STEPS TO INTEGRATE DOCKER_DESKTOP WITH K8s

27 May 2025 17:36

FOR MORE DETAILS :

CREATION OF DJANGO APPLICATION >> https://www.w3schools.com/django/django_create_app.php

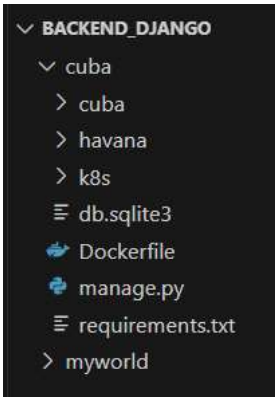
REQUIREMENTS

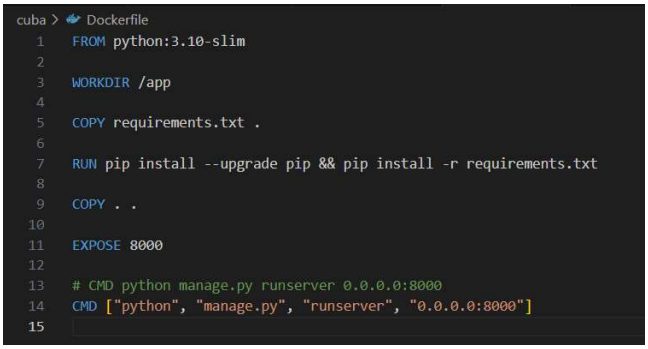
1. Docker Desktop.(For Docker & In built Kubernetes)
2. Kubectl.(For Command Line tool)
3. Lens.(For GUI)
4. VS Code.(For Code Customization)

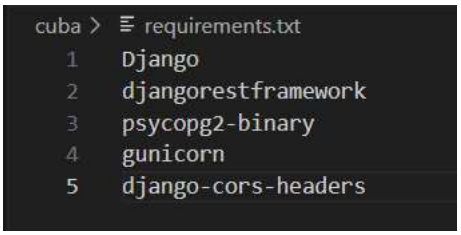
STEP : 1

Create a project (Lets say a 1 tier/2 tier/3 tier Application)
For Example here I'm just developing a single tier backend Application (Python Django With Single GET API) in the following order

Create a docker file & requirements.txt in root
"E:\PROJECTS\Devops\Docker_Kubernetes_Integration\Backend_Django\cuba"

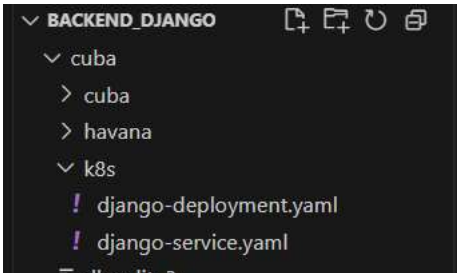


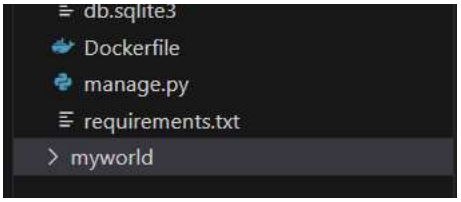




STEP : 2

Also for K8s create a folder named as K8s/ with two .yaml files in following image for deployment & service





cuba > k8s > ! django-deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: django-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: django
  template:
    metadata:
      labels:
        app: django
    spec:
      containers:
      - name: django-container
        image: django-backend-cuba
        imagePullPolicy: Never
        ports:
        - containerPort: 8000

```

cuba > k8s > ! django-service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: django-service
spec:
  type: NodePort
  selector:
    app: django
  ports:
    - name: http
      protocol: TCP
      port: 8000
      targetPort: 8000
      nodePort: 30001

```

STEP : 3

In order to containerize my django project, I want to build the project along with docker file for getting docker images,

NOTE : While building a project, always ensure that docker is running....

