
SEASON 5
SUPER AI ENGINEER
AI INNOVATOR • AI ENGINEER • AI RESEARCHER
AI
AI
AI

Fast API and Docker

Sivaphong Niyomphanich

SuperAI Season#5

About Me

- Sivaphong Niyomphanich
- Pong+

- Hospital Management Services
- PRINC Group

- SuperAI Engineer SS#2

Agenda

- Fast API
- Docker
- Related Fundamental

Pre-Require Tools

- VSCode
- Postman
- Python (Mini-conda and conda-forge)
- WSL
- Docker
- Ngrok

- Plese refer document “FastAPI-Docker_ToolsInstall-Manual”

About Docker License Concern

- Docker Engine
  - Option I: Windows OS
    - WSL with Ubuntu and Docker (CE)
    - Or Docker Desktop (License concern*)
  - Option II: Mac OS
    - Docker Desktop (License concern*)
  - Option III: Linux/Unix
    - Docker (CE)

*License concern: Docker Desktop is free for small businesses (fewer than 250 employees AND less than $10 million in annual revenue), personal use, education, and non-commercial open source projects. Governance also purchase the license.

About Anaconda License Concern

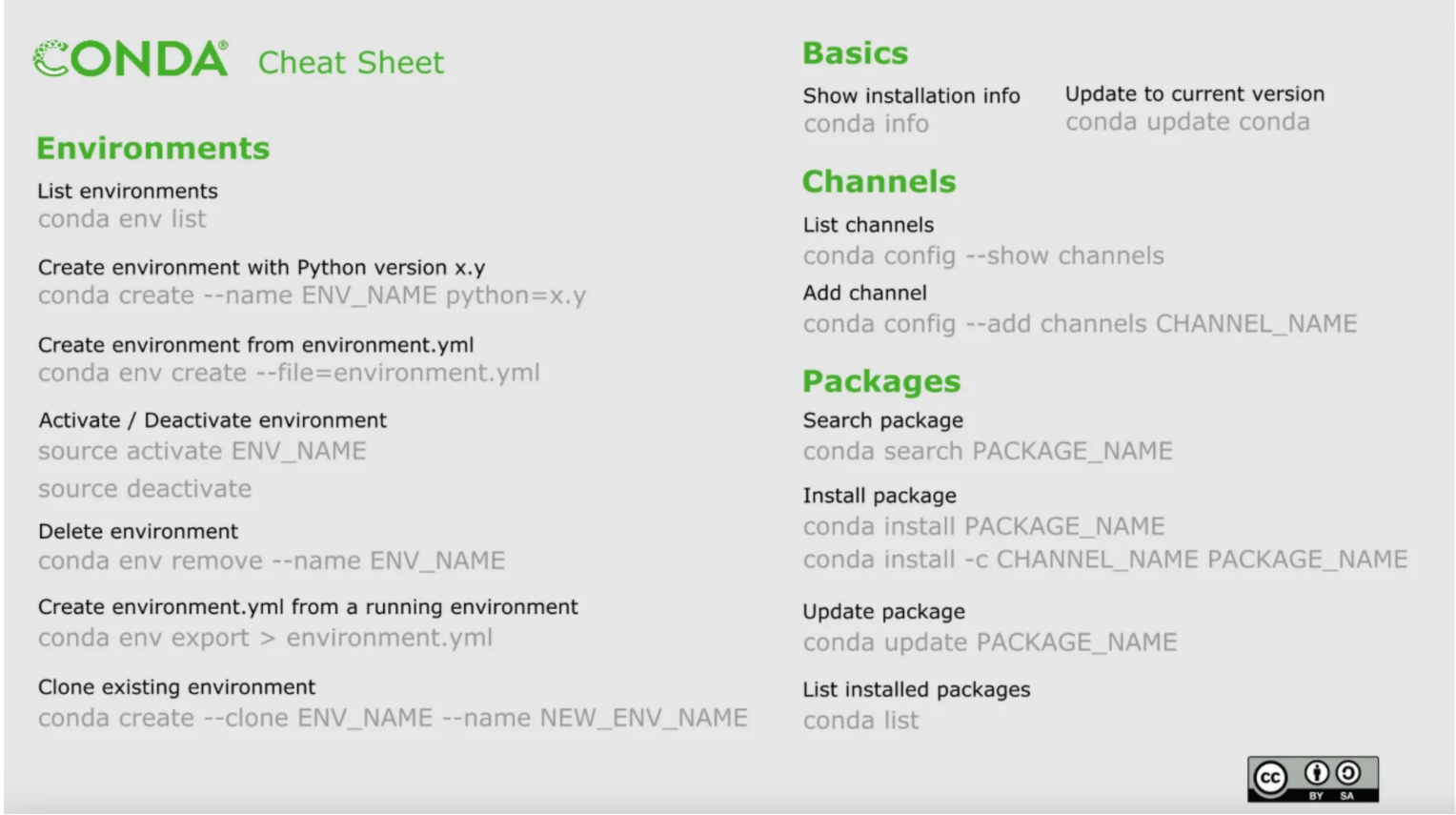
- Anaconda/Conda
  - Option I: Anaconda default repository channel
    - License concern**
  - Option II: Anaconda with conda-forge repository channel
    - The conda-forge channel provides a vast array of packages and is completely open-source, which helps in avoiding any potential licensing issues.
  - Option III: Using other Python
    - Using other package management and
    - environment management.

**Anaconda's licensing policies have made some organizations with more than 200 employees, look for alternatives. Who wants to pay 600USD for one license.

Python and Conda

Conda command

- Conda is Python package management and environment management.



CONDA Cheat Sheet

Environments

List environments
conda env list

Create environment with Python version x.y
conda create --name ENV_NAME python=x.y

Create environment from environment.yml
conda env create --file=environment.yml

Activate / Deactivate environment
source activate ENV_NAME
source deactivate

Delete environment
conda env remove --name ENV_NAME

Create environment.yml from a running environment
conda env export > environment.yml

Clone existing environment
conda create --clone ENV_NAME --name NEW_ENV_NAME

Basics

Show installation info
conda info

Update to current version
conda update conda

Channels

List channels
conda config --show channels

Add channel
conda config --add channels CHANNEL_NAME

Packages

Search package
conda search PACKAGE_NAME

Install package
conda install PACKAGE_NAME
conda install -c CHANNEL_NAME PACKAGE_NAME

Update package
conda update PACKAGE_NAME

List installed packages
conda list

Image source: https://dcordero.medium.com/conda-cheat-sheet-e95ca5aeda85

Preparing Python for development

- Activate Conda
  - source ~/miniconda3/bin/activate
- List conda environment
  - conda env list

```
pongier@Windows11:~$ source ~/miniconda3/bin/activate
(base) pongier@Windows11:~$ conda env list

# conda environments:
#
base                  * /home/pongier/miniconda3
```

- List Python version
  - conda search python

```
(base) pongier@Windows11:~$ conda search python
Loading channels: done
# Name                       Version           Build  Channel
python                         1.0.1               0  conda-forge
python                           1.2               0  conda-forge
python                           1.3               0  conda-forge
python                           1.4               0  conda-forge
python                         1.5.2               0  conda-forge
python                           1.6               0  conda-forge
python                           2.0               0  conda-forge
python                         2.6.9               0  conda-forge
python                        2.7.12               0  conda-forge
python                        2.7.12               1  conda-forge
python                        2.7.12               2  conda-forge
python                        2.7.13               0  conda-forge
python                        2.7.13               1  conda-forge
```

Preparing Python for development

- Create conda environment
  - conda create --name superapi python=3.12.9

```
(base) pongier@Windows11:~$ conda create --name superapi python=3.12.9
Channels:
 - conda-forge
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done
```

- Deactivate conda environment
  - conda deactivate

```
(base) pongier@Windows11:~$ conda deactivate
pongier@Windows11:~$
```

Preparing Python for development

- Activate superapi environment
  - source ~/miniconda3/bin/activate superapi

```
pongier@Windows11:~$ source ~/miniconda3/bin/activate superapi
(superapi) pongier@Windows11:~$
```

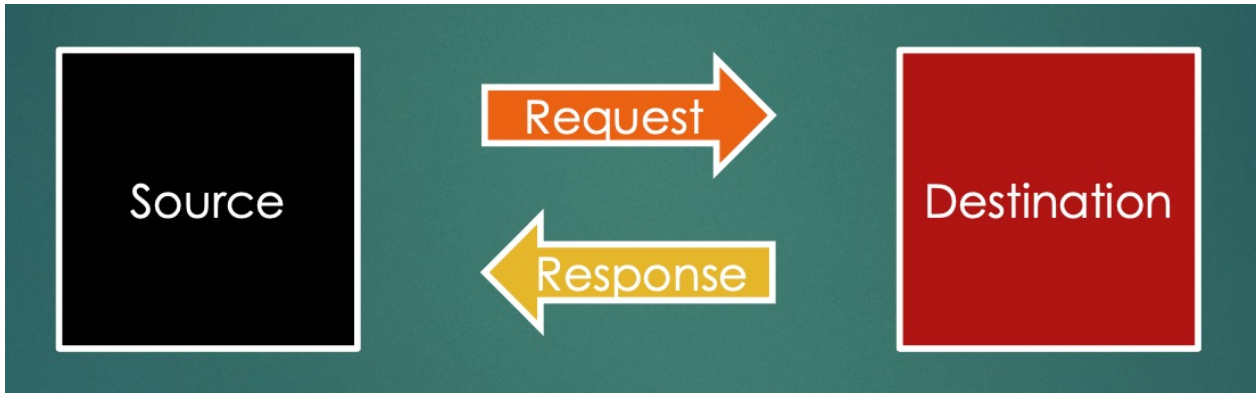
- Install package/library
  - conda install numpy
  - Or pip install numpy

```
(superapi) pongier@Windows11:~$ conda install numpy
Channels:
 - conda-forge
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done
```

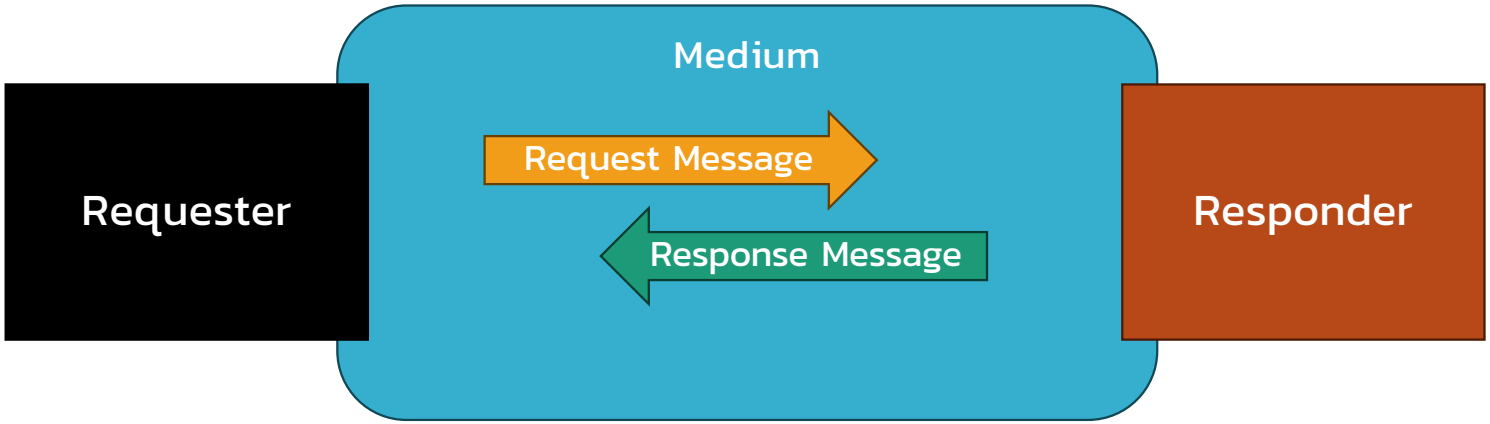
Fast API

API (Application Programming Interface) Overview

- APIs are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols.



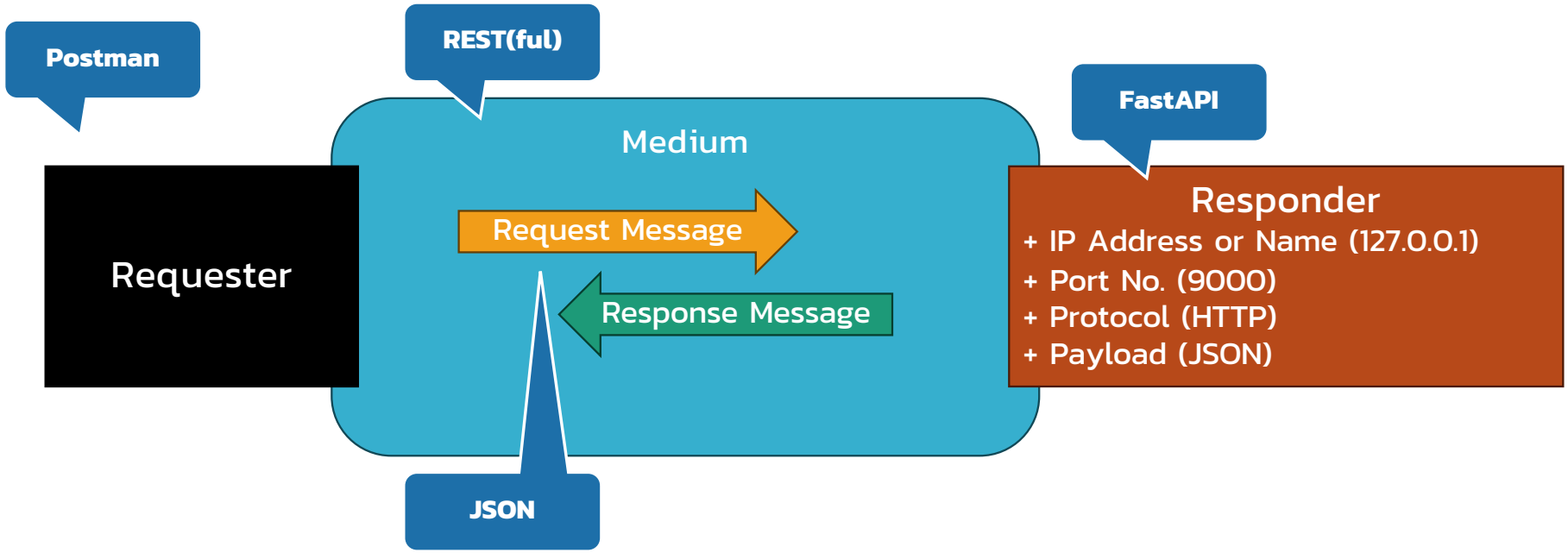
API Architecture



API Architecture

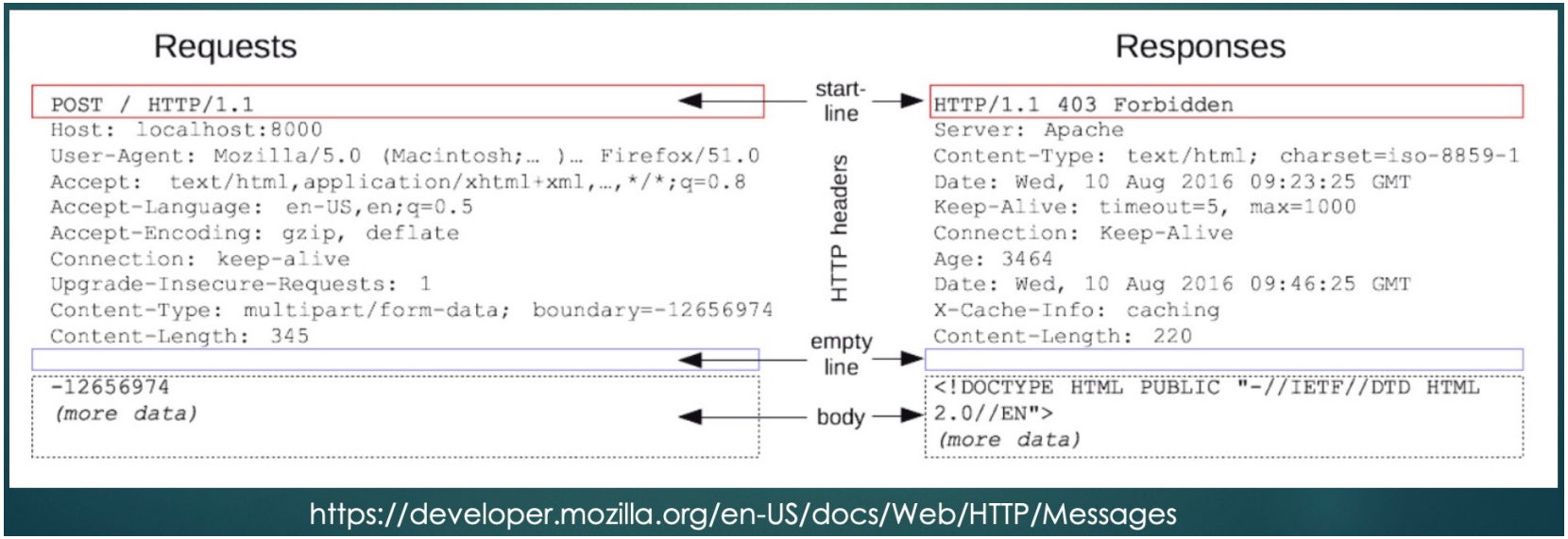
- Requester
  - Initial caller
- Message
  - JSON
  - XML
- Medium
  - SOAP
  - RPC
  - Web Socket
  - REST
- Responder
  - Response result

