth
- Result: Child nodes cannot connect

Security Alternatives

- **Firewall Rules:** Restrict access to trusted IPs only
- **VPN Access:** Access Director via WireGuard VPN
- **Netdata Cloud:** Use official Netdata Cloud service
- **Network Isolation:** Keep Director on private network

Stream API Key Security

- API key provides authentication for child nodes
- Keep key secret (only share with trusted servers)
- Director validates key before accepting streams
- Can use separate keys per child node (advanced)

Alert Configuration

Apprise Integration

If Apprise is enabled, Director sends alerts for ALL child nodes:

- Centralized alert management
- Alerts include server hostname
- Single notification channel for all servers
- Config file: `health_alarm_notify.conf`

Alert Flow

1. Child node detects issue
2. Alert sent to Director
3. Director forwards to Apprise
4. Apprise sends to configured channels

Data Retention

Retention Periods

- **High-resolution:** 1 hour (1-second granularity)
- **Mid-resolution:** 1 day (1-minute granularity)
- **Low-resolution:** 30 days (15-minute granularity)

Storage

- Metrics stored in `dbengine` mode
- Configurable retention in `netdata.conf`
- Storage scales with number of child nodes

Child Node Connection

Connection Process

1. Install Netdata on child server (Chapter 17)
2. Enable streaming during installation
3. Provide Director hostname/IP
4. Provide Director port (default: 19999)
5. Provide Stream API key
6. Child node connects automatically

Connection Verification

- Wait 1-2 minutes for connection to establish
- Check Director dashboard dropdown for child node
- Verify metrics appearing in Director
- Check Director logs: `docker logs netdata-director`

Troubleshooting

Child Nodes Not Connecting

- Verify API key is correct
- Check network connectivity (firewall rules)
- Verify Director port is accessible
- Check child node logs for connection errors
- Check Director logs: `docker logs netdata-director`

Streaming Issues

- Verify `stream.conf` configuration
- Check API key matches between child and parent
- Verify Director is accepting connections
- Check network connectivity

Production Considerations

- **Access Method:** Use Traefik mode for production (trusted SSL)
- **Security:** Implement firewall rules or VPN access (cannot use Basic Auth)
- **Network:** Ensure Director is accessible from all child nodes
- **Storage:** Plan storage capacity based on number of child nodes
- **API Key Management:** Use separate keys per child for enhanced security
- **Monitoring:** Monitor Director itself (resource usage, connectivity)

Advanced Configuration

Multiple API Keys

Create separate API keys for different child nodes:

```
# In stream.conf, add multiple sections:  
[api-key-1]  
    enabled = yes  
    allow from = 192.168.1.10  
  
[api-key-2]  
    enabled = yes  
    allow from = 192.168.1.20
```

IP Restrictions

Restrict which IPs can connect:

```
[$STREAM_API_KEY]  
    enabled = yes
```

```
allow from = 192.168.1.0/24 # Only allow from this subnet
```

Integration with Infinity Tools

Netdata Director complements Infinity Tools by providing:

- Centralized monitoring of all Infinity Tools servers
- Unified alerting for entire infrastructure
- Historical data retention for capacity planning
- Cross-server performance comparison

Next Steps

Netdata Director is now operational. Use it to:

- Connect child nodes from all your servers
- Monitor entire infrastructure from one dashboard
- Set up centralized alerting
- Analyze performance trends across servers
- Plan capacity based on historical data

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

Revision #3

Created 17 November 2025 17:17:13 by bjoern

Updated 17 November 2025 17:22:23 by bjoern