

VM vs Containers

18 May 2020 10:00

<https://www.linkedin.com/learning/docker-for-dot-net-developers-with-visual-studio/work-with-containers>

VMs:

We start with

- 1) Host operating system
- 2) We add VMs that we need to the host OS
 - o Decide on the CPU, RAM,..
- 3) Install the OS for each one of the VMs
- 4) Install the software to the run the App on these VMs
- 5) Install the application

The VM **takes as much resources as it has been allocated even while it is not running**

Containers:

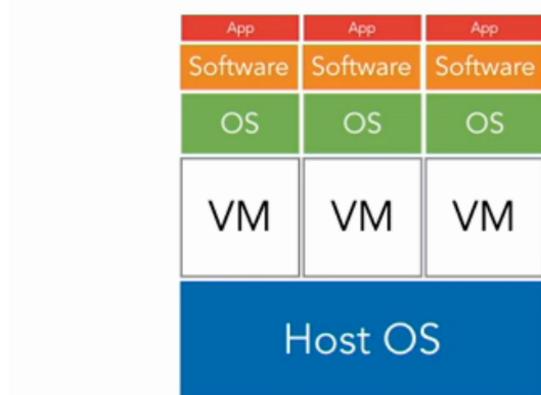
We start with

- 1) Host OS
- 2) We put as many containers as we need on the Host OS
 - o The app is included in the container
 - o No need to decide on the resource allocation (No decision anxiety :))
 - o The container just **takes as little or as much RAM it needs while it is running**
 - o When it is not running, **It consumes almost 0 resources**

The reason we can do this is because **It shares the Host OS resources**

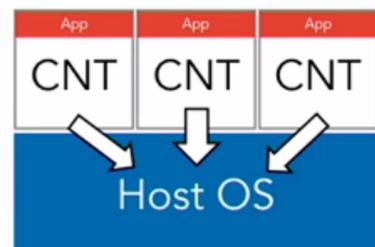
Why Containers?

Virtual Machines (VMs)



vs.

Containers



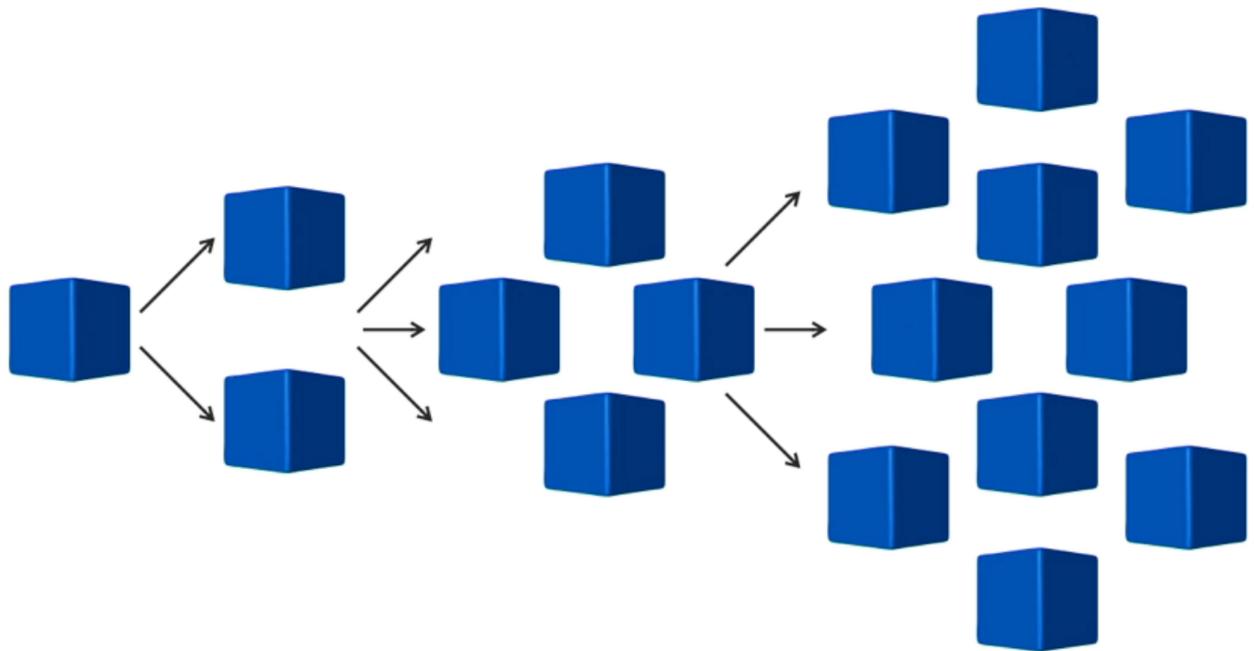
In Containers, everything is included and so it is easy to move around

We can move containers to a new server (as long as docker tools are installed) and everything works as it should

**Everything is included
in the container.**



Since Containers are lighter than VMs, whether we need one, few or many. It is **easy to scale**



Standardize containers

Transporting

Scaling

Tooling

You currently can not run Windows containers on *nix-based machines.

Docker originally used [LinuX Containers](#) (LXC), but later switched to [runC](#) (formerly known as [libcontainer](#)), which runs in the same operating system as its host. This allows it to share a lot of the host operating system resources. Also, it uses a layered filesystem ([AuFS](#)) and manages networking.

AuFS is a layered file system, so you can have a read only part and a write part which are merged together. One could have the common parts of the operating system as read only (and shared amongst all of your containers) and then give each container its own mount for writing.

So, let's say you have a 1 GB container image; if you wanted to use a full VM, you would need to have 1 GB x number of VMs you want. With Docker and AuFS you can share the bulk of the 1 GB between all the containers and if you have 1000 containers you still might only have a little over 1 GB of space for the containers OS (assuming they are all running the same OS image).

A full virtualized system gets its own set of resources allocated to it, and does minimal sharing. You get more isolation, but it is much heavier (requires more resources). With Docker you get less isolation, but the containers are lightweight (require fewer resources). So you could easily run thousands of containers on a host, and it won't even blink. Try doing that with Xen, and unless you have a really big host, I don't think it is possible.

A full virtualized system usually takes minutes to start, whereas Docker/LXC/runC containers take seconds, and often even less than a second.

There are pros and cons for each type of virtualized system. If you want full isolation with guaranteed resources, a full VM is the way to go. If you just want to isolate processes from each other and want to run a ton of them on a reasonably sized host, then Docker/LXC/runC seems to be the way to go.

For more information, check out [this set of blog posts](#) which do a good job of explaining how LXC works.

Why is deploying software to a docker image (if that's the right term) easier than simply deploying to a consistent production environment?

Deploying a consistent production environment is easier said than done. Even if you use tools like [Chef](#) and [Puppet](#), there are always OS updates and other things that change between hosts and environments.

Docker gives you the ability to snapshot the OS into a shared image, and makes it easy to deploy on other Docker hosts. Locally, dev, qa, prod, etc.: all the same image. Sure you can do this with other tools, but not nearly as easily or fast.

This is great for testing; let's say you have thousands of tests that need to connect to a database, and each test needs a pristine copy of the database and will make changes to the data. The classic approach to this is to reset the database after every test either with custom code or with tools like [Flyway](#) - this can be very time-consuming and means that tests must be run serially. However, with Docker you could create an image of your database and run up one instance per test, and then run all the tests in parallel since you know they will all be running against the same snapshot of the database. Since the tests are running in parallel and in Docker containers they could run all on the same box at the same time and should finish much faster. Try doing that with a full VM.

From comments...

Interesting! I suppose I'm still confused by the notion of "snapshot[ting] the OS". How does one do that without, well, making an image of the OS?