API Architecture for today



Understanding HTTP Protocol

- HTTP Header
- HTTP Body



Understanding HTTP URL

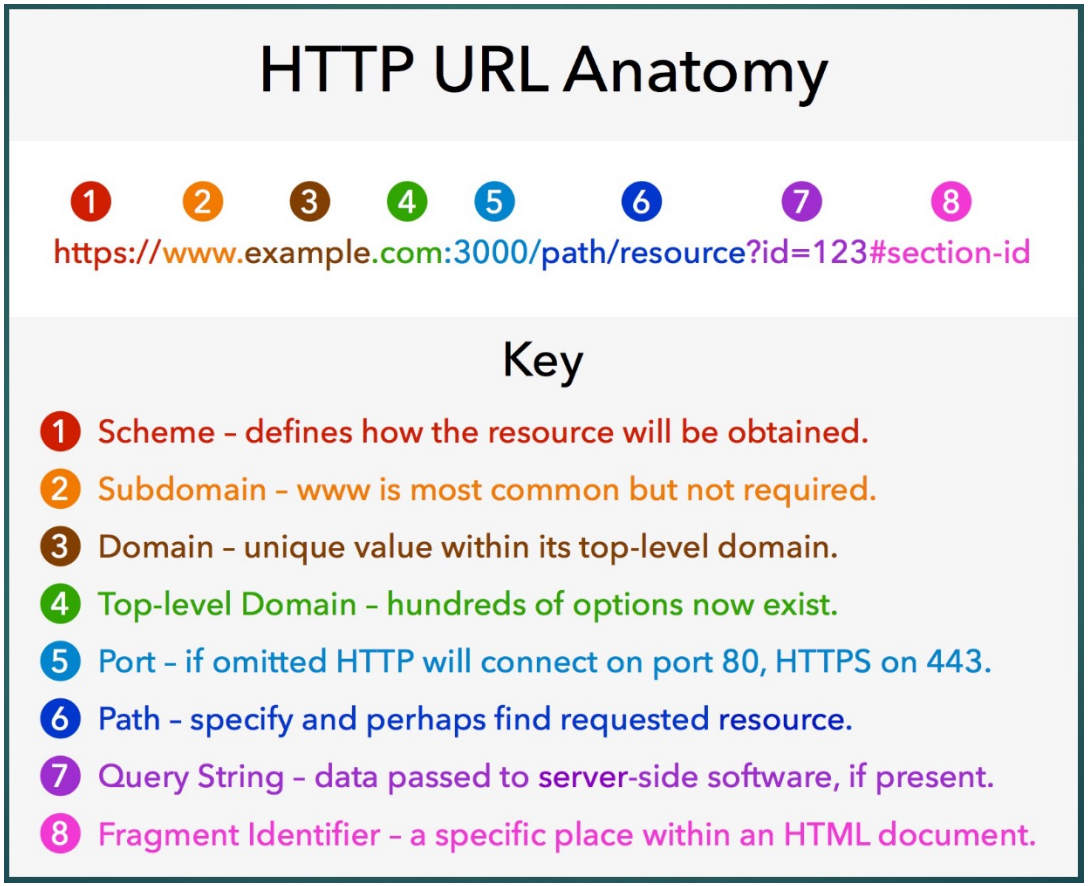
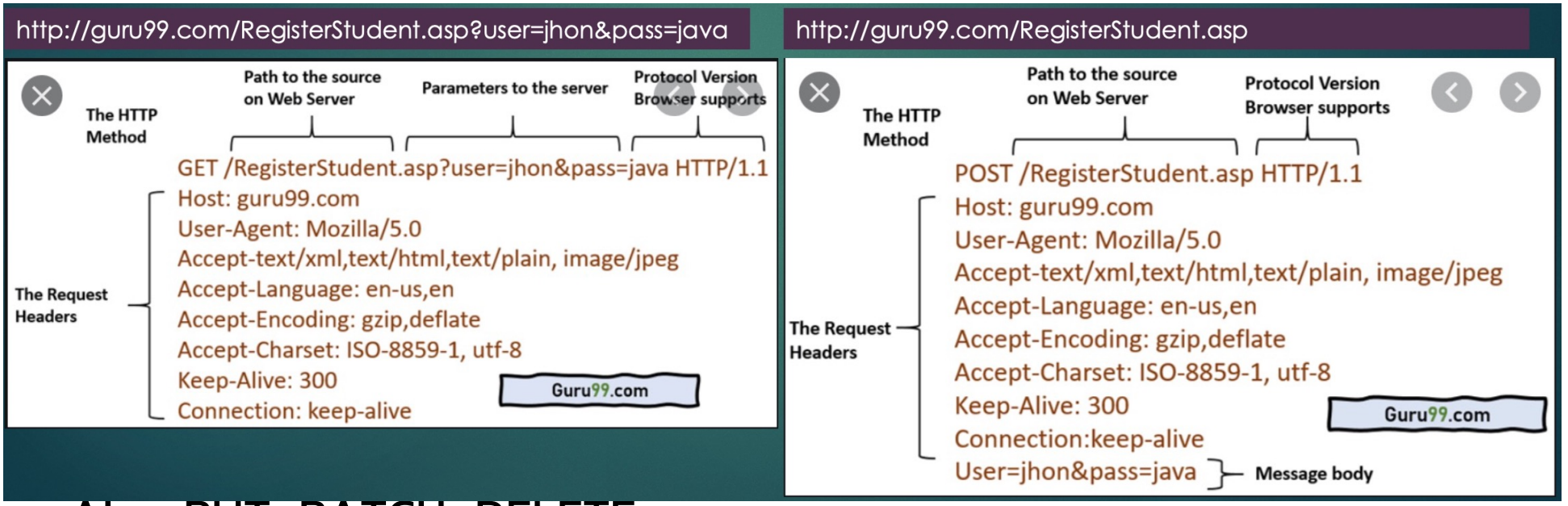


Image source: https://cascadingmedia.com/insites/2015/02/url-anatomy-seo-tips.html

Understanding HTTP Request Method

- GET Method
- POST Method



- Also PUT, PATCH, DELETE

Important when working with (REST) API

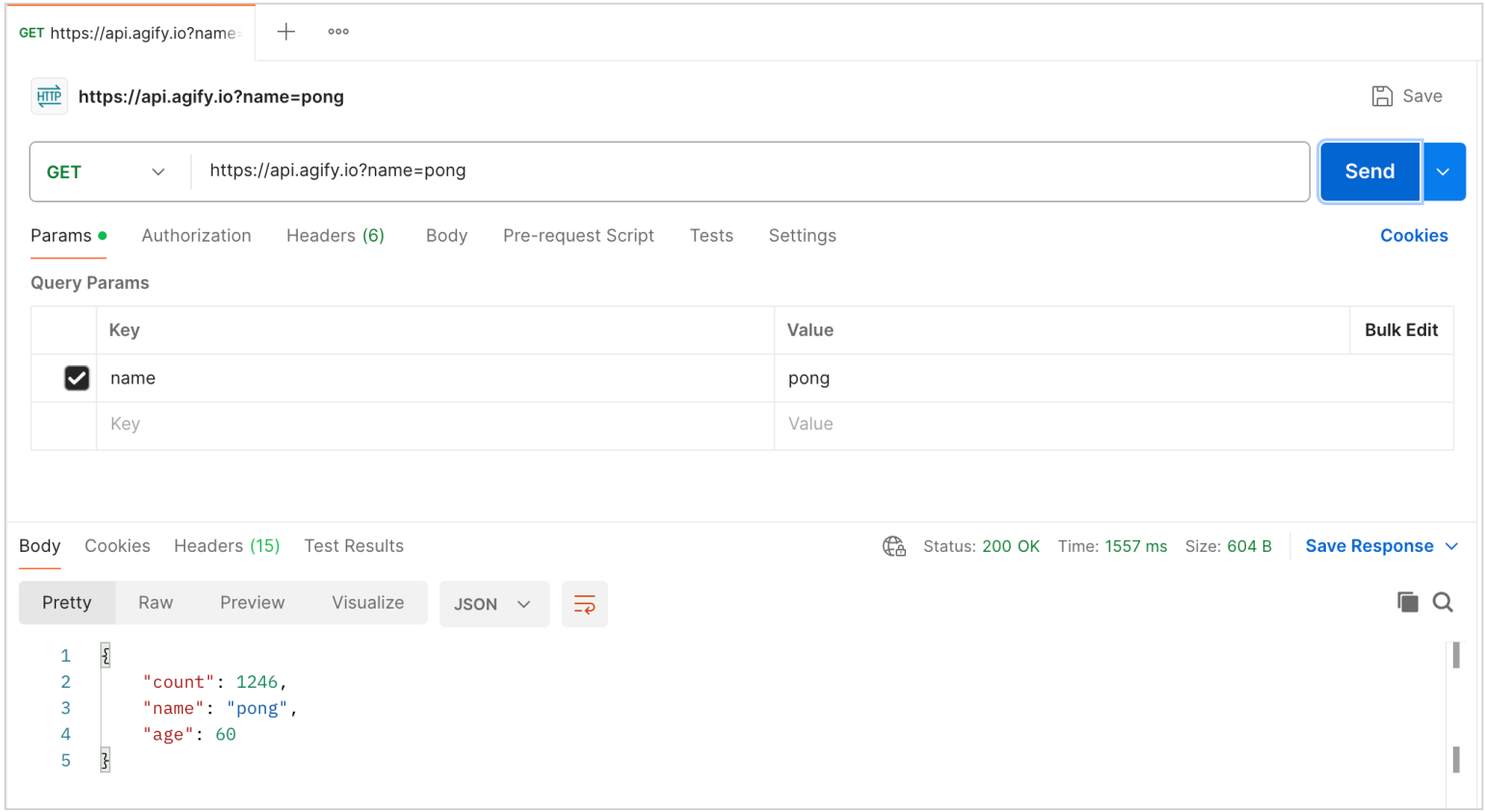
- Request URL (IP_Name/Port No./Path)
- HTTP Request Method
- Parameter/Payload
  - Request/Response
- Is need subscriber (e.g. token, key)

Call REST API with Postman

- Simple API
- https://agify.io/documentation

- Request URL (IP_Name/Port No./Path)
  - https://api.agify.io?name=michael
- HTTP Request Method
  - GET
- Parameter/Payload
  - name=michael
- Request/Response
  - { "count": 298219, "name": "michael", "age": 62 }
- Is need subscriber (e.g. token, key)
  - No

Call REST API with Postman



Call REST API with Postman

- More complex API
- https://reqbin.com/req/ala1tavu/sample-api-post-request

- Request URL (IP_Name/Port No./Path)
  - https://reqbin.com/echo/post/json
- HTTP Request Method
  - POST
- Parameter/Payload

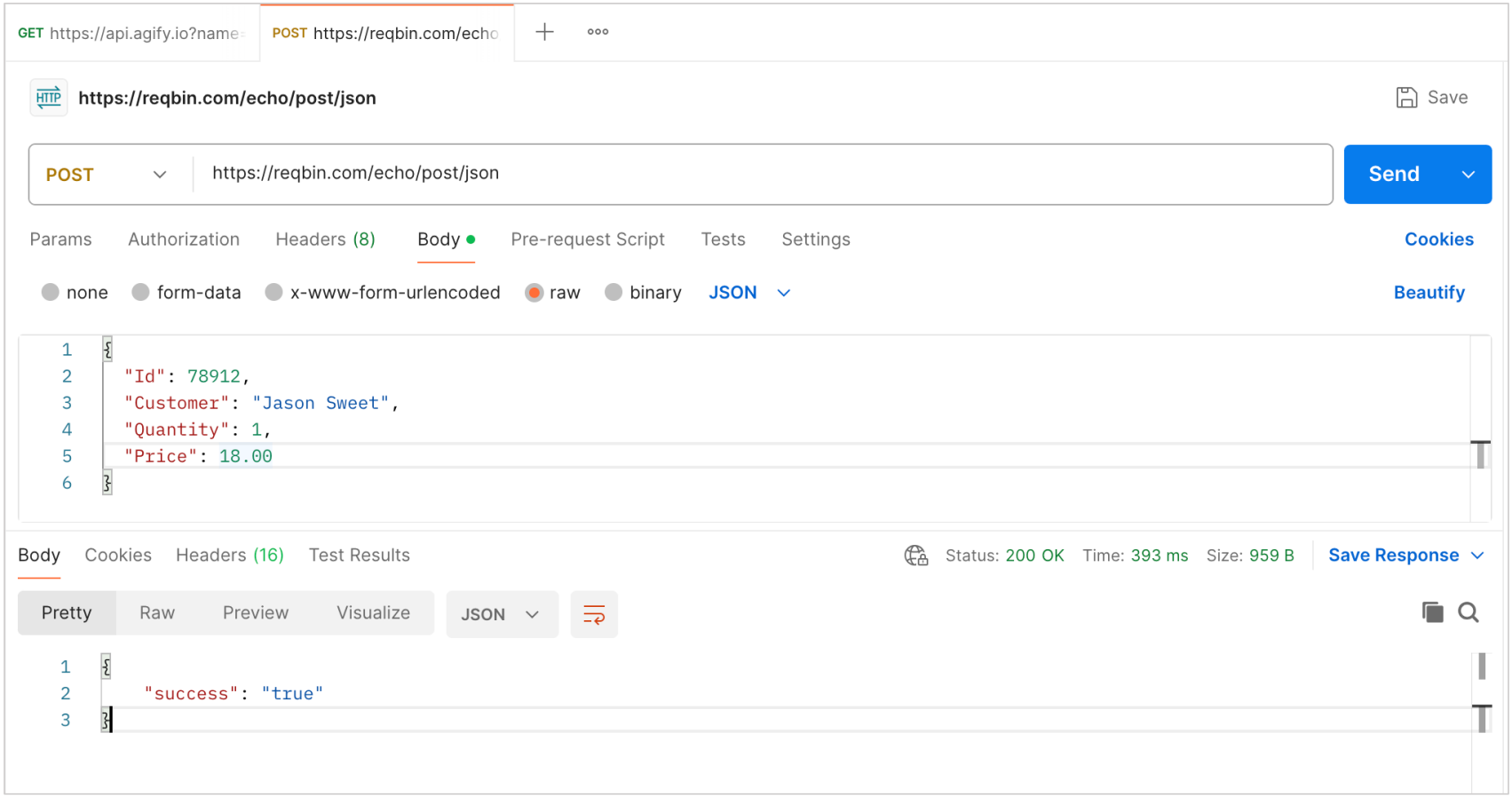
```
{
"Id": 78912,
"Customer": "Jason Sweet",
"Quantity": 1,
"Price": 18.00
}
```

- Request/Response

```
{
"success": "true"
}
```

- Is need subscriber (e.g. token, key)
  - No

Call REST API with Postman



Create our API with FastAPI

- FastAPI is a modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints.

- https://fastapi.tiangolo.com

Preapring for Coding

- What is your computer?
  - Windows user may using WSL with Ubuntu and Mini-conda
  - Mac user may using MacOSX and Mini-conda
  - Linux user may using Linux OS and Mini-conda

- Coding at
  - VSCode

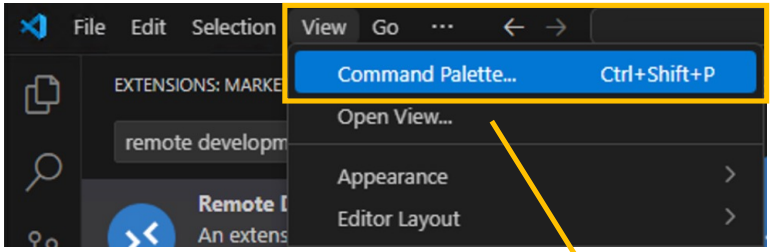
- Testing tool by
  - Postman

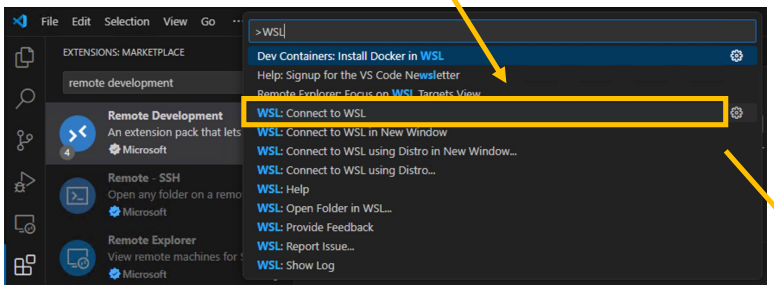
Preapring for Coding

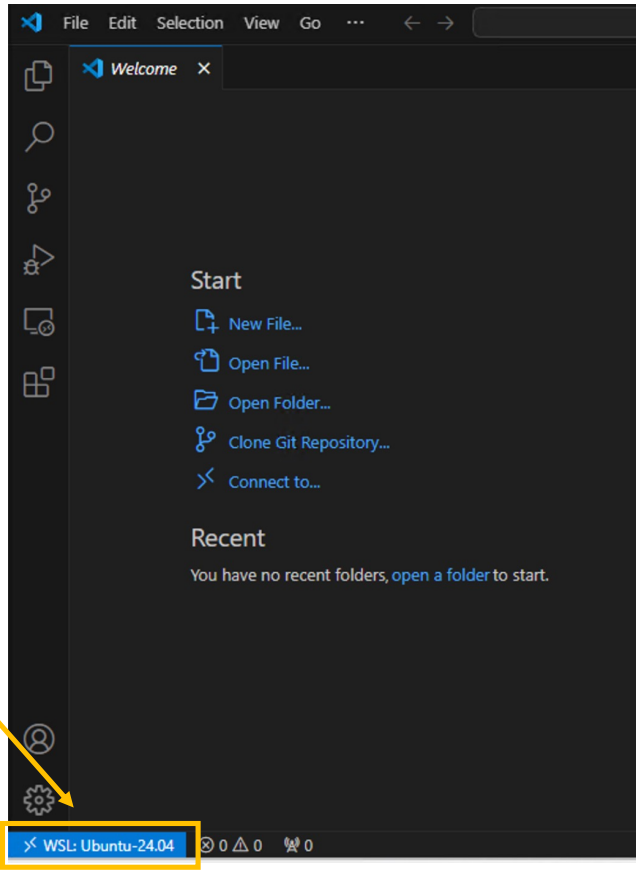
- Python and Environment
  - **Using python environment named superapi**
  - Using package/library
    - "fastapi[standard]"

Open VSCode and connect to WSL

- Open VSCode
- Connect to WSL

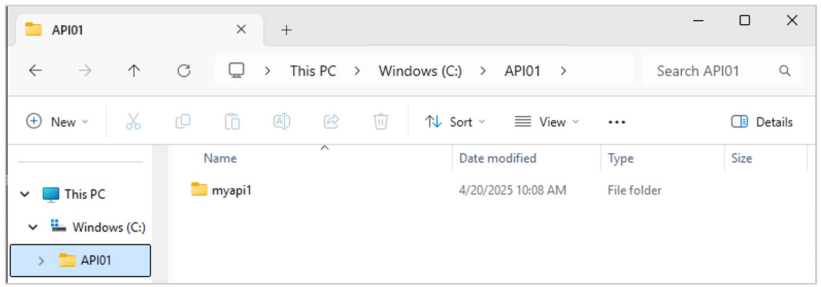


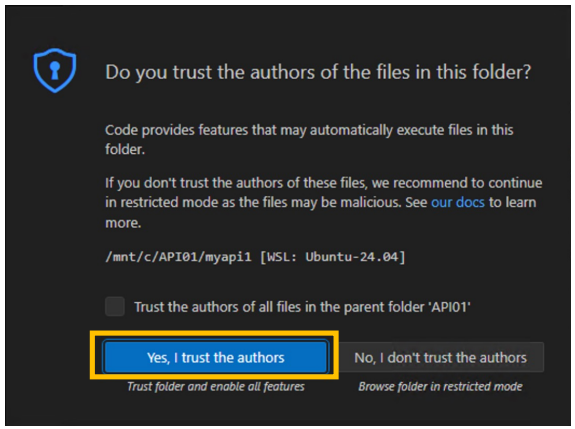




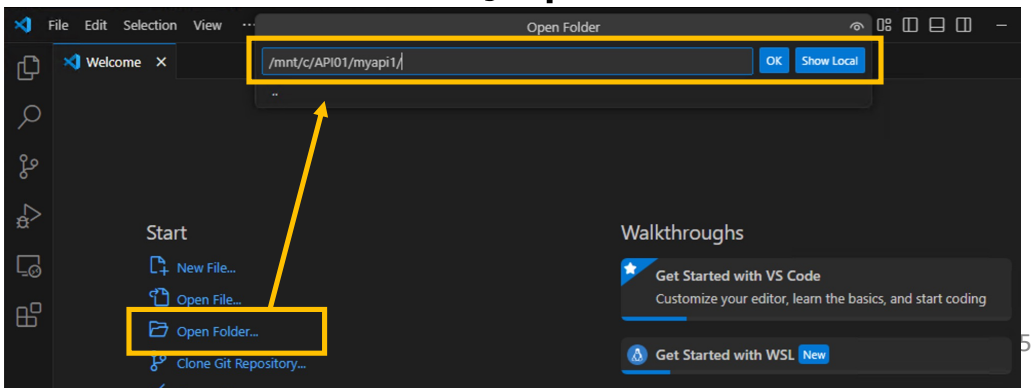
Create API01\myapi1 folder on C:\

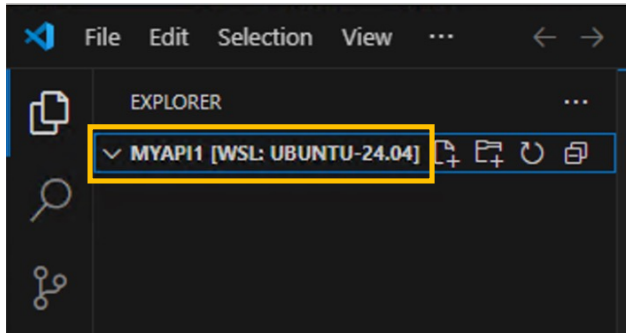
- Create API01\myapi1 folder on C:\ at Windows OS





