



Installing Docker Desktop 2.5.0.1 (49550)



Configuration

- ☒ Enable Hyper-V Windows Features
- ☒ Install required Windows components for WSL 2
- ☒ Add shortcut to desktop

Ok



Installing Docker Desktop 2.5.0.1 (49550)



Docker Desktop 2.5.0.1

Unpacking files...

```
Unpacking file: resources/docker.tar
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/docker
Unpacking file: resources/ddvp.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/CHANGELOG.md
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/.gitignore
Unpacking file: InstallerCli.pdb
Unpacking file: InstallerCli.exe.config
Unpacking file: frontend/vk_swiftshader_icd.json
Unpacking file: frontend/v8_context_snapshot.bin
Unpacking file: frontend/snapshot_blob.bin
```



Installing Docker Desktop 2.5.0.1 (49550)



Docker Desktop 2.5.0.1

Installation succeeded

You must restart Windows to complete installation.

Close and restart



Sign in



Get started with Docker in a few easy steps!

⌚ ESTIMATED TIME: 2 minutes

Start

[Skip tutorial](#)

We send usage statistics. Check your [privacy settings](#).

✓ Clone

2 Build

3 Run

4 Share

Now, build the image

A Docker image is a private file system just for your container. It provides all the files and code your container needs.

```
cd getting-started
docker build -t docker101tutorial .
```

>>

Next Step

Skip tutorial

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS C:\Users\Administrator> docker run --rm --repo alpine/git t clone https://github.com/docker/getting-started.git

Unable to find image 'alpine/git:latest' locally

latest: Pulling from alpine/git

188c0c94c7c5: Pull complete

702923bb7eae: Pull complete

fad1dae2adf8: Pull complete

Digest: sha256:f3a0a5f4b156e34708a9315b175f1bc7c54e7fb913a2d068c438859e92ac408a

Status: Downloaded newer image for alpine/git:latest

Cloning into 'getting-started'...

PS C:\Users\Administrator> docker cp repo:/git/getting-started/ .

