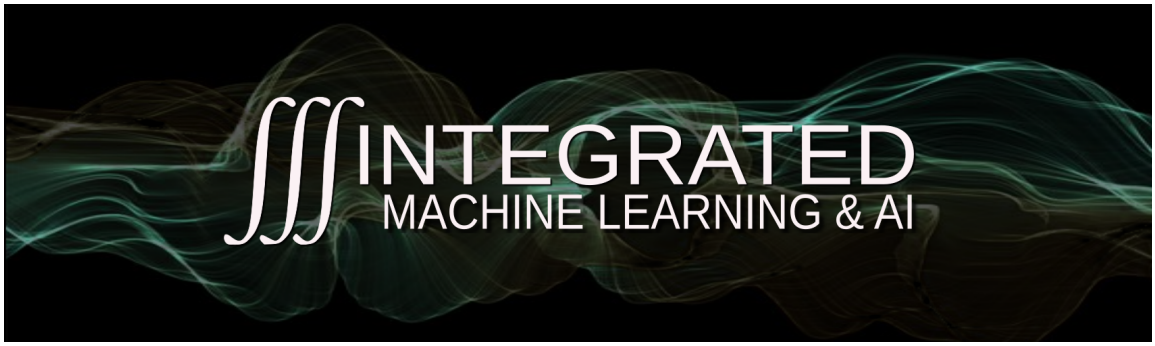
Basic ML Ops For REST APIs With Docker And Jenkins

> by Thom Ives, Ph.D.



Overview

1. Develop Your REST API Using FastAPI On Your **Dev** System
2. Ensure That You Have Docker Installed And Install It If You Do Not
3. Install The Latest Jenkins LTS Version On Your **Host** System Using The "Generic Java Package (.war)" Method
4. Create A GitHub Repo To Hold Your Code
    A. Your Dev System Will Hold A Clone
    B. Make Changes / Improvements And Commit And Push From Your **Dev** System
5. Clone Your Initial Code To Your **Host** System In The Correct Location For Working With Jenkins
6. Create A Docker Image And Run Its Container For Your API
    A. Write Your Dockerfile That Will Create Your Image
    B. Build Your Image
    C. Run Your Image In A Docker Container With A Directory From Your Machine Mounted To Your Container
7. Create A Jenkins Pipeline That Will:
    A. Update The Code On Your **Host** System
    B. When You Push Changes From Your **Dev** System
    C. THIS Is Our SIMPLE CI-CD For REST APIs

Develop Your REST API Using FastAPI On Your **Dev** System

There is an initial repo for this HERE called "Basic_Model_Serving_With_FastAPI".

Let's go look at it.