- In VSCode -> “Open Folder”
  - /mnt/c/API01/myapi1/

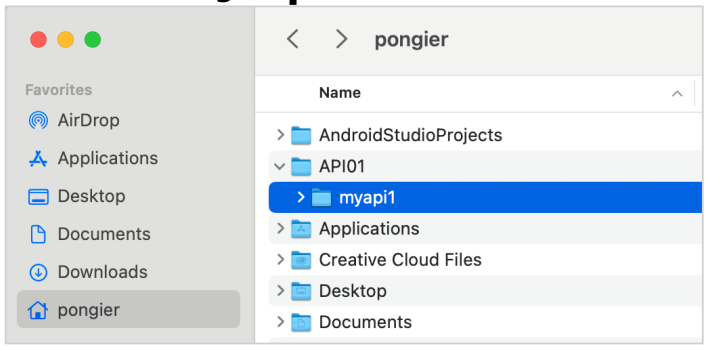




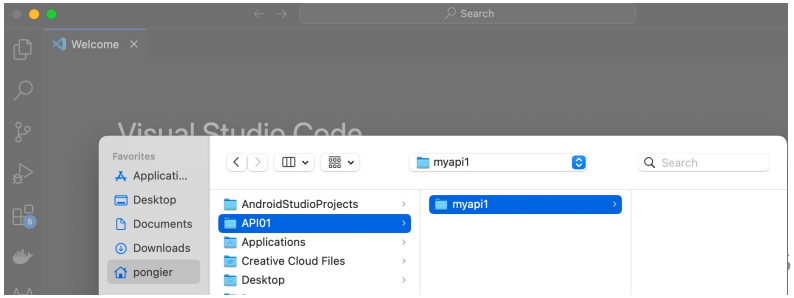
Mac User

Open VSCode and Create directory on

- Create API01\myapi1 folder on /Users/YOUR_NAME at MacOSX



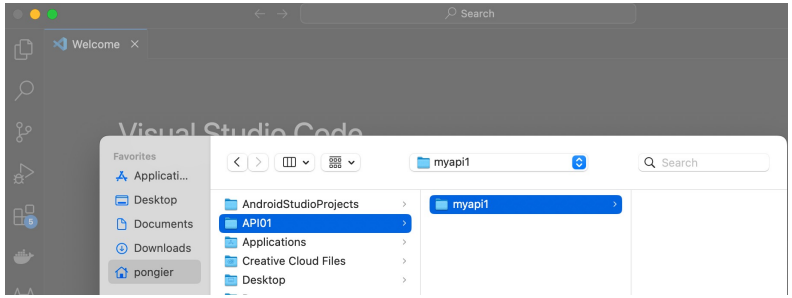
- In VSCode -> “Open”
  - /Users/YOUR_NAME/API01/myapi1/



Open VSCode and Create directory on

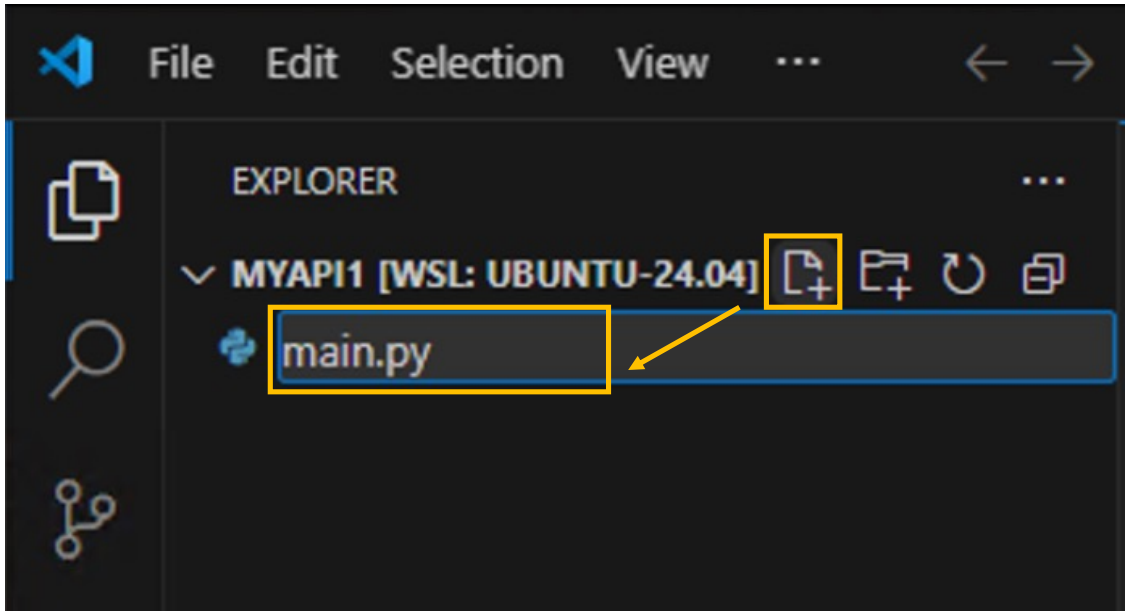
- Create API01\myapi1 folder on /home/YOUR_NAME at Linux

- In VSCode -> “Open”
  - /home/YOUR_NAME/API01/myapi1/



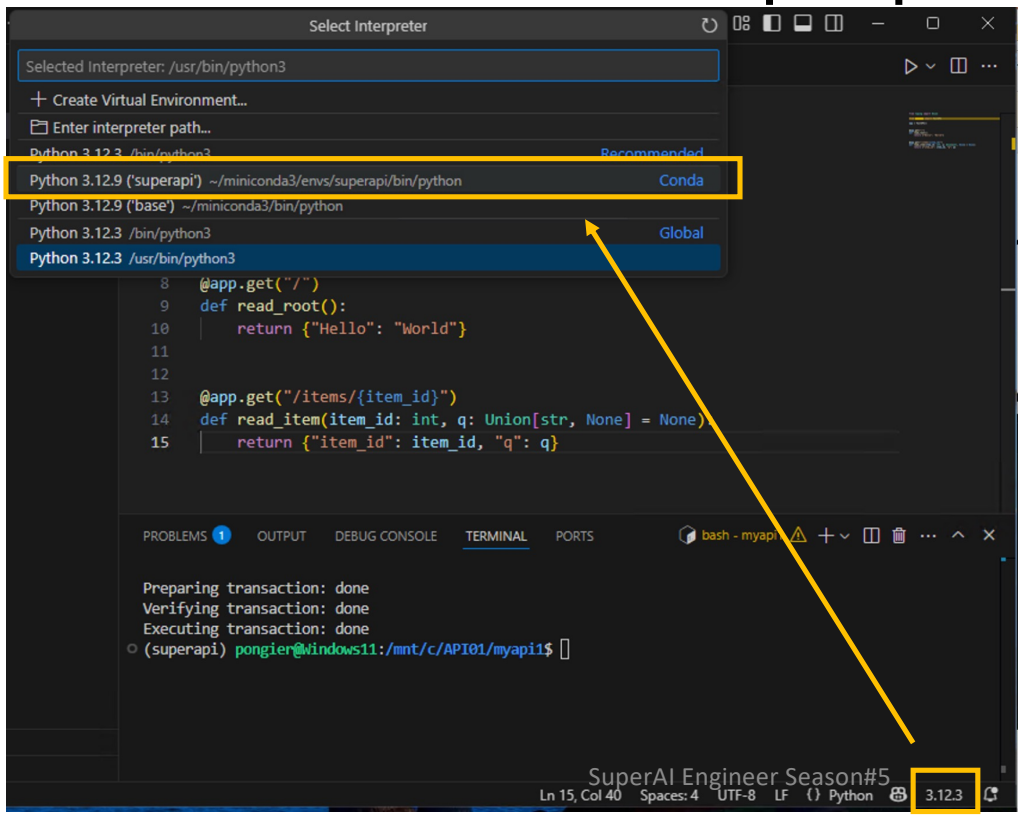
Create a new file named main.py

- Create a new file named “main.py” in VSCode



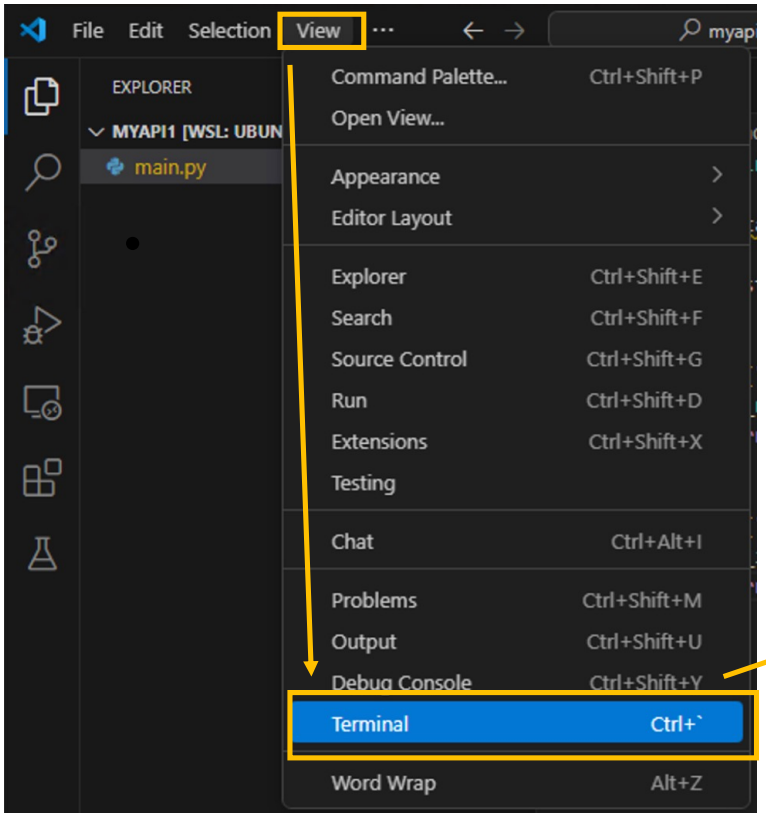
Select Python/Conda environment

- Select environment named “superapi” in VSCode

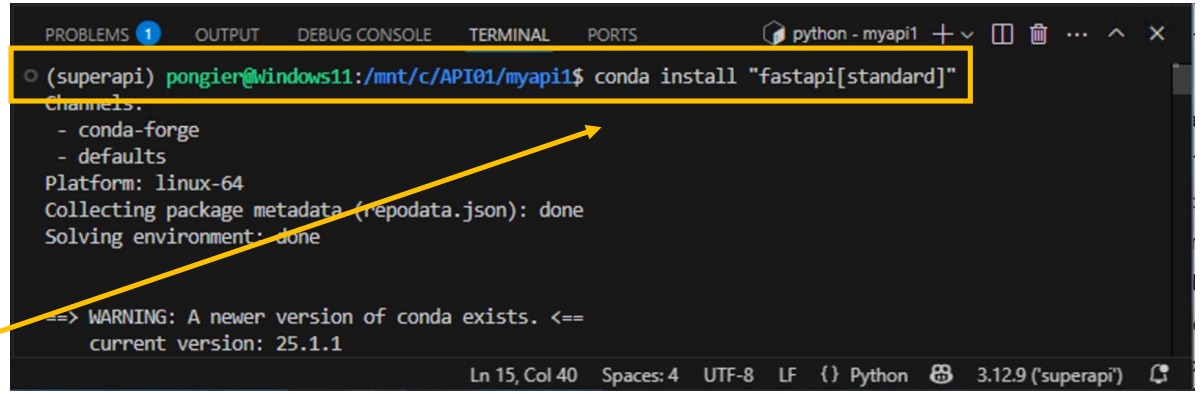


Install fastapi package/library

- Install package/library named fastapi[standard] in VSCode terminal



conda install "fastapi[standard]"

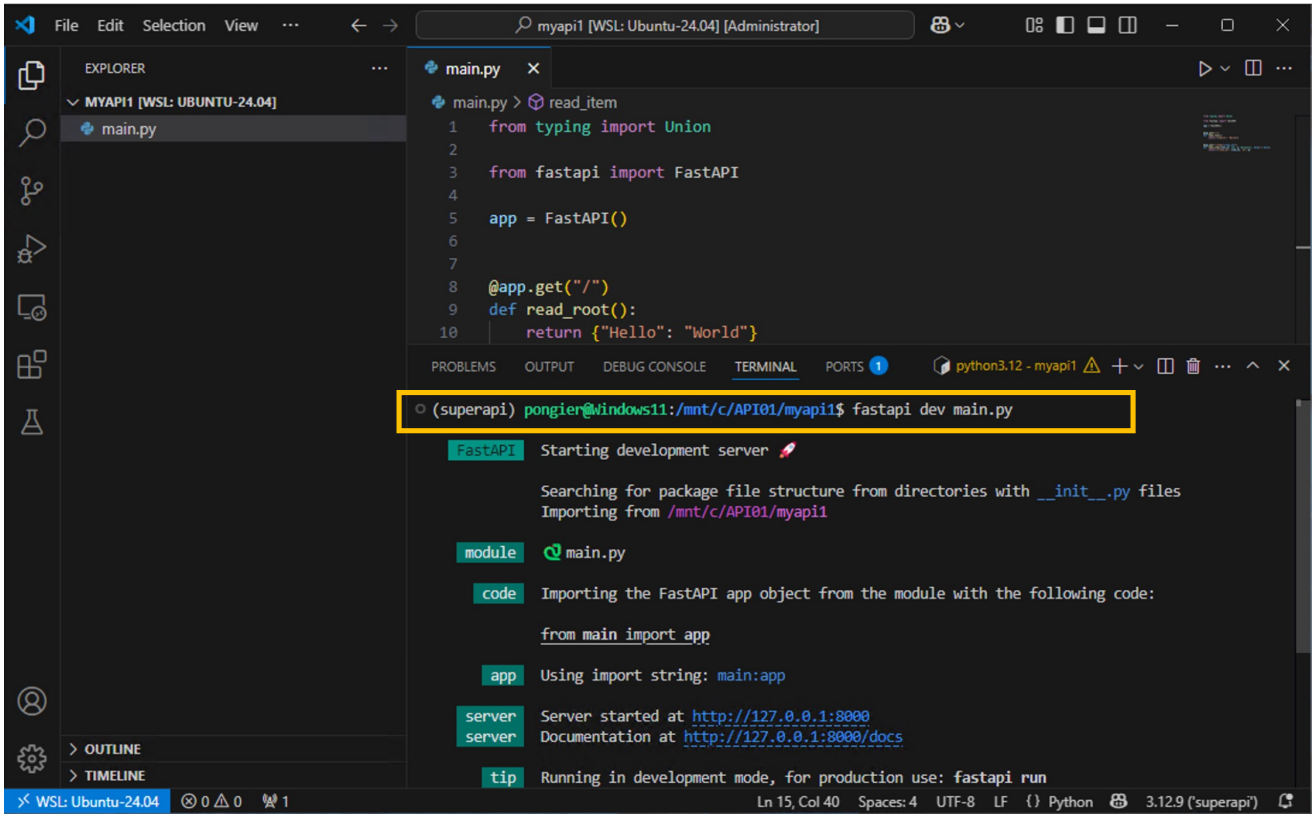


Coding in file main.py

```
from typing import Union

from fastapi import FastAPI

app = FastAPI()


@app.get("/")
def read_root():
    return {"Hello": "World"}


@app.get("/items/{item_id}")
def read_item(item_id: int, q: Union[str, None] = None):
    return {"item_id": item_id, "q": q}
```