PS C:\Users\Administrator>

Creating webapp.py application

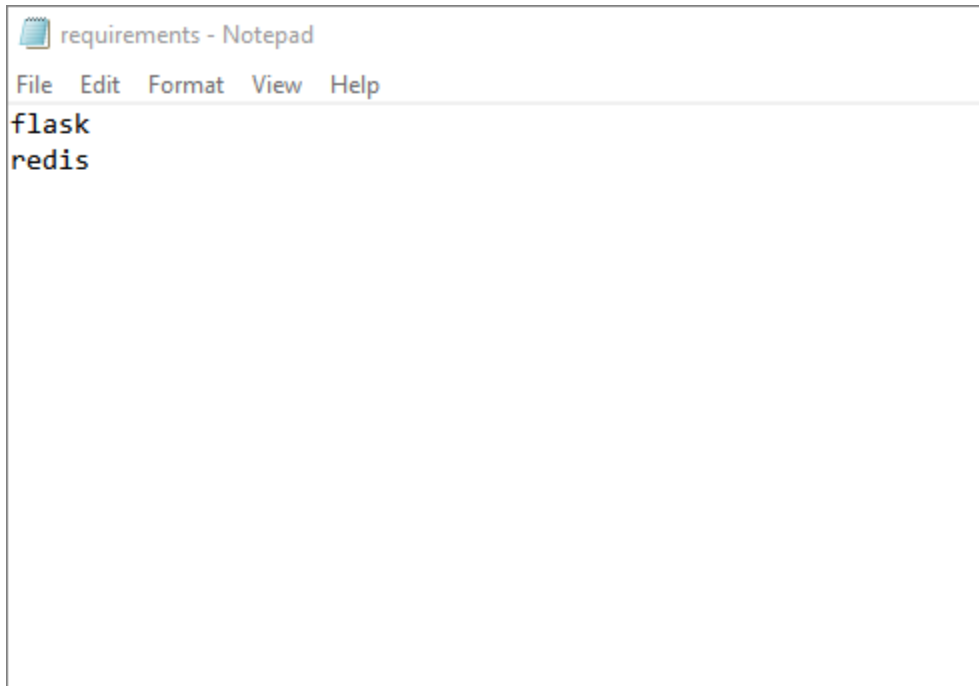
docker-compose.yml

webapp.py X

C: > Users > Administrator > myapp > webapp.py > get_hit_count

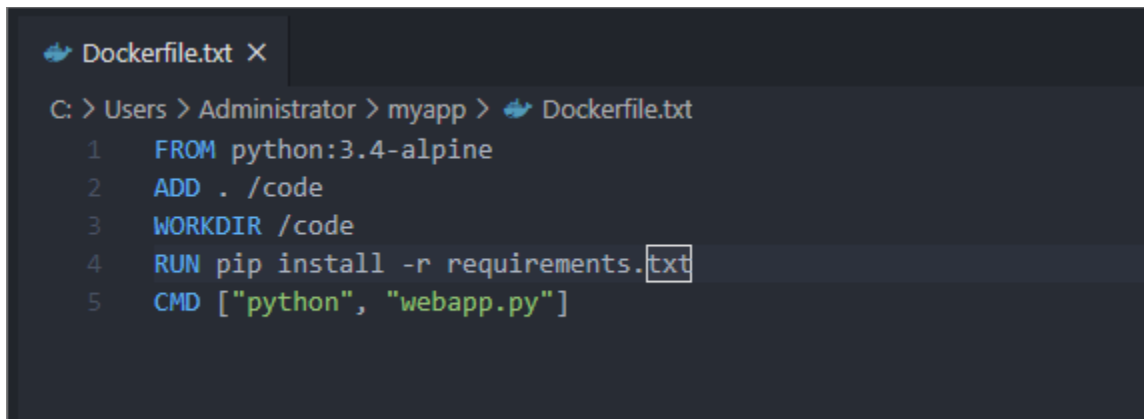
```
1 import time
2 import redis
3 from flask import Flask
4 app = Flask(__name__)
5 cache = redis.Redis(host='redis', port=6379)
6 def get_hit_count():
7     retries = 5
8     while True:
9         try:
10             return cache.incr('hits')
11         except redis.exceptions.ConnectionError as exc:
12             if retries == 0:
13                 raise exc
14             retries -= 1
15             time.sleep(0.5)
16
17 @app.route('/')
18 def hello():
19     count = get_hit_count()
20     return('Hello World! I have been seen {} times.'.format(count))
21
22 if __name__ == "__main__":
23     app.run(host="0.0.0.0", debug=True)
```

Creating requirements.txt file

A screenshot of a Notepad window titled "requirements - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text content of the file is "flask" on the first line and "redis" on the second line.

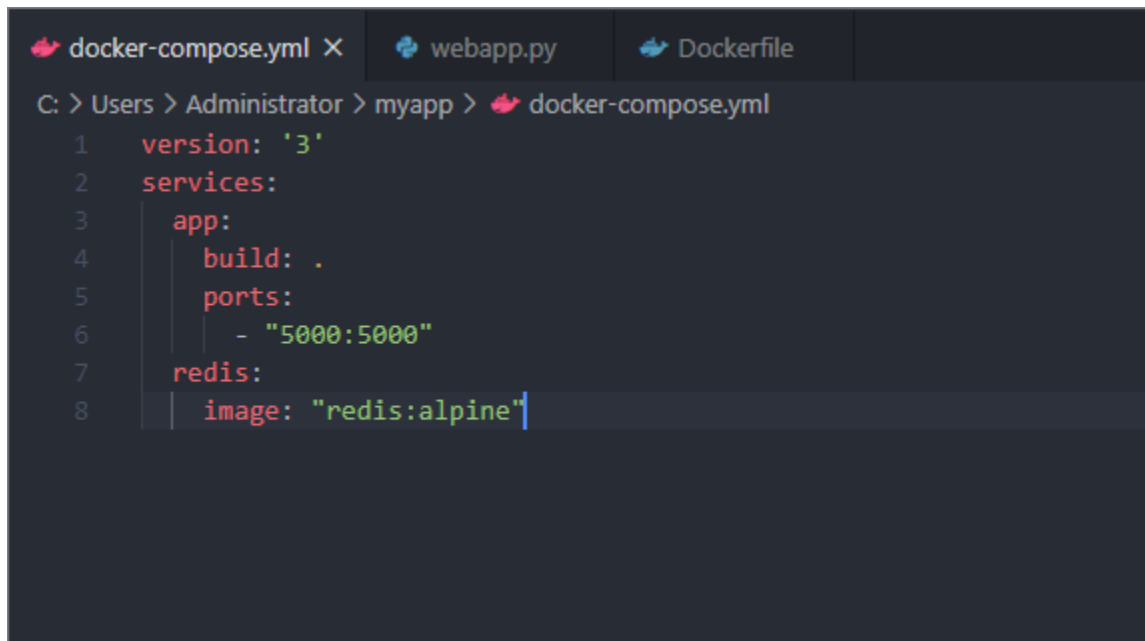
```
requirements - Notepad
File Edit Format View Help
flask
redis
```