PROJECT + DOCKERFILE ==> BUILD (*docker build –t <anyname_u_want> <path_of_the_file_u_want_to_build>*) ==> DOCKER IMAGE

```
E:\PROJECTS\Devops\Docker_Kubernetes_Integration\Backend_Django\cuba>docker build -t django-backend-cuba .
[+] Building 3.9s (11/11) FINISHED                                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                              0.0s
 => => transferring dockerfile: 307B                                                                              0.0s
 => [internal] load metadata for docker.io/library/python:3.10-slim                                               3.4s
 => [auth] library/python:pull token for registry-1.docker.io                                                     0.0s
 => [internal] load .dockerignore                                                                                 0.0s
 => => transferring context: 2B                                                                                   0.0s
 => [1/5] FROM docker.io/library/python:3.10-slim@sha256:49454d2bf78a48f217eb25ecbcb4b5face313fea6a6e8270646 0.0s
 => => resolve docker.io/library/python:3.10-slim@sha256:49454d2bf78a48f217eb25ecbcb4b5face313fea6a6e8270646 0.0s
 => [internal] load build context                                                                                 0.0s
 => => transferring context: 1.67kB                                                                               0.0s
 => CACHED [2/5] WORKDIR /app                                                                                     0.0s
 => CACHED [3/5] COPY requirements.txt .                                                                          0.0s
 => CACHED [4/5] RUN pip install --upgrade pip && pip install -r requirements.txt                                 0.0s
 => [5/5] COPY . .                                                                                                0.0s
 => exporting to image                                                                                            0.2s
 => => exporting layers                                                                                           0.1s
 => => exporting manifest sha256:bfc9d3129daf5bde3719ef8b355c8ddb545d1c38230c3e7086ffb2e951bb6ce2                 0.0s
 => => exporting config sha256:1c22391be1c332f5ca293786995317ac44f8caf39ce94aa4e837181bb5ba84a4                   0.0s
 => => exporting attestation manifest sha256:399dca28718862c2fe1e1b5aed18fd88af8a491af14a338322852080634097 0dc   0.0s
 => => exporting manifest list sha256:39f01fecbc0c30a88b493ffeb6f7ae1f3331fe0aaa57aae29d11d1c2185e98c06           0.0s
 => => naming to docker.io/library/django-backend-cuba:latest                                                     0.0s
 => => unpacking to docker.io/library/django-backend-cuba:latest                                                  0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/k4yk7ffzr72iwdu5ec011h5f
```

```
E:\PROJECTS\Devops\Docker_Kubernetes_Integration\Backend_Django\cuba>
```

After build, run the build to make a container

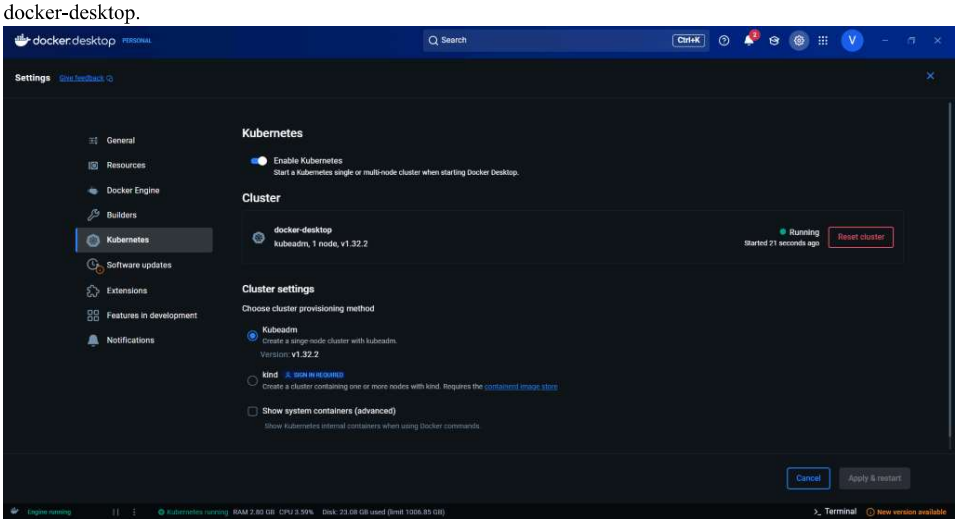
```
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/k4yk7ffzr72iwdus5ec0l1h5f

E:\PROJECTS\Devops\Docker_Kubernetes_Integration\Backend_Django\cuba>docker run -p 8000:8000 django-backend-cuba
Watching for file changes with StatReloader
```

Once your docker is running use command `docker ps` or `docker ps -a` to visualize containers in cmd prompt

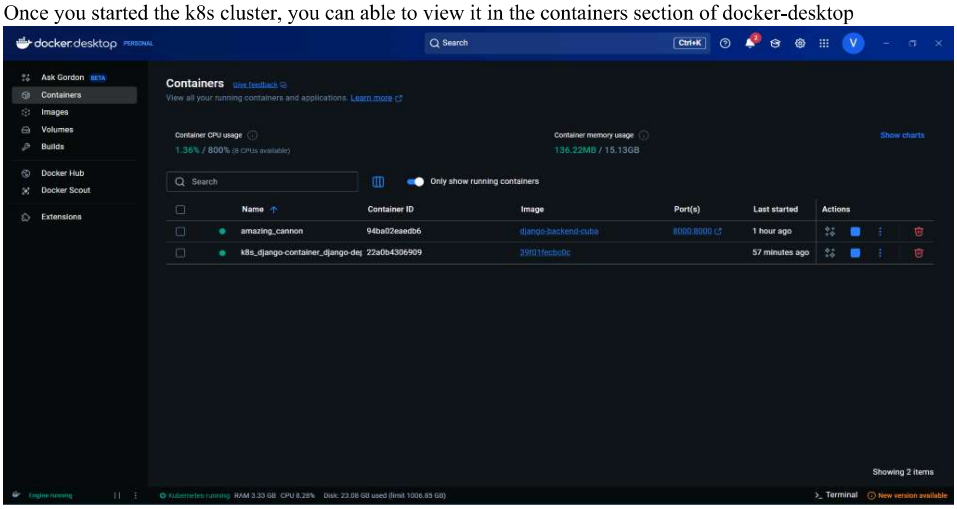
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Microsoft Windows [Version 10.0.26100.4061]
(c) Microsoft Corporation. All rights reserved.