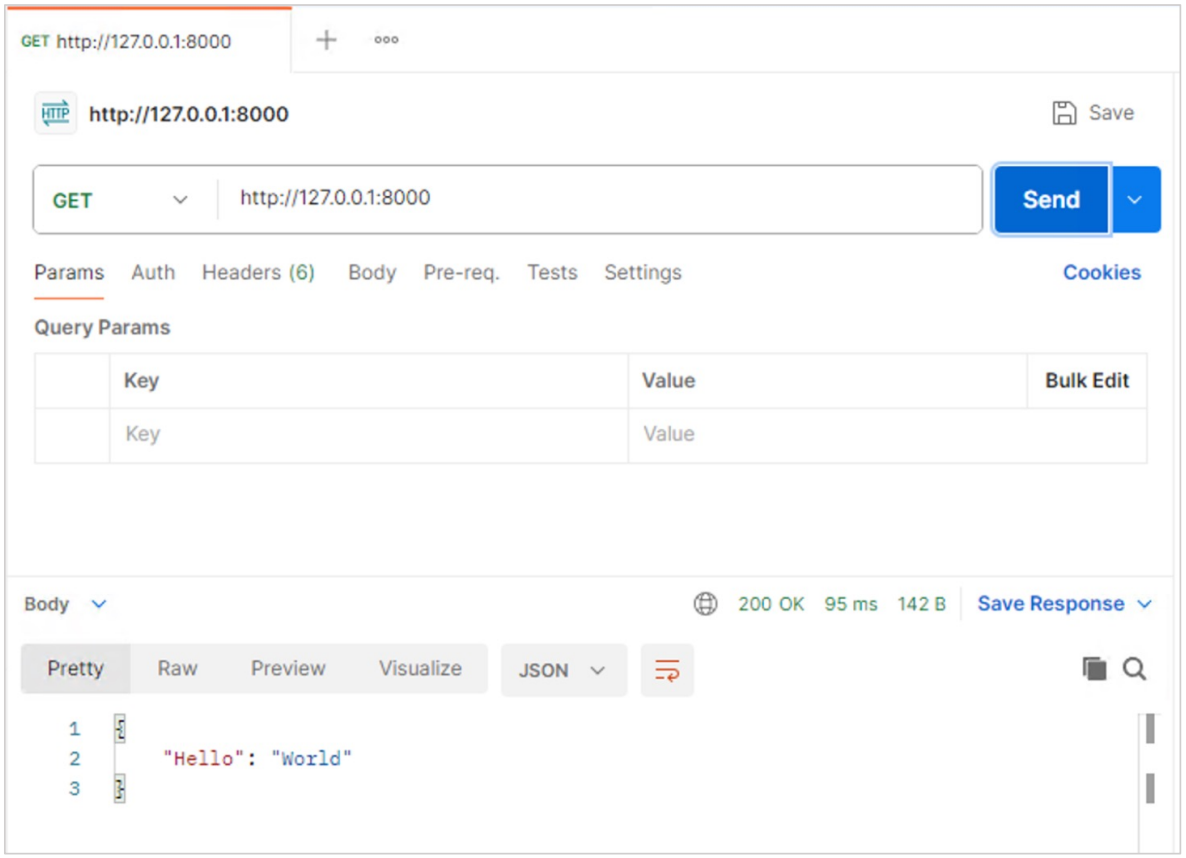
Running your code

- Enter command in VSCode terminal
  - fastapi dev main.py



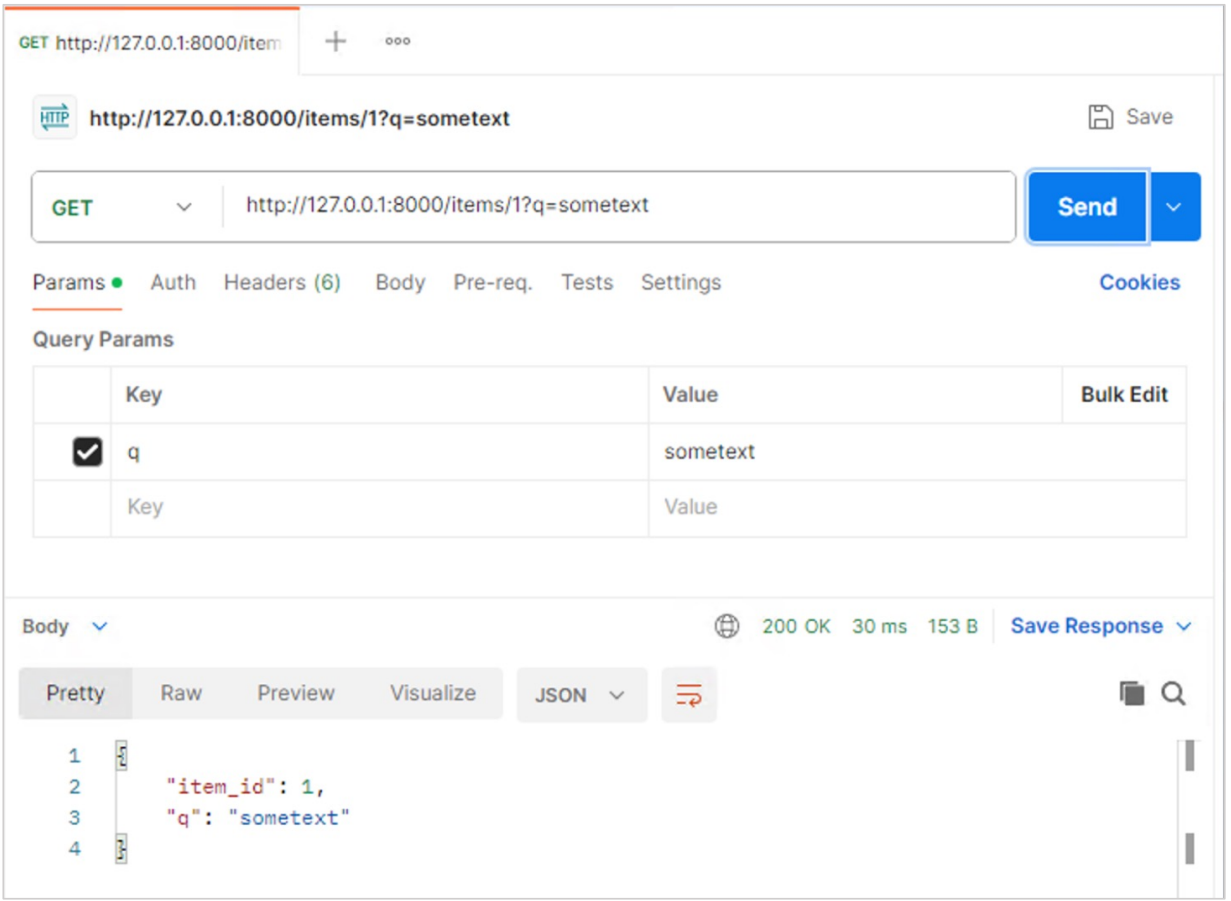
Testing your API

- Testing your API with Postman
  - Request URL (IP_Name/Port No./Path)
    - http://127.0.0.1:8000
  - HTTP Request Method
    - GET
  - Parameter/Payload
    - No
  - Request/Response
    - { "Hello": "World" }
  - Is need subscriber (e.g. token, key)
    - No



Testing your API

- Testing your API with Postman
  - Request URL (IP_Name/Port No./Path)
    - http://127.0.0.1:8000/items/1?q=x
  - HTTP Request Method
    - GET
  - Parameter/Payload
    - items
    - q
  - Request/Response
    - {
    - "item_id": 1,
    - "q": "sometext"
    - }
  - Is need subscriber (e.g. token, key)
    - No



Understanding the code (FastAPI)

```
from typing import Union

from fastapi import FastAPI

app = FastAPI()


@app.get("/")      #Method, Path
def read_root():   #Defined function
    return {"Hello": "World"}  #Result that return


@app.get("/items/{item_id}") #Method, Path, Parameter/Variable
def read_item(item_id: int, q: Union[str, None] = None):  #Defined function
    return {"item_id": item_id, "q": q} #Result that return
```

Understanding JSON

- JSON (Java Script Object Notation)

- Key:Value
- [] is array

personal.json

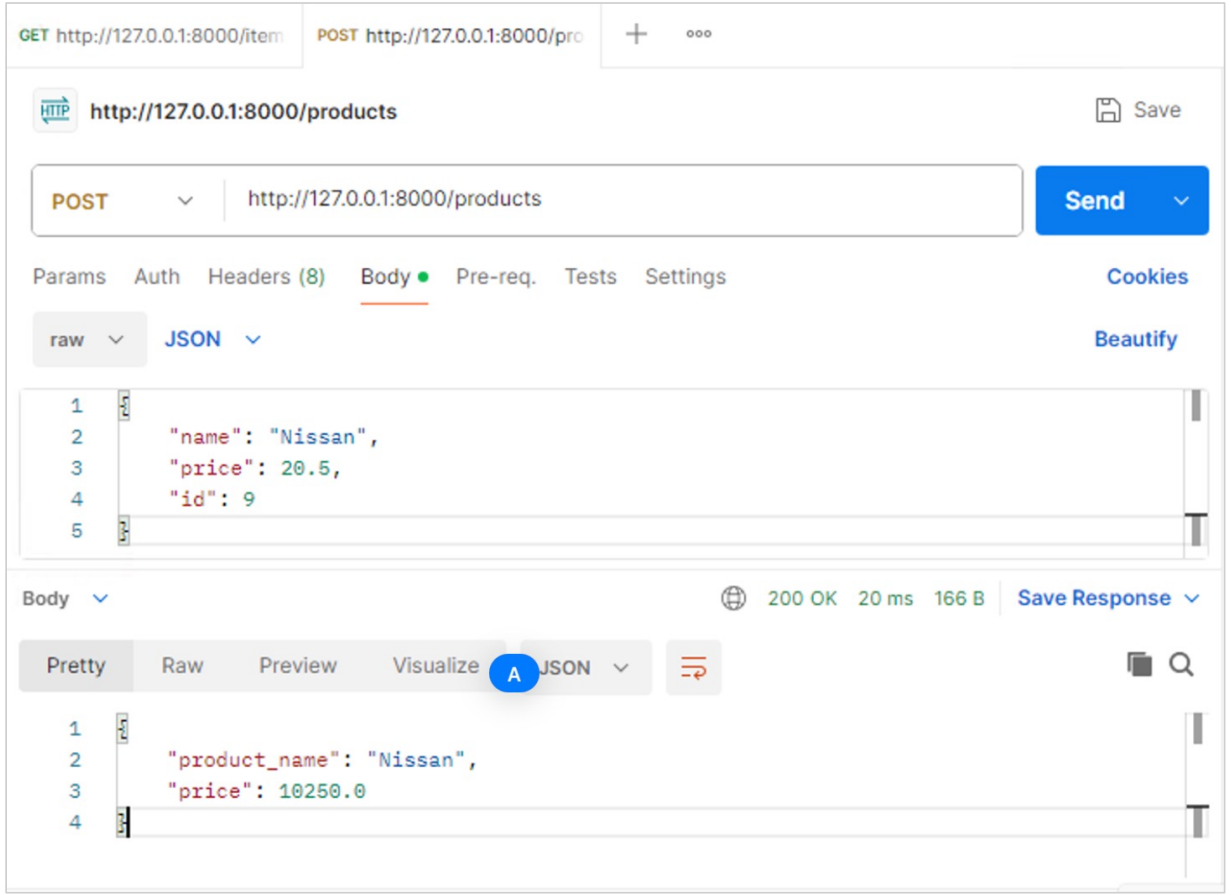
```
{
    "$schema": "personal-schema.json",
    "personnel": {
        "person": [
            {
                "id": "Big.Boss",
                "name": {
                    "family": "Boss",
                    "given": "Big"
                },
                "email": "chief@oxygenxml.com",
                "link": {
                    "subordinates": [
                        "one.worker",
                        "two.worker",
                        "three.worker",
                        "four.worker",
                        "five.worker"
```

More complex your API

```
from typing import Union

from fastapi import FastAPI
from pydantic import BaseModel

app = FastAPI()


class Product(BaseModel):
    name: str
    price: float
    id: int


@app.get("/")
def read_root():
    return {"Hello": "World"}


@app.get("/items/{item_id}")
def read_item(item_id: int, q: Union[str, None] = None):
    return {"item_id": item_id, "q": q}


@app.post("/products")
def add_product(product: Product):
    return {"product_name": product.name, "price": product.price*500}
```

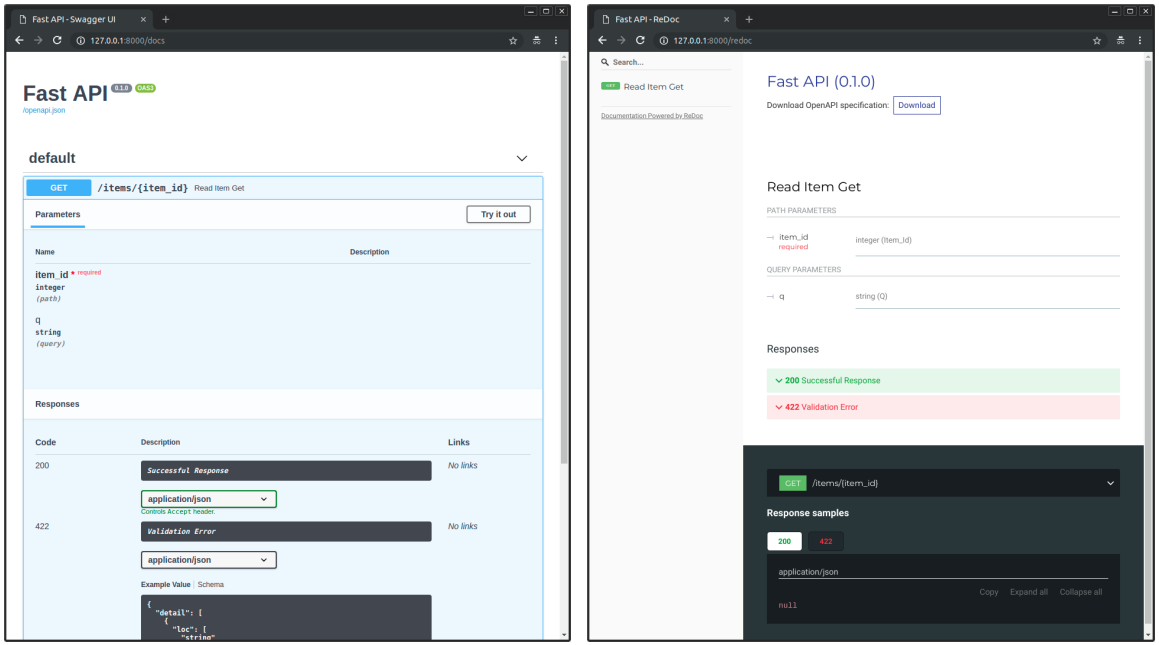
Testing your API

- Testing your API with Postman
  - Request URL (IP_Name/Port No./Path)
    - http://127.0.0.1:8000/products
  - HTTP Request Method
    - POST
  - Parameter/Payload
    - {
    - "name": "Nissan",
    - "price": 20.5,
    - "id": 9
    - }
  - Request/Response
    - {
    - "product_name": "Nissan",
    - "price": 10250.0
    - }
  - Is need subscriber (e.g. token, key)
    - No



Something fantastic

- http://127.0.0.1:8000/docs
- http://127.0.0.1:8000/redoc



Custom Port and Running in Production

- Using uvicron
  - Bind all IP and Port is 9000
  - Multi worker is 2
- Install package/library
  - conda install "uvicorn[standard]"

```
(superapi) pongier@Windows11:/mnt/c/API01/myapi1$ conda install "uvicorn[standard]"
Retrieving notices: done
Channels:
 - conda-forge
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done
```

- Running API
  - uvicorn main:app --host 0.0.0.0 --port 9000 --workers 2

```
(superapi) pongier@Windows11:/mnt/c/API01/myapi1$ uvicorn main:app --host 0.0.0.0 --port 9000 --workers 2
INFO:     Uvicorn running on http://0.0.0.0:9000 (Press CTRL+C to quit)
INFO:     Started parent process [6290]
INFO:     Started server process [6292]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Started server process [6293]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

Entenable complexities your API

- FastAPI stands on the shoulders of giants:
  - Starlette for the web parts.
  - Pydantic for the data parts.

- Additional optional Pydantic dependencies:
  - pydantic-settings - for settings management.

- Used by FastAPI / Starlette:
  - uvicorn - for the server that loads and serves your application. This includes
  - uvicorn[standard], which includes some dependencies (e.g. uvloop) needed for high performance serving.

HTTP Method for API

HTTP Verb	CRUD	Entire Collection (e.g. /customers)	Specific Item (e.g. /customers/{id})
POST	Create	201 (Created), 'Location' header with link to /customers/{id} containing new ID.	404 (Not Found), 409 (Conflict) if resource already exists..
GET	Read	200 (OK), list of customers. Use pagination, sorting and filtering to navigate big lists.	200 (OK), single customer. 404 (Not Found), if ID not found or invalid.
PUT	Update/Replace	405 (Method Not Allowed), unless you want to update/replace every resource in the entire collection.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
PATCH	Update/Modify	405 (Method Not Allowed), unless you want to modify the collection itself.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
DELETE	Delete	405 (Method Not Allowed), unless you want to delete the whole collection—not often desirable.	200 (OK). 404 (Not Found), if ID not found or invalid.

Image source: https://www.restapitutorial.com/lessons/httpmethods.html

Creating API with your AI Model

- Understand the example AI Model
  - Input are Age, Sex and Embarked
  - Output is Prediction = 0,1

- Install require package/library
  - conda install numpy
  - conda install pandas
  - conda install scikit-learn

Model Code

Source: https://www.datacamp.com/tutorial/machine-learning-models-api-python

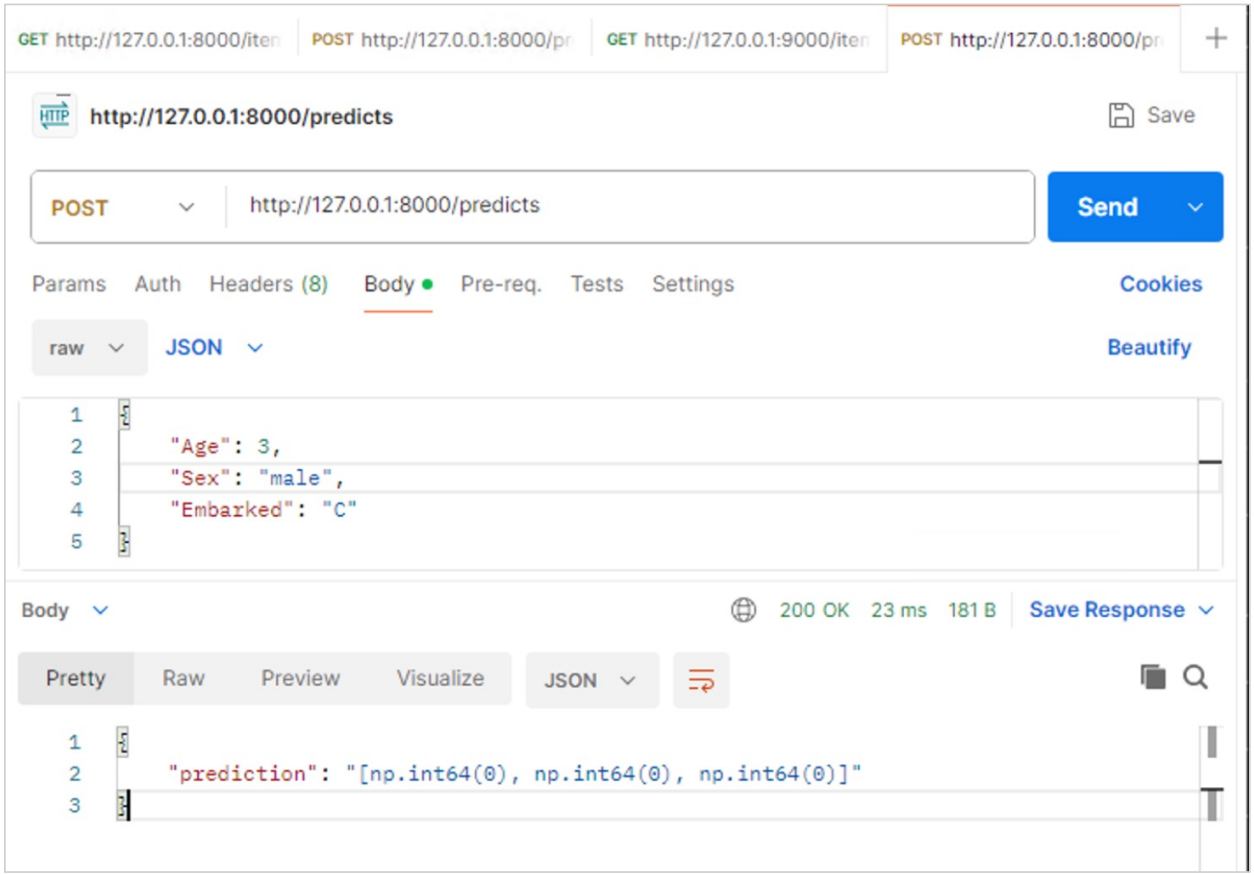
```
# Import dependencies
import pandas as pd
import numpy as np