Creating Dockerfile. Note on saving .txt extension was removed

A screenshot of a VS Code editor window titled "Dockerfile.txt". The editor shows the following Dockerfile content:

```
Dockerfile.txt X
C: > Users > Administrator > myapp > Dockerfile.txt
1 FROM python:3.4-alpine
2 ADD . /code
3 WORKDIR /code
4 RUN pip install -r requirements.txt
5 CMD ["python", "webapp.py"]
```

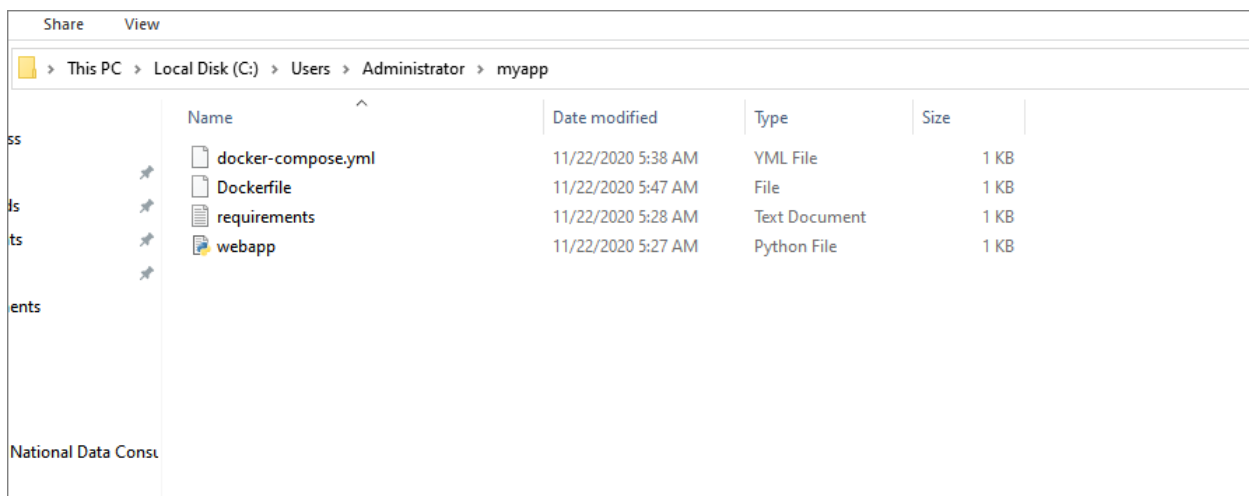
Create YAML file called docker-compose



The screenshot shows a code editor with three tabs: 'docker-compose.yml', 'webapp.py', and 'Dockerfile'. The 'docker-compose.yml' tab is active, displaying the following YAML configuration:

```
1 version: '3'
2 services:
3   app:
4     build: .
5     ports:
6       - "5000:5000"
7   redis:
8     image: "redis:alpine"
```

Folder containing all 4 files required to run docker-compose up command



The screenshot shows a Windows File Explorer window with the address bar displaying the path: This PC > Local Disk (C:) > Users > Administrator > myapp. The main area shows a list of four files:

Name	Date modified	Type	Size
docker-compose.yml	11/22/2020 5:38 AM	YML File	1 KB
Dockerfile	11/22/2020 5:47 AM	File	1 KB
requirements	11/22/2020 5:28 AM	Text Document	1 KB
webapp	11/22/2020 5:27 AM	Python File	1 KB

