ZTP Instructions

Todd Wintermute

2023-08-26

Contents

1 Overview

1.1 What is ZTP

ZTP stands for Zero-Touch Provisioning. It is a way of provisioning telecommunications devices without the need of manual intervention.

1.2 What is this software package?

This tool is a Docker image and set of GitHub files.

The Docker image contains a server application which can be used in two different methods to provision devices including update the Operating System and/or load a configuration file.

The GitHub files contain scripts used to make managing the Docker image easier as well as the source code of the Docker image.

1.3 Who is this for?

This tool if for anyone who has a need to update the operating system and/or load a configuration on a factory default Juniper or Aruba networking device.

1.4 What PC platforms are supported?

This ZTP Docker image is designed to work using a Linux operating system. Specifically, it was designed and tested on Ubuntu 22.04 Jammy Jellyfish.

It is also required to install `Docker`. It is recommended to have `Git` installed, but not required. On many Linux systems `Git` is already installed. For instructions on installing Docker, please see the section Install Docker.

The reason Windows and MacOS are not supported is because this application makes use of the Docker `MACVLAN` network feature which Docker only supports on Linux. If you would like to try to get it to work on Windows or MacOS, you might have some success using Docker `host networking`. For more information about the different types of Docker network drivers see this Docker page. Another option is running a Linux Virtual Machine on these platforms. See the section Running inside a Virtual Machine

1.5 How does it work?

First, the device vendor has to include the option for provisioning their devices via ZTP. Juniper and Aruba do. So do other vendors like Cisco, but this tool only includes the DHCP options and code for Juniper and Aruba.

When the ZTP capable device boots up from a factory default configuration, it looks for a DHCP message which includes certain options telling it the location of the file server and the file names it needs to download. It can also include the transfer protocol to use such as ftp or tftp. It uses these options to download the files. Then it installs these files all by itself.

This Docker image contains a DHCP, FTP, and TFTP server as well as scripts to put everything together. It contains additional applications and scripts so that you can view the transfers in your web browser and monitor the progress.

Be aware, although a file transfer might be complete, the device might still need 15 or 20 minutes to finish installing.

So it sounds interesting. You want to try it out, but how do you get the information into ZTP? Well, there are two methods. . .

1.5.1 Two provisioning methods

This tool supports two different methods to provision devices.

1. DHCP Vendor Class Identifiers
    - `vendor_class_defaults.csv`
2. MAC addresses
    - `ztp.csv`

Samples of each of these files are created and put in the `ftp` folder/volume the first time the container is started.

1.5.1.1 Provisioning method 1: Vendor Class Identifier

Devices send a vendor class identifier string in DHCP requests to the DHCP server. This string can be used to provide the OS and/or configuration file to that specific type of device.

This option is a good idea if you want to load the same OS and/or a starting configuration to many of the same device types.

On Aruba devices you can obtain this string with the command `show dhcp client vendor-specific`.

On Juniper devices, a zeroized device or device with factory defaults should have this string listed in the configuration under the interfaces section on the interfaces configured for DHCP.

To use this method, fill out the information in `vendor_class_defaults.csv`

1.5.1.1.1 Vendor Class ID file: `vendor_class_defaults.csv`

A default file is created and put in the `ftp` folder/volume the first time the container is started.

You can edit it using a text editor or using a program like Microsoft Excel or LibreOffice Calc. Just remember to save it as a CSV file as it might prompt you to save it in a different format.