Well, let's see if I can explain. You start with a base image, and then make your changes, and commit those changes using docker, and it creates an image. This image contains only the differences from the base. When you want to run your image, you also need the base, and it layers your image on top of the base using a layered file system: as mentioned above, Docker uses AuFS. AuFS merges the different layers together and you get what you want; you just need to run it. You can keep adding more and more images (layers) and it will continue to only save the diffs. Since Docker typically builds on top of ready-made images from a [registry](#), you rarely have to "snapshot" the whole OS yourself.

From <<https://stackoverflow.com/questions/16047306/how-is-docker-different-from-a-virtual-machine>>

Docker Basics - Images and Containers

18 May 2020 10:16

Like in Object Oriented Programming Classes are blueprints for making objects

Images are blueprints for making a container

- They are actually containers frozen in time
- Used for stamping/cloning/creating multiple containers

Images

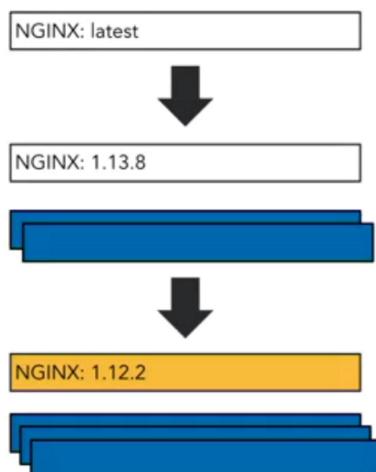
- Images are blueprints for making containers
- Containers frozen in time
- Used for “stamping out” multiple containers

Containers are actual running machines

Containers

- The actual running “machines”
- Where you put your application
- What will be deployed

Images are Made Up of Layers



Images are made up of layers.

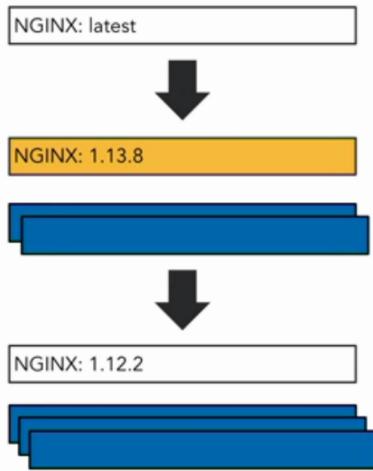
If we have downloaded NGINX: 1.12.2 Image

And then we pull NGINX: 1.13.8 (Only the additional layers would get downloaded and not the entire image)

We don't need to download all the layers which are same in between 1.12.2 and 1.13.8

We just need to download the 2 layers that are different between 1.12.2 and 1.13.8

Images are Made Up of Layers



This will save a lot of hard disk space

For downloading an image, go to dockerhub

A screenshot of the Docker Hub website. The top navigation bar includes links for Dashboard, Explore, Organizations, Create, and a user profile for 'leebrandt'. The search bar contains the query 'nginx'. Below the search bar, there is a dropdown menu showing 'leebrandt' and a 'Repositories' tab, which is currently selected. Other tabs include 'Stars' and 'Contributed'. To the right, a message indicates 'Private Repositories: Using 0 of 1' and a 'Get more' link. The main content area is titled 'Welcome to Docker Hub' and features a message: 'Here are a few things to get you started.' Below this are three buttons: 'Create Repository' (with a notepad icon), 'Create Organization' (with a people icon), and 'Explore Repositories' (with a magnifying glass icon).

Docker Store is the new place to discover public Docker content. [Check it out →](#)

  nginx

Dashboard Explore Organizations Create  leebrandt

Repositories (29605)

		All		
	nginx official	7.9K STARS	10M+ PULLS	
	jwilder/nginx-proxy public automated build	1.3K STARS	10M+ PULLS	
	jrcs/letsencrypt-nginx-proxy-companion public automated build	296 STARS	10M+ PULLS	
	richarvey/nginx-php-fpm public automated build	510 STARS	1M+ PULLS	

Official image at the TOP

It is recommended to use the official image, unless we know the owners or know that the other images are exactly what you want

nginx is now available in the Docker Store, the new place to discover public Docker content. [Check it out →](#)

  nginx

Dashboard Explore Organizations Create  leebrandt

OFFICIAL REPOSITORY

nginx 

Last pushed: a month ago

Repo Info	Tags
Short Description	Docker image for Nginx
Official build of Nginx.	
Full Description	
Supported tags and respective Dockerfile links <ul style="list-style-type: none"> • 1.13.8, mainline, 1, 1.13, latest (mainline/stretch/Dockerfile) • 1.13.8-perl, mainline-perl, 1-perl, 1.13-perl, perl (mainline/stretch-perl/Dockerfile) • 1.13.8-alpine, mainline-alpine, 1-alpine, 1.13-alpine, alpine (mainline/alpine/Dockerfile) • 1.13.8-alpine-perl, mainline-alpine-perl, 1-alpine-perl, 1.13-alpine-perl, alpine-perl (mainline/alpine-perl/Dockerfile) • 1.12.2, stable, 1.12 (stable/stretch/Dockerfile) • 1.12.2-perl, stable-perl, 1.12-perl (stable/stretch-perl/Dockerfile) 	
Docker Pull Command	<code>docker pull nginx</code>

Full Description

Supported tags and respective Dockerfile links

- 1.13.8 , mainline , 1 , 1.13 , latest ([mainline/stretch/Dockerfile](#))
- 1.13.8-perl , mainline-perl , 1-perl , 1.13-perl , perl ([mainline/stretch-perl/Dockerfile](#))
- 1.13.8-alpine , mainline-alpine , 1-alpine , 1.13-alpine , alpine ([mainline/alpine/Dockerfile](#))
- 1.13.8-alpine-perl , mainline-alpine-perl , 1-alpine-perl , 1.13-alpine-perl , alpine-perl ([mainline/alpine-perl/Dockerfile](#))

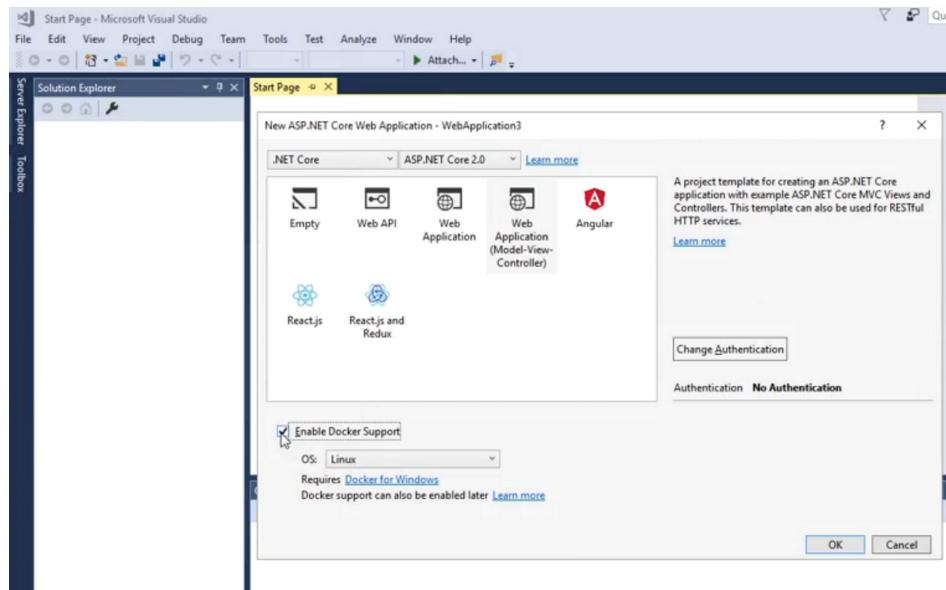
That mean 1.13 is the latest version

Docker Basics - List and Pull Images

18 May 2020 10:53

VS 2015 has enabled support for docker

We can enable it by checking the "Enable Docker Support" check box



But if we do not know what VS would generate for us, it becomes difficult for us to modify it.
Let's start from scratch - do it from the command line