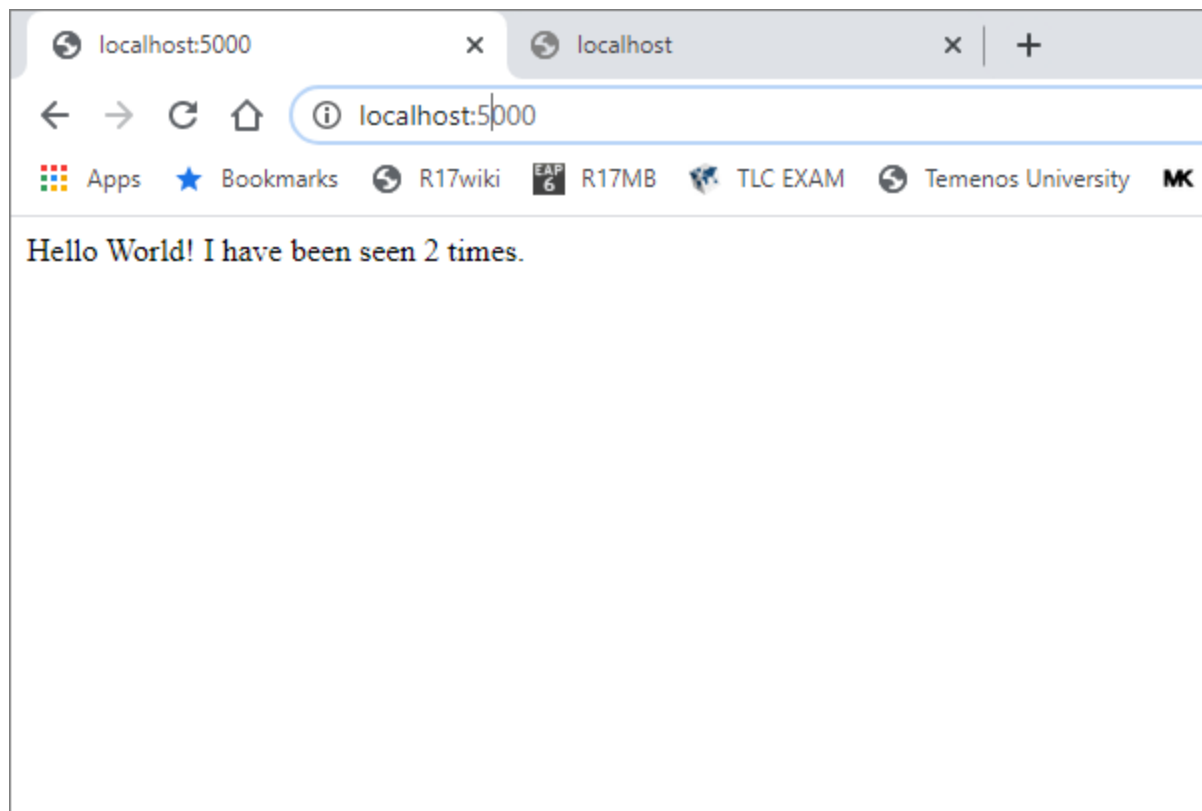
Running **docker-compose up** command

```
Administrator: Windows PowerShell
PS C:\Users\Administrator\myapp> docker-compose up
Building app
Step 1/5 : FROM python:3.4-alpine
3.4-alpine: Pulling from library/python
8e402f1a9c57: Pull complete
cda9ba2397ef: Pull complete
aaefcf9bbbfd: Pull complete
bc2e7e266629: Pull complete
e1977129b756: Pull complete
Digest: sha256:c210b660e2ea553a7afa23b41a6ed112f85dbce25cbcb567c75dfe05342a4c4b
Status: Downloaded newer image for python:3.4-alpine
--> c06adcf62f6e
Step 2/5 : ADD . /code
--> 8199a2302fc5
Step 3/5 : WORKDIR /code
--> Running in 7c2ec1747510
Removing intermediate container 7c2ec1747510
--> 40ef9a1af27e
Step 4/5 : RUN pip install -r requirements.txt
--> Running in 0b7ea3bab360
DEPRECATION: Python 3.4 support has been deprecated, pip 19.1 will be the last one supporting it. Please upgrade your Python as Python 3.4 won't be maintained after March 2019 (cf PEP 429).
Collecting flask (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/d8/94/7350820ae209ccdba073f83220cea1c376f2621254d1e0e82609c9a65e58/Flask-1.0.4-py2.py3-none-any.whl (92kB)
Collecting redis (from -r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/32/ae/28613a62eea0d53d3db3147f8715f90da07667e99baeedf1010eb400f8c0/redis-3.3.11-py2.py3-none-any.whl (66kB)
Collecting Jinja2>=2.10 (from flask->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/65/e0/eb35e762802015cab1ccee04e8a277b03f1d8e53da3ec3106882ec42558b/Jinja2-2.10.3-py2.py3-none-any.whl (125kB)
Collecting itsdangerous>=0.24 (from flask->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/76/ae/44b03b253d6fade317f32c24d100b3b35c2239807046a4c953c7b89fa49e/itsdangerous-1.1.0-py2.py3-none-any.whl
Collecting Werkzeug>=0.14 (from flask->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/c2/e4/a859d2fe516f466642fa5c6054fd9646271f9da26b0cac0d2f37fc858c8f/Werkzeug-0.16.1-py2.py3-none-any.whl (327kB)
Collecting click>=5.1 (from flask->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/fa/37/45185cb5abbc30d7257104c434fe0b07e5a195a6847506c074527aa599ec/Click-7.0-py2.py3-none-any.whl (81kB)
Collecting MarkupSafe>=0.23 (from Jinja2>=2.10->flask->-r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/b9/2e/64db92e53b86efccfaea71321f597fa2e1b2bd3853d8ce658568f7a13094/MarkupSafe-1.1.1.tar.gz
Building wheels for collected packages: MarkupSafe
  Building wheel for MarkupSafe (setup.py): started
  Building wheel for MarkupSafe (setup.py): finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/f2/aa/04/0edf07a1b8a5f5f1aed7580fffb69ce8972edc16a505916a77
Successfully built MarkupSafe
Installing collected packages: MarkupSafe, Jinja2, itsdangerous, Werkzeug, click, flask, redis
```

