

# README

Todd Wintermute

2023-11-03

## Contents

<b>1</b>	<b>toddwint/rpc</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 . . . . .	2

## 1 toddwint/rpc

### 1.1 Info

rpc docker image for simple lab testing applications.

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

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

### 1.2 Overview

Docker image containing rpcbind and can be used as the base image for other images.

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`.

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:
  - rpcbind
  - libtirpc3
  - netbase
  - tmux
  - python3-minimal
  - iproute2
  - 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 "rpc01"
```

Create a management macvlan interface.

```
sudo ip link add "rpc01" link "eth0" type macvlan mode bridge
sudo ip link set "rpc01" 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 "rpc01"
sudo ip route add "192.168.10.0/24" dev "rpc01"
```

## 1.5 Sample docker run command

```
docker run -dit \
  --name "rpc01" \
  --network "rpc01" \
  --ip "192.168.10.1" \
  -h "rpc01" \
  -p "192.168.10.1:111:111/tcp" \
  -p "192.168.10.1:111:111/udp" \
  -e TZ="UTC" \
  -e HOSTNAME="rpc01" \
  -e APPNAME="rpc" \
  "toddwint/rpc"
```

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

```
name: rpc01

services:
  rpc:
    image: toddwint/rpc
    hostname: rpc01
    ports:
      - "192.168.10.1:111:111/tcp"
      - "192.168.10.1:111:111/udp"
    networks:
      default:
        ipv4_address: 192.168.10.1
    environment:
      - HOSTNAME=rpc01
      - TZ=UTC
      - APPNAME=rpc
```

```
tty: true
```

```
networks:
```

```
  default:
```

```
    name: "rpc01"
```

```
    external: true
```