Powershell/GitbashShell/..

```
Docker Version
PS C:\Users\Gokav> docker -v
Docker version 19.03.8, build afacb8b
PS C:\Users\Gokav>
```

List the images on host machine

```
Windows PowerShell
PS C:\Users\Gokav> docker image list
REPOSITORY TAG IMAGE ID CREATED SIZE
myfirstdockerapp dev 2954b23d3622 7 months ago 523MB
microsoft/dotnet 2.2-aspnetcore-runtime-nanoserver-1803 4b3b08e06374 8 months ago 523MB
PS C:\Users\Gokav>
```

Repository : (The image)
Tag: dev (version/.)
ImageID: 15 digit hash

Think of it like github where we have a repository and tag different versions of the repository

If we need an image that we do not have in the host machine, we pull it from Dockerhub

```
PS C:\windows\system32> docker pull ...
Mention the image that we want to pull along with the tag
```

```
PS C:\windows\system32> docker pull nginx:1.13.8
```

My DockerId:
kalyangokavarapu

The screenshot shows the Docker Hub homepage. At the top, there's a search bar with placeholder text "Search for great content (e.g., mysql)". Below it, a navigation bar includes "Explore", "Repositories", "Organizations", "Get Help", and a user dropdown for "kalyangokavarapu". A "Create Repository" button is visible. On the left, a repository card for "kalyangokavarapu / cheers2019" is shown, updated 8 months ago, with 0 stars and 6 forks. A tip message says "Tip: Not finding your repository? Try switching namespace via the top left dropdown." To the right, there's a "Create an Organization" section with a "Manage Docker Hub repositories with your team" link. At the bottom, there are "Download Docker Desktop" and "Secure, Private Repo Pricing" buttons.

The screenshot shows the Docker Hub search results for "nginx". The search bar at the top has "nginx" entered. The results page shows a single repository card for "NGINX nginx" under "Docker Official Images", described as an "Official build of Nginx". It has 1B+ stars. Below the card are tabs for "Description", "Reviews", and "Tags". To the right, there's a "Copy and paste to pull this image" field containing "docker pull nginx" and a "View Available Tags" link.

The screenshot shows the detailed view of the "nginx" repository. It includes sections for "Quick reference" (with links to maintainers and help forums), "Supported tags and respective Dockerfile links" (listing tags like 1.17.10, mainline, 1.17, latest, 1.17.10-perl, mainline-perl, 1-perl, 1.17-perl, perl, 1.17.10-alpine, mainline-alpine, 1-alpine, 1.17-alpine, alpine, 1.17.10-alpine-perl, mainline-alpine-perl, 1-alpine-perl, 1.17-alpine-perl, alpine-perl), and a "Description" tab which contains the text: "Once we start downloading".

```
PS C:\Windows\system32> docker pull nginx:1.13.8
1.13.8: Pulling from library/nginx
e7bb522d92ff: Pulling fs layer
6edc05228666: Pulling fs layer
cd866a17e81f: Pulling fs layer
```

It will start downloading the layers.

```
PS C:\Windows\system32> docker pull nginx:1.13.8
1.13.8: Pulling from library/nginx
e7bb522d92ff: Downloading [=====] 7.11MB/22.49MB
6edc05228666: Downloading [=====] 15.81MB/21.95MB
cd866a17e81f: Download complete
```

```
PS C:\Windows\system32> docker pull nginx:1.13.8
1.13.8: Pulling from library/nginx
e7bb522d92ff: Extracting [=====] 12.39MB/22.49MB
6edc05228666: Download complete
cd866a17e81f: Download complete
```

```
PS C:\Windows\system32> docker pull nginx:1.13.8
1.13.8: Pulling from library/nginx
e7bb522d92ff: Pull complete
6edc05228666: Pull complete
cd866a17e81f: Pull complete
Digest: sha256:285b49d42c703fdf257d1e2422765c4ba9d3e37768d6ea83d7fe2043dad6e63d
Status: Downloaded newer image for nginx:1.13.8
PS C:\Windows\system32>
```

Once the image is downloaded, The image will get a SHA256 hash
And the status

```
PS C:\Windows\system32> docker image list
REPOSITORY          TAG        IMAGE ID      CREATED       SIZE
nginx              1.13.8    3f8a4339aadd  5 weeks ago   108MB
hello-world        latest    f2a91732366c  2 months ago  1.85kB
PS C:\Windows\system32>
```

Lets assume I have downloaded a wrong image,

To Remove an image that I have already downloaded on to my host

rmi : remove image

Either with

- o name_and_tag or with
 - PS C:\Windows\system32> docker rmi nginx:1.13.8

- o Image_ID (you need not type the entire image ID, just enough of it to make it unique in the list)

In the above number 3 is sufficient to identify a unique image

- PS C:\Windows\system32> docker rmi 3

For Production we can use first 3 chars

```
PS C:\Windows\system32> docker rmi 3f8
Untagged: nginx:1.13.8
Untagged: nginx@sha256:285b49d42c703fdf257d1e2422765c4ba9d3e37768d6ea83d7fe2043dad6e63d
Deleted: sha256:3f8a4339aadda5897b744682f5f774dc69991a81af8d715d37a616bb4c99edf5
Deleted: sha256:bb528503f6f01b70cd8de94372e1e3196fad3b28da2f69b105e95934263b0487
Deleted: sha256:410204d28a96d436e31842a740ad0c827f845d22e06f3b1ff19c3b22706c3ed4
Deleted: sha256:2ec5c0a4cb57c0af7c16ceda0b0a87a54f01f027ed33836a5669ca266cafe97a
```

Tried the same in my machine

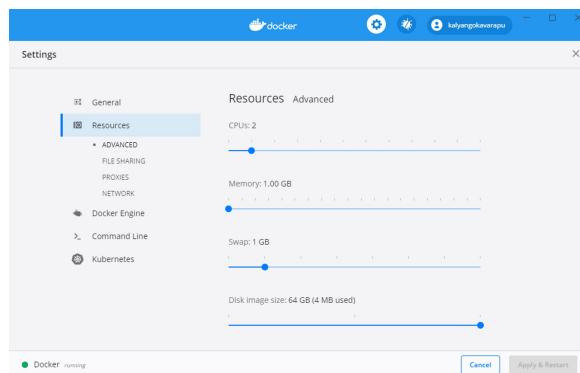
```
PS C:\Users\Gokav> docker image list
REPOSITORY TAG IMAGE ID CREATED SIZE
myfirstdockerapp dev 2954b23d3622 7 months ago 523MB
microsoft/dotnet 2.2-aspnetcore-runtime-nanoserver-1803 4b3b08e06374 8 months ago 523MB
PS C:\Users\Gokav> docker rmi 2
Untagged: myfirstdockerapp:dev
Deleted: sha256:2954b23d36228770eff30044c3d4b7035732e8adeada0257ebded842bd0d5cad3
Deleted: sha256:90c9a7768f18ff5eda9255e1900a26fb86091b4b36617889ef1bf6d292333
Deleted: sha256:7b86a496b46ba0a5e0fe4d2c06e7c386549bd03e77b75df37eb6391b61951c2
Deleted: sha256:1419d8ed3ecc9b8742accf8e09d8f48a9af1ded974a37b8643d9ce0e07516aa3
Deleted: sha256:b1204683d01947fd9b98d6982e7c13f587595dc197680cd217f1dc65ebelbf5c
Deleted: sha256:6aecc73804534f16c87306a0d1e32abb08df8c190653bfe37b64e75378ae6
PS C:\Users\Gokav> docker image list
REPOSITORY TAG IMAGE ID CREATED SIZE
microsoft/dotnet 2.2-aspnetcore-runtime-nanoserver-1803 4b3b08e06374 8 months ago 523MB
PS C:\Users\Gokav> docker rmi 4
Untagged: microsoft/dotnet:2.2-aspnetcore-runtime-nanoserver-1803
Untagged: microsoft/dotnet@sha256:781eb8a5ed77f7276467582803f391027f4c74ed34c35c80f6a5c1c423b3e5f0
Deleted: sha256:4b3b08e0637453d4ac723d2a9292cb9b9f3dc8ad4a1e27c216016fb6165
Deleted: sha256:7490776c11be7e2a96770988fea32d139f40576f816b1964251dec3535bf74f1
Deleted: sha256:239162195a2f5b13a34a3abb510dc92e5c76a4f95c170e77d1c992ea1afcf0fc
Deleted: sha256:d996bc7e85d4fa8d3b10c4f550dd229d8c332d897953be2c29e34e53193b714
Deleted: sha256:9ab73f7e4cccc428c42f545c9a3c7f6e831efef21225d6368c6e81f6326b849
Deleted: sha256:6541114e7e6d6d5883ae3eb5c0aac0c83d30a854a1e66ec1951a91112c37666a
Deleted: sha256:7fcf86341c5cc15d7a6301c260dd9470a838b25598bef1fad7461ac3761fiae5d
Deleted: sha256:bc06b2070c8fb7c81b24025f0d67ed17390339fd8506cc46ce0224ec9c6e73
PS C:\Users\Gokav> docker image list
REPOSITORY TAG IMAGE ID CREATED SIZE
PS C:\Users\Gokav>
```

