

# README

Todd Wintermute

2023-11-03

## Contents

<b>1</b>	<b>toddwint/rtr</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/rtr

### 1.1 Info

`rtr` docker image for simple lab testing applications.

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

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

### 1.2 Overview

Docker image for a quick single physical interface router which can route multiple connected subnets plus static routes.

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 files in that directory. Specify connected interfaces in `addrs.csv` and static routes in `routes.csv`. 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:
  - `iputils-arping`
  - `tmux`
  - `python3-minimal`
  - `iproute2`

- `iputils-ping`
- `tzdata`
- `tttyd`
  - ◊ 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=169.254.255.240/28 --gateway=169.254.255.241 \
  --aux-address="mgmt_ip=169.254.255.253" -o parent="eth0" \
  --attachable "rtr01"
```

Create a management macvlan interface.

```
sudo ip link add "rtr01" link "eth0" type macvlan mode bridge
sudo ip link set "rtr01" up
```

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

```
sudo ip addr add "169.254.255.253/32" dev "rtr01"
sudo ip route add "169.254.255.240/28" dev "rtr01"
```

## 1.5 Sample docker run command

```
docker run -dit \
  --name "rtr01" \
  --network "rtr01" \
  --ip "169.254.255.254" \
  -h "rtr01" \
  -v "${PWD}/upload:/opt/rtr/upload" \
  -p "169.254.255.254:8080:8080" \
  -p "169.254.255.254:8081:8081" \
  -p "169.254.255.254:8082:8082" \
  -p "169.254.255.254:8083:8083" \
  -e TZ="UTC" \
  -e MGMTIP="169.254.255.253" \
  -e GATEWAY="169.254.255.241" \
  -e HUID="1000" \
  -e HGID="1000" \
  -e HTTPPORT1="8080" \
  -e HTTPPORT2="8081" \
  -e HTTPPORT3="8082" \
  -e HTTPPORT4="8083" \
  -e HOSTNAME="rtr01" \
  -e APPNAME="rtr" \
  --cap-add=NET_ADMIN \
  "toddwint/rtr"
```

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

```
name: rtr01

services:
  rtr:
    image: toddwint/rtr
    hostname: rtr01
    ports:
      - "169.254.255.254:8080:8080"
      - "169.254.255.254:8081:8081"
      - "169.254.255.254:8082:8082"
      - "169.254.255.254:8083:8083"
    networks:
      default:
        ipv4_address: 169.254.255.254
    environment:
      - MGMTIP=169.254.255.253
      - GATEWAY=169.254.255.241
      - HUID=1000
      - HGID=1000
      - HOSTNAME=rtr01
      - TZ=UTC
      - HTTPPORT1=8080
      - HTTPPORT2=8081
      - HTTPPORT3=8082
      - HTTPPORT4=8083
      - APPNAME=rtr
    privileged: true
    cap_add:
      - NET_ADMIN
    volumes:
      - "${PWD}/upload:/opt/rtr/upload"
    tty: true

networks:
  default:
    name: "rtr01"
    external: true
```