

# README

Todd Wintermute

2023-12-21

## Contents

<b>1</b>	<b>toddwint/snmp</b>	<b>1</b>
1.1	Info . . . . .	1
1.2	Overview . . . . .	1
1.3	Features . . . . .	1
1.4	Sample commands to create the <code>macvlan</code> . . . . .	2
1.5	Sample <code>docker run</code> command . . . . .	2
1.6	Sample <code>docker compose</code> ( <code>compose.yaml</code> ) file . . . . .	3

## 1 toddwint/snmp

### 1.1 Info

`snmp` docker image for simple lab testing applications.

Docker Hub: <https://hub.docker.com/r/toddwint/snmp>

GitHub: <https://github.com/toddwint/snmp>

### 1.2 Overview

Docker image for receiving SNMP trap messages.

Pull the docker image from Docker Hub or, optionally, build the docker image from the source files in the `build` directory.

Create and run the container using `docker run` commands, `docker compose` commands, or by downloading and using the files here on github in the directories `run` or `compose`.

**NOTE: A volume named `upload` is created the first time the container is started. Modify the file in that directory with SNMP users and/or authentication and privacy passwords. Then restart the container.**

Manage the container using a web browser. Navigate to the IP address of the container and one of the HTTPPORTs.

**NOTE: Network interface must be UP i.e. a cable plugged in.**

Example `docker run` and `docker compose` commands as well as sample commands to create the `macvlan` are below.

### 1.3 Features

- Ubuntu base image
- Plus:
  - `fzf`
  - `iproute2`
  - `iputils-ping`
  - `libsnmp-dev`
  - `python3-minimal`

- snmp
- snmp-mibs-downloader
- snmptrapd
- tmux
- tzdata
- [ttyd](#)
  - ◊ View the terminal in your browser
- [frontail](#)
  - ◊ View logs in your browser
  - ◊ Mark/Highlight logs
  - ◊ Pause logs
  - ◊ Filter logs
- [tailon](#)
  - ◊ View multiple logs and files in your browser
  - ◊ User selectable **tail**, **grep**, **sed**, and **awk** commands
  - ◊ Filter logs and files
  - ◊ Download logs to your computer

## 1.4 Sample commands to create the macvlan

Create the docker macvlan interface.

```
docker network create -d macvlan --subnet=192.168.10.0/24 --gateway=192.168.10.254 \
  --aux-address="mgmt_ip=192.168.10.2" -o parent="eth0" \
  --attachable "snmp01"
```

Create a management macvlan interface.

```
sudo ip link add "snmp01" link "eth0" type macvlan mode bridge
sudo ip link set "snmp01" up
```

Assign an IP on the management macvlan interface plus add routes to the docker container.

```
sudo ip addr add "192.168.10.2/32" dev "snmp01"
sudo ip route add "192.168.10.0/24" dev "snmp01"
```

## 1.5 Sample docker run command

```
docker run -dit \
  --name "snmp01" \
  --network "snmp01" \
  --ip "192.168.10.1" \
  -h "snmp01" \
  -v "${PWD}/upload:/opt/snmp/upload" \
  -p "192.168.10.1:161:161/udp" \
  -p "192.168.10.1:161:161/tcp" \
  -p "192.168.10.1:162:162/udp" \
  -p "192.168.10.1:162:162/tcp" \
  -p "192.168.10.1:8080:8080" \
  -p "192.168.10.1:8081:8081" \
  -p "192.168.10.1:8082:8082" \
  -p "192.168.10.1:8083:8083" \
  -e TZ="UTC" \
  -e HUID="1000" \
  -e HGID="1000" \
  -e HTTPPORT1="8080" \
  -e HTTPPORT2="8081" \
  -e HTTPPORT3="8082" \
```

```
-e HTTPPORT4="8083" \  
-e HOSTNAME="snmp01" \  
-e APPNAME="snmp" \  
"toddwint/snmp"
```

## 1.6 Sample docker compose (compose.yaml) file

```
name: snmp01  
  
services:  
  snmp:  
    image: toddwint/snmp  
    hostname: snmp01  
    ports:  
      - "192.168.10.1:161:161/udp"  
      - "192.168.10.1:161:161/tcp"  
      - "192.168.10.1:162:162/udp"  
      - "192.168.10.1:162:162/tcp"  
      - "192.168.10.1:8080:8080"  
      - "192.168.10.1:8081:8081"  
      - "192.168.10.1:8082:8082"  
      - "192.168.10.1:8083:8083"  
    networks:  
      default:  
        ipv4_address: 192.168.10.1  
    environment:  
      - HUID=1000  
      - HGID=1000  
      - HOSTNAME=snmp01  
      - TZ=UTC  
      - HTTPPORT1=8080  
      - HTTPPORT2=8081  
      - HTTPPORT3=8082  
      - HTTPPORT4=8083  
      - APPNAME=snmp  
    volumes:  
      - "${PWD}/upload:/opt/snmp/upload"  
    tty: true  
  
networks:  
  default:  
    name: "snmp01"  
    external: true
```