Lets download latest NGINX

```
PS C:\Users\Gokav> docker pull nginx:1.17.10
1.17.10: Pulling from library/nginx
no matching manifest for windows/amd64 10.0.18362 in the manifest list entries
PS C:\Users\Gokav>
```

We couldn't as supported windows image isn't available

The above was due to resource issue.. Where I had to reduce the allotted RAM from 2GB to 1GB



```
PS C:\WINDOWS\system32> docker pull hello-world:latest
latest: Pulling from library/hello-world
2e03bdc26d7: Pull complete
Digest: sha256:6a65f928fb1fcfbc963f7aa6d57c8eeb426ad9a20c7ee045538ef34847f44f1
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
```

```
PS C:\WINDOWS\system32> docker pull hello-world:latest
latest: Pulling from library/hello-world
0e03bdc26d7: Pull complete
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
PS C:\WINDOWS\system32> docker pull nginx:latest
latest: Pulling from library/nginx
4fb6ec6fd1c1: Pull complete
99c53mb9b92: Pull complete
1fa5a20fc0c8: Pull complete
Digest: sha256:30dfa439718a17baaffefad16c5e7c9d0aicde97b4fd84f63b69e13513be7097
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
PS C:\WINDOWS\system32>
```

Docker Basics - Containers

18 May 2020 13:38

We have 2 images in our host

```
PS C:\WINDOWS\system32> docker image list
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
nginx          latest    9beeba249f3e  2 days ago   127MB
hello-world    latest    bf756fb1ae65  4 months ago 13.3kB
PS C:\WINDOWS\system32> ■
```

Now to **RUN a Container** from the NGINX image

To do that we use <docker run> command with some extra information

- **-it** (2 switches combined together)
 - -i (run container interactively)
 - -t (sudo tty shell in the terminal - meaning what ever we type in the terminal gets run in the container)
- <image>
In this case **nginx:latest**
- <command> that we want to run at processId1 in the new container
In this case it is **/bin/bash** (just the bash shell)
If not familiar with Linux, ProcessId1 is the thing that keeps the machine running

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> docker run -it nginx:latest /bin/bash
root@77e47d8e0399:/#
```

When we run the command , It would immediately drop us in the new container.

This is the command line for the new container created

Lets do a directory listing in the new container

The O/p: looks just a normal linux machine

```
PS C:\WINDOWS\system32> docker run -it nginx:latest /bin/bash
root@77e47d8e0399:/# ls -al
total 72
drwxr-xr-x  1 root root 4096 May 18 08:16 .
drwxr-xr-x  1 root root 4096 May 18 08:16 ..
-rw-r--r--  1 root root    0 May 18 08:16 .dockerenv
drwxr-xr-x  2 root root 4096 May 14 14:50 bin
drwxr-xr-x  2 root root 4096 May  2 16:39 boot
drwxr-xr-x  5 root root 360 May 18 08:16 dev
drwxr-xr-x  1 root root 4096 May 18 08:16 etc
drwxr-xr-x  2 root root 4096 May  2 16:39 home
drwxr-xr-x  1 root root 4096 May 15 20:15 lib
drwxr-xr-x  2 root root 4096 May 14 14:50 lib64
drwxr-xr-x  2 root root 4096 May 14 14:50 media
drwxr-xr-x  2 root root 4096 May 14 14:50 mnt
drwxr-xr-x  2 root root 4096 May 14 14:50 opt
dr-xr-xr-x 126 root root    0 May 18 08:16 proc
drwx-----  2 root root 4096 May 14 14:50 root
drwxr-xr-x  3 root root 4096 May 14 14:50 run
drwxr-xr-x  2 root root 4096 May 14 14:50 sbin
drwxr-xr-x  2 root root 4096 May 14 14:50 srv
dr-xr-xr-x 12 root root    0 May 18 08:16 sys
drwxrwxrwt  1 root root 4096 May 15 20:15 tmp
drwxr-xr-x  1 root root 4096 May 14 14:50 usr
drwxr-xr-x  1 root root 4096 May 14 14:50 var
root@77e47d8e0399:/#
```

Lets open another Powershell window to see the containers running in the host machine

```

root@77e47d8e0399:/# ls -al
total 72
drwxr-xr-x 1 root root 4096 May 18 08:16 .
drwxr-xr-x 1 root root 4096 May 18 08:16 ..
drwxr-xr-x 1 root root 0 May 18 08:16 .dockerenv
drwxr-xr-x 2 root root 4096 May 14 14:50 bin
drwxr-xr-x 2 root root 4096 May 2 16:39 boot
drwxr-xr-x 5 root root 360 May 18 08:16 dev
drwxr-xr-x 1 root root 4096 May 18 08:16 etc
drwxr-xr-x 2 root root 4096 May 18 08:16 home
drwxr-xr-x 1 root root 4096 May 18 08:16 lib
drwxr-xr-x 2 root root 4096 May 18 08:16 lib64
drwxr-xr-x 2 root root 4096 May 18 08:16 media
drwxr-xr-x 2 root root 4096 May 18 08:16 mnt
drwxr-xr-x 2 root root 4096 May 18 08:16 opt
dr-xr-xr-x 126 root root 0 May 18 08:16 proc
drwxr-xr-x 2 root root 4096 May 18 08:16 root
drwxr-xr-x 4 root root 4096 May 18 08:16 run
drwxr-xr-x 2 root root 4096 May 18 08:16 sbin
drwxr-xr-x 2 root root 4096 May 18 08:16 srv
dr-xr-xr-x 13 root root 0 May 18 08:16 sys
drwxrwxrwt 1 root root 4096 May 18 08:16 tmp
drwxr-xr-x 1 root root 4096 May 18 08:16 usr
drwxr-xr-x 1 root root 4096 May 18 08:16 var
root@77e47d8e0399:#