# Load the dataset in a dataframe object and include only four features as mentioned
url = "http://s3.amazonaws.com/assets.datacamp.com/course/Kaggle/train.csv"
df = pd.read_csv(url)
include = ['Age', 'Sex', 'Embarked', 'Survived'] # Only four features
df_ = df[include]

# Data Preprocessing
categoricals = []
for col, col_type in df_.dtypes.items():
     if col_type == 'O':
          categoricals.append(col)
     else:
          df_[col].fillna(0, inplace=True)

df_ohe = pd.get_dummies(df_, columns=categoricals, dummy_na=True)

# Logistic Regression classifier
from sklearn.linear_model import LogisticRegression
dependent_variable = 'Survived'
x = df_ohe[df_ohe.columns.difference([dependent_variable])]
y = df_ohe[dependent_variable]
lr = LogisticRegression()
lr.fit(x, y)

# Save your model
import joblib
joblib.dump(lr, 'model.pkl')
print("Model dumped!")

# Load the model that you just saved
lr = joblib.load('model.pkl')

# Saving the data columns from training
model_columns = list(x.columns)
joblib.dump(model_columns, 'model_columns.pkl')
print("Models columns dumped!")
```

API Code

```
from typing import Union

from fastapi import FastAPI
from pydantic import BaseModel

import joblib
import pandas as pd
import numpy as np

app = FastAPI()

class Product(BaseModel):
    name: str
    price: float
    id: int

class Idata(BaseModel):
    Age: int
    Sex: str
    Embarked: str

lr = joblib.load("model.pkl") # Load "model.pkl"
model_columns = joblib.load("model_columns.pkl") # Load "model_columns.pkl"

@app.get("/")
def read_root():
    return {"Hello": "World"}

@app.get("/items/{item_id}")
def read_item(item_id: int, q: Union[str, None] = None):
    return {"item_id": item_id, "q": q}

@app.post("/products")
def add_product(product: Product):
    return {"product_name": product.name, "price": product.price*500}

@app.post("/predicts")
def ai_predict(indata: Idata):
    query = pd.get_dummies(pd.DataFrame(data=indata))
    query = query.reindex(columns=model_columns, fill_value=0)
    prediction = list(lr.predict(query))
    return {'prediction': str(prediction)}
```

Testing your API

- Testing your API with Postman
  - Request URL (IP_Name/Port No./Path)
    - http://127.0.0.1:8000/predicts
  - HTTP Request Method
    - POST
  - Parameter/Payload
    - {
    - "Age": 24,
    - "Sex": ”female",
    - "Embarked": "C"
    - }
  - Request/Response
    - {
    - "prediction": "[np.int64(0), np.int64(0), np.int64(0)]"
    - }
  - Is need subscriber (e.g. token, key)
    - No



See More...

- FastAPI
  - https://fastapi.tiangolo.com
- WSL with VSCode
  - https://code.visualstudio.com/docs/remote/wsl
- Miniconda
  - https://www.anaconda.com/docs/getting-started/miniconda/main

Docker

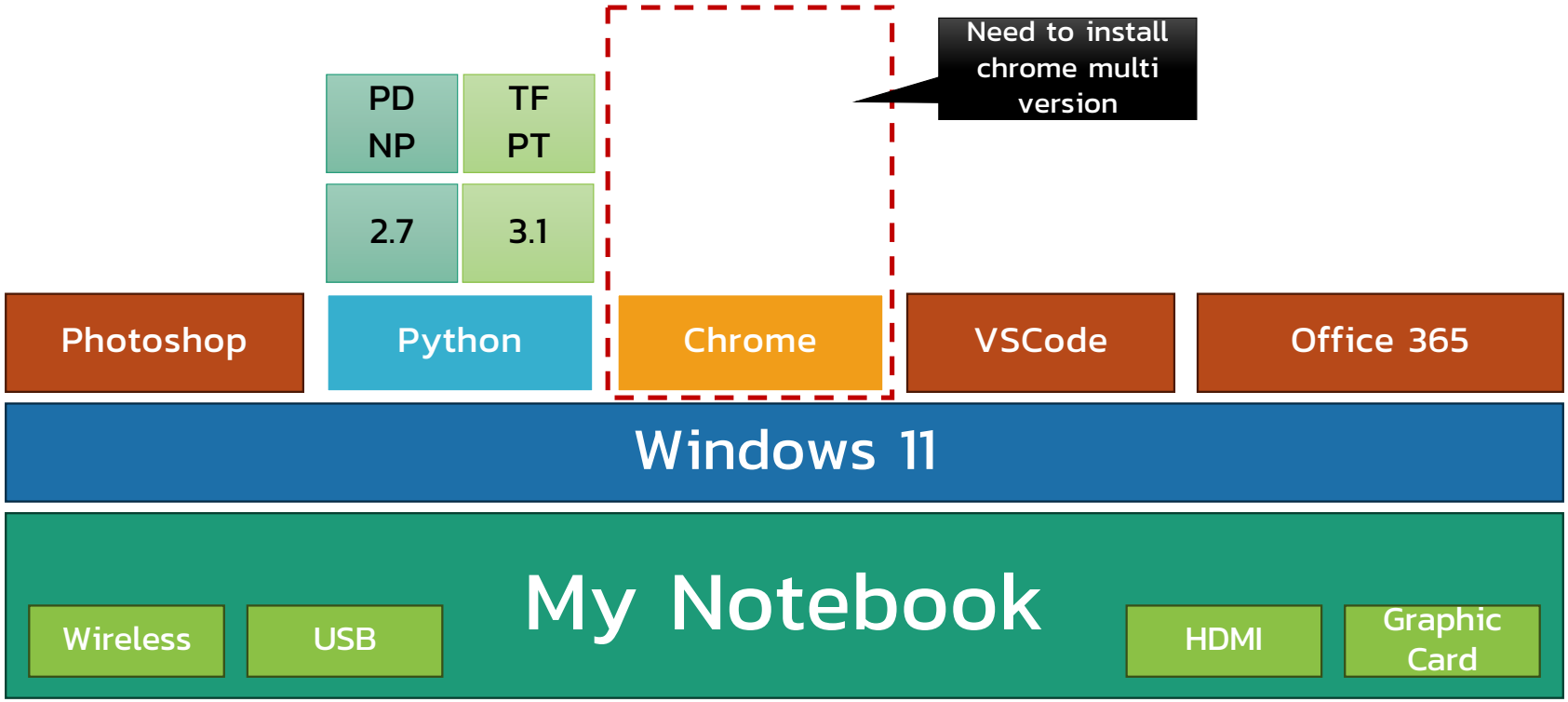
Deploy API in modern world

- Why Docker
  - Running our API on Server or other Computer???
- How Docker
  - Docker usage and command
- Docker Compose
  - Docker and Compose for easier life

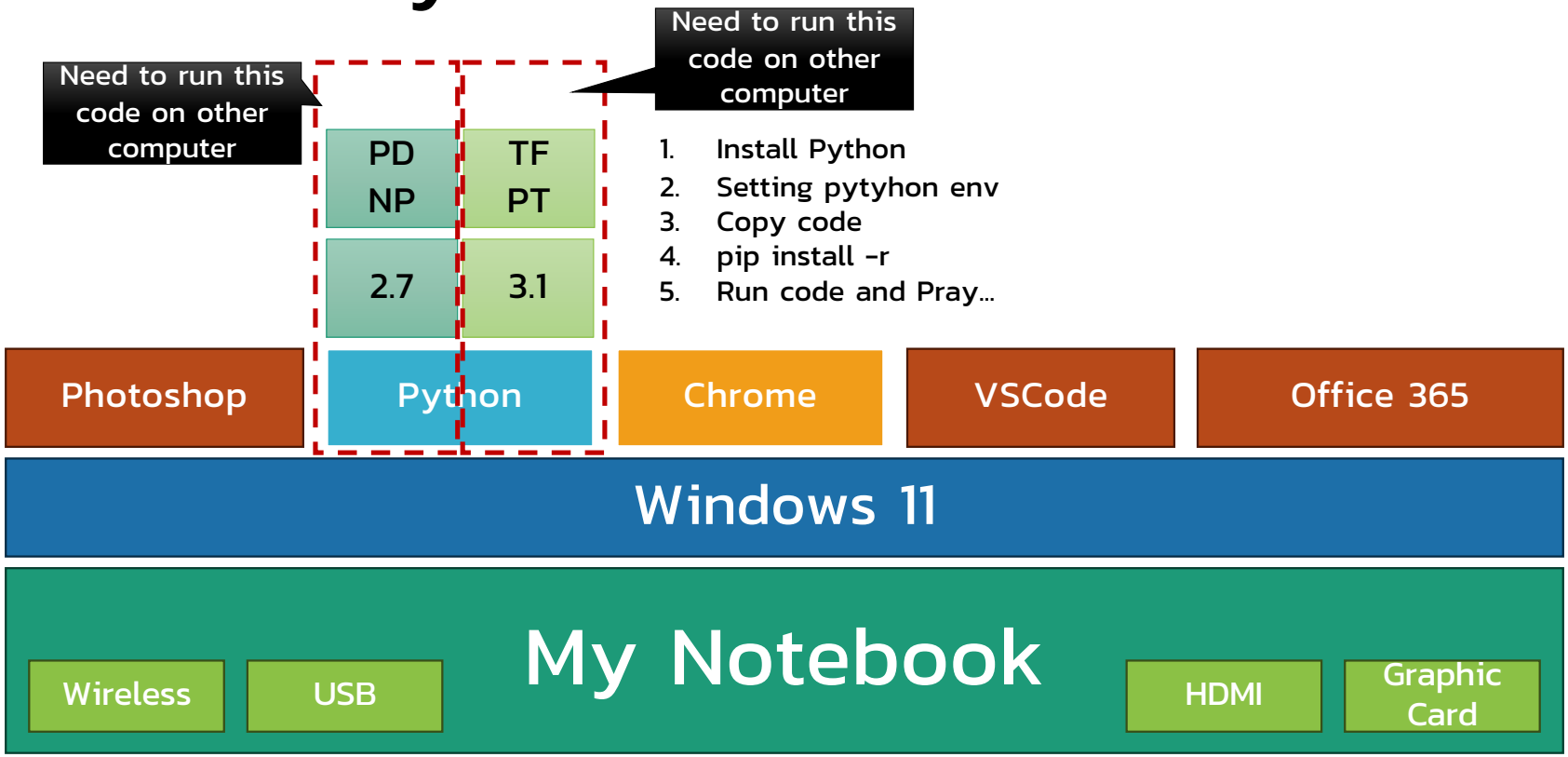
Major Problem on Application Deployment

- Deploy our API or App on Server or Other Computer
- Scaling our API or App for large concurrent of users
- Deployment Strategies and Rollback

Docker: Why


Need to install chrome multi version
PD
NP
TF
PT
2.7
3.1
Photoshop
Python
Chrome
VSCode
Office 365
Windows 11
My Notebook
Wireless
USB
HDMI
Graphic Card

Docker: Why


Need to run this code on other computer
Need to run this code on other computer
PD
NP
TF
PT
2.7
3.1
1. Install Python
2. Setting pytyhon env
3. Copy code
4. pip install -r
5. Run code and Pray…
Photoshop
Python
Chrome
VSCode
Office 365
Windows 11
My Notebook
Wireless
USB
HDMI
Graphic Card

Scale-up and Scale-out

Image source: https://inverodigital.com/insights/digital-transformation-the-importance-of-scalability/

Deployment Strategies

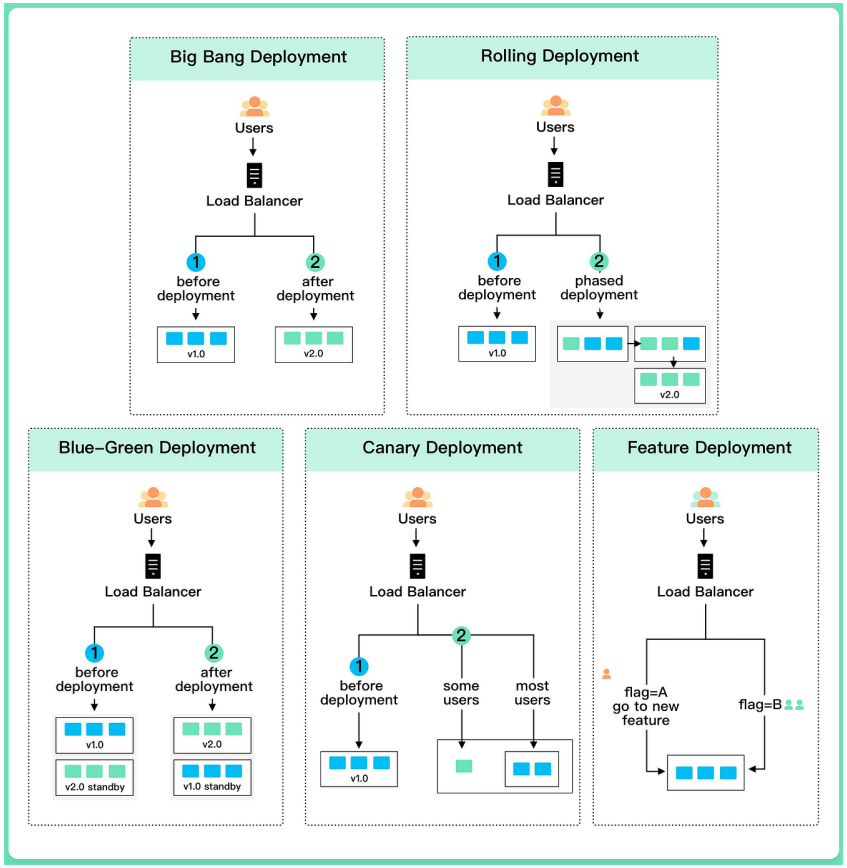


Image source: https://harshityadav95.medium.com/notes-top-5-deployment-strategies-cc4a04ace01c

Docker: Alternative

- https://signoz.io/comparisons/docker-alternatives/

Docker: Overall application development workflow

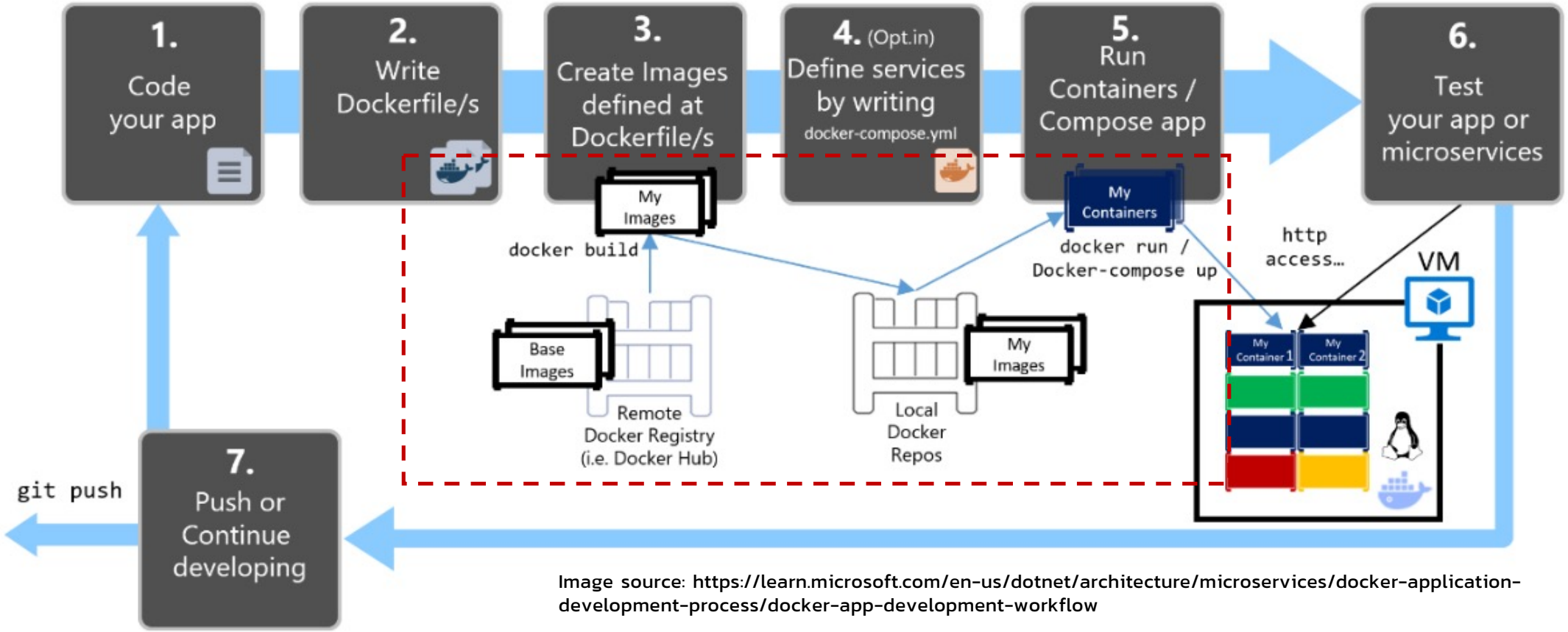


Image source: https://learn.microsoft.com/en-us/dotnet/architecture/microservices/docker-application-development-process/docker-app-development-workflow

Image source: https://cto.ai/blog/docker-image-vs-container-vs-dockerfile/

Docker: Basic Concept

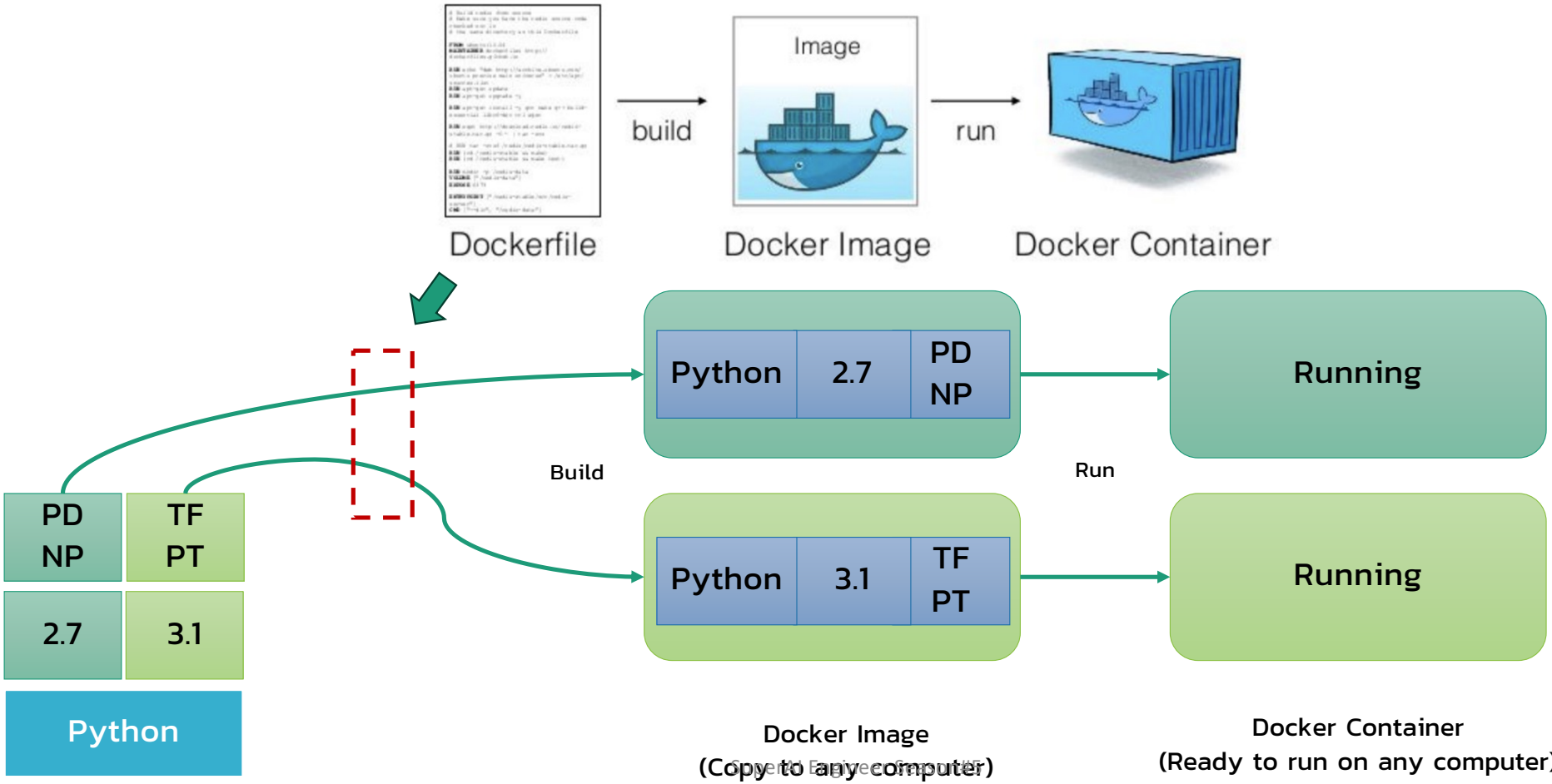
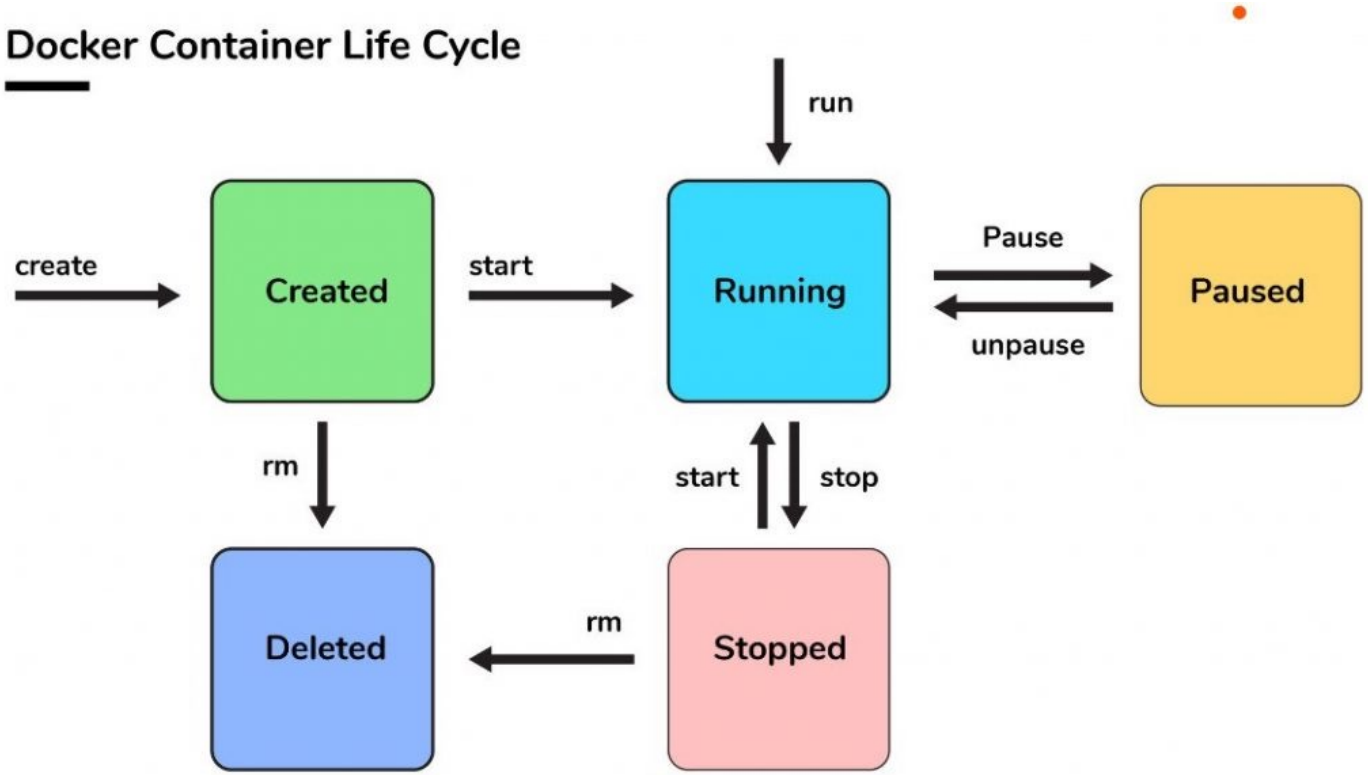
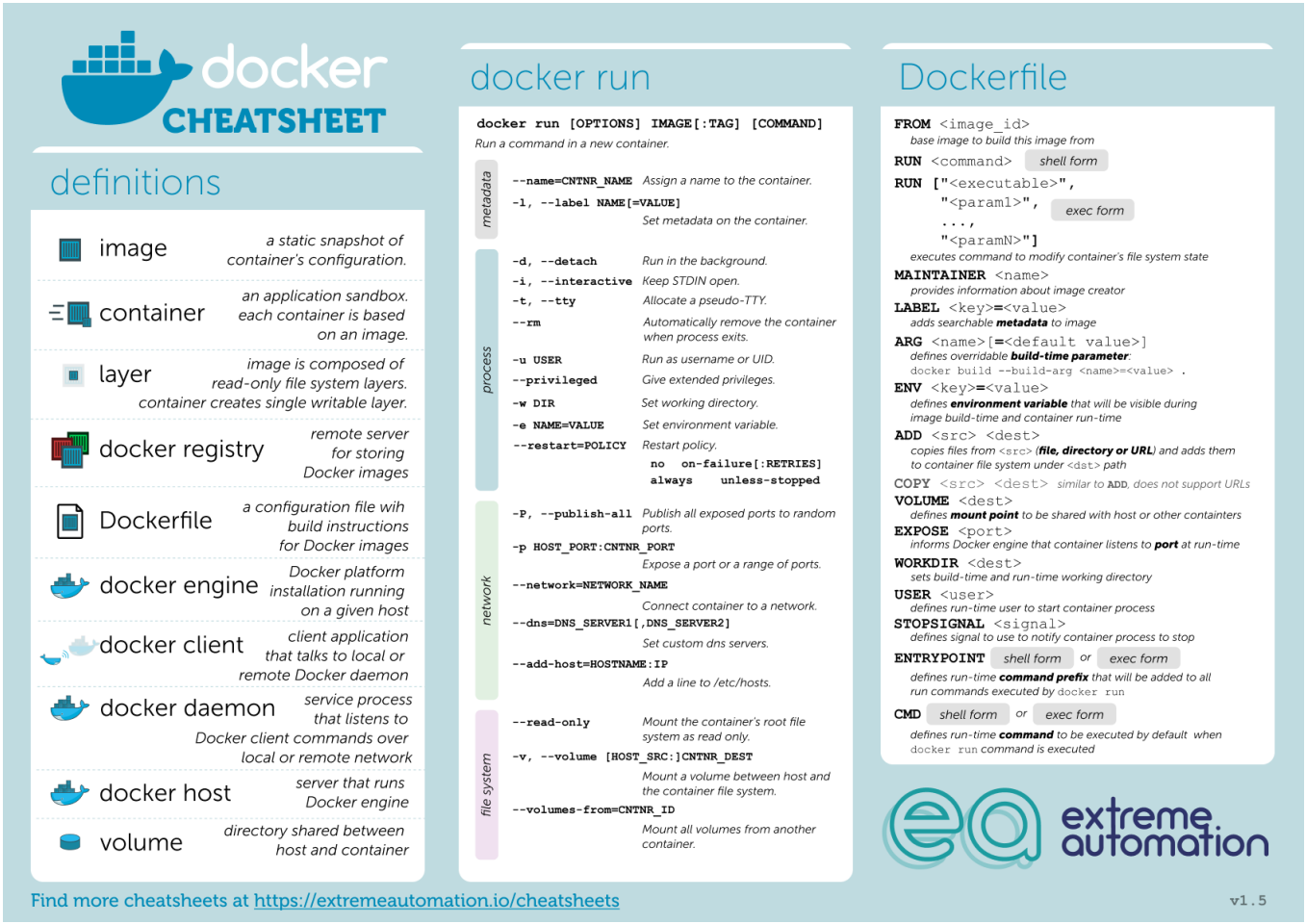


Image source: https://www.devopsschool.com/blog/lifecycle-of-docker-containers/

Docker: Container Life Cycle



Docker Cheat Sheet (Basic)



Docker Cheat Sheet (Command)

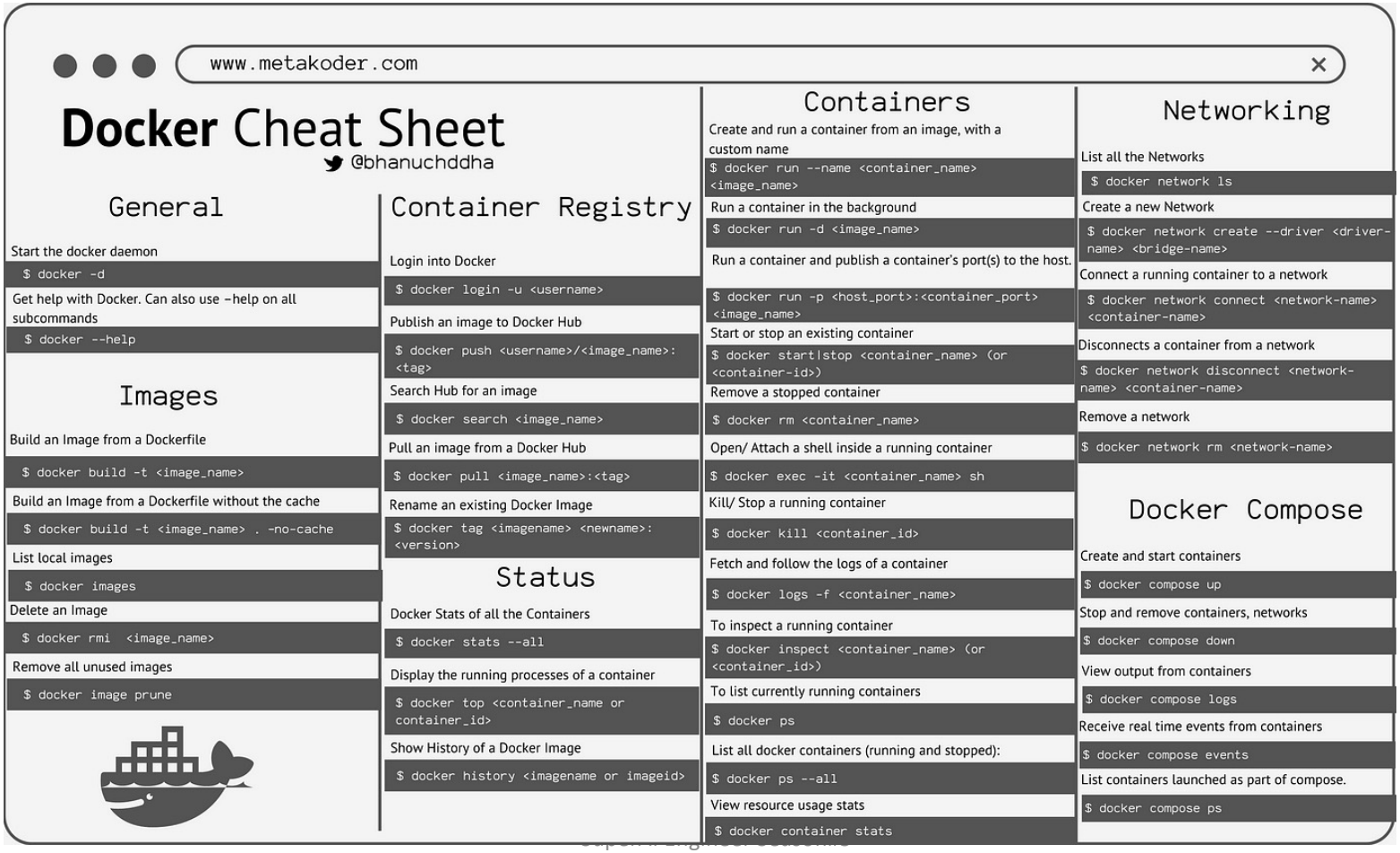
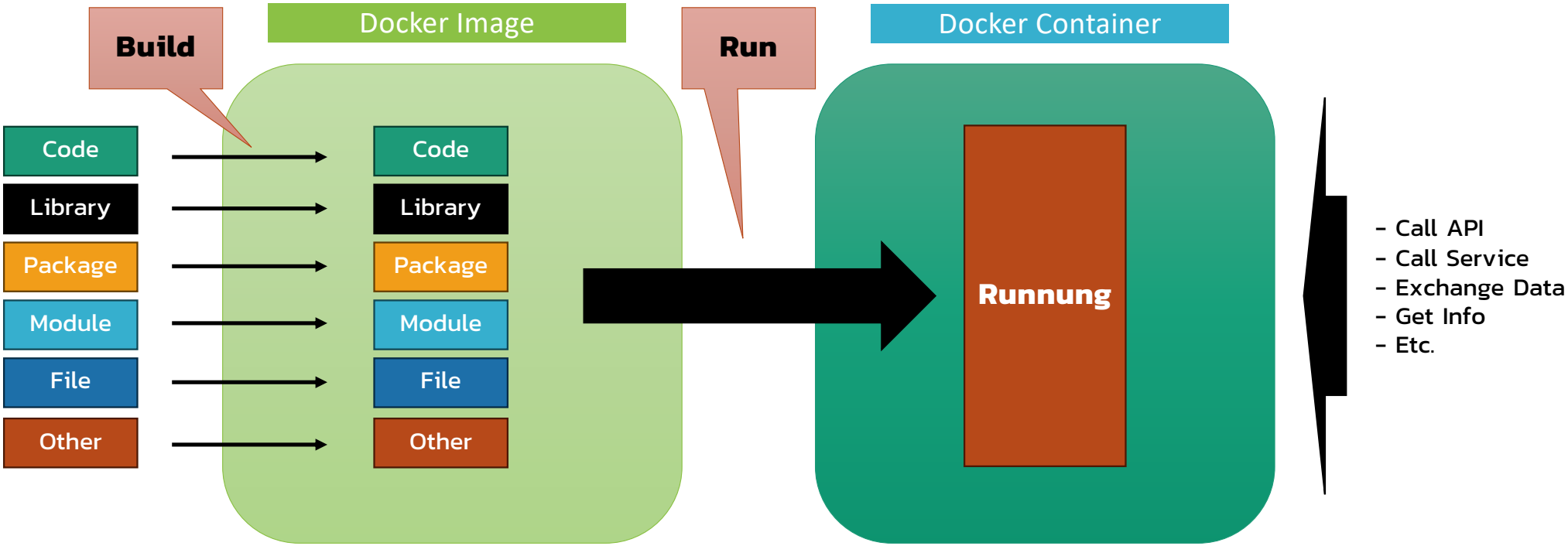


Image source: https://medium.com/@bhanuchaddha/unlocking-docker-the-ultimate-cheat-sheet-and-guide-79c3dfdee233

Wrap-up Docker Flow



Basic Docker Command

Windows user	-> WSL Terminal
Mac user	-> Terminal
Linux user	-> Terminal

- List Docker images
  - sudo docker images