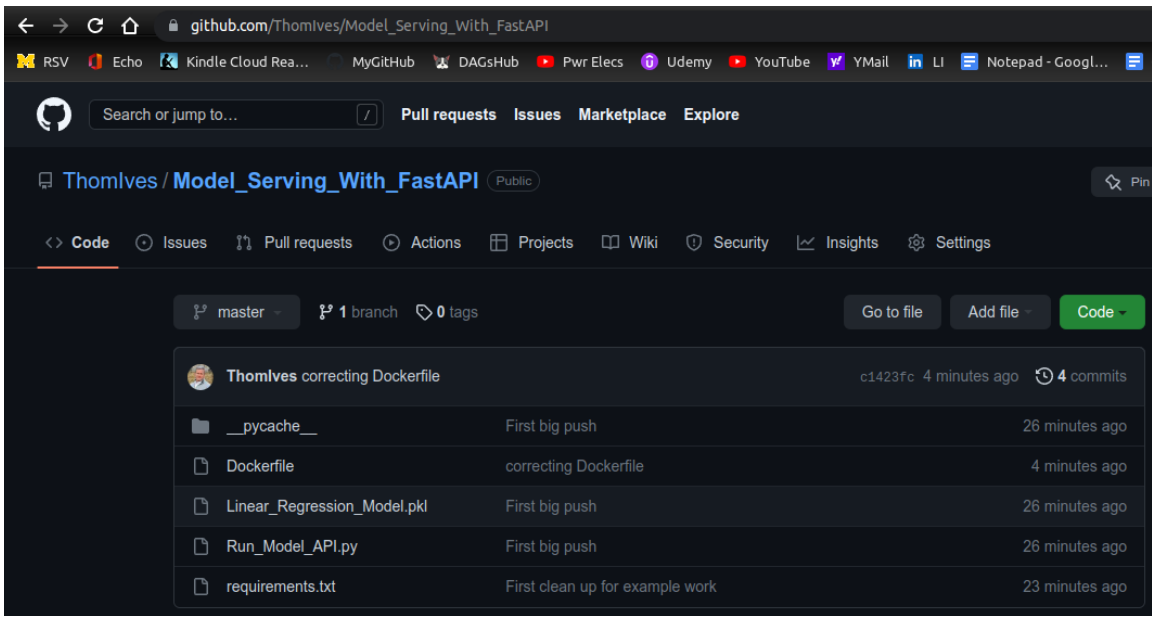
Ensure That You Have Docker Installed

Please see my repository covering my studies on Docker, which include how to install it.

Install The Latest Jenkins LTS Version On Your HOST System
Using The "Generic Java Package (.war)" Method

Go HERE and read carefully. The simplest and recommended way to install Jenkins is with "Generic Java Package (.war)" described on that page.

Create A GitHub Repo To Hold Your Code



Your Dev System Will Hold A Clone

Show Development File Manager

Make Changes / Improvements And Commit And Push From Your Dev System

Clone Your Initial Code To Your HOST System In The Correct Location For Working With Jenkins

Show Jenkins Clone Of Code

We Also Build And Run Our Docker Image And Container From The Host

Create A Docker Image And Run Its Container For Your API

Write Your Dockerfile That Will Create Your Image

FROM python:3.8

WORKDIR /src

COPY ./requirements.txt /src COPY ./main.py /src COPY ./Linear_Regression_Model.pkl /src

RUN pip install -r requirements.txt

EXPOSE 8000:8000

CMD ["uvicorn", "Run_Model_API:app", "--reload", "--host", "0.0.0.0"]

Build Your Image

```
$ sudo docker image build . -t api:1
```

Run Your Image In A Docker Container With A Directory From Your Machine Mounted To Your Container

```
$ sudo docker run -d -p 8000:8000 -v /home/thom/.jenkins/workspace
/Model_Serving_With_FastAPI/src:/src api:1
```

A Simpler Example For CI CD

The FastAPI REST API Code - main.py

In [ ]:
```python
# main.py
from fastapi import FastAPI

app = FastAPI()


@app.get("/")
async def root():
    return {"message": "This is a simple API."}
```

The Dockefile

In [ ]:
```
FROM python:3.8

WORKDIR /src

COPY ./requirements.txt /src
COPY ./main.py /src

RUN pip install -r requirements.txt

EXPOSE 8000:8000

CMD ["uvicorn", "main:app", "--reload", "--host", "0.0.0.0"]
```

The requirements.txt File

In [ ]:
```
fastapi
uvicorn
```

The Jenkins File

In [ ]:
```
pipeline {
    agent any
    stages {
        stage ("Go To Git Directory"){
            steps {
                sh "cd /home/thom/.jenkins/workspace/Docker_Jenkins_Cloud_AP
            }
        }
        stage ("Pull Changes From Repo"){
            steps {
                script{
                    checkout([$class: 'GitSCM', branches: [[name: 'master']]
                }
            }
        }
        stage ("Report Status"){
            steps {
                echo "Updates Complete."
            }
        }
    }
}
```

Summary

1. Develop REST API With FastAPI
2. Dockerize Your API Or Not - Serve It Either Way From A Host - On Prem Or From Cloud VM
3. Setup Jenkins To Watch Your Cloud Git Repo To Update Your Host Files
4. Changes Will Pass To Controlling Directory On Host
5. Uvicorn Will See Changes And Restart