Our application is running now

```
redis_1 | 1:M 22 Nov 2020 01:29:08.163 # Server initialized
redis_1 | 1:M 22 Nov 2020 01:29:08.163 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
redis_1 | 1:M 22 Nov 2020 01:29:08.163 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and memory usage issues with Redis. To fix this issue run the command 'echo madvise > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain the setting after a reboot. Redis must be restarted after THP is disabled (set to 'madvise' or 'never').
redis_1 | 1:M 22 Nov 2020 01:29:08.164 * Loading RDB produced by version 6.0.9
redis_1 | 1:M 22 Nov 2020 01:29:08.164 * RDB age 45 seconds
redis_1 | 1:M 22 Nov 2020 01:29:08.164 * RDB memory usage when created 0.77 Mb
redis_1 | 1:M 22 Nov 2020 01:29:08.164 * DB loaded from disk: 0.000 seconds
redis_1 | 1:M 22 Nov 2020 01:29:08.164 * Ready to accept connections
app_1 | * Serving Flask app "webapp" (lazy loading)
app_1 | * Environment: production
app_1 | WARNING: This is a development server. Do not use it in a production deployment.
app_1 | Use a production WSGI server instead.
app_1 | * Debug mode: on
app_1 | * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
app_1 | * Restarting with stat
app_1 | * Debugger is active!
app_1 | * Debugger PIN: 110-616-467
app_1 | 172.18.0.1 - - [22/Nov/2020 01:29:18] "GET / HTTP/1.1" 200 -
app_1 | 172.18.0.1 - - [22/Nov/2020 01:29:18] "GET /favicon.ico HTTP/1.1" 404 -
app_1 | 172.18.0.1 - - [22/Nov/2020 01:29:29] "GET / HTTP/1.1" 200 -
```

On browser we can check. On refreshing again we see that the counter has changed



docker ps shows containers that are running

```
PS C:\Users\Administrator> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
93bf0df8188f   myapp_app     "python webapp.py"      4 minutes ago Up 4 minutes  0.0.0.0:5000->5000/tcp   myapp_app_1
5c6badb7f8fd   redis:alpine  "docker-entrypoint.s..." 44 minutes ago Up 4 minutes  6379/tcp                myapp_redis_1
PS C:\Users\Administrator>
```

Q) What are the methods to create a docker swarm?

A docker swarm is a cluster where multiple containers are running on different nodes connected and communicated over a network. These can be physical machines or virtual machines

To create a docker swarm, we need at least one machine called manager node and other as worker node depending on how many nodes are needed. An IP address must be assigned to a network interface available to the host operating system. All nodes in the swarm need to connect to the manager at the IP address which should be fixed

1. Create Docker machines by running command:
“**docker-machine create --driver hyperv manager1**” on one machine &
“**docker-machine create --driver hyperv worker1**” on another machine

2. Get manager1 ip by running below command

“**docker-machine ip manager1**”

After that run below command

“**docker-machine ssh manager1**” and initialize docker swarm:

“**docker swarm init --advertise-addr <MANAGER-IP>**”

Once we run this command, docker swarm will be initialized and a link to join docker swarm is displayed. We will use this link on worker nodes so they can join. The link will look something like this:

**docker swarm join **

**--token SWMTKN-1-49nj1cmqI0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743ojnwacrr2e7c **

192.168.99.100:2377

3. On worker node run “**docker-machine ssh worker1**” and then enter the link obtained from step 2. This was worker1 node will become part of docker swarm
4. We will deploy a service and thereby creating replicas