```
pongier@Windows11:~$ sudo docker images
[sudo] password for pongier:
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
hello-world   latest    74cc54e27dc4   3 months ago    10.1kB
```

- List Docker container
  - sudo docker ps -a

```
pongier@Windows11:~$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED      STATUS                     PORTS     NAMES
89c38ee218df   hello-world   "/hello"   3 days ago   Exited (0) 3 days ago              inspiring_shamir
```

- Useful command
  - sudo docker inspect CONTAINER
  - sudo docker logs CONTAINER
  - sudo docker stat

Start with Readymade Docker image

- Meet with Docker Hub
  - https://hub.docker.com

- Try busybox from Docker Hub
  - sudo docker pull busybox
  - sudo docker run busybox
  - sudo docker run busybox echo "hello from busybox"
  - sudo docker run -it busybox sh

- Check Docker image/Check Docker container

Start with Readymade Docker image

```
pongier@Windows11:~$ sudo docker pull busybox
[sudo] password for pongier:
Using default tag: latest
latest: Pulling from library/busybox
97e70d161e81: Pull complete
Digest: sha256:37f7b378a29ceb4c551b1b5582e27747b855bbfaa73fa11914fe0df028dc581f
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest
pongier@Windows11:~$ sudo docker run busybox
pongier@Windows11:~$ sudo docker run busybox echo "hello from busybox"
hello from busybox
pongier@Windows11:~$ sudo docker run -it busybox sh
/ # ls
bin    dev    etc    home   lib    lib64  proc   root   sys    tmp    usr    var
/ # exit
pongier@Windows11:~$
```

Basic Docker Command