Here is a look at the example `vendor_class_defaults.csv`:

```
hardware,vendor_class_string,os,config
srx345,Juniper-srx345,,
srx1500,Juniper-srx1500,,
2930f,Aruba JL253A 2930F-24G-4SFP+ Switch dslforum.org,,
2930f,Aruba JL254A 2930F-48G-4SFP+ Switch dslforum.org,,
2930f,Aruba JL255A 2930F-24G-PoE+-4SFP+ Switch dslforum.org,,
2930f,Aruba JL256A 2930F-48G-PoE+-4SFP+ Switch dslforum.org,,
2930f,Aruba JL260A 2930F-48G-4SFP Switch dslforum.org,,
2930f,Aruba JL263A 2930F-24G-PoE+-4SFP+-TAA Switch dslforum.org,,
2930f,Aruba JL264A 2930F-48G-PoE+-4SFP+-TAA Switch dslforum.org,,
```

Or in an human viewable way

hardware	vendor_class_string	os	config
srx345	Juniper-srx345		
srx1500	Juniper-srx1500		
2930f	Aruba JL253A 2930F-24G-4SFP+ Switch dslforum.org		
2930f	Aruba JL254A 2930F-48G-4SFP+ Switch dslforum.org		
2930f	Aruba JL255A 2930F-24G-PoE+-4SFP+ Switch dslforum.org		
2930f	Aruba JL256A 2930F-48G-PoE+-4SFP+ Switch dslforum.org		
2930f	Aruba JL260A 2930F-48G-4SFP Switch dslforum.org		
2930f	Aruba JL263A 2930F-24G-PoE+-4SFP+-TAA Switch dslforum.org		
2930f	Aruba JL264A 2930F-48G-PoE+-4SFP+-TAA Switch dslforum.org		

And here is a description of each column:

- hardware

  - A unique string that identifies the specific make and model of the device. This string is used to determine whether the device is Juniper or Aruba and use the appropriate DHCP option codes as well as the transfer protocol for the device i.e. ftp or tftp.
- vendor_class_string
  - The vendor-id the device(s) will provide to the DHCP server in a DHCP request.
- os_image
  - The operating system file for the device(s). Junipers usually use `.tgz` and Aruba usually uses `.swi` files.
- config_file
  - The configuration file for the device(s). Note that Aruba devices must include a specific string in the header as well as the correct module. Also, Aruba LAN switches will continue to perform ZTP on themselves unless the following commands are added to the configuration file: `no dhcp config-file-update` and `no dhcp image-file-update`

1.5.1.2 Provisioning method 2: MAC addresses

The device's MAC address is a unique identifier which can be used to provide the OS and/or configuration file to that specific device.

This option is a good idea if you want to load a specific configuration and OS per device.

The MAC addresses can be obtained from the rear sticker on the device. An affordable Bluetooth barcode scanner from Amazon.com and a Google Sheet open on your phone can be very handy in making this task a lot easier. Another good option is using a smartphone barcode scanner application.

To use this method, fill out the information in `ztp.csv`.

1.5.1.2.1 MAC method file: `ztp.csv`

A default file is created and put in the `ftp` folder/volume the first time the container is started.

You can edit it using a text editor or using a program like Microsoft Excel or LibreOffice Calc. Just remember to save it as a CSV file as it might prompt you to save it in a different format.

Here is a look at the example `ztp.csv`:

```
hardware,mac,os,config
2930f,888888eeeee1,aruba-2930f.swi,switch1.cfg
ex2300,888888eeeee2,junos-ex2300.tgz,switch2.cfg
ex4100,888888eeeee3,junos-ex4100.tgz,switch3.cfg
srx1500,888888eeeee4,junos-srx1500.tgz,router1.cfg
srx345,888888eeeee5,junos-srx345.tgz,router2.cfg
acx7024,888888eeeee6,junos-acx7024.tgz,router3.cfg
```

Or in an human viewable way

hardware	mac	os	config
2930f	888888eeeee1	aruba-2930f.swi	switch1.cfg
ex2300	888888eeeee2	junos-ex2300.tgz	switch2.cfg
ex4100	888888eeeee3	junos-ex4100.tgz	switch3.cfg
srx1500	888888eeeee4	junos-srx1500.tgz	router1.cfg
srx345	888888eeeee5	junos-srx345.tgz	router2.cfg
acx7024	888888eeeee6	junos-acx7024.tgz	router3.cfg

And here is a description of each column:

- hardware
  - A unique string that identifies the specific make and model of the device. This string is used to determine whether the device is Juniper or Aruba and use the appropriate DHCP option codes as well as the transfer protocol for the device i.e. ftp or tftp.
- mac

