Anson Learns Docker

I'm learnding about whales

- [Baby's 1st Steps](#)

Baby's 1st Steps

Let's Learn Docker

Phil likes Docker and it's something useful to learn. And LastPass is taking away my free service so I need to DIY a password manager. Here's some notes

References

- General Docker Documentation
- Docker run
- Docker compose
- bitwardenrs wiki

Bitwardenrs as the jumping off point

Bitwarden basic commands - Imperative vs Declarative

Let's use this command to install bitwardenrs but also as a way to learn basic docker stuff

```
sudo docker run -d --name bitwarden -v ~/bwdata/:/data/ --restart=always -p 80:80 bitwardenrs/server:latest
```

- `docker` = command like sudo
- `run` = verb - it provides actions and creates the container by downloading an image and then tries to start it. docker has command options so the -X options. For example the -zavh options you specified in rsync
- `-p 8999:80` = port map from container (docker)to the host (server aka rpi). so 8999 is my IRL port that is on my network. I'd port forward it on the router if i wanted to expose the service to the actual internet. I don't have to do this. 80 is the port the container thinks it's seeing.

- `-v urhostfolder:urcontainerfolder` = volume mount aka what folders are you using. Like the port map, you setup a folder path location in the host (`~/docker` for example) and then a folder path location for in the new container.
- `restart=always` = this sets it so the container restarts whenever it goes down. Could be because of power outage, restarting the server, etc.
  - If I want to stop the container myself, I'd run `docker stop`
- `--name bitwarden` = this is the name of hte container. it's there for me to reference later
- `:latest` = this specifies the version of the image I'm looking for. in this case, i want the latest bitwardenrs

So I could run docker with all these options but there's another way. Above is the imperative way of running docker. So I'm telling docker EXACTLY what I want it to do. I could also run things declarative. So I could tell docker what I'd like it to do and have docker fill in the details. See this link for more info. So think C vs Python. That terminology is lifted from programming I guess.

Let's try it again but now using `docker-compose`.

1. Install docker-compose
   - For normal people: `sudo curl -L "https://github.com/docker/compose/releases/download/1.28.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose`
   - For RPi: `sudo curl -L --fail https://raw.githubusercontent.com/linuxserver/docker-docker-compose/master/run.sh -o /usr/local/bin/docker-compose`
2. Create a docker-compose.yaml in `/usr/local/bin/docker-compose`
   - In my case I'd be setting up for bitwarden. Put this in the folder you've setup for all docker info to live. for me it's ~/docker

```
services:
  bitwarden:
    image: bitwardenrs/server:latest
    volumes:
      - <hostfolder>:<containerfolder>
    ports:
      - "8999:80"
```

3. Run `docker-compose up -d` in `/usr/local/bin/docker-compose`
4. Hopefully you did the thing! In my case, I have bitwarden up on `<ur-rpi-ip-here>:8999`. If you install and run docker on the same machine (I'm running it through my server), then you could use `localhost/#/`

Need to setup ssl authentication - SWAG server

So I can't use bitwarden as is because it needs SSL. So now I need to setup a SWAG SSL authentication server so I don't have to worry about this again.

First problem is that Nextcloud is using port 443. This is what SWAG will use for the DNS authentication so I need to move the Nextcloud server to a different port.

1. Run `sudo netstat -apn | grep LIST` to find all ports being used and pick a port under 20000 that isn't used. I chose 8888 for nextcloud.
2. Port forward port 8888 in the router.
   - For me it's 192.168.0.1 -> Port Forwarding -> Create IPV4 -> leave the left side alone and right side put in the RPi's internal network IP address (`ipconfig` to find out) and then 8888 in the two boxes below
3. Edit these three files:
   - `sudo vim /etc/apache2/sites-enabled/000-default.conf` - make the first line
   - `sudo vim /etc/apache2/ports.conf` - comment out `Listen 80`, add in a line `Listen 8888` underneath, set the other two listen to 1443 ports
   - `sudo vim /etc/apache2/sites-enabled/nextcloud.conf` - update second line
4. Run `sudo systemctl restart apache2` to restart apache2
5. Run `docker restart swag`
6. See if Nextcloud is up and running again

Now install your service with docker-compose.

1. Create a docker-compose.yaml file
2. Edit this yaml file using this as an example

```
services:
  bitwarden:
    image: bitwardenrs/server:latest
    volumes:
      - /data:/data
    ports:
      - "8999:80"
    restart: unless-stopped
```

3. Save and run `docker-compose up -d`
4. You did it!

Now we can get the docker subdomain working from the proxy confs folder

1. Look for all the subdomains available: `sudo ls /home/tactilezine/docker/swag/nginx/proxy-confs/`
2. You should see file names like `nextcloud.subdomain.conf.sample`
3. To get the subdomains working, change the file from `.conf.sample` to `.conf`
4. Then edit the file `sudo vim /home/tactilezine/docker/swag/nginx/proxy-confs/nextcloud.subdomain.conf`

   - find the line starting with `$upstream_app` and then put in your host ip
   - find the line starting with `$upstream_port` and change the port to `1443`
   - find the line starting with `$upstream_porto` and change it to `https`

5. Run `docker restart swag` and it should now yield your service and subdomain!
6. Go to cloudflare or where ever you manage subdomains and add in a new CNAME for subdomain. In my case it was bitwarden under Name and tactilezine.xyz under Content.