README

Todd Wintermute

2023-09-13

Contents

1 toddwint/ztp

1.1 Info

`ZTP` (Zero-Touch Provisioning) docker image for Juniper SRX345, SRX1500, ACX7024, EX2300, EX4100, and HPE Aruba 2930F devices.

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

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

For more detailed information, please view the `ZTP Instructions` files: `ZTP Instructions.md`, `ZTP Instructions.html`, or `ZTP Instructions.pdf`.

1.2 Overview

Docker image for performing Zero-Touch Provisioning of network devices.

- Supports the following devices:
  - Juniper SRX345
  - Juniper SRX1500
  - Juniper ACX7024
  - Juniper EX2300
  - Juniper EX4100
  - HPE Aruba 2930F

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 `ftp` is created the first time the container is started and contains default files. Modify these files with your information and restart the container.

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:
  - rsyslog
  - isc-dhcp-server
  - ftp
  - vsftpd
  - tftp-hpa
  - tftpd-hpa
  - webfs
  - bsdmainitils
  - fzf
  - tmux
  - python3-minimal
  - iputils-ping
  - 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=172.21.0.0/16 --gateway=172.21.255.254 \
    --aux-address="mgmt_ip=172.21.255.253" -o parent="eth0" \
    --attachable "eth0-macvlan"
```

Create a management macvlan interface.

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

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

```
sudo ip addr add "172.21.255.253/32" dev "eth0-macvlan"
sudo ip route add "172.21.0.0/16" dev "eth0-macvlan"
```

1.5 Sample `docker run` command

```
docker run -dit \
    --name "ztp01" \
    --network "eth0-macvlan" \
    --ip "172.21.255.252" \
    -h "ztp01" \
    -v "${PWD}/ftp:/opt/ztp/ftp" \
    -e TZ="UTC" \
    -e MGMTIP="172.21.255.253" \
```

```
    -e GATEWAY="172.21.255.254" \
    -e HUID="1000" \
    -e HGID="1000" \
    -e HTTPPORT1="8080" \
    -e HTTPPORT2="8081" \
    -e HTTPPORT3="8082" \
    -e HTTPPORT4="8083" \
    -e HOSTNAME="ztp01" \
    -e APPNAME="ztp" \
    "toddwint/ztp"
```

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

```
name: ztp01

services:
  ztp:
    image: toddwint/ztp
    hostname: ztp01
    ports:
        - "172.21.255.252:8080:8080"
        - "172.21.255.252:8081:8081"
        - "172.21.255.252:8082:8082"
        - "172.21.255.252:8083:8083"
    networks:
        default:
            ipv4_address: 172.21.255.252
    environment:
        - HUID=1000
        - HGID=1000
        - HOSTNAME=ztp01
        - TZ=UTC
        - MGMTIP=172.21.255.253
        - GATEWAY=172.21.255.254
        - HTTPPORT1=8080
        - HTTPPORT2=8081
        - HTTPPORT3=8082
        - HTTPPORT4=8083
    privileged: true
    cap_add:
      - NET_ADMIN
    volumes:
      - "${PWD}/ftp:/opt/ztp/ftp"
    tty: true

networks:
    default:
        name: "eth0-macvlan"
        external: true
```