  - The MAC address of the device. Specifically the MAC obtained from the rear sticker. This tool includes an increment amount for some devices to increment the base MAC of the device and configure the DHCP server for the MAC address of the first port on the device.
- os_image
  - The operating system file for the device(s). Junipers usually use `.tgz` and Aruba usually uses `.swi` files.
- config_file
  - The configuration file for the device(s). Note that Aruba devices must include a specific string in the header as well as the correct module. Also, Aruba LAN switches will continue to perform ZTP on themselves unless the following commands are added to the configuration file: `no dhcp config-file-update` and `no dhcp image-file-update`

1.5.1.3 How to specify which provisioning method you want to use

Both methods will be checked when the application starts, and they can both be used at the same time.

There is a default precedence to choose the more specific entry over the more generic ones. That means if you specify a configuration file using the MAC method and the Vendor Class method, the file listed in the MAC method (`ztp.csv`) will be loaded to the device.

If you only want to use one method, do not specify the OS file, configuration file, etc. in the other file and it will not be used.

1.5.2 Supported devices

Currently only Juniper and Aruba are supported.

Not all Juniper and Aruba devices are supported either.

The supported device models are :

1. Aruba 2930F LAN switches
   - JL253A
   - JL254A
   - JL255A
   - JL256A
   - JL260A
   - JL263A
   - JL264A
2. Juniper SRX345
3. Juniper SRX1500
4. Juniper AXC7024
5. Juniper EX2300
6. Juniper EX4100

1.5.3 Expanding the devices supported

If your device is not in the list, it still might be possible to use this tool on it.

However, only DHCP codes for Juniper and Aruba are supported by this tool.

1.5.3.1 Viewing supported devices or expanding the list: `supported_device_models.json`

You can view the currently supported devices by opening the file `supported_device_models.json`

It looks like this:

```
{
 "srx345": {
  "vendor": "juniper",
  "incr_mac": 1
 },
 "srx1500": {
  "vendor": "juniper",
```

```
  "incr_mac": null
 },
 "acx7024": {
  "vendor": "juniper",
  "incr_mac": 1023
 },
 "2930f": {
  "vendor": "aruba",
  "incr_mac": null
 },
 "ex2300": {
  "vendor": "juniper",
  "incr_mac": null
 },
 "ex4100": {
  "vendor": "juniper",
  "incr_mac": null
 }
}
```

If you have a Juniper or Aruba device not in the list, you can try adding it to the list. Chances are it will work.

Make sure to add the item in correct `json` format including the comma after the previous entry's `}` (close curly brace).

Let's look at one entry and explain what the fields do.

```
 "srx345": {
  "vendor": "juniper",
  "incr_mac": 1
 },
```

- "srx345"
  - The unique model for the device. Use this value in the `hardware` column in `ztp.csv` and `vendor_class_defaults.csv`.
- "vendor": "juniper"
  - Choices are either `juniper` or `aruba`. Tells the application which DHCP codes to use and which transfer protocol to use (ftp vs tftp).
- "incr_mac": 1
  - Tells the application if it needs to increment the MAC address entered (usually scanned from the rear sticker) for the specific Ethernet port on the device being used.
    - ◇ On Juniper srx345 this means enter the MAC from the sticker on the back and use port `ge-0/0/0`.

1.5.4 Where to plug in the cables

Use the following ports on the devices when connecting to the ZTP application.

1. Aruba 2930F LAN switches
   - any port
2. Juniper SRX345
   - `ge-0/0/0`
3. Juniper SRX1500
   - `ge-0/0/0`
4. Juniper AXC7024
   - `MGMT` port
5. Juniper EX2300
   - any port
6. Juniper EX4100
   - any port

1.5.5 Files and folders inside the `ftp` folder/volume

- vendor_class_defaults.csv
  - See Provisioning method 1: Vendor Class Identifier
- ztp.csv
  - See Provisioning method 2: MAC addresses
- supported_device_models.json
  - See Expanding the devices supported
- csv_filter.py
  - A python script which can be used to create additional CSV files filtered by make and model. Run `csv_filter.py -h` for more info.
- os_images/
  - The folder where the operating system files should be placed.
- config_files/
  - The folder where the configuration files should be placed.
- .exists
  - This file is added automatically and tells ZTP not to override the files inside the `ftp` folder. Erase this file and restart the container to receive a fresh copy of all the files.

