

C:\Users\User\Documents\TP4>**docker login**

Authenticating with existing credentials... [Username: ineshk]

i Info → To login with a different account, run 'docker logout' followed by 'docker login'

2025/11/12 22:33:36 notifying Desktop of credentials store update: Post

"http://ipc/registry/credstore-updated": context deadline exceeded

Login Succeeded

C:\Users\User\Documents\TP4>**docker build -t tp3-image:1.0 .**

[+] Building 7.1s (11/11) FINISHED	docker:desktop-linux
=> [internal] load build definition from Dockerfile	0.1s
=> => transferring dockerfile: 199B	0.0s
=> [internal] load metadata for docker.io/library/python:3.11-slim	
4.1s	
=> [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:3.11-	
slim@sha256:e4676722fba839e2e5cdb844a52262b43e90e56dbd55b7ad	0.2s
=> => resolve docker.io/library/python:3.11-	
slim@sha256:e4676722fba839e2e5cdb844a52262b43e90e56dbd55b7ad	0.2s
=> [internal] load build context	0.2s
=> => transferring context: 242B	0.1s
=> CACHED [2/5] WORKDIR /app	0.0s
=> CACHED [3/5] COPY requirements.txt ./	0.0s
=> CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt	
0.0s	
=> CACHED [5/5] COPY ..	0.0s
=> exporting to image	0.9s
=> => exporting layers	0.0s
=> => exporting manifest	
sha256:ac114f87a1233a40c0543a8adc5d95dc4ae2de8f3b10f1ff356c84e4256288a5	
0.0s	

```
=> => exporting config  
sha256:1f161b635025197d3f0d26388610098d6118694c4d822547aace165c1702af0e  
0.0s  
=> => exporting attestation manifest  
sha256:908ceec76be123784d4f66ba7a565a7dba84de2deb5e12ac1b73cb9d3485 0.2s  
=> => exporting manifest list  
sha256:2b7e9472fa435363ae41aab7fea76e8ca899dbdea7ed44eba8923b9fbff632e3  
0.2s  
=> => naming to docker.io/library/tp3-image:1.0 0.1s  
=> => unpacking to docker.io/library/tp3-image:1.0 0.1s
```

C:\Users\User\Documents\TP4>**docker tag tp3-image:1.0 ineshk/tp3:1.0**

C:\Users\User\Documents\TP4>**docker push ineshk/tp3:1.0**

The push refers to repository [docker.io/ineshk/tp3]

6acebaeb6c5c: Layer already exists

d7ecded7702a: Layer already exists

f002d17b63fe: Layer already exists

cc65c7af130b: Layer already exists

84bff50156f4: Layer already exists

b13c8d1fd7c8: Layer already exists

9ad5ed543b0d: Pushed

65868b001a40: Layer already exists

1ee9c106547f: Layer already exists

1.0: digest:

sha256:2b7e9472fa435363ae41aab7fea76e8ca899dbdea7ed44eba8923b9fbff632e3 size:
856

C:\Users\User\Documents\TP4>**docker pull ineshk/tp3:1.0**

1.0: Pulling from ineshk/tp3

Digest:

sha256:2b7e9472fa435363ae41aab7fea76e8ca899dbdea7ed44eba8923b9fbff632e3

Status: Image is up to date for ineshk/tp3:1.0

docker.io/ineshk/tp3:1.0

C:\Users\User\Documents\TP4>**docker create --name app1 -p 5001:5000 ineshk/tp3:1.0**

Error response from daemon: Conflict. The container name "/app1" is already in use by container

"705d51f48a04d2e863e28f352344fbf72ea0918a01cff6d9991b797244a6702c". You have to remove (or rename) that container to be able to reuse that name.

C:\Users\User\Documents\TP4>**docker create --name app2 -p 5002:5000 ineshk/tp3:1.0**

Error response from daemon: Conflict. The container name "/app2" is already in use by container

"96aa08050cf2479b2b622f34c9da06215177de3dfd187bb972ac136993a07808". You have to remove (or rename) that container to be able to reuse that name.

C:\Users\User\Documents\TP4>**docker create --name app3 -p 5003:5000 ineshk/tp3:1.0**

Error response from daemon: Conflict. The container name "/app3" is already in use by container

"bb73a01802b3931723fb7ce56dbd91983bb1e937dd5d051efeecc841374206e0e". You have to remove (or rename) that container to be able to reuse that name.

C:\Users\User\Documents\TP4>**docker network create tp3-net**

Error response from daemon: network with name tp3-net already exists

C:\Users\User\Documents\TP4>**docker network connect tp3-net app1**

Error response from daemon: endpoint with name app1 already exists in network tp3-net

C:\Users\User\Documents\TP4>**docker network connect tp3-net app2**

Error response from daemon: endpoint with name app2 already exists in network tp3-net

C:\Users\User\Documents\TP4>**docker network connect tp3-net app3**

Error response from daemon: endpoint with name app3 already exists in network tp3-net

C:\Users\User\Documents\TP4>**docker start app1 app2 app3**

app1

app2

app3

C:\Users\User\Documents\TP4>**docker ps**

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
NAMES					
bb73a01802b3	8b5668f84f1b	"python app.py"	8 hours ago	Up 8 hours	
0.0.0.0:5003->5000/tcp, [::]:5003->5000/tcp	app3				
96aa08050cf2	8b5668f84f1b	"python app.py"	8 hours ago	Up 8 hours	
0.0.0.0:5002->5000/tcp, [::]:5002->5000/tcp	app2				
705d51f48a04	8b5668f84f1b	"python app.py"	8 hours ago	Up 8 hours	
0.0.0.0:5001->5000/tcp, [::]:5001->5000/tcp	app1				

C:\Users\User\Documents\TP4>**curl http://localhost:5001/**

TP3 running in Docker!