PS C:\Users\Gokav> docker container list
CONTAINER ID        IMAGE               COMMAND      CREATED          STATUS           PORTS
NAMES
77e47d8e0399        nginx:latest       "/bin/bash"   5 minutes ago   Up 5 minutes    80/tcp
wizardly_cartwright
PS C:\Users\Gokav>

```

//C:\Users\Gokav>
wizardly_cartwright

- Name if not provided then it would generate
- o List of adjectives
 - o List of famous computer scientists

If we **EXIT** the bash shell, we are **STOPPING** the container

STOPPING a Container

- o Docker stop <containerId/Names>

```

PS C:\Windows\system32> docker container list
CONTAINER ID        IMAGE               COMMAND      CREATED          STATUS           PORTS
NAMES
219efb014106        nginx:1.13.8       "/bin/bash"
gracious_feynman
PS C:\Windows\system32> docker stop 219

```

- o In the Powershell terminal and type **exit** in the command line

```

PS C:\Windows\system32> docker run -it nginx:1.13.8 /bin/bash
root@219efb014106:/# ls -al
total 72
drwxr-xr-x 1 root root 4096 Feb 6 00:19 .
drwxr-xr-x 1 root root 4096 Feb 6 00:19 ..
-rw-rxr-xr-x 1 root root 0 Feb 6 00:19 .dockerenv
drwxr-xr-x 2 root root 4096 Dec 10 00:00 bin
drwxr-xr-x 2 root root 4096 Nov 19 15:25 boot
drwxr-xr-x 5 root root 360 Feb 6 00:19 dev
drwxr-xr-x 1 root root 4096 Feb 6 00:19 etc
drwxr-xr-x 2 root root 4096 Nov 19 15:25 home
drwxr-xr-x 1 root root 4096 Dec 10 00:00 lib
drwxr-xr-x 2 root root 4096 Dec 10 00:00 lib64
drwxr-xr-x 2 root root 4096 Dec 10 00:00 media
drwxr-xr-x 2 root root 4096 Dec 10 00:00 mnt
drwxr-xr-x 2 root root 4096 Dec 10 00:00 opt
dr-xr-xr-x 126 root root 0 Feb 6 00:19 proc
drwxr-xr-x 2 root root 4096 Dec 10 00:00 root
drwxr-xr-x 4 root root 4096 Dec 10 00:00 run
drwxr-xr-x 2 root root 4096 Dec 10 00:00 sbin
drwxr-xr-x 2 root root 4096 Dec 10 00:00 srv
dr-xr-xr-x 13 root root 0 Feb 5 23:08 sys
drwxrwxrwt 1 root root 4096 Dec 26 18:16 tmp
drwxr-xr-x 1 root root 4096 Dec 10 00:00 usr
drwxr-xr-x 1 root root 4096 Dec 10 00:00 var
root@219efb014106:/# exit
exit
PS C:\Windows\system32>

```

This will exit bin/bash, which will exit the container

Now if we check the list of containers. There are none running

```

PS C:\Windows\system32> docker container list
CONTAINER ID        IMAGE               COMMAND      CREATED          STATUS           PORTS
NAMES
219efb014106        nginx:1.13.8       "/bin/bash"
gracious_feynman
PS C:\Windows\system32> docker container list
CONTAINER ID        IMAGE               COMMAND      CREATED          STATUS           PORTS
NAMES
PS C:\Windows\system32>

```

Try in my machine

```

PS C:\Users\Gokav> docker container list
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
77e47d8e0399        nginx:latest       "/bin/bash"            5 minutes ago      Up 5 minutes      80/tcp
wizardly_cartwright
PS C:\Users\Gokav> docker stop 77e
77e
PS C:\Users\Gokav>

```

Automatically exited the terminal

```

total 72
drwxr-xr-x  1 root root 4096 May 18 08:16 .
drwxr-xr-x  1 root root 4096 May 18 08:16 ..
-rw-rxr-xr-x  1 root root    0 May 18 08:16 .dockerenv
drwxr-xr-x  2 root root 4096 May 14 14:50 bin
drwxr-xr-x  2 root root 4096 May  2 16:39 boot
drwxr-xr-x  5 root root 360 May 18 08:16 dev
drwxr-xr-x  1 root root 4096 May 18 08:16 etc
drwxr-xr-x  2 root root 4096 May  2 16:39 home
drwxr-xr-x  1 root root 4096 May 15 20:15 lib
drwxr-xr-x  2 root root 4096 May 14 14:50 lib64
drwxr-xr-x  2 root root 4096 May 14 14:50 media
drwxr-xr-x  2 root root 4096 May 14 14:50 mnt
drwxr-xr-x  2 root root 4096 May 14 14:50 opt
dr-xr-xr-x 126 root root    0 May 18 08:16 proc
drwx----- 2 root root 4096 May 14 14:50 root
drwxr-xr-x  3 root root 4096 May 14 14:50 run
drwxr-xr-x  2 root root 4096 May 14 14:50 sbin
drwxr-xr-x  2 root root 4096 May 14 14:50 srv
dr-xr-xr-x 12 root root    0 May 18 08:16 sys
drwxrwxrwt  1 root root 4096 May 15 20:15 tmp
drwxr-xr-x  1 root root 4096 May 14 14:50 usr
drwxr-xr-x  1 root root 4096 May 14 14:50 var
root@77e47d8e0399:/# exit
PS C:\WINDOWS\system32>

```

No containers running

```

PS C:\Users\Gokav> docker container list
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
PS C:\Users\Gokav>

```

This is not completely true.

The containers are just not running but they are still present

<docker container list -a>

```

PS C:\Users\Gokav> docker container list
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
PS C:\Users\Gokav> docker container list -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
77e47d8e0399        nginx:latest       "/bin/bash"            16 minutes ago     Exited (0) 2 minutes ago
wizardly_cartwright
5cf11ad5765c        nginx:latest       "nginx -g 'daemon of..." 17 minutes ago     Exited (0) 17 minutes ago
fervent_beaver
PS C:\Users\Gokav>

```

If I want to REMOVE the container

```

PS C:\Windows\system32> docker container list -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
TS
219efb014106        nginx:1.13.8      "/bin/bash"            4 minutes ago
gracious_feynman

```

```

PS C:\Windows\system32> docker rm 219

```

Try in my machine

```

PS C:\Users\Gokav> docker container list -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
77e47d8e0399        nginx:latest       "/bin/bash"            16 minutes ago     Exited (0) 2 minutes ago
wizardly_cartwright
5cf11ad5765c        nginx:latest       "nginx -g 'daemon of..." 17 minutes ago     Exited (0) 17 minutes ago
fervent_beaver

PS C:\Users\Gokav> docker rm 77e
77e
PS C:\Users\Gokav> docker rm 5cf
5cf
PS C:\Users\Gokav> -

```

```

PS C:\Users\Gokav> docker container list
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
PS C:\Users\Gokav> -