2 Running ZTP

2.1 Install the requirements

2.1.1 Install Docker

On a Linux computer, install Docker.

If you are using Ubuntu Linux, follow these instructions:

Docker.com: Install Docker Engine on Ubuntu

To run Docker without requiring root, follow the following instructions:

Docker.com: Manage Docker as a non-root user

These steps can also be found in this document near the end in Installing Docker on Ubuntu Linux

2.1.2 Install git (optional, but recommended)

Git is usually already installed.

For instructions for specific platforms, see GitHub's page on installing git: GitHub.com: Install Git.

For Ubuntu, see the section Install git on Ubuntu Linux

2.2 Download the files

The Docker image contains everything to run the application.

Download the Docker image with the following command:

```
docker pull toddwint/ztp
```

To verify the image has downloaded please view the section Download the ZTP Docker image.

The GitHub files are optional, but recommended as it makes it easier to run and manage.

Download the GitHub project to the current folder with the following command:

```
git clone https://github.com/toddwint/ztp
```

There are two main folders in the GitHub project. Here is a brief description:

- build

  - Contains the source code and Docker files which is everything you need if you want to build the application offline.
  - You are also free to modify the code and tailor it to your own needs.
  - Most users won't need these files. They can be discarded.
- run
  - Contains files to assist in creating and managing the container.
  - I recommend renaming this folder to the hostname of your ZTP server (as configured in `config.txt`) such as `ztp01` and moving it to a location on your computer which is easy to access.

NOTE: If you do not require the `ztp/build` folder and files, I recommend reorganizing the folder structure as such:

- Remove the directory `ztp/build`
- Rename the `ztp/run` folder (e.g. `ztp01`)
- Create a docs folder and copy the `README` and `ZTP Instructions` files to it.

For example commands, see the section Download the GitHub files.

2.3 Create the Docker container

All the files from here on will be found in the `run` folder.

2.3.1 Modify container configuration parameters: `config.txt`

Before creating the container, it is important to review the default settings in `config.txt`. Each option has a description above it. Most of the defaults should be fine.

You will probably want to update the timezone. This will make the syslog messages match your local timezone. That section looks like this:

```
# To get a list of timezones view the files in `/usr/share/zoneinfo`
TZ=UTC
```

As an example, for my location I enter `America/Chicago`.

The most important variable is `INTERFACE`. Record the name of your Ethernet adapter in Linux and change the variable in this section:

```
# The interface on which to set the IP. Run `ip -br a` to see a list
INTERFACE=eth0
```

NOTE: The script will take care of setting the IP address on your Ethernet adapter so that you can communicate with the Docker container. It is not recommended to set an IP on your Ethernet adapter, and it can actually cause issues.

2.3.2 Run the script to create the container: `create_container.sh`

After configuring the options in `config.txt`, you are ready to create the container.

To create the container run the following command:

```
./create_container.sh
```

The script does a few things more than just create the container. It sets up some environmental variables and loads the parameter in `config.txt` so that those parameters can be passed to the container and also uses those parameters to create a new `MACVLAN` network adapter on your computer. Then it creates the container. Finally it creates an HTML launch page for the application by modifying a template and prompts the user to open this HTML file.

At this point, the ZTP Docker image you downloaded is now a Docker container instance. Also, the folder `ftp` should exist which is the Docker volume for this container.

Since at this point the container won't be doing anything useful as it does not have *your* information, I recommend stopping the container.

Run this command:

```
./stop.sh
```

Inside of the newly created `ftp` folder, you should see the files as explained already in Files and folders inside the `ftp` folder/volume. Modify these files with your information.