E:\PROJECTS\Devops\Docker_Kubernetes_Integration\Backend_Django>docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED          STATUS                         PORTS                    NAMES
94ba02eaedb6   django-backend-cuba  "python manage.py ru…"   3 minutes ago    Up 3 minutes                   0.0.0.0:8000->8000/tcp   amazing_cannon
9f758398eb0d   nginx:alpine         "/docker-entrypoint.…"   11 days ago      Restarting (1) 21 seconds ago                           3_tier_application_dock
er_build_deploy-nginx-1

E:\PROJECTS\Devops\Docker_Kubernetes_Integration\Backend_Django>docker ps -a
CONTAINER ID   IMAGE                                                COMMAND                  CREATED          STATUS                      PORTS
     NAMES
94ba02eaedb6   django-backend-cuba                                  "python manage.py ru…"   3 minutes ago    Up 3 minutes                0.0.0.0:8000->8000/t
cp   amazing_cannon
5d10a8beefb2   c301ea2cb523                                         "python manage.py ru…"   22 hours ago     Exited (0) 21 hours ago
     sleepy_moser
144487cf4382   c301ea2cb523                                         "/bin/sh -c 'python …"   22 hours ago     Exited (2) 22 hours ago
     sad_mendel
ad0475db7e39   c301ea2cb523                                         "/bin/sh -c 'python …"   22 hours ago     Exited (2) 22 hours ago
     priceless_montalcini
f7a05dcfc97c   c301ea2cb523                                         "ls -R /app"             22 hours ago     Exited (0) 22 hours ago
     upbeat_bartik
7256fdf35a52   c301ea2cb523                                         "/bin/sh -c 'python …"   23 hours ago     Exited (2) 23 hours ago
     confident_hermann
60c1af81a399   c301ea2cb523                                         "/bin/sh -c 'python …"   23 hours ago     Exited (2) 23 hours ago
     inspiring_vaughan
0a07d5e68075   3_tier_application_docker_build_deploy-backend       "/bin/sh -c 'python …"   10 days ago      Exited (137) 10 days ago
     3_tier_application_docker_build_deploy-backend-1
8fdbb6582111   3_tier_application_docker_build_deploy-frontend      "docker-entrypoint.s…"   10 days ago      Exited (1) 10 days ago
     3_tier_application_docker_build_deploy-frontend-1
9f758398eb0d   nginx:alpine                                         "/docker-entrypoint.…"   11 days ago      Restarting (1) 32 seconds ago
     3_tier_application_docker_build_deploy-nginx-1
51499e8c19a1   postgres:latest                                      "docker-entrypoint.s…"   13 days ago      Exited (0) 10 days ago
     3_tier_application_docker_build_deploy-db-1

E:\PROJECTS\Devops\Docker_Kubernetes_Integration\Backend_Django>
```

STEP : 4

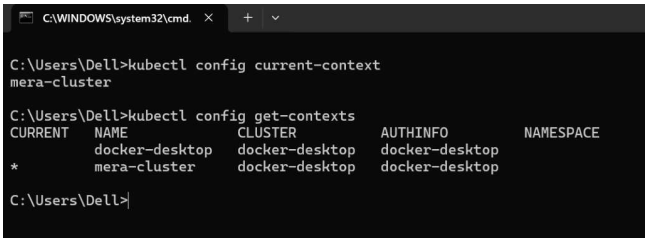
Now that the docker part is over, Here begins the kubernetes with docker-desktop begins by enabling a toggle in settings of docker-desktop.



Once you started the k8s cluster, you can able to view it in the containers section of docker-desktop



Also you can view the list of all clusters & currently running clusters using following command in cmd prompt

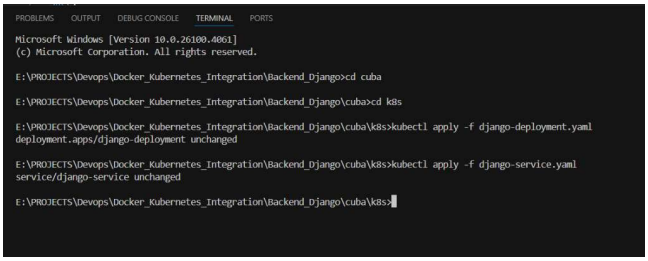


Since you have already have the .yaml files for kubernetes, use the following kubectl commands to apply the yaml files.

```
kubectl apply –f <file-name>
```

Here for currently taken example use the following commands

```
kubectl  apply –f django-deployment.yaml
Kubectl apply –f django-service.yaml
```



STEP : 5

Once K8s configurations were done check the API in browser
http://localhost:30001/havana

STEP : 6

In order to facilitate the visualization of k8s management , we have GUI Tool for k8s called Lens