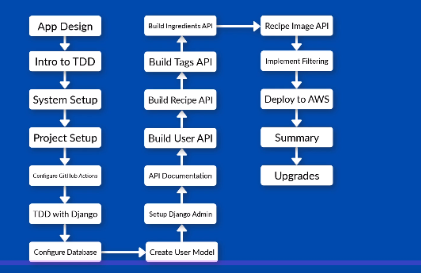
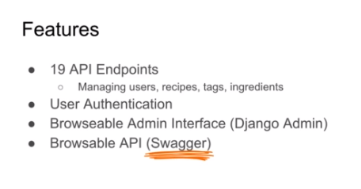
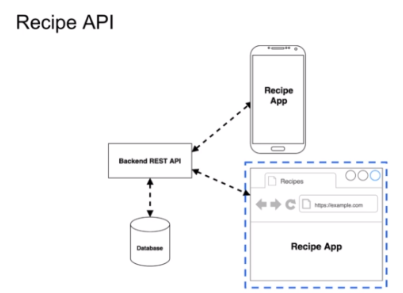
**Django API**

**Course Structure**

****

**Our Project**

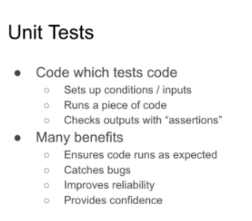
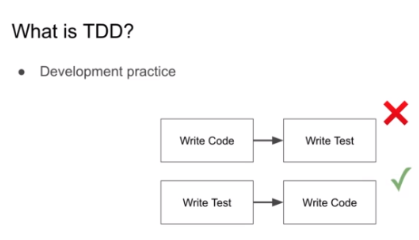
****

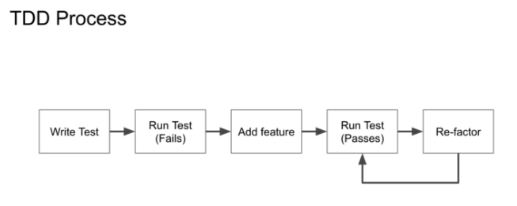
We use **Docker** a containerization software which allow us to run different application services

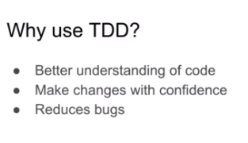
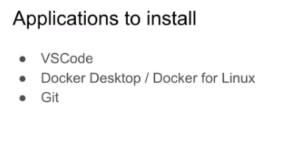
**Swagger API** generate automated documentation for our api

**Our project structure**

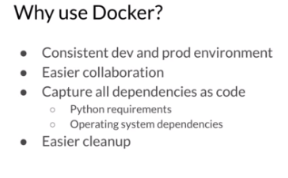
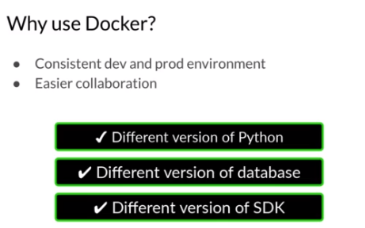


**What is Test Driven Development? **

****

**** ****

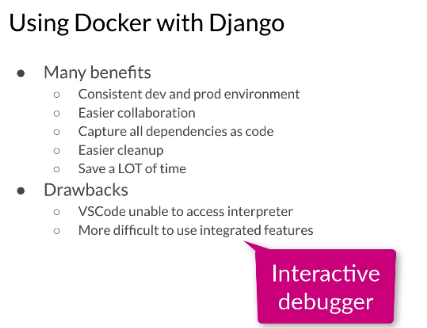
**** ****

****

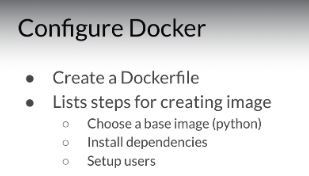
**** ****

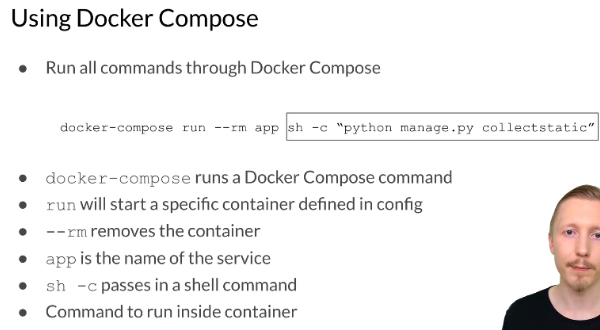
**In Docker Hub We will create a acces token to our project**

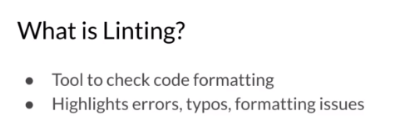
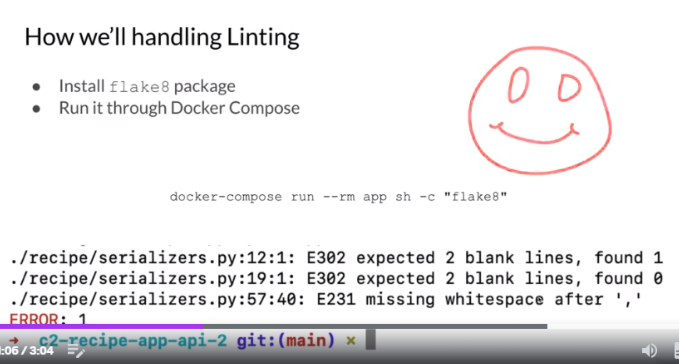
**And we will create secret key in our github account for our project**