Don't forget to copy your OS and configuration files for your devices to `ftp/os_images/` and `ftp/config_files/`

Once you have modified your provisioning files and copied the OS and/or configuration files for your devices to the appropriate folders, it is time to see it work.

2.4 Restart the container

Now that we have everything set up for our devices, it is time to start the container, and let it provision all your devices.

Run this command:

```
./start.sh
```

If you did not stop the container from before, run this command:

```
./restart.sh
```

The container starts. If the cables are plugged in and your devices powered on with a factory default configuration, it should be provisioning them.

2.5 View the progress

To view the progress, open one of the links in the `webadmin.html` page. This page should have opened by default when you ran `create_container.sh`. If you closed the page or if it never opened, simply click on `webadmin.html`

Inside of the `webadmin.html` launch page, there are several options. Here is a description of each option:

- ttyd
  - A view only page that displays the file transfer report.
- ttyd (w/tmux)
  - An interactive terminal with multiple windows.
    - ◇ Window 1 displays the file transfer report.
    - ◇ Window 2 displays the log file.
    - ◇ Window 3 displays the completed ftp transfers log
    - ◇ Window 4 displays the log file filtered for ftp traffic
    - ◇ Window 5 displays the log file filtered for tftp traffic
    - ◇ Window 6 is access to the terminal.
      - You can run commands from inside the Docker container. Alternatively, you can press `CTRL-C` on any other windows to gain access to the terminal.
      - There are several scripts available in the starting directory (`debug`). Run `ls` to see them. Run them by typing `./` followed by the script name.
- frontail
  - Displays the log file and has filtering capability.
- tailon
  - Displays multiple log files (using the drop down) and also includes filtering capability.
- webfs
  - You can view the files in the `ftp` folder and download them.

You can also interact with the Docker container by using the scripts inside the `exec` folder. Here is a description:

- `bash.sh`
  - Access to the terminal.
    - ◇ You can run commands from inside the Docker container. Alternatively, you can press `CTRL-C` on any other windows to gain access to the terminal.
    - ◇ There are several scripts available in the starting directory (`debug`). Run `ls` to see them. Run them by typing `./` followed by the script name.

- `tail.sh`
  - Displays the log file.
- `tmux.sh`
  - The same as `ttyd (w/tmux)`, but in the command line instead of the web browser.
- `transfer_report.sh`
  - Displays the file transfer report.

2.6 Stop the container

Hopefully, your devices are loaded now. If not, skip ahead to the section Common issues.

To stop the container, run this command:

```
./stop.sh
```

If you are finished with the container, run this command:

```
./delete_container.sh
```

Both of these commands will stop the container and copy a timestamped version of the transfer report to your `ftp` folder. Deleting the container will additionally delete the container and remove the networking adapter created with the `create_container.sh` script and the `webadmin.html` file.

Do not be afraid of deleting the container. The files in the `ftp` folder will not be overwritten the next time the container is created as it checks for a `.exists` file inside that folder which is created the first time the container is created. I actually hardly ever use the `stop.sh`, `restart.sh`, or `start.sh` scripts. Instead, I use `create_container.sh` and `delete_container.sh`.

3 Common issues

3.1 Links in `webadmin.html` display the message *"The connection has timed out"* or *"This Site Can't Be Reached"*

Check that your Ethernet adapter link's status is *UP* or connect your Ethernet adapter to a network device.

Because of the way `MACVLAN` works, you cannot access the host directly. To work around this, a second `MACVLAN` interface is created on the same Ethernet adapter and a route to the Docker container is added. However, if that physical Ethernet adapter's link status is *DOWN*, then it will not use that adapter. Rather than having a cable plugged in to view the management page, another option is to purchase a USB-to-Ethernet adapter with a built in 4-Port Ethernet switch. See Running inside a Virtual Machine for more information.

3.2 You receive errors when running `start.sh`, `restart.sh`, `stop.sh`

This could be because the network no longer exists. Did your computer get restarted? The network adapter won't persist across a reboot.

