

# Munin Workshop

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME
1.0	27.11.2009		PS

---

## Contents

<b>1</b>	<b>Workshop</b>	<b>1</b>
<b>2</b>	<b>Hinweis</b>	<b>1</b>
<b>3</b>	<b>Allgemein</b>	<b>1</b>
<b>4</b>	<b>Funktionsweise</b>	<b>2</b>
<b>5</b>	<b>Installation</b>	<b>2</b>
<b>6</b>	<b>Konfiguration</b>	<b>2</b>
6.1	Node . . . . .	2
6.2	Master . . . . .	3
<b>7</b>	<b>Sicherheit</b>	<b>3</b>
<b>8</b>	<b>Plugin</b>	<b>4</b>
8.1	Allgemein . . . . .	4
8.2	Ein einfaches Plugin . . . . .	4
8.3	Testen . . . . .	4
8.4	Wildcard Plugin . . . . .	5
8.5	Multigraph plugins . . . . .	5
<b>9</b>	<b>Munin Protokol</b>	<b>5</b>
<b>10</b>	<b>Links</b>	<b>5</b>

---

## 1 Workshop

**Titel:**

Munin Workshop

**Datum:**

27.11.2009

**Referent:**Philipp Seidel (<http://www.dinotools.de/>)**Ort:**

ComputerTreff Chemnitz

## 2 Hinweis

Dieses Material dient der Ergänzung zum Munin Workshop und stellt nur eine Sammlung einiger der gezeigten Beispiele dar.

## 3 Allgemein

- Monitoring-Tool
  - es gibt einen Master und einen oder mehrere Nodes
  - Darstellung mit RRDtools
  - Auflösung des Diagramms
    - 1. Tag: 1 Wert pro 5 Minuten
    - 1. Woche: 1 Wert pro 30 Minuten
    - 1. Monat: 1 Wert pro 2 Stunden
    - 1. Jahr: 1 Wert pro Tag
  - einfache Konfiguration
  - Plugin-System
    - leicht erweiterbar
    - können relativ einfach Selbst geschrieben werden
  - unterstützte Betriebssysteme
    - Computer mit Linux/Unix
    - Router mit OpenWRT(Linux)
    - Portierung auf Windows existiert
    - auch andere Systeme, so lange Perl vorhanden
-

## 4 Funktionsweise

- Installation eines Masters
- Installation eines oder mehrerer Nodes(Clients)
- Abfrage der Daten wird auf Master durch CronJob ausgelöst
  - Master baut Verbindung zu allen Nodes auf und holt die Daten ab
  - Master schreibt die Daten in RRD-Archive
  - Master erstellt Diagramme zur Auswertung
- Darstellung der Diagramme per Web-Interface
  - erzeugt werden nur statische Dateien → keine besonderen Anforderungen an Webserver
  - HTTPS oder Authentifikation muss bei Webserver eingerichtet werden

## 5 Installation

- viele Linux Distributionen bringen fertige Pakete mit
  - Debian, Ubuntu: munin, munin-node
- kann auch von Hand installiert werden
  - Quellen gibt es auf <http://munin.projects.linpro.no/>

## 6 Konfiguration

### 6.1 Node

- zuerst sollte Node konfiguriert werden, bevor der Master dessen Werte abholt
- Konfigurationsdateien liegen unter /etc/munin
  - munin-node.conf - eigentliche Konfigurationsdatei
  - plugin-conf.d - Ordner mit zusätzlicher Konfiguration von Plugins
  - plugins - Ordner mit Symlinks zu den Plugins; nur diese Plugins gelten als aktiv und werden vom Node berücksichtigt, wenn Master Daten abholen will
- nach jeder Änderung der Konfiguration sollte die munin-node Anwendung neu gestartet werden.

---

**Note**

Die einzelnen Optionen in der Datei munin-node.conf werden im Workshop genauer erläutert. Zum nachlesen: <http://munin.projects.linpro.no/wiki/munin-node.conf>

---

**Note**

Die Einstellungen im Verzeichnis plugin-conf.d werden im Workshop genauer erläutert. Zum nachlesen: <http://munin.projects.linpro.no/wiki/plugin-conf.d>

---

Mit `munin-node-configure` können Informationen zur möglichen Konfiguration des Nodes angezeigt werden.