```
pongier@Windows11:~$ sudo docker ps -a
CONTAINER ID   IMAGE         COMMAND                  CREATED         STATUS                     PORTS     NAMES
f3f40c4a80b7   busybox       "sh"                     3 minutes ago   Exited (0) 3 minutes ago             focused_germa
in
86b88b0bf112   busybox       "echo 'hello from bu…"   3 minutes ago   Exited (0) 3 minutes ago             tender_gould
8ee177988d90   busybox       "sh"                     4 minutes ago   Exited (0) 4 minutes ago             optimistic_mc
clintock
89c38ee218df   hello-world   "/hello"                 3 days ago      Exited (0) 3 days ago                inspiring_sha
mir
```

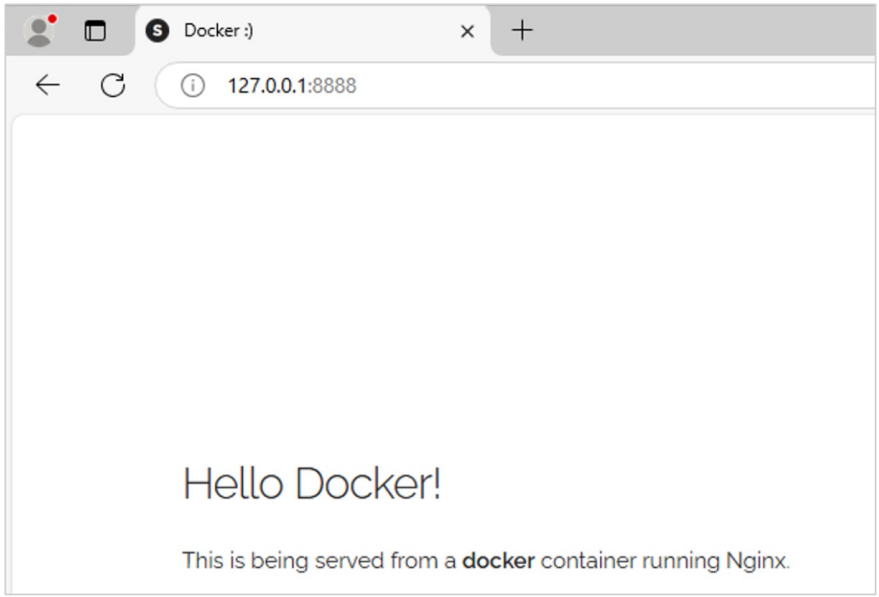
- Remove Docker container
  - sudo docker rm CONTAINER
  - sudo docker rm f3f40c4a80b7

- Remove Docker image
  - sudo docker rmi IMAGE
  - sudo docker rmi hello-world

Example of running web with docker

- Readymade Docker named “prakhar1989/static-site”
  - sudo docker pull prakhar1989/static-site
  - sudo docker run -p **8888**:**80** prakhar1989/static-site
  - Try to open your browser with http://127.0.0.1:8888/

```
pongier@Windows11:~$ sudo docker run -p 8888:80 prakhar1989/static-site
Nginx is running...
```



Improvement Docker Running

- Exit from current Docker Container running
  - Ctrl + c

- Find and Remove above
  - sudo docker ps -a
  - sudo docker rm CONTAINER

```
pongier@Windows11:~$ sudo docker ps -a
CONTAINER ID   IMAGE                  COMMAND                  CREATED          STATUS                      PORTS
 NAMES
8f9f925f364a   prakhar1989/static-site   "./wrapper.sh"           38 seconds ago   Exited (0) 13 seconds ago
 sleepy_visvesvaraya
f3f40c4a80b7   busybox                "sh"                     10 hours ago     Exited (0) 10 hours ago
 focused_germain
86b88b0bf112   busybox                "echo 'hello from bu…"   10 hours ago     Exited (0) 10 hours ago
 tender_gould
8ee177988d90   busybox                "sh"                     10 hours ago     Exited (0) 10 hours ago
 optimistic_mcclintock
89c38ee218df   hello-world            "/hello"                 3 days ago       Exited (0) 3 days ago
 inspiring_shamir
pongier@Windows11:~$ sudo docker rm 8f9f925f364a
8f9f925f364a
```

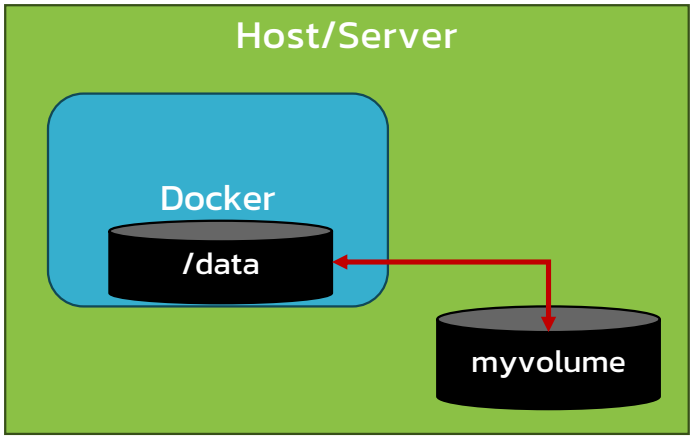
- Improve by running command (-d is Detach)
  - sudo docker run **-d** -p 8888:80 prakhar1989/static-site

```
pongier@Windows11:~$ sudo docker run -d -p 8888:80 prakhar1989/static-site
f20f4ca9b4f8754d917b2b9ba6103c07d91323bdac16ab17e4c6bd3bd9a35ea1
```

Also Storage/Volumn Mapping

- Because Docker Container is not Persistant
  - When container remove all change data is looooose!!!

- Understanding command (-v is Volume)
  - sudo docker run -d **-v myvolume:/data** example/docker
    - myvolume is Host/Server created volume
    - /data is Container volume



Wrap-up: Overall application development workflow

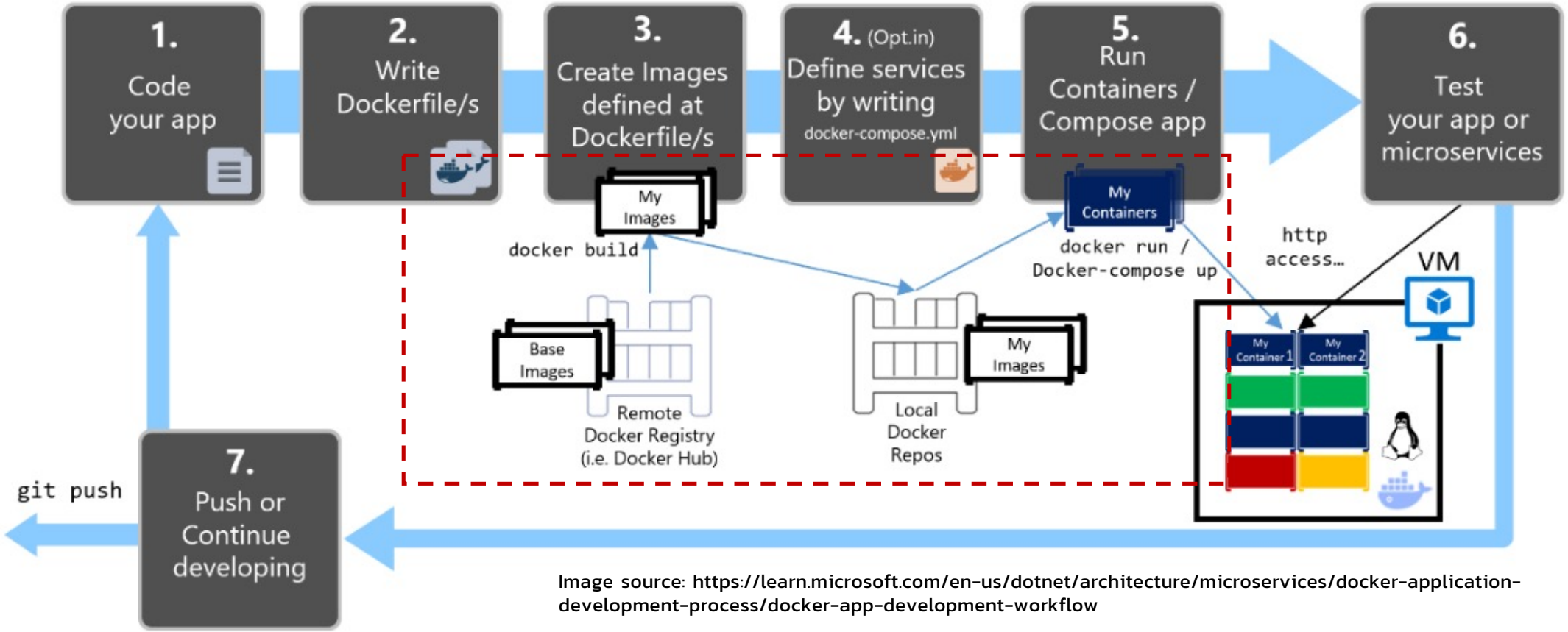


Image source: https://learn.microsoft.com/en-us/dotnet/architecture/microservices/docker-application-development-process/docker-app-development-workflow

Build Your Own Docker Image

- Code your app or Get some code

Windows User: (Working on WSL Terminal)

```
cd /mnt/c/API01/
```

Mac User : (Working on MacOSX Terminal)

```
cd /home/YOUR_NAME/API01/
```

Linux User : (Working on Linux OS Terminal)

```
cd /home/YOUR_NAME/API01/
```

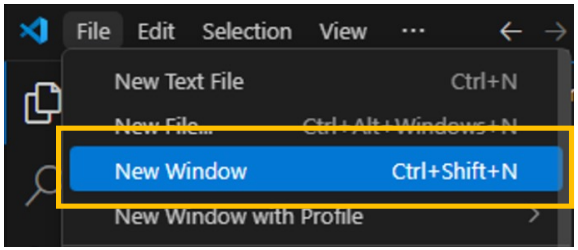
Build Your Own Docker Image

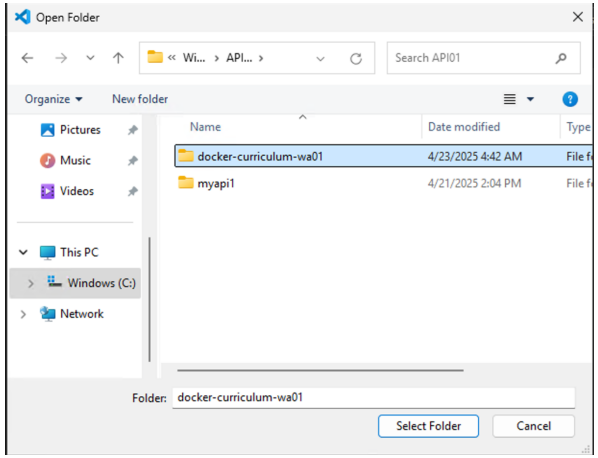
- 1. Code your app or Get some code
  - FOLLOWING YOUR OS USER COMMAND
  - git clone https://github.com/pongier/docker-curriculum-wa01.git
  - cd docker-curriculum-wa01/flask-app/

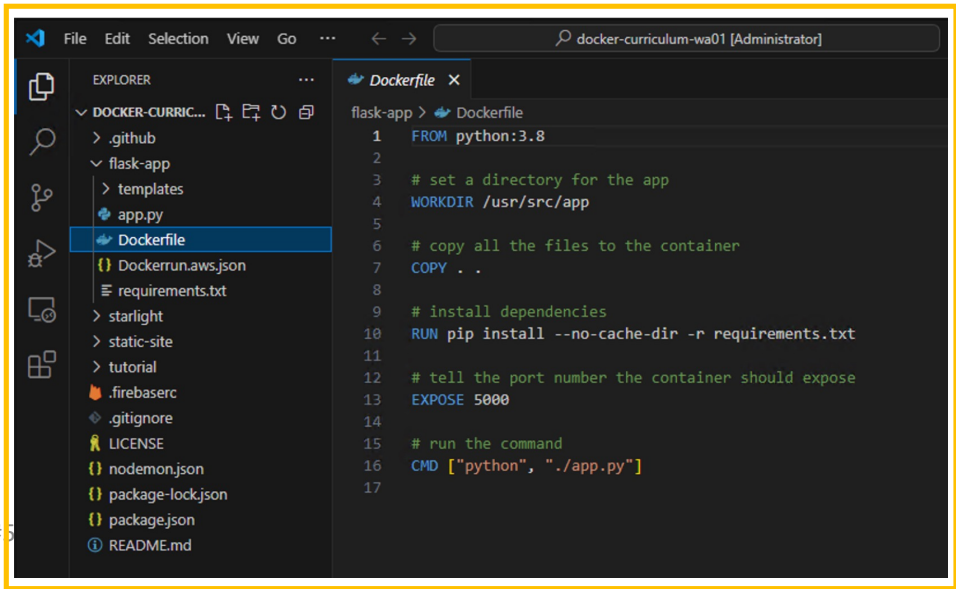
```
pongier@Windows11:~$ cd /mnt/c/API01/
pongier@Windows11:/mnt/c/API01$ git clone https://github.com/pongier/docker-curriculum-wa01.git
Cloning into 'docker-curriculum-wa01'...
remote: Enumerating objects: 1649, done.
remote: Counting objects: 100% (57/57), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 1649 (delta 43), reused 23 (delta 23), pack-reused 1592 (from 3)
Receiving objects: 100% (1649/1649), 9.12 MiB | 5.58 MiB/s, done.
Resolving deltas: 100% (932/932), done.
Updating files: 100% (172/172), done.
pongier@Windows11:/mnt/c/API01$ cd docker-curriculum-wa01/flask-app/
pongier@Windows11:/mnt/c/API01/docker-curriculum-wa01/flask-app$
```

Build Your Own Docker Image

- 2. Writing your Dockerfile or Customize some Dockerfile
  - Using VSCode to open Dockerfile in docker-curriculum/flask-app







Understanding Dockerfile

```
FROM python:3.8   #Base Docker Image

# set a directory for the app
WORKDIR /usr/src/app

# copy all the files to the container
COPY . .

# install dependencies
RUN pip install --no-cache-dir -r requirements.txt

# tell the port number the container should expose
EXPOSE 5000

# run the command
CMD ["python", "./app.py"]
```

Build Your Own Docker Image

- 3. Build your Docker Image (Defined from Dockerfile)
  - sudo docker build -t yourusername/yourapp .

```
pongier@Windows11:/mnt/c/API01/docker-curriculum-wa01/flask-app$ sudo docker build -t yourusername/yourapp .
[+] Building 8.7s (9/9) FINISHED                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                        0.1s
 => => transferring dockerfile: 340B                                                                        0.0s
```

  - sudo docker images

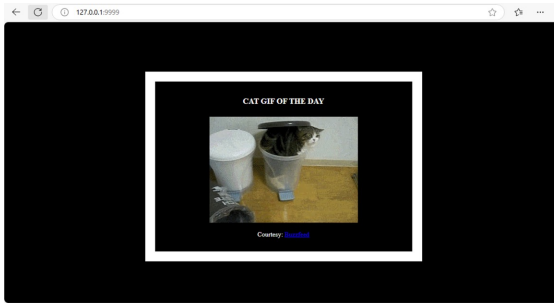
```
pongier@Windows11:/mnt/c/API01/docker-curriculum-wa01/flask-app$ sudo docker images
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
yourusername/yourapp      latest    8e548b66559b   3 minutes ago    1.01GB
<none>                    <none>    485e449b3813   16 minutes ago   1.01GB
<none>                    <none>    db4b44bda01e   34 minutes ago   1.01GB
hello-world               latest    74cc54e27dc4   3 months ago     10.1kB
busybox                   latest    ff7a7936e930   6 months ago     4.28MB
prakhar1989/static-site   latest    f01030e1dcf3   9 years ago      134MB
```

Run Your Own Docker Container

- 5. Docker Runnnnnnn
  - sudo docker run -p 9999:5000 yourusername/yourapp

```
pongier@Windows11:/mnt/c/API01/docker-curriculum-wa01/flask-app$ sudo docker run -p 9999:5000 yourusername/yourapp
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.3:5000
Press CTRL+C to quit
```

- 6. Testing by open Web browser http://127.0.0.1:9999



Checking Docker Container

- Checking something
  - sudo docker ps -a
- Improve something
  - sudo docker run -d -p 9999:5000 yourusername/yourapp

Your API with Your Docker

- Remember Requirement
  - Python Version
    - python=3.12.9
  - Required Package
    - "fastapi[standard]"
    - "uvicorn[standard]"
    - install numpy
    - pandas
    - scikit-learn
  - Port
    - 10000

Understanding requirements.txt

- requirements.txt (Create in same directory of project)
  - is a file that contains a list of packages or libraries needed to work on a project that can all be installed with the file.

- List all packages or libraries that installed in this environment
  - pip freeze

```
(superapi) pongier@Windows11:/mnt/c/API01/myapi1$ pip freeze
annotated-types @ file:///home/conda/feedstock_root/build_artifacts/annotated-types_1733247046149/work
anyio @ file:///home/conda/feedstock_root/build_artifacts/bld/rattler-build_anyio_1742243108/work
certifi @ file:///home/conda/feedstock_root/build_artifacts/certifi_1739515848642/work/certifi
```

requirements.txt of your API

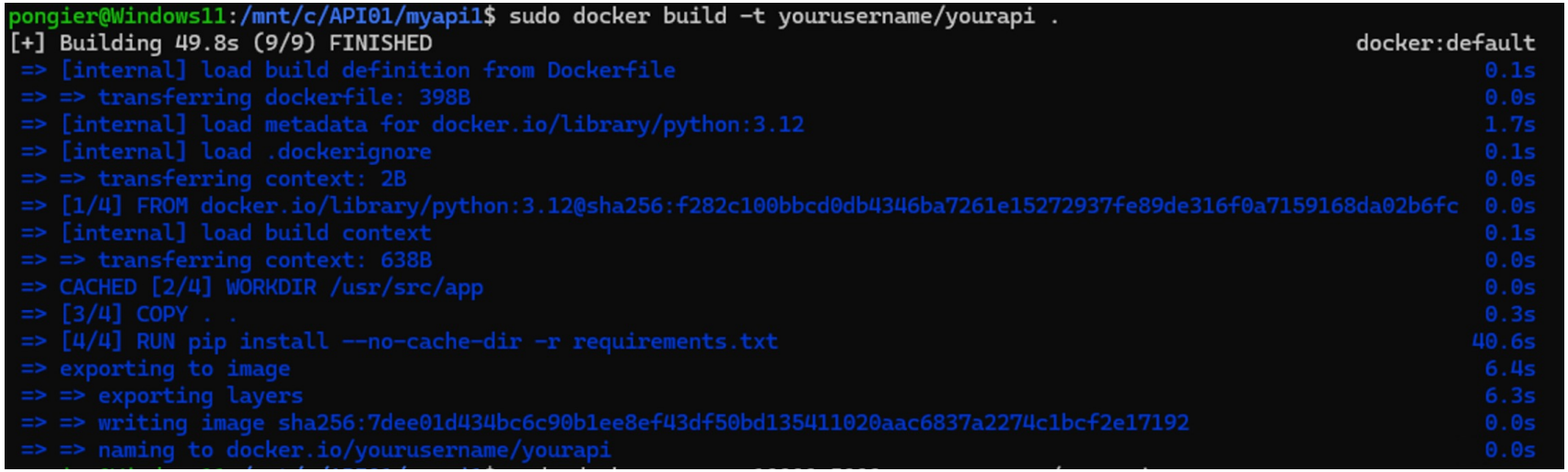
```
numpy==2.2.5
pandas
scikit-learn
fastapi[standard]
uvicorn[standard]
```

Create Dockerfile for your API

