1. Project Setup

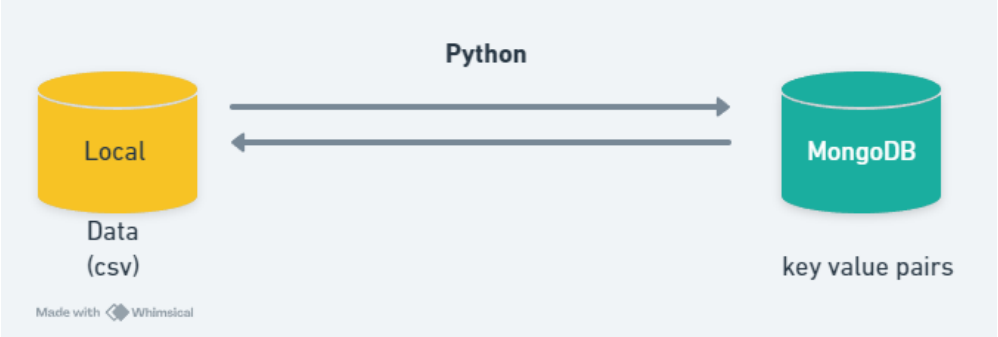
2. Create a directory in your local
3. Open it in the vscode
4. Create a read me file
5. Publish the branch and create a repo in the github
6. In github create the lisence and .gitignore file
7. Create an env
8. Template.py
9. And requirements.txt
10. Then setup.py

2. MongoDB Setup

Mongodb:

1. Setup
    a) Log into MongoDB
    b) Project → new project
    c) Project name : _____ → next
    d) Keep everything in default → create a project
    e) Deploy the model → MO , AWS and keep everything in default
    f) Deploy the model
    g) Username : annaelsaluiz & password: 3vppgFTRlC4SMjVk
    h) Create database user
    i) If we are using 0.0.0.0/0 we can acces the cluster from any server
    j) Using python client to connect with DB and push the data and reteive it from the DB.
    k) Connect→ Drivers → python : 3.6 or later
    l) Copy the connection String : mongodb+srv://name:<password>@cluster0.m4dqupu.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0

2. How to store the data into mongoDB



Convert the data stored in the csv format to dictionary before pushing it to MongoDB

```
DB_NAME= "Restaurant Rating"
COLLECTION_NAME = "zomato"
CONNECTION_URL=
"mongodb+srv://                                @cluster0.h4yexmb.mongodb.net/?ret
ryWrites=true&w=majority&appName=Cluster0"
```

```
import pymongo
```

```
client = pymongo.MongoClient(CONNECTION_URL)
data_base = client[DB_NAME]
collection = data_base[COLLECTION_NAME]
record = collection.insert_many(data)
```

Go to atlas → database → browse collection

Can see the data in MongoDB atlas

To retrieve data from DB ;

```
records = collection.find()
records
```

```
for i , j in enumerate(records):
    print(f"{i} :{j}")
```

```
data = pd.DataFrame(list(collection.find()))
data.head()
```

3.AWS Setup

Deployment

In VS code

1. Update the DockerFile
2. Update the .dockerignore
3. Create a folder .github→ inside the folder create a folder workflows
    - If you want to perform CI/CD with the help of github actions you must create this two folder
4. Inside that create aws.yaml
    - Inside this file we will mention all the CI /CD related commands

In AWS account

1. Log into AWS console
2. IAM → users→ create user → <username > next → attach policies directly → Administrator access → next → create user
3. User → security credentials → create access key → Command Line Interface (CLI) → Confirmation → next → create access key → download .csv file → done

ECR

1. Elastic Container Registry (ECR) → us east 1
2. Create repository → private → <repository name > create repository

EC2

1. EC2 → launch instance → <name:usvisa-machine>→ ubuntu → Ubuntu 20.04 LTS (HVM) , SSD Volume Type (choose the older one)
2. Instance type : t2.large
3. Key pair → create new key pair → <usvisakey> → create key pair
4. Select alooww ssh traffic form , allow HTTPS traffic from the internet
5. Configure storage : atleast 32 gb
6. Launch instance
7. View instance → click on the instance running → connect → coonect
8. Terminal appears : →
   ls: to check if there are some files
   touch test.txt : to create a file

Since the ubuntu machine is new , we have to upgrade :

I. sudo apt -get update -y
II. sudo apt -get upgraded

Now the machine is upgraded and we have to download and install the docker in the machine

III. curl -fsSL https:/get.docker.com -c get-docker.sh (to download)

IV. sudo sh get-docker.sh(to install)

This is the production server

Now we are addinf docker

V. sudo usermod -aG docker ubuntu

VI. To ckeck whether docker is running docker --version

Connect Aws to github

1. Github → settings → actions → runners → new self-hosted runner → linux
2. Copy the command one by one and execute in the ubuntu machine in EC2
3. GITHUB ACTIONS
4. Press Enter
5. self-hosted (it is in aws.yaml)→ Enter
6. Keep it in default → Enter
7. Enter
8. Run the last command : ./run.sh
9. CONNECTED TO GITHUB AND LISTENING FOR JOBS

10. Go to Github →settings → actions → runners → Self-hosted : idle

11. If you disconnect it (ctrl + C) → it shows offline instead of idle

12. If you want to connect it again run the command ./run.sh

Setup the github secret key

1. There are 4 secret keys : AWS ACCESSS KEY ID , AWS SECRET ACCESS KEY , AWS DEAFULT REGION , ECR REPO
2. To get the secret keys: setttings → secrets and variables → actions → Repository secret → new repository secret
3. Add each secret key one by one

Everything is configured now and we are ready to push

In vscode

1. Push the code into github repo
2. Workflow has started

In AWS EC2

1. EC2 → instances → security → security groups → edit inbound rules → custom TCP : 8080: 0.0.0.0/0 → save rules

2. Copy the public ip and add the port number :8080

How to stop the instance?

1. EC2 → select the instance → actions →terminate
2. ECR → select the Registry → delete
3. S3→ select the bucket
   Delete the model inside the bucket → permanently delete → delete object
   Now delete the bucket →