C:\Users\User\Documents\TP4>**curl http://localhost:5002/**

TP3 running in Docker!

C:\Users\User\Documents\TP4>**curl http://localhost:5003/**

TP3 running in Docker!

C:\Users\User\Documents\TP4>**docker network inspect tp3-net**

```
[  
  {  
    "Name": "tp3-net",  
    "Id": "258cbfe320ab4f9f53658e7bd1f98971c078ce9695f0a9570ea5c2dbd1d87ddd",  
    "Created": "2025-11-05T00:20:00.164943614Z",  
    "Scope": "local",  
    "Driver": "bridge",  
    "EnableIPv4": true,  
    "EnableIPv6": false,  
    "IPAM": {  
      "Driver": "default",  
      "Options": {},  
      "Config": [  
        {  
          "Subnet": "172.18.0.0/16",  
          "Gateway": "172.18.0.1"  
        }  
      ]  
    }  
  }]
```

```
},
  "Internal": false,
  "Attachable": false,
  "Ingress": false,
  "ConfigFrom": {
    "Network": ""
  },
  "ConfigOnly": false,
  "Containers": {
    "705d51f48a04d2e863e28f352344fbf72ea0918a01cff6d9991b797244a6702c": {
      "Name": "app1",
      "EndpointID": "5640e7c10e5e76a16808d85fc5d6167e118c1768bf28e0cf49941ab23e4841e",
      "MacAddress": "9e:18:2b:d9:f0:52",
      "IPv4Address": "172.18.0.2/16",
      "IPv6Address": ""
    },
    "96aa08050cf2479b2b622f34c9da06215177de3df187bb972ac136993a07808": {
      "Name": "app2",
      "EndpointID": "67a75f0a81d479211721ce82283eaace9d34e6c0dcd00d10addd68782b609064",
      "MacAddress": "1e:a7:3a:46:f6:51",
      "IPv4Address": "172.18.0.3/16",
      "IPv6Address": ""
    },
    "bb73a01802b3931723fb7ce56dbd91983bb1e937dd5d051efeecc841374206e0e": {
      "Name": "app3",
      "EndpointID": "adf9e2a2c753b3e370d5efa52196e401f8f483c750262be2898627a9474d47aa",
      "MacAddress": "1e:be:2c:b8:86:4c",
      "IPv4Address": "172.18.0.4/16",
      "IPv6Address": ""
    }
  }
},
```

```

"Options": {
    "com.docker.network.enable_ipv4": "true",
    "com.docker.network.enable_ipv6": "false"
},
"Labels": {}
}
]

```

Dockerfile :

```

FROM python:3.11-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY ..
EXPOSE 5000
CMD ["python","app.py"]

```

The screenshot shows the Docker Desktop interface. On the left, there's a sidebar with options like Ask Gordon (Beta), Containers (selected), Images, Volumes, Kubernetes, Builds, Models, MCP Toolkit (Beta), Docker Hub, Docker Scout, and Extensions. The main area is titled 'Containers' with a 'Give feedback' link. It displays container usage statistics: Container CPU usage (0.16% / 400%) and Container memory usage (98.91MB / 3.69GB). A chart button is labeled 'Show charts'. Below this, a search bar and a 'Only show running containers' toggle are present. A table lists four containers:

	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	app1	705d51f48a04	ineshk/tp3	5001:5000	0.07%	8 hours ago	
<input type="checkbox"/>	app2	96aa08050cf2	ineshk/tp3	5002:5000	0.03%	8 hours ago	
<input type="checkbox"/>	app3	bb73a01802b3	ineshk/tp3	5003:5000	0.03%	8 hours ago	
<input type="checkbox"/>	laughing_elbaky	d7e65b941886	ineshk/tp3		0.03%	4 minutes ago	

At the bottom, status indicators show 'Engine running', system resources (RAM 0.98 GB, CPU 1.77%, Disk 1.99 GB used / limit 1006.85 GB), and links for 'Terminal' and 'Update available'.

Docker Desktop Personal

Images Local My Hub

159.89 MB / 751.16 MB in use 5 images Last refresh: 9 hours ago

	Name	Tag	Image ID	Created	Size	Actions
○	ineshk/tp3	latest	fd795eb26592	7 days ago	211.72 MB	⋮
○	postgres	16	21f6013073bc	8 days ago	635.48 MB	⋮
○	ineshenka/tp3	1.0	0efc9f284130	7 days ago	211.72 MB	⋮
●	<none>	<none>	8b5668f84f1b	7 days ago	211.72 MB	⋮
●	ineshk/tp3	1.0	2b7e9472fa43	7 days ago	211.72 MB	⋮
●	tp3-image	1.0	2b7e9472fa43	7 days ago	211.72 MB	⋮

Showing 6 items

Engine running RAM 0.98 GB CPU 0.75% Disk: 1.99 GB used (limit 1006.86 GB)

TP3 running in Docker!

https://hub.docker.com/repository/docker/ineshk/tp3/general

Repositories / tp3 / General

Using 0 of 1 private repositories.

ineshk/tp3

Last pushed about 14 hours ago • Repository size: 49.4 MB • ⭐0 • ↓79

Add a description Add a category

Docker commands

To push a new tag to this repository:

```
docker push ineshk/tp3:tagname
```

General	Tags	Image Management	Collaborators	Webhooks	Settings
Tags					
This repository contains 2 tag(s).					
Tag	OS	Type	Pulled	Pushed	
1.0		Image	less than 1 day	about 14 hours	
latest		Image	less than 1 day	8 days	

buildcloud

Build with Docker Build Cloud

Accelerate image build times with access to cloud-based builders and shared cache.

Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation.