To resolve this issue, try running `delete_container.sh` followed by `create_container.sh`

3.3 You receive errors when running `create_container.sh` or `delete_container.sh`

- Is the name of your Ethernet adapter in `config.txt` correct?
- Do you already have a container running? Run `./is_running.sh` to see.
- Did you change the hostname in `config.txt`? If it differs from the currently running hostname, it won't be able to delete it until you set it back or use Docker commands.

3.4 My Aruba device says *"Connection Refused"*.

Is the OS already at the version you are trying to load?

If so, Aruba LAN switches will try to download the file 5 or 6 times before moving onto the configuration file.

3.5 My device doesn't appear to be loading

- Verify the OS file is for the correct device and the file is not corrupt. Try loading it manually to verify it is correct.
- Verify the configuration file is valid by trying to load it manually.
- Did you create a network storm? Connecting too many switches together can create a network loop where the switch utilization goes to 100% and the network is no longer accessible.
- Did you add the MAC address to the list? Some times new devices get racked, but they don't automatically enter themselves into your files.
- Is the Vendor Class ID string correct?
- Is the device restored to factory defaults?
  - Juniper: `request system zeroize`
  - Aruba: `erase startup-config`
- Verify you are using the correct port on the device. See Where to plug in the cables
- If your files contain spaces in their names, try renaming them without spaces.
- Did you specify the full filename including the file extension? If you forgot, the script will search the directory for files with the same name and pick the first one it finds. It also searches for duplicate MAC addresses and configuration files. It will write messages to the log file for all the issues it finds.
- Use the log file to gather more information. See View the progress for more information about viewing the log file.

4 Additional Information

4.1 Links to Docker and GitHub pages

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

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

4.2 Running inside a Virtual Machine

Using a virtual machine is a good way to use this application if you currently run Windows or MacOS.

Configure your networking to `bridge` mode will yield the best results. However, I suggest using a USB-to-Ethernet adapter and attaching it directly to your virtual machine. Even better, *Cable Matters* makes a USB-to-4-Port-Gigabit-Ethernet-Switch in both standard USB type A and USB type C connectors. The nice thing about these adapters (other than having 4 Ethernet ports directly connected to your laptop) is that it has an internal switch which means the Ethernet interface will always show link status of *UP*.

4.3 Installing Docker on Ubuntu Linux

Below are the commands taken from Docker's website (here and here) how to install Docker on Ubuntu Linux.

1. Update the `apt` package index and install packages to allow `apt` to use a repository over HTTPS:

```
sudo apt-get update
sudo apt-get install -y ca-certificates curl gnupg lsb-release
```

2. Add Docker's official GPG key:

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \
  sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

3. Set up the repository:

```
echo \
  "deb [arch=$(dpkg --print-architecture) " \
  "signed-by=/etc/apt/keyrings/docker.gpg] " \
  "https://download.docker.com/linux/ubuntu \
```

```
  $(lsb_release -cs) stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

4. Update the apt package index:

```
sudo apt-get update
```

5. Install Docker Engine, containerd, and Docker Compose.

```
sudo apt-get install -y \
    docker-ce \
    docker-ce-cli \
    containerd.io \
    docker-compose-plugin
```

6. Add your user to the `docker` group to manage Docker as a non-root user

```
sudo usermod -aG docker $USER
```

7. Log out and log back in so that your group membership is re-evaluated.

4.4 Install git on Ubuntu Linux

Use the following command to install `git` on Ubuntu Linux:

```
sudo apt update
sudo apt install git
```

4.5 Download the ZTP Docker image

Download the Docker image to your Docker image files folder with the following command:

```
docker pull toddwint/ztp
```

The docker image downloads to your computer. Verify with the following command:

```
docker image ls
```

Example output:

```
REPOSITORY              TAG       IMAGE ID       CREATED         SIZE
toddwint/ztp            latest    dfe47ddf7ccf   5 weeks ago     150MB
```

4.6 Download the GitHub files

Download the GitHub project to the current folder with the following command:

```
git clone https://github.com/toddwint/ztp
```

For users that want to run the docker image and never build it from source, the following commands can be used to remove the source files make the directories easier to mange:

```
mv ztp/run/ ztp01
mkdir ztp01/docs
mv ztp/@(README|ZTP Instructions)* ztp01/docs/
rm -rf ztp/
```

4.7 Give the current user access to `ip` command without requiring a password (optional)

For Ubuntu Linux, add these commands if you do not wish to enter a password when the container starts and creates the network.