```
FROM python:3.12

# set a directory for the app
WORKDIR /usr/src/app

# copy all the files to the container
COPY . .

# install dependencies
RUN pip install --no-cache-dir -r requirements.txt

# tell the port number the container should expose
EXPOSE 5000

# run the command
CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "9000", "--workers", "2"]
```

Building Docker Image

- Buildinggggg...
  - sudo docker build -t yourusername/yourapi .



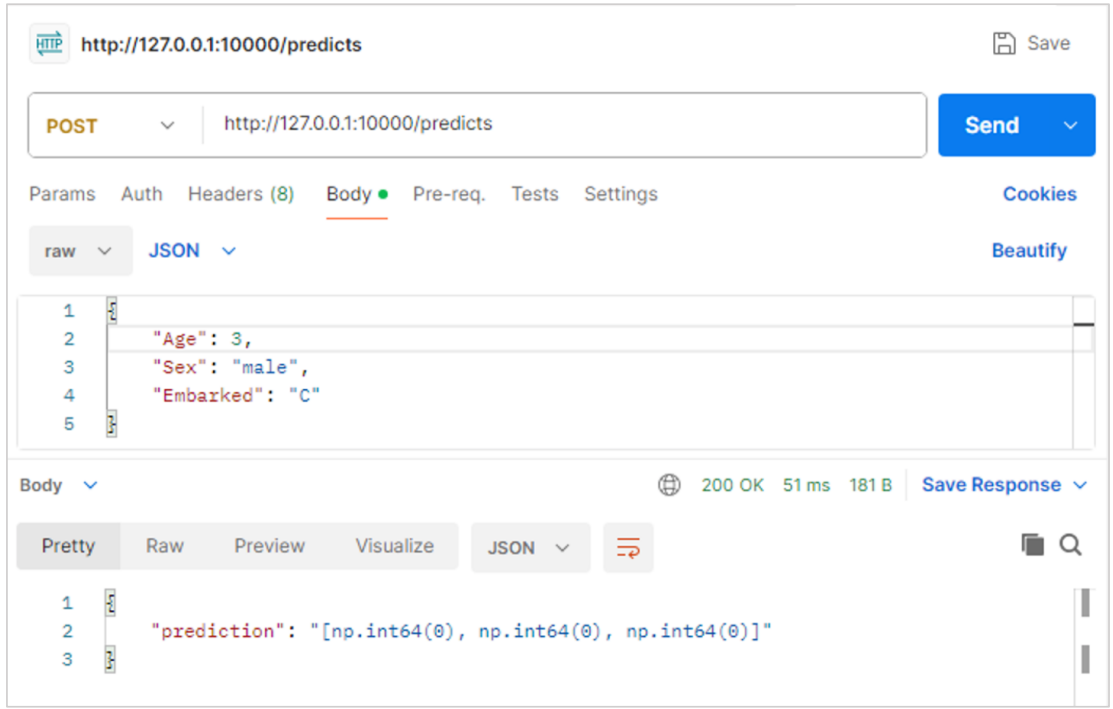
Running

- Runninggggg...
  - sudo docker run -p 10000:9000 yourusername/yourapi

```
pongier@Windows11:/mnt/c/API01/myapi1$ sudo docker run -p 10000:9000 yourusername/yourapi
INFO:     Uvicorn running on http://0.0.0.0:9000 (Press CTRL+C to quit)
INFO:     Started parent process [1]
INFO:     Started server process [10]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Started server process [9]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     172.17.0.1:54388 - "POST /predicts HTTP/1.1" 200 OK
```

Testing Your API

- Testinggggg...
  - Postman
  - http://127.0.0.1:10000/predicts



Your work, Image API and Docker

- Hint...
  - https://docs.leonardo.ai/docs/getting-started
  - https://github.com/enriquecatala/fastapi-ai-template
  - https://pub.aimind.so/building-a-image-generator-with-stable-diffusion-fastapi-and-modern-web-tech-3b5ff2b37c71

Something in Advanced

- Security
  - HTTPS
  - API Authentication/Authorization/Accounting
  - CORS
- Running on startup
- Restarts
- Multi Workers (the number of processes running)
- More advanced...In the real worlddddd.

See More...

- Docker
  - https://docs.docker.com/manuals/
- Docker Tutorial
  - https://docker-curriculum.com
- Play With Docker
  - https://labs.play-with-docker.com/
- Docker Compose (Recommend)
  - https://docs.docker.com/compose/

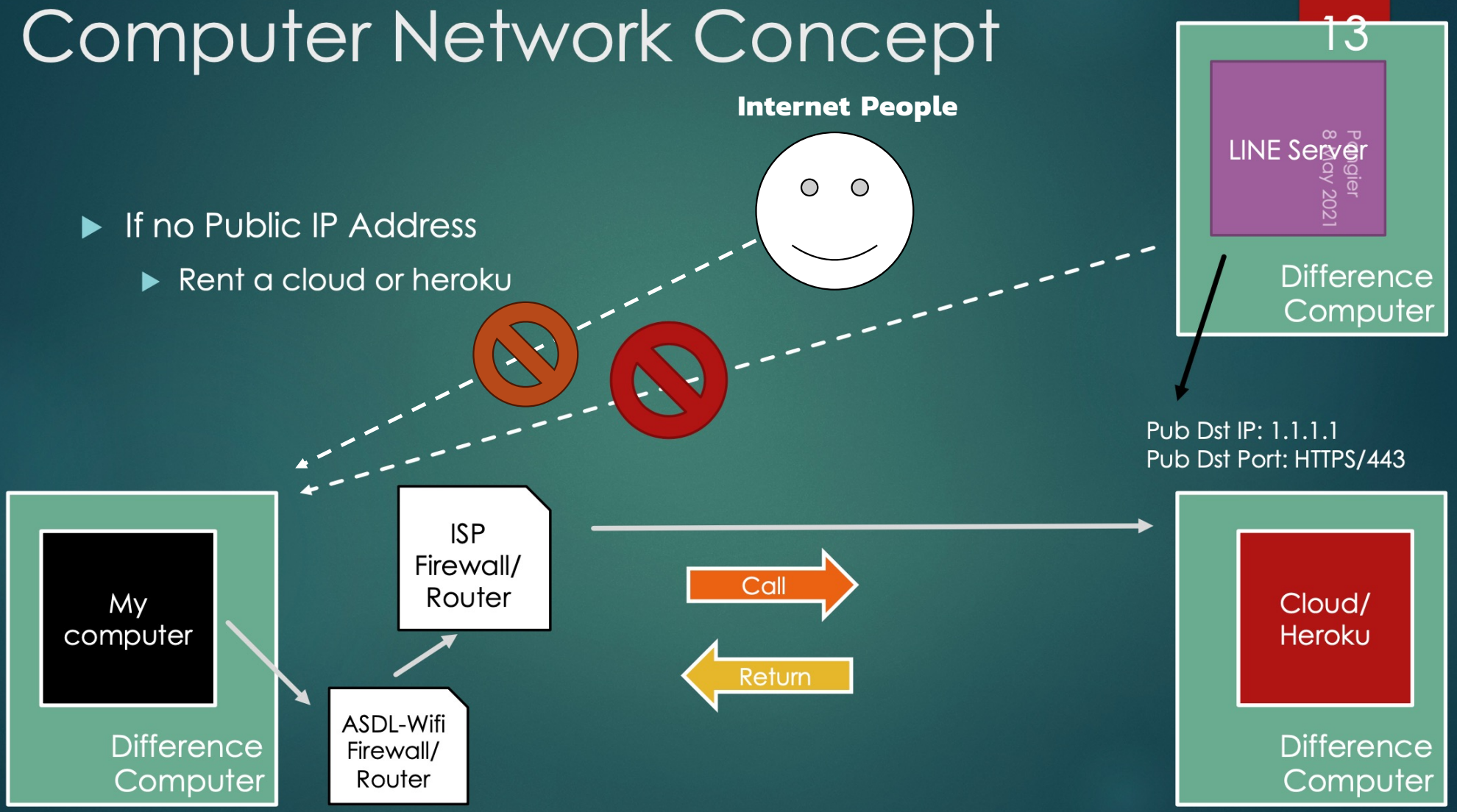
Ngrok

Explore The World

- Only have Home Internet (e.g. ADSL)
- But need Internet People could access your Application/API
- Beware!!!
  - Internet People in the world can see you.

Computer Network Concept

- If no Public IP Address
  - Rent a cloud or heroku

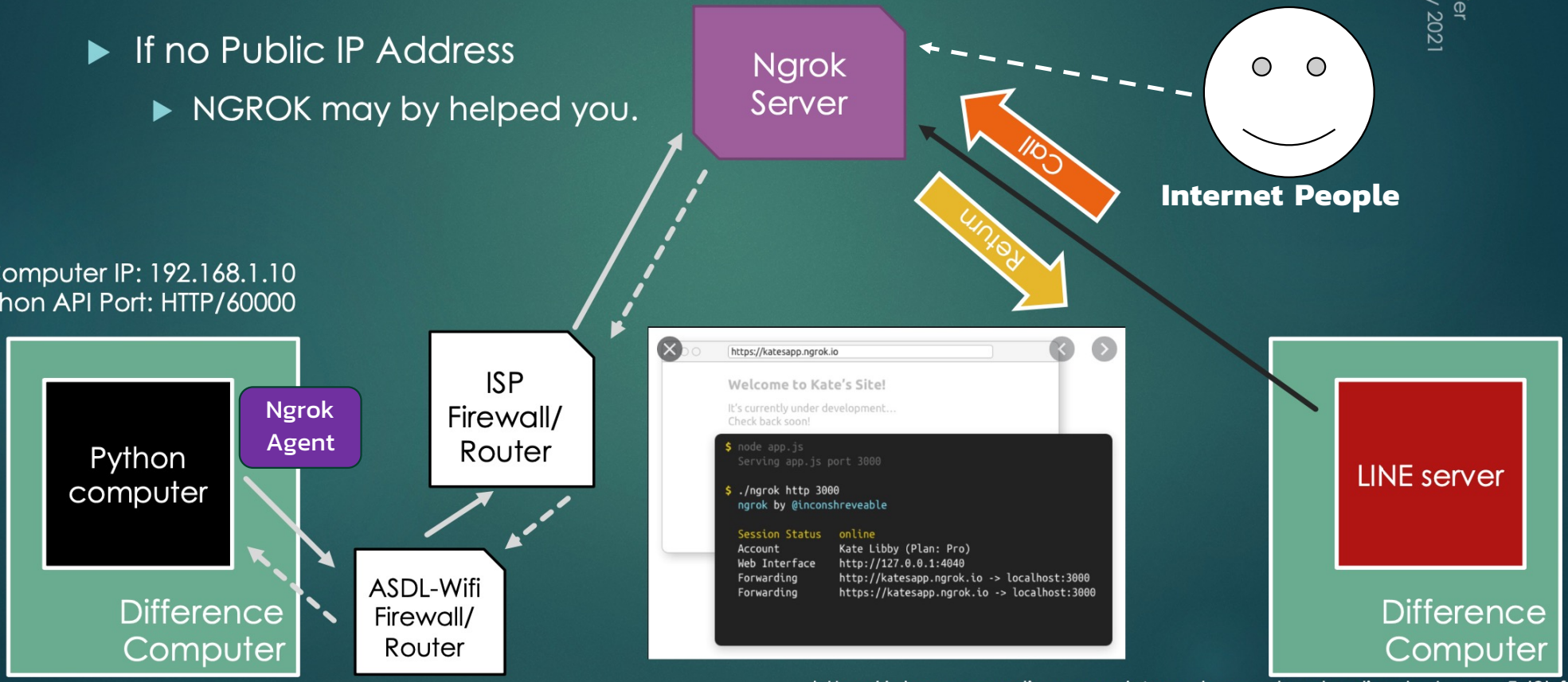


Computer Network Concept

Pongjer
8 May 2021

Ngrok Pub Src IP: 203.204.205.99 (Assign with name)
Ngrok Pub Src Port: HTTP/10000 (Assigh per source computer)

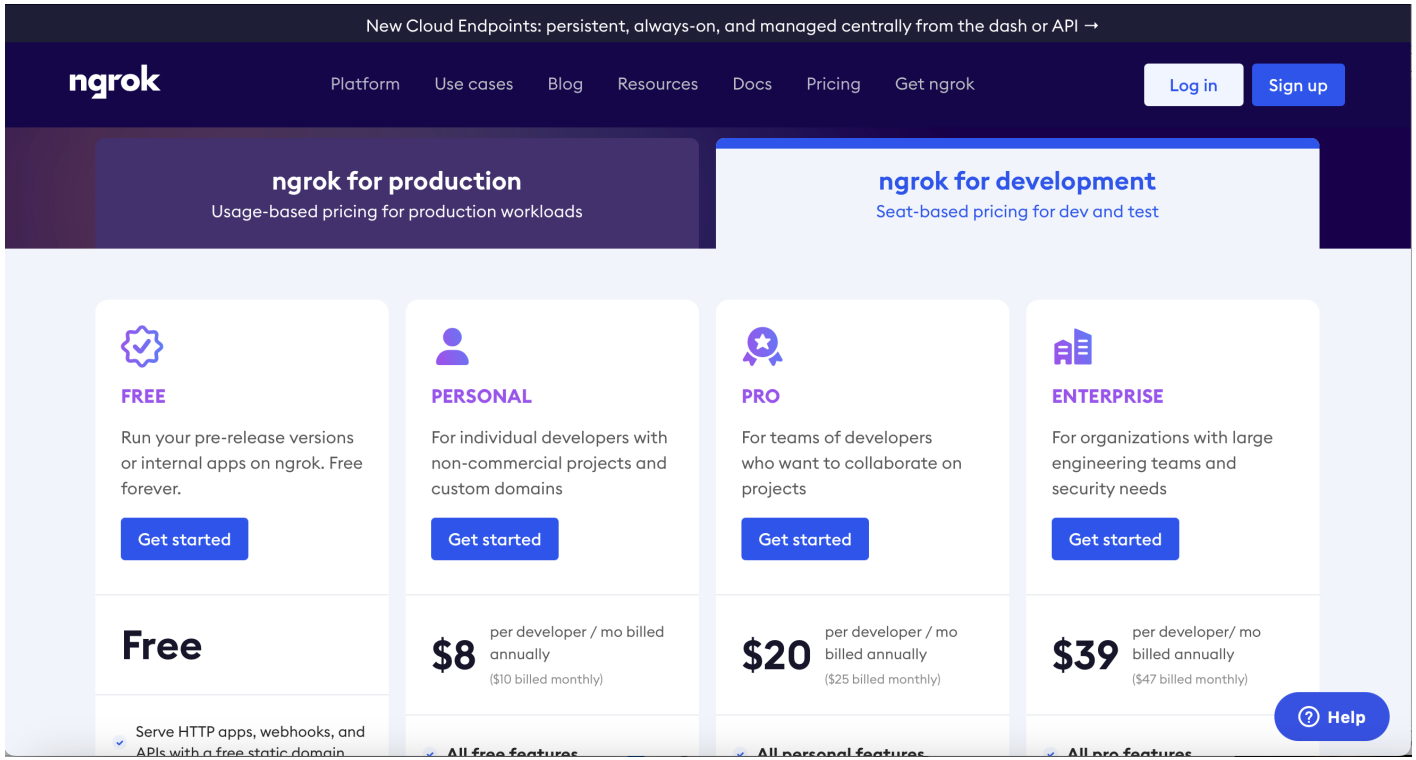
- If no Public IP Address
  - NGROK may by helped you.



https://phayao.medium.com/ใช้-ngrok-กับการ-develop-line-bot-อย่างง่าย-5d8bf12e968f

Ngrok might help ypu

- Only have Home Internet (e.g. ADSL)
  - https://ngrok.com



Make sure your API or Docker is online

- Checking
  - sudo docker ps -a

```
(superapi) pongier@Windows11:/mnt/c/API01/myapi1$ sudo docker ps -a
[sudo] password for pongier:
CONTAINER ID   IMAGE                     COMMAND                  CREATED       STATUS                  PORT
S                                                  NAMES
e5f9d913c9f5   yourusername/yourapi      "uvicorn main:app --…"   2 hours ago   Up 2 hours              5000
/tcp, 0.0.0.0:10000->9000/tcp, [::]:10000->9000/tcp   youthful_hypatia
```

Getting Start with Ngrok

- https://ngrok.com/docs/getting-started/?os=windows

- Register and Login for an ngrok account
  - https://dashboard.ngrok.com/login

- Copy your **ngrok authtoken** from your ngrok dashboard

Getting Start wit Ngrok

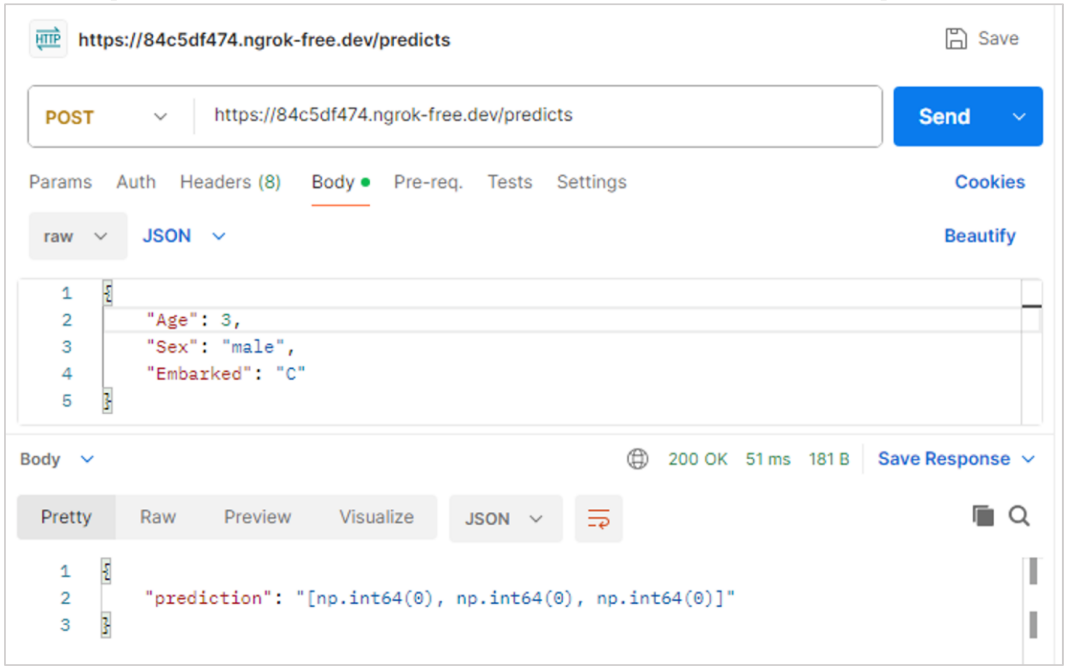
- Run the following command in your terminal to install the authtoken and connect the Ngrok agent to your account.
  - ngrok config add-authtoken TOKEN

- Put your app online
  - ngrok http http://127.0.0.1:8080

```
ngrok                                                    (Ctrl+C to quit)

Session Status                online
Account                       inconshreveable (Plan: Free)
Version                       3.0.0
Region                        United States (us)
Latency                       78ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://84c5df474.ngrok-free.dev -> http://localhost:8080

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Testing Your API same as Internet Poeple

- Testing
  - Postman
  - https://84c5df474.ngrok-free.dev/predicts



Q&A

If any problem,
Please feel free to contact me.


SEASON 5
SUPER AI ENGINEER
AI INNOVATOR • AI ENGINEER • AI RESEARCHER
AI
AI
AI