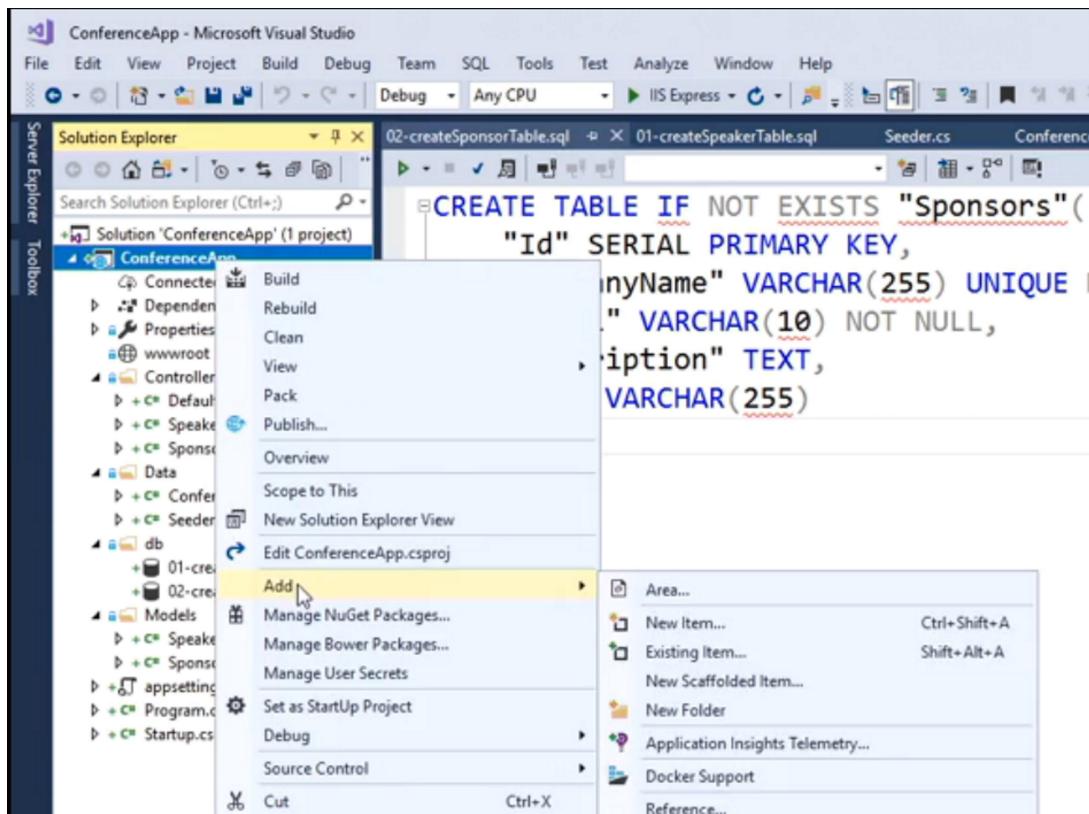
```

Lot more things can be done in command line, refer to the documentation

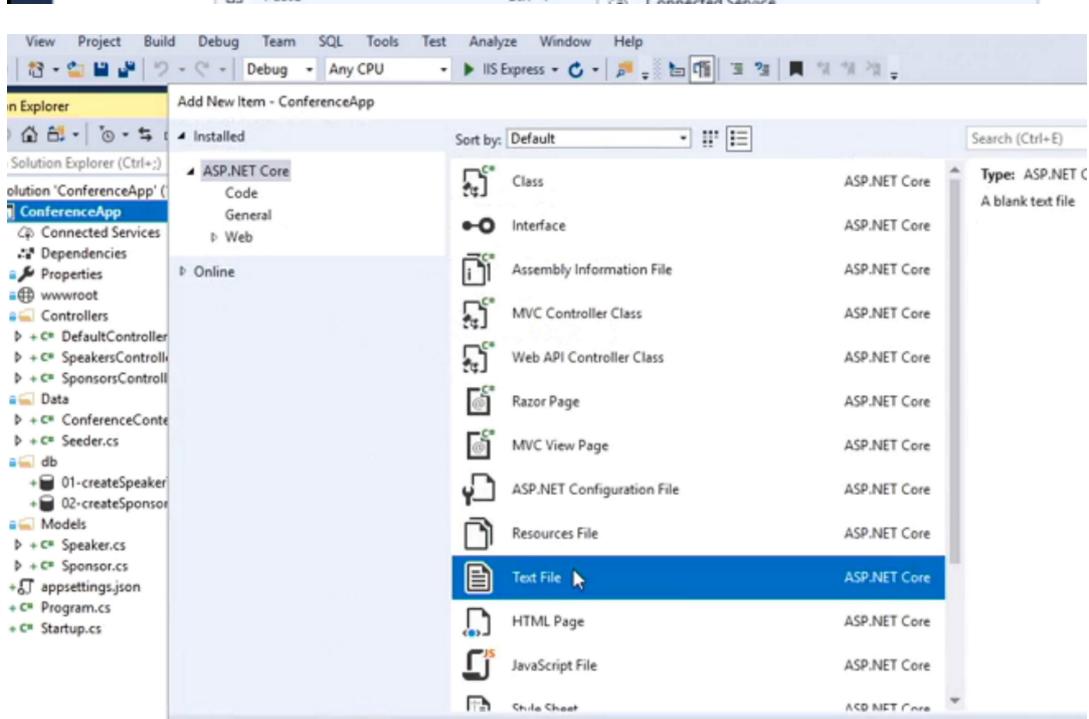
The screenshot shows the Docker documentation website. The top navigation bar includes links for 'docker docs', a search bar, and categories like 'Guides', 'Product manuals', 'Glossary', 'Reference', and 'Samples'. On the left, a sidebar menu is open under 'Get started', specifically 'Get started with Docker'. It lists several sections: 'Part 1: Orientation' (which is currently selected and highlighted in blue), 'Part 2: Containers', 'Part 3: Services', 'Part 4: Swarms', 'Part 5: Stacks', 'Part 6: Deploy your app', 'Learn by example', 'Docker overview', 'Develop with Docker', 'Configure networking', and 'Manage application data'. The main content area features a large heading 'Get Started, Part 1: Orientation and setup'. Below it, a sub-headline says 'Estimated reading time: 4 minutes'. A horizontal navigation bar at the bottom of the main content area contains six tabs: '1: Orientation' (selected), '2: Containers', '3: Services', '4: Swarms', '5: Stacks', and '6: Deploy your app'. The main text on the page starts with 'Welcome! We are excited that you want to learn Docker. The *Docker Get Started Tutorial* teaches you how to:' followed by a numbered list from 1 to 6. The first item in the list is '1. Set up your Docker environment (on this page)'. To the right of the main content, there is a sidebar with the text 'Docker is a platform for developers and sysadmins to **develop, deploy, and run** applications with containers. The use of Linux containers to deploy applications is called *containerization*. Containers are not new, but their use for easily deploying applications is.' and a small note below it: 'Containerization is increasingly popular because containers are:'.

Dockerfile - Proj

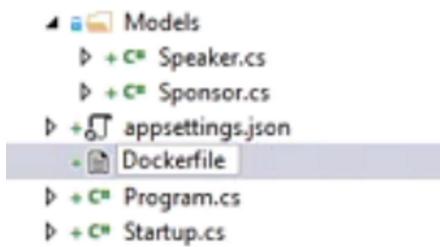
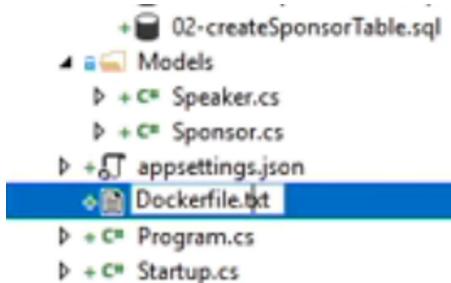
18 May 2020 14:10



The screenshot shows the Microsoft Visual Studio interface with the 'ConferenceApp' project open. In the Solution Explorer, there are several files and folders: 'ConferenceApp.csproj', '01-createSpeakerTable.sql', '02-createSponsorTable.sql', 'Seeder.cs', 'ConferenceController.cs', 'Controller.cs', 'Properties', 'wwwroot', 'Data', 'db', 'Models', 'SpeakersController.cs', 'SponsorsController.cs', 'appsettings.json', 'Program.cs', and 'Startup.cs'. A context menu is open over the '02-createSponsorTable.sql' file, showing options like 'Add New Item...', 'Manage NuGet Packages...', 'Manage Bower Packages...', 'Manage User Secrets...', 'Set as StartUp Project', 'Debug', 'Source Control', 'Cut', and 'Paste'.



The 'Add New Item' dialog is displayed, showing the 'ASP.NET Core' category under 'Installed'. The 'Text File' item is selected, highlighted with a blue background. Other items listed include 'Class', 'Interface', 'Assembly Information File', 'MVC Controller Class', 'Web API Controller Class', 'Razor Page', 'MVC View Page', 'ASP.NET Configuration File', 'Resources File', 'HTML Page', 'JavaScript File', and 'StyleSheet'. The 'Type: ASP.NET Core' dropdown is set to 'A blank text file'.



With no extension

The Docker File should start with **FROM**

There might be comments above but the **first executable command is FROM**

