README

Todd Wintermute

2023-12-21

Contents

1 toddwint/tcpdump

1.1 Info

`tcpdump` docker image for simple lab testing applications.

Docker Hub: https://hub.docker.com/r/toddwint/tcpdump

GitHub: https://github.com/toddwint/tcpdump

1.2 Overview

Docker image which runs tcpdump to capture network traffic and save to files timestamped at intervals set by the user via a configuration environment variable.

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 `captures` is created the first time the container is started.

Manage the container using a web browser. Navigate to the IP address of the container and one of the `HTTPPORT`s.

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
  - python3-minimal
  - tcpdump
  - 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 "tcpdump01"
```

Create a management macvlan interface.

```
sudo ip link add "tcpdump01" link "eth0" type macvlan mode bridge
sudo ip link set "tcpdump01" 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 "tcpdump01"
sudo ip route add "192.168.10.0/24" dev "tcpdump01"
```

1.5 Sample `docker run` command

```
docker run -dit \
    --name "tcpdump01" \
    --network "tcpdump01" \
    --ip "192.168.10.1" \
    -h "tcpdump01" \
    -v "${PWD}/captures:/opt/tcpdump/captures" \
    -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 ROTATE_SECONDS="300" \
    -e EXPRESSION="" \
    -e EXPRESSION_TYPE="" \
    -e HUID="1000" \
    -e HGID="1000" \
    -e HTTPPORT1="8080" \
    -e HTTPPORT2="8081" \
    -e HTTPPORT3="8082" \
    -e HTTPPORT4="8083" \
    -e HOSTNAME="tcpdump01" \
    -e APPNAME="tcpdump" \
    "toddwint/tcpdump"
```

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

```yaml
name: tcpdump01

services:
  tcpdump:
    image: toddwint/tcpdump
    hostname: tcpdump01
    ports:
        - "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:
        - ROTATE_SECONDS=300
        - EXPRESSION=
        - EXPRESSION_TYPE=
        - HUID=1000
        - HGID=1000
        - HOSTNAME=tcpdump01
        - TZ=UTC
        - HTTPPORT1=8080
        - HTTPPORT2=8081
        - HTTPPORT3=8082
        - HTTPPORT4=8083
        - APPNAME=tcpdump
    volumes:
      - "${PWD}/captures:/opt/tcpdump/captures"
    tty: true

networks:
    default:
        name: "tcpdump01"
        external: true
```