Liste aller Plugins und was noch für die Nutzer dieser benötigt wird oder welche Parameter übergeben werden können.

```
$ munin-node-configure --suggest
Plugin          | Used | Suggestions
-----
acpi            | no   | [ACPI program not found]
apache_accesses | no   | [LWP::UserAgent not found]
apache_processes | no   | [LWP::UserAgent not found]
apache_volume   | no   | [LWP::UserAgent not found]
courier_mta_mailqueue | no   | [spooldir not found]
courier_mta_mailstats | no   | [could not find executable]
courier_mta_mailvolume | no   | [could not find executable]
if_             | yes  | -eth0
if_err_        | yes  | -eth0
```

Liste aller Symlinks, die erstellt werden können.

```
munin-node-configure --shell
ln -s /usr/share/munin/plugins/df_inode /etc/munin/plugins/df_inode
ln -s /usr/share/munin/plugins/memory /etc/munin/plugins/memory
ln -s /usr/share/munin/plugins/open_files /etc/munin/plugins/open_files
ln -s /usr/share/munin/plugins/swap /etc/munin/plugins/swap
ln -s /usr/share/munin/plugins/vmstat /etc/munin/plugins/vmstat
```

## 6.2 Master

- ist ein Node fertig konfiguriert, kann er beim Master eingetragen werden
- Konfigurationsdateien liegen unter `/etc/munin`
  - `munin.conf` - eigentliche Konfigurationsdatei
  - `templates` - Templates aus denen die Webseite erstellt wird

---

### Note

Die einzelnen Optionen in der Datei `munin.conf` werden im Workshop genauer erläutert. Zum nachlesen: <http://munin.projects.linpro.no/wiki/munin.conf>

---

## 7 Sicherheit

- Kommunikation ist unverschlüsselt
    - ab Version 1.4 gibt es TLS Unterstützung
    - nutzen von `stunnel`
    - nutzen von `ssh` Portweiterleitung
  - Zugriffsrechte
    - IPs von denen auf Node zugegriffen werden darf kann eingestellt werden
  - weiterer Schutz
    - eigenes Subnetz
    - eigenes VLAN
-

- iptables

---

**Tip**

Infos zu TLS <http://munin.projects.linpro.no/wiki/MuninTLSSetup>

---

## 8 Plugin

### 8.1 Allgemein

- einfache ausführbare Programme
  - meist Shell, Perl, AWK, etc.
  - selten irgendwelche C, C++, etc. Programme

### 8.2 Ein einfaches Plugin

- Load Graph

---

**Tip**

Infos zu Load bei Linux: <http://de.wikipedia.org/wiki/Load>

---

```
#!/bin/sh

case $1 in
    config)
        cat <<'EOF'
graph_title Load average
graph_vlabel load
load.label load
graph_category system
graph_scale no
EOF
        ;;
    *)
        awk '{print "load.value "$2}' /proc/loadavg
esac
```

---

**Tip**

Sehr ähnliches Beispiel auf: <http://munin.projects.linpro.no/wiki/HowToWritePlugins>

---

### 8.3 Testen

Plugin ausführen.

```
$ munin-run test_load
load.value 0.15
```

Plugin Konfiguration ausgeben.

---

```
$ munin-run test_load config
graph_title Load average
graph_vlabel load
load.label load
graph_category system
graph_scale no
```

---

**Tip**

SNMP ist sehr ähnlich: <http://munin.projects.linpro.no/wiki/HowToWriteSNMPPlugins>

---

## 8.4 Wildcard Plugin

- sehr ähnlich dem normalem Plugin
- Name endet auf \_ (Bsp.: if\_)
- Konfiguration mit passendem Name des Symlinks
  - Beispiel: if\_eth0, if\_eth1
- Plugin holt sich Informationen aus Plugin name

## 8.5 Multigraph plugins

- ab Munin 1.4
- Realisierung über multigraph Keyword

```
multigraph if_eth0
send.value 10
recv.value 10

multigraph if_eth1
send.value 20
recv.value 20
```

## 9 Munin Protokoll

---

**Note**

Life Demo zum Workshop. Zum Nachlesen: <http://munin.projects.linpro.no/wiki/network-protocol>

---

## 10 Links

- [Munin Projekt Seite](#)
-