```
Dockerfile* 02-createSponsorTable.sql 01-cr
1 FROM
```

Mentions from which image

We can type **FROM scratch** but is **not advisable**

```
Dockerfile* 02-createSponsorTable.sql
1 FROM scratch
```

We'll build Ours from Microsoft aspnet core build image (2.0 version)

```
Dockerfile* 02-createSponsorTable.sql 01-createSpeakerTable.sql Seeder.cs
1 FROM microsoft/aspnetcore-build:2.0
```

And **alias it as build** (we are going to use it later in the docker file)

Because we need to give it a name we can refer to by .. Later in the file

```
Dockerfile* 02-createSponsorTable.sql 01-createSpeakerTable.sql Seeder.cs ConferenceContext.cs
1 FROM microsoft/aspnetcore-build:2.0 AS build
```

The first thing we are going to do is setting up the working directory inside the newly created image
Using the **WORKDIR** command

```
Dockerfile* 02-createSponsorTable.sql 01-createSpeakerTable.sql Seeder.cs ConferenceContext.cs Sp
1 FROM microsoft/aspnetcore-build:2.0 AS build
2 WORKDIR /build
```

This would be the build folder inside the container

Anything below the WORKDIR line would be running inside the working directory /build folder in our container

Now some commands

COPY all our code from

Current working directory of our **host** ---to---> working directory of the **container**

```
Dockerfile* 02-createSponsorTable.sql 01-createSpeakerTable.sql Seeder.cs ConferenceContext.cs
1 FROM microsoft/aspnetcore-build:2.0 AS build
2 WORKDIR /build
3 COPY . .
```

Restore and build our code

RUN dotnet restore

RUN dotnet publish

Publish because we are building a production website (this needs switches)

RUN dotnet publish -c Release -o output

- -c : configuration type is Release
- -o : output (this will put the published inside the build folder in a folder by name output)

The **dotnet restore** command uses NuGet to restore dependencies as well as project-specific tools that are specified in the project file. In most cases, you don't need to explicitly use the **dotnet restore** command, since a NuGet restore is run implicitly if necessary when you run the following commands:

- [dotnet new](#)
- [dotnet build](#)
- [dotnet build-server](#)
- [dotnet run](#)
- [dotnet test](#)
- [dotnet publish](#)
- [dotnet pack](#)

From <<https://docs.microsoft.com/en-us/dotnet/core/tools/dotnet-restore>>

dotnet publish compiles the application, reads through its dependencies specified in the project file, and publishes the resulting set of files to a directory. The output includes the following assets:

- Intermediate Language (IL) code in an assembly with a *dll* extension.
- A *deps.json* file that includes all of the dependencies of the project.
- A *runtimeconfig.json* file that specifies the shared runtime that the application expects, as well as other configuration options for the runtime (for example, garbage collection type).
- The application's dependencies, which are copied from the NuGet cache into the output folder.

The **dotnet publish** command's output is ready for deployment to a hosting system (for example, a server, PC, Mac, laptop) for execution. It's the only officially supported way to prepare the application for deployment. Depending on the type of deployment that the project specifies, the hosting system may or may not have the .NET Core shared runtime installed on it. For more information, see [Publish .NET Core apps with the .NET Core CLI](#).

Implicit restore

You don't have to run [dotnet restore](#) because it's run implicitly by all commands that require a restore to occur, such as dotnet new, dotnet build, dotnet run, dotnet test, dotnet publish, and dotnet pack. To disable implicit restore, use the --no-restore option.

The dotnet restore command is still useful in certain scenarios where explicitly restoring makes sense, such as [continuous integration builds in Azure DevOps Services](#) or in build systems that need to explicitly control when the restore occurs.

For information about how to manage NuGet feeds, see the [dotnet restore documentation](#).

MSBuild

The dotnet publish command calls MSBuild, which invokes the Publish target. Any parameters passed to dotnet publish are passed to MSBuild. The -c and -o parameters map to MSBuild's Configuration and OutputPath properties, respectively.

The dotnet publish command accepts MSBuild options, such as -p for setting properties and -l to define a logger. For example, you can set an MSBuild property by using the format: -p:<NAME>=<VALUE>. You can also set publish-related properties by referring to a *.pubxml* file, for example:

.NET Core CLI

```
dotnet publish -p:PublishProfile=Properties\PublishProfiles\FolderProfile.pubxml  
From <https://docs.microsoft.com/en-us/dotnet/core/tools/dotnet-publish>
```

Arguments

• PROJECT|SOLUTION

The project or solution to publish.

- PROJECT is the path and filename of a C#, F#, or Visual Basic project file, or the path to a directory that contains a C#, F#, or Visual Basic project file. If the directory is not specified, it defaults to the current directory.
- SOLUTION is the path and filename of a solution file (.sln extension), or the path to a directory that contains a solution file. If the directory is not specified, it defaults to the current directory. Available since .NET Core 3.0 SDK.

Options

• -c|--configuration <CONFIGURATION>

Defines the build configuration. The default for most projects is Debug, but you can override the build configuration settings in your project.

• -f|--framework <FRAMEWORK>

Publishes the application for the specified [target framework](#). You must specify the target framework in the project file.

• --force

Forces all dependencies to be resolved even if the last restore was successful.

Specifying this flag is the same as deleting the *project.assets.json* file.

• -h|--help

Prints out a short help for the command.

• --interactive

Allows the command to stop and wait for user input or action. For example, to complete authentication. Available since .NET Core 3.0 SDK.

• --manifest <PATH_TO_MANIFEST_FILE>

Specifies one or several [target manifests](#) to use to trim the set of packages published with the app. The manifest file is part of the output of the [dotnet store command](#). To specify multiple manifests, add a --manifest option for each manifest.

• --no-build

Doesn't build the project before publishing. It also implicitly sets the --no-restore flag.

• --no-dependencies

Ignores project-to-project references and only restores the root project.

• --nologo

Doesn't display the startup banner or the copyright message. Available since .NET Core 3.0 SDK.

• --no-restore

Doesn't execute an implicit restore when running the command.

• -o|--output <OUTPUT_DIRECTORY>

From <<https://docs.microsoft.com/en-us/dotnet/core/tools/dotnet-publish>>

Problem is the `aspnetcore-build:2.0` image is almost 2GB, this is not good for production website.
To fix this we can add another FROM line to build from lighter image (`aspnetcore:2.0` is only 300mb)

Add another FROM line and build it from a lighter image `aspnetcore:2.0` is only 300mb
This is because it doesn't have all command line options in it

```
Dockerfile* 02-createSponsorTable.sql 01-createSpeakerTable.sql Seeder.cs ConferenceContext.cs
1 FROM microsoft/aspnetcore-build:2.0 AS build
2 WORKDIR /build
3 COPY .
4 RUN dotnet restore
5 RUN dotnet publish -c Release -o output
6
7 FROM microsoft/aspnetcore:2.0
```

Set the working directory "/app"

```
Dockerfile* 02-createSponsorTable.sql 01-createSpeakerTable.sql Seeder.cs ConferenceContext.cs
1 FROM microsoft/aspnetcore-build:2.0 AS build
2 WORKDIR /build
3 COPY .
4 RUN dotnet restore
5 RUN dotnet publish -c Release -o output
6
7 FROM microsoft/aspnetcore:2.0
8 WORKDIR /app
```

Now we will copy all of our code from the original build directory to the app directory (current working directory)