```
sudo touch /etc/sudoers.d/ip_cmd
sudo chmod 0440 /etc/sudoers.d/ip_cmd
sudo tee /etc/sudoers.d/ip_cmd > /dev/null << EOF
Cmnd_Alias IP_CMD=/usr/sbin/ip,/usr/bin/link,/usr/sbin/route
$USER   ALL=(ALL:ALL) NOPASSWD:IP_CMD
%$USER  ALL=(ALL:ALL) NOPASSWD:IP_CMD
EOF
sudo visudo -c
```

4.8 Updating to a new version

If you wish to download the latest Docker container and GitHub files, perform the following steps:

1. Stop and delete the current container.
   - You can view all the currently running Docker containers by running the command `docker ps` or `docker container ls`
2. Remove the current Docker file image and download the latest one with these commands:
   - You can view all the Docker file images on your system by running the command `docker image ls`

```
docker rmi toddwint/ztp
docker pull toddwint/ztp
```

3. Delete the current `ztp` directory, and download the latest ones. Don't forget to copy any important files elsewhere before deleting it. Here are some example commands to do that:

```
rm -rf ztp/
git clone https://github.com/toddwint/ztp
```

4.9 Running more than one container at a time.

It is possible to run more than one container at a time. To do so follow these steps.

- Make a copy of the `run` folder which I often rename to my container hostname such as `ztp01`.
- In `config.txt` change the Ethernet adapter to a new name (you won't be able to use the same Ethernet adapter) and the hostname. Also, change the IP addresses. I would increment the 2nd octet.

4.10 Changing the default IP scheme

Different size subnets should be fine (/16, /20, /22, etc). Don't put anything smaller than a `/28` network.

The image will reserve the last 4 IPs for the container IP, management host IP, spare IP, and the gateway IP. It will reserve 2 IPs lower than that range for the DHCP range of unknown hosts, but after running the python script will expand that to the IP after your last device.

If you do want to use the first IPs in the subnet range, it should be fine. It will check and skip the static DHCP assignments if those IPs are set.

4.11 Renaming predictable network interface names to user defined values

Predictable network interface names look like `ens33`, `eno1`, `enp1s0`, `enx78e7d1ea46da`

If you wish to change this behavior and rename them to names like `lan0` or `lan1`, follow these steps.

NOTE: Do not use kernel reserved names like `eth0` or `wlan0`. Those names are reserved. If you run into issues, that is why.

In this example the Ethernet adapter name is: `eno1`. Replace `eno1` with the name of your adapter.

Example: `eno1`

```
INTF=eno1
```

Find the pci path of the device. Use `grep` to filter for `ID_PATH` and awk to grab the value.

```
udevadm info /sys/class/net/$INTF | grep ID_PATH= | awk -F= '{print $2}'
```

Sample output:

```
pci-0000:03:00.0
```

Save the ID_PATH value to a variable.

```
IDPATH=$(udevadm info /sys/class/net/$INTF | grep ID_PATH= | awk -F= '{print $2}')
```

Decide a new name for the interface and save that name to a variable.

Example: `lan1`

```
INTFNAME=lan1
```

Run the following command to create the configuration file using the variables created in the previous steps.

```
sudo tee /etc/systemd/network/10-${INTFNAME}.link > /dev/null << EOF
[Match]
Path=$IDPATH
[Link]
Name=$INTFNAME
EOF
```

Reboot and verify the names.

Don't see the change? Verify which rule is being used with:

```
udevadm info /sys/class/net/eno1
```

If it looks correct, then perform:

```
sudo update-initramfs -u
```

Reboot and check again.