**ML Project - From Inueron  
  
Class: 57**There was an intro about GitHub – repo, branches…etc.

**Docker**:  
It is used to portable the code by put it in a docker container and making it a docker image.   
By making the project a docker image then that should be uploaded in the docker hub.   
If that project needs to run in another system, then install docker in that system download the docker image using the commands then run the project using commands in the terminal so that project can be run in the system.   
Benefit of using this - no matter what the OS is they can be run without downloading project dependent libraries, environment, IDE.  
  
Now comes the Heroku, the project can be run in Heroku (VM Cloud). Build/download the docker image in the Heroku and run the image using commands. And we will be accessing the project/app using the URL which was generated by Heroku.  
  
**CICD**:  
Continuous Integration - Developing code and pushing it into GitHub by multiple developer.  
Continuous Deployment – Whenever the code is merged into main branch and gets git release then it triggers to create docker image and deployed into Heroku.  
  
Heroku will allow to create 5 application without cost.  
  
To setup CI/CD pipeline in Heroku we need 3 information  
1. HEROKU\_EMAIL =   
2. HEROKU\_API\_KEY =   
3. HEROKU\_APP\_NAME =   
  
  
1. Create Dockerfile in the project directory to write instruction for to create a docker image.  
  
In the DockerFile:  
FROM python:3.7 🡪Which version of python  
COPY ./app 🡪 The entire files in the current directory will be copied to the app folder.  
WORKDIR /app 🡪 Set the work directory.  
RUN pip install -r requirements.txt 🡪 installing the requirements for the project  
EXPOSE $PORT 🡪   
CMD gunicorn –workers=4 –bind 0.0.0.0:$PORT app:app 🡪 whatever the python file name app:app and object name in the python code  
  
Now the docker commands are done in the docker file.  
Install docker in the local system   
  
Now need to build the docker image by using the below command.  
docker build -t <image\_name>:<tagname>) 🡪 Note: image name for docker should be small letters and tagname can be anything  
docker images 🡪 To list down the docker image.  
docker run -p 5000:5000 -e PORT=5000 <image id> 🡪 Run docker image  
Now the search host <localhost5000> in browser  
docker ps 🡪 To check running container in docker  
docker stop <container\_id> 🡪 to stop docker container  
  
2. Create dockerignore file and write down file name to ignore file while creating docker image.  
  
Now to push code for deployement:  
Create .github\workflows folder and in that create main.yaml file.  
We need to create trigger which will be triggered after code is getting pushed in main branch by right some commands in main.yaml file  
  
And deploy it into any cloud service.!  
  
**Class: 59**  
What is gunicorn.?

Folder structure for a project  
In housing folder:  
1. \_\_init\_\_.py  
2. Exception 🡪To handle unexpected exception things  
 \_\_init\_\_.py  
3. Logger 🡪 To track everything  
 \_\_init\_\_.py  
4. Pipeline  
 \_\_init\_\_.py  
5. Component  
 \_\_init\_\_.py  
6. Config  
 \_\_init\_\_.py  
7. Entity  
 \_\_init\_\_.py  
  
First we need to work on logger and exception folder.  
Need to know more about logging  
exception – 2:47  
  
**Class: 60**An intro about Exception and Logger function.  
  
**Class: 61**  
Assigning input parameter for the config in the config entity.py file in the entity folder- 1:14