

Administration GW2-Batman

Alles die Administration von [GW2.Batman](#) betreffend.

Es ist das Paket [etckeeper](#) mit Git als Backend installiert um den Vorteil einer Versionsverwaltung für /etc zu haben. Nach administrativen Aufgaben einfach kurz

```
sudo etckeeper commit
```

und kurz eine Commit Message eingeben. Das Repository ist ausschließlich lokal vorhanden und wird nicht nach draußen gepusht!

Administratoren

(aka Nutzer mit sudo Rechten)

- [tux](#)
- nold
- kwasir

Installierte Dienste

- YANIC
- NODE Exporter

Dienste Konfigurartion

YANIC

Install

Install GO

```
cd /usr/local/  
wget https://dl.google.com/go/go1.13.1.linux-amd64.tar.gz -O go-release-  
linux-amd64.tar.gz  
tar xvf go-release-linux-amd64.tar.gz  
rm go-release-linux-amd64.tar.gz
```

Export Path

```
export GOPATH=/opt/go
export PATH=$PATH:/usr/local/go/bin:$GOPATH/bin
```

Download and Build Yanic

```
go get -v -u github.com/FreifunkBremen/yanic
```

Copy Sample Config

```
cp /opt/go/src/github.com/FreifunkBremen/yanic/config_example.toml
/etc/yanic.conf
```

create needed folders & files

```
mkdir -p /var/lib/yanic
mkdir -p /var/www/html/meshviewer/data
touch /var/log/yanic.log
```

Create User and Group

```
sudo groupadd --system yanic
sudo useradd -s /sbin/nologin --system -g yanic yanic
```

Own this

```
sudo chown yanic /var/log/yanic.log /var/lib/yanic
/var/www/html/meshviewer/data
```

create Systemd service

```
cp /opt/go/src/github.com/FreifunkBremen/yanic/contrib/init/linux-
systemd/yanic.service /lib/systemd/system/yanic.service
systemctl daemon-reload
```

!Before Start edit /etc/yanic.conf Start yanic

```
systemctl start yanic
```

Enable Startup

```
systemctl enable yanic
```

Config File

RespondD Interface

```
[[respondd.interfaces]]
# name of interface on which this collector is running
```

```
ifname = "mesh-vpn"
# ip address which is used for sending
# (optional - without definition used a address of ifname - preferred link
local)
#ip_address = "fe80::1ce8:33ff:fe05:3f9f"
# disable sending multicast respondd request
# (for receiving only respondd packages e.g. database respondd)
#send_no_request = false
# multicast address to destination of respondd
# (optional - without definition used default ff05::2:1001)
multicast_address = "ff02::1"
# define a port to listen
# if not set or set to 0 the kernel will use a random free port at its own
port = 45123
```

Forward Respondd to "Master" Yanic

```
# respondd (yanic)
# forward collected respondd package to a address
# (e.g. to another respondd collector like a central yanic instance or
hopglass)
[[database.connection.respondd]]
enable    = true
# type of network to create a connection
type      = "udp"
# destination address to connect/send respondd package
address   = "10.11.12.2:11001"
```

Node Exporter

Download and Extract

```
curl -s
https://api.github.com/repos/prometheus/node_exporter/releases/latest \
| grep browser_download_url \
| grep linux-amd64 \
| cut -d '"' -f 4 \
| wget -qi -
tar xzf node_exporter-0.18.1.linux-amd64.tar.gz
cd node_exporter-0.18.1.linux-amd64/
sudo cp node_exporter /usr/local/bin/
```

Create Startup Script

```
sudo mkdir /opt/node_exporter
sudo vi /opt/node_exporter/node_exporter.sh
```

Content of Startup Script

```
#!/bin/sh
/usr/local/bin/node_exporter \
--collector.cpu \
--collector.diskstats \
--collector.filesystem \
--collector.loadavg \
--collector.meminfo \
--collector.filefd \
--collector.netdev \
--collector.stat \
--collector.netstat \
--collector.systemd \
--collector.uname \
--collector.vmstat \
--collector.time \
--collector.mdadm \
--collector.zfs \
--collector.tcpstat \
--collector.bonding \
--collector.hwmon \
--collector.arp \
--web.listen-address=:9100 \
--web.telemetry-path="/metrics"
```

Startup Script

```
sudo vi /etc/init.d/node_exporter

#!/bin/sh
### BEGIN INIT INFO
# Provides: node_exporter
# Required-Start: $local_fs $network $named $time $syslog
# Required-Stop: $local_fs $network $named $time $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Description:
### END INIT INFO
SCRIPT=/opt/node_exporter/node_exporter.sh
RUNAS=root
PIDFILE=/var/run/node_exporter.pid
LOGFILE=/var/log/node_exporter.log
start() {
if [ -f "$PIDFILE" ] && kill -0 $(cat "$PIDFILE"); then
echo 'Service already running' >&2
return 1
fi
echo 'Starting service...' >&2
local CMD="$SCRIPT &> \"$LOGFILE\" && echo \#! > $PIDFILE"
su -c "$CMD" $RUNAS > "$LOGFILE"
echo 'Service started' >&2
}
```

```
stop() {
if [ ! -f "$PIDFILE" ] || ! kill -0 $(cat "$PIDFILE"); then
echo 'Service not running' >&2
return 1
fi
echo 'Stopping service' >&2
kill -15 $(cat "$PIDFILE") && rm -f "$PIDFILE"
echo 'Service stopped' >&2
}
uninstall() {
echo -n "Are you really sure you want to uninstall this service? That
cannot be undone. [yes|No] "
local SURE
read SURE
if [ "$SURE" = "yes" ]; then
stop
rm -f "$PIDFILE"
echo "Notice: log file is not be removed: '$LOGFILE'" >&2
update-rc.d -f remove
rm -fv "$0"
fi
}
case "$1" in
start)
start
;;
stop)
stop
;;
uninstall)
uninstall
;;
retart)
stop
start
;;
*)
echo "Usage: $0 {start|stop|restart|uninstall}"
esac
```

Engage

```
sudo chmod +x /opt/node_exporter/node_exporter.sh
sudo chmod +x /etc/init.d/node_exporter
```

Changelog

From:

<https://wiki.netz39.de/> - **Netz39**

Permanent link:

<https://wiki.netz39.de/freifunk:server:gw2.batman>

Last update: **2020-04-15 19:56**

