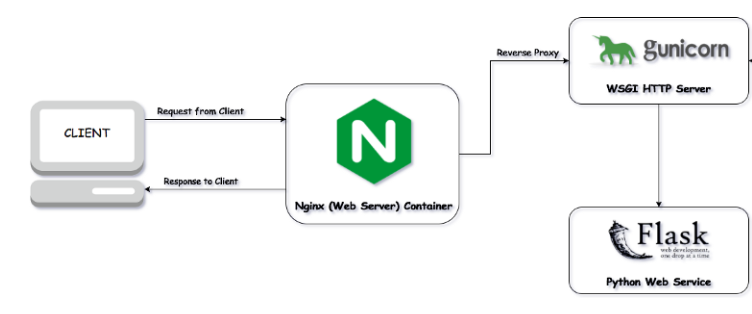
**REST API based on Flask, Gunicorn and NGINX:**

**Why Flask?**

Since, we wish to build multi-value lookup infra for real-time IoC comparisons in Graylog, we opted for *HTTPJSONPath* as a data adapter (Lookup Table). This data adapter demands rest API creation and access over HTTP to yield multi-values for a given lookup. Since it’s a simple use-case (not a complete web app), we chose Flask over Django.

***Architecture:***



**Why NGINX?**

Exposing Flask services straight away without any webserver in front is a security flaw; and, flask handles one request/thread at a time. In prod environment, we need queuing capabilities that Flask lacks. NGINX solves these issues.

**What is Gunicorn?**

Interface between NGINX and Flask.

**So, enough theory. Let’s do it?**

1. sudo yum install epel-release
2. sudo yum install python-pip python-devel gcc nginx
3. pip install gunicorn flask
4. ti\_rest.py (Flask app /rest api) resides here: /opt/scripts/TI\_REST/

Refer: ti\_rest.py for the code

1. wsgi.py (WSGI Entry Point) resides here: /opt/scripts/TI\_REST/

Refer: wsgi.py for the code

1. Now, bind wsgi.py with Gunicorn like below:

gunicorn --bind 0.0.0.0:8000 wsgi

1. Create a service for Gunicorn.

Refer: ti\_rest.service

1. Now, reload daemon and start the service.

systemctl daemon-reload

systemctl start ti\_rest.service

1. Configure NGINX.

Add below to /etc/nginx/nginx.conf

**Refer**: *nginx.conf* in main director*y*

* + 1. Now, check the validity of config by running below command:

nginx -t

* + 1. Now start the NGINX:

systemctl start nginx

**Note**: Please make sure host hosting these TI scripts is not exposed to outside world.