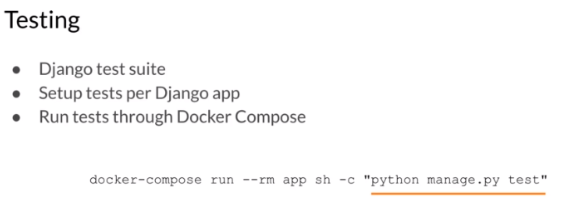
****

**Configuring Docker**

****

****

* **We have create requirement file and dockerfile for our project**
* **Docker composer consist of one or more services on it**
* **\recipe-app-api>docker-compose build**
* **\recipe-app-api>docker build .**
* 
* 

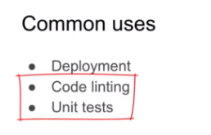
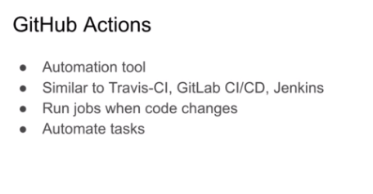


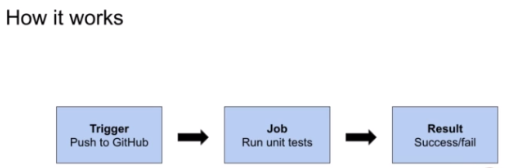
* **to run linting –**
* **docker-compose run --rm app sh -c "flake8"**
* **in order to create django project in app folder docker-compose run --rm app sh -c "django-admin startproject app ."**
* **. means in the current directory and after executing this command a new app folder will be created in app folder**
* **now we will run our django project using**
* **docker-compose up – will start all docker services**

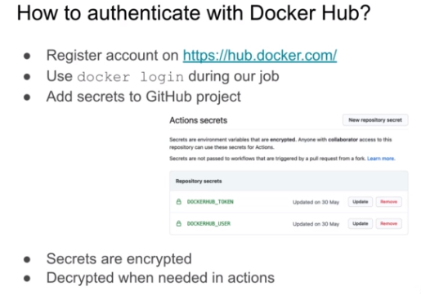
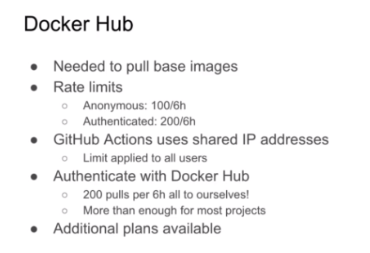
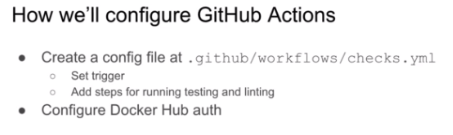
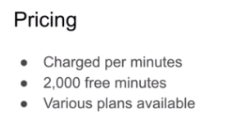
**Summary in this section**

* **we start by creating new project in github and in this setting we put docker hub secret key**
* **then we create project in our machine and create docker file docker-compose file and create requirement file for both deployement and development**
* **we are also create linting tool in app folder**
* **and create new blank django project in app folder**
* **and use docker-compose up to start our django application**

**Github Action**







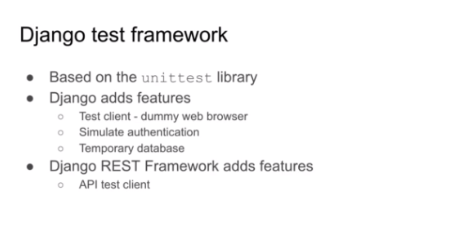
* **we start with \_\_\_ three dashes to start yml file**
* **this command used to run unit test**

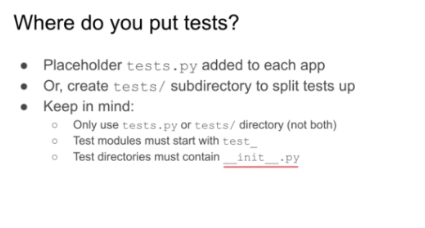
**docker-compose run --rm app sh -c "python manage.py test"**

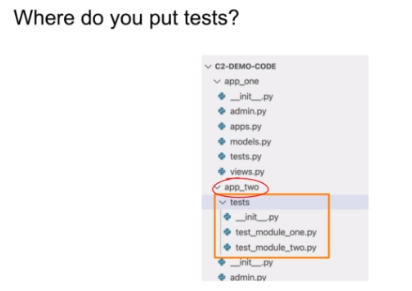
* **the below code also used to test lintng**
* run: docker-compose run --rm app sh -c "flake8"

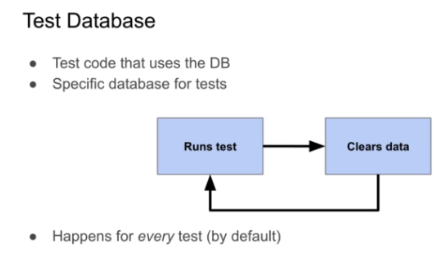
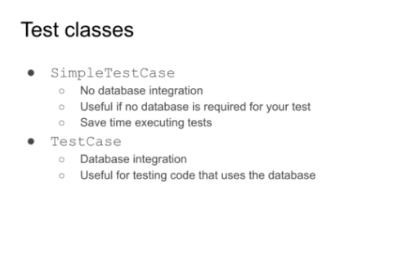
**in github/workflow/check.yml file we write a job that perform which push is performed in the github**

**Test driven Development**





**Sample example** 

* **We don’t need to create a test DB in our production db we will get a temporary test db by default**
* 
* 
* **Sample test code**

