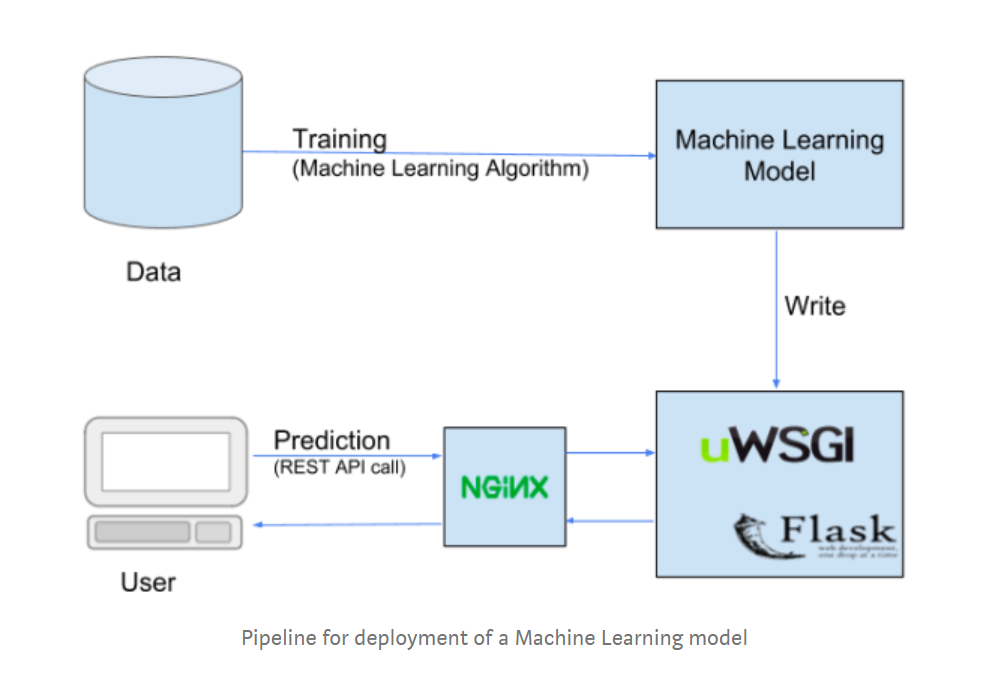
**ML Model deployment**



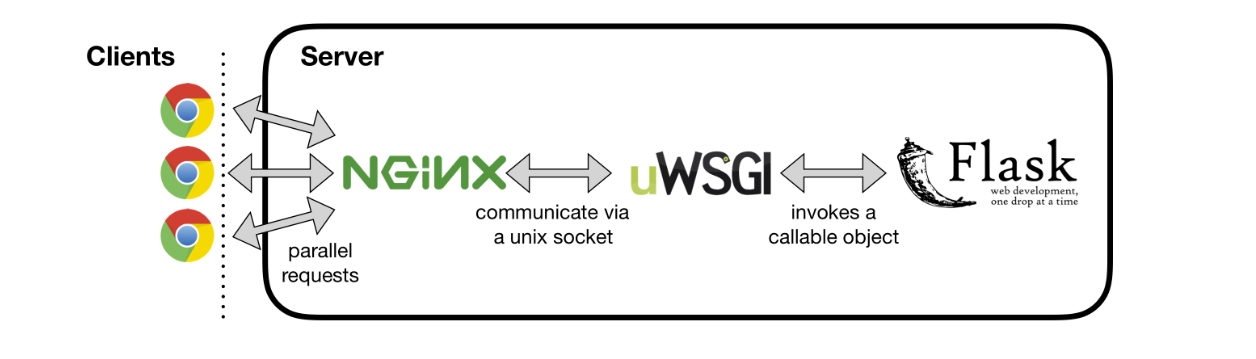
REST API increases the reach of your ML model to wider audience and as it can be called from any Application – mobile app, Java application, web application, .Net application, PHP/javascript, Python etc.

The most common Python framework, which is used to create a REST API in Python is, Flask.

Flask is a lightweight micro web framework written in Python, which can be used to create small web applications in Python.

For production purposes, you can deploy this REST service (written using Python and Flask framework) on a more matured WSGI protocol compliant application server like Gunicorn or **uWSGI** along with **Nginx** as the web server.

* Flask framework as something like Spring or Jersey framework – to create REST API
* Nginx as like Apache Tomcat  – web server
* uWSGI (or Gunicorn) as like JBoss or WebSphere  – application server



[**Anaconda**](https://anaconda.org/?ref=hackernoon.com): for managing package installation and creating an isolated Python 3 environment.

[**Keras**](http://keras.io/?ref=hackernoon.com): a high-level neural networks API, that is capable of running on top of [TensorFlow](https://github.com/tensorflow/tensorflow?ref=hackernoon.com), [CNTK](https://github.com/Microsoft/cntk?ref=hackernoon.com), or [Theano](https://github.com/Theano/Theano?ref=hackernoon.com).

[**Flask**](http://flask.pocoo.org/?ref=hackernoon.com): a minimalistic python framework for building RESTful APIs. Despite being easy to use, Flask’s built-in server serves only one request at a time by default; hence [it is not suitable](http://flask.pocoo.org/docs/deploying/?ref=hackernoon.com) on its own for deployment in production.

[**nginx**](https://nginx.org/en/?ref=hackernoon.com): the highly stable web server, which provides benefits such as load-balancing, SSL configuration, etc.

[**uWSGI**](https://uwsgi-docs.readthedocs.io/en/latest/?ref=hackernoon.com): a highly configurable WSGI server (Web Server Gateway Interface) that allows forking multiple workers to serve multiple requests at a time.

[**systemd**](https://en.wikipedia.org/wiki/Systemd?ref=hackernoon.com): an init system used in multiple Linux distributions to manage system processes after booting.

Nginx will be our interface to the internet, and it will be the one handling clients’ requests. Nginx has native support for the binary uWSGI protocol, and they communicate via Unix sockets. In turn, the uWSGI server will be invoking a callable object within our Flask application directly. That is the way that requests will be served.

**Different Approach:**

