Splunk Project - Tracking System Logs

By Jack Walton

1. Make a website and host it

a. I chose to make a Flask website that colorizes black and white images using machine learning. Currently there is no db.
b. The backend is a DigitalOcean Ubuntu droplet with 2 GB Memory / 2 Premium Intel vCPUs / 25 GB Disk (Probably can downsize once the website is complete)
c. Used docker to containerize the Flask application and its dependencies. (This is why I initially chose a server with more resources)
d. Left open ports 80 (http which nginx redirects to https), 443 (https), 22 (ssh) and gave Nginx full access to all ports. See below
e. Used nginx for the reverse proxy, Namecheap for the domain name and certs, and LetsEncrypt's 'certbot' tool for encrypting traffic over TLS.
f. Download the Splunk Universal Forwarder and set up a monitor for the directories of logs desired to track. Here's some docs. Or look here at some examples for this case. NOTE: It's recommended that you set up the receiver before the forwarder (See section 2 below).

2. Make a splunk instance on the same network

a. Backend host has to be from the same provider in the same VPC Network if you don't want to deal with creating a local network for both hosts.
    i. If you would like to use your local machine for Splunk Enterprise, you'll have to use the OpenVPN cli tool or something similar on both machines to create a local network. More here.
b. The backend is another DigitalOcean Ubuntu droplet with 2 GB Memory / 2 Intel vCPUs / 25 GB Disk
    i. Due to limited memory and CPU (Full Splunk Enterprise recommends 8 cores), I had to limit the size of _internal and _introspection indexes once splunk was running.
c. Install Splunk Enterprise using the correct wget <url> for your architecture.
d. Navigate to /splunk/bin and run:

```
sudo ./splunk start
```

e. Start splunk and navigate to HOST_IP:8000 and configure a receiver (Navigate to Settings > Data > Forwarding and Receiving; Usually a receiver is on port 9997)
f. You can check to see if the host is listening on port 9997 and from what IP address range with:

```
> netstat -auntp | grep 9997
```

Examples:

Docker Commands:

1. Dockerfile

```
# Use the slim-ish Python image from the Docker Hub
FROM python:3.11-slim-bullseye

# Set the working directory in the container
WORKDIR /app

# Copy the requirements.txt file first for better caching
COPY requirements.txt .

# Install the required Python packages
RUN pip install --no-cache-dir --timeout=1000 -r requirements.txt

# Copy the rest of the application code to the container
COPY . .

# Expose the port my app runs on
EXPOSE 5000

# run Flask app via gunicorn
CMD ["gunicorn", "--bind", "0.0.0.0:8080", "--workers", "1", "--timeout",
"300", "bw_webapp:app"]
```

2. Docker build (locally)

```
sudo docker build -t <org>/<repoName>:<tag> <srcDirectory>
sudo docker build -t jackwalton1/bw-webapp-img:v1.0 .
```

3. Docker run (locally first before we put it on the server)

```
sudo docker run -d –name <containerName> -p <hostPort>:<dockerPort>
<org>/<repoName>:<tag>
sudo docker run -d --name bw-webapp -p 8080:8080
jackwalton1/bw-webapp-img:v1.0
```

4. Push to the docker.io registry

```
sudo docker push jackwalton1/bw-webapp-img:v1.0
```

5. Pull from the host

```
sudo docker pull jackwalton1/bw-webapp-img:v1.0
```

6. Run on the host (if nginx is redirecting traffic to 8080, and docker is running the app on 8080 respectively)

```
sudo docker run -d --name bw-webapp -p 8080:8080
jackwalton1/bw-webapp-img:v1.0
```

Nginx Config (/etc/nginx/sites-available/default):

NOTE: Must make a default config with domain names without TLS, run certbot to get the certification files, and then restart nginx with this config.

DID NOT FEEL COMFORTABLE PUTTING THIS ON GITHUB

Configuring Splunk Universal Forwarder:

1. Download via the link in section 1 for the correct architecture of your host.
2. Add monitors for the directories of logs you'd like to forward
    a. Must make your index on the receiver
    b. Found in opt/splunkforwarder/etc/apps/SplunkUniversalForwarder/default

```
[monitor:///var/log/nginx/access.log]
disabled = false
index = bw_webapp

[monitor:///var/log/nginx/error.log]
disabled = false
index = bw_webapp
```

3. From /opt/splunkforwarder/bin/ Run:

```
sudo ./splunk add forward-server <RecieverIP>:9997
```

4. Can check status of forward server with:

```
sudo ./splunk list forward-server
```

5. Start splunk:

```
sudo ./splunk start
```

6. If you already set up the receiver, you can navigate to <RecieverIP>:9997 in your browser