```
Dockerfile* 02-createSponsorTable.sql 01-createSpeakerTable.sql Seeder.cs ConferenceContext.cs
1 FROM microsoft/aspnetcore-build:2.0 AS build
2 WORKDIR /build
3 COPY .
4 RUN dotnet restore
5 RUN dotnet publish -c Release -o output
6
7 FROM microsoft/aspnetcore:2.0
8 WORKDIR /app
9 COPY --from=build /build/output .
```

<code>build</code>	is the original image
<code>/build/output</code>	is the folder
.	is the current directory (" <code>/app</code> ")

Now let us give this image an `ENTRYPOINT`
This is the thing we want to run from `Process Id 1` when we create a container from this image

It uses the exact style command, which is an array

First element of the array	Command that we want to run	In our case it is <code>dotnet</code>
All the other elements are	Arguments that we want to pass	Is the <code>dll</code> that has been built from the original build image

```
1 FROM microsoft/aspnetcore-build:2.0 AS build
2 WORKDIR /build
3 COPY . .
4 RUN dotnet restore
5 RUN dotnet publish -c Release -o output
6
7 FROM microsoft/aspnetcore:2.0
8 WORKDIR /app
9 COPY --from=build /build/output .
10 ENTRYPOINT ["dotnet", "ConferenceApp.dll"]
```

We have Docker File ready.

We are ready to build an image from the Docker file.

Lot more that can be done with the docker file

Explore..

The screenshot shows a section of the Docker documentation titled "Dockerfile reference". On the left, there's a sidebar with a "File formats" dropdown menu containing options like "Dockerfile reference" (which is selected), "Compose file reference", "Cloud stack file reference", "Command-Line Interfaces (CLIs)", "Application Programming Interfaces (APIs)", "Drivers and specifications", and "Compliance control references". To the right of the sidebar, the main content area has a header "Estimated reading time: 89 minutes" and a large title "Dockerfile reference". Below the title, there's a paragraph about Dockerfiles and a link to "Dockerfile Best Practices". Further down, there's a section titled "Usage" with a paragraph about the docker build command and its context, followed by a code snippet showing the command being run in a terminal.

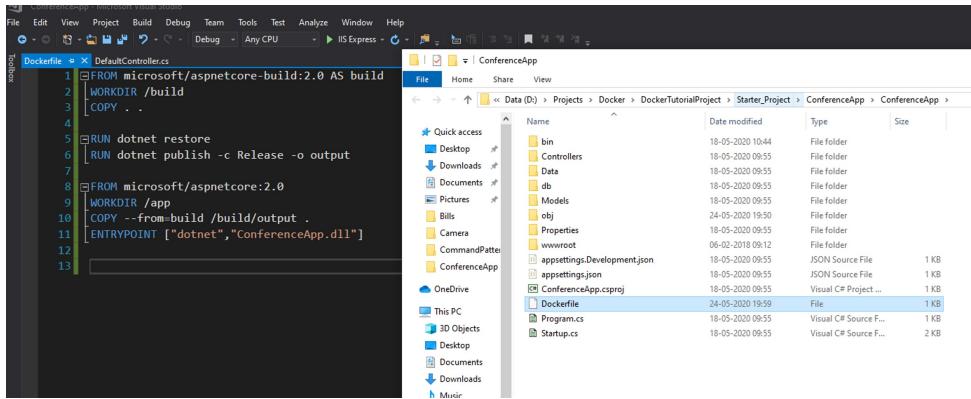
<https://docs.docker.com/engine/reference/builder>

Dockerfile - Build an image

18 May 2020 14:12

We can spin up containers from an existing image.
The way we do this by using [Docker build command](#)

Make sure we are in the directory where the docker file is



```
Administrator: Windows PowerShell
PS C:\WINDOWS> cd D:\Projects\Docke\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp
PS D:\Projects\Docke\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp
```

docker build -t conference/api.

-t	is for tagging the image to give it a name
Conference/api	Is the name of the image
.	To look for a Dockerfile file in the directory

```
PS D:\Projects\Docke\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker build -t conference/api .
Sending build context to Docker daemon 1.029MB
Step 1/9 : FROM microsoft/aspnetcore-build:2.0 AS build
2.0: Pulling from microsoft/aspnetcore-build
55cbf04beb70: Downloading [=====]
1607093a898c: Downloading [=====]
9a8ea045c926: Downloading [=====]
d4ee24d4dac: Waiting
996a3dbf48f: Waiting
e626b8d73a89: Waiting
4d833e0fa218: Waiting
leebc8a262e7: Waiting
4867c32ba582: Waiting
c0ece037099c: Waiting
-
```

```
PS D:\Projects\Docke\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker build -t conference/api .
Sending build context to Docker daemon 1.029MB
Step 1/9 : FROM microsoft/aspnetcore-build:2.0 AS build
2.0: Pulling from microsoft/aspnetcore-build
55cbf04beb70: Pull complete
1607093a898c: Pull complete
9a8ea045c926: Pull complete
d4ee24d4dac: Pull complete
996a3dbf48f: Pull complete
e626b8d73a89: Downloading [=====>]
4d833e0fa218: Downloading [=>]
leebc8a262e7: Downloading [=>]
4867c32ba582: Waiting
c0ece037099c: Waiting
-
```

```
PS D:\Projects\Docke\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker build -t conference/api .
Sending build context to Docker daemon 1.029MB
Step 1/9 : FROM microsoft/aspnetcore-build:2.0 AS build
2.0: Pulling from microsoft/aspnetcore-build
55cbf04beb70: Pull complete
1607093a898c: Pull complete
9a8ea045c926: Pull complete
d4ee24d4dac: Pull complete
996a3dbf48f: Pull complete
e626b8d73a89: Downloading [=----->]
4d833e0fa218: Downloading [=----->]
leebc8a262e7: Download complete
4867c32ba582: Download complete
c0ece037099c: Downloading [=---->]
-
```

```
PS D:\Projects\Docke\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker build -t conference/api .
Sending build context to Docker daemon 1.029MB
Step 1/9 : FROM microsoft/aspnetcore-build:2.0 AS build
2.0: Pulling from microsoft/aspnetcore-build
55cbf04beb70: Pull complete
1607093a898c: Pull complete
9a8ea045c926: Pull complete
d4ee24d4dac: Pull complete
996a3dbf48f: Pull complete
e626b8d73a89: Pull complete
4d833e0fa218: Extracting [=----->]
leebc8a262e7: Download complete
4867c32ba582: Download complete
c0ece037099c: Download complete
-
```

```
PS D:\Projects\Docke\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker build -t conference/api .
Sending build context to Docker daemon 1.029MB
Step 1/9 : FROM microsoft/aspnetcore-build:2.0 AS build
2.0: Pulling from microsoft/aspnetcore-build
55cbf04beb70: Pull complete
1607093a898c: Pull complete
9a8ea045c926: Pull complete
d4ee24d4dac: Pull complete
996a3dbf48f: Pull complete
e626b8d73a89: Pull complete
4d833e0fa218: Pull complete
leebc8a262e7: Pull complete
4867c32ba582: Pull complete
c0ece037099c: Extracting [=----->] 91.21MB/91.21MB
-
```

```

PS D:\Projects\ Docker\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker build -t conference/api .
Sending build context to Docker daemon 1.029MB
Step 1/9 : FROM microsoft/aspnetcore-build:2.0 AS build
2.0: Pulling from microsoft/aspnetcore-build
55cbf04beb70: Pull complete
1607093a898c: Pull complete
9a8ea045c926: Pull complete
d4eee24d4dac: Pull complete
9996a3dbf48f: Pull complete
e62608d73a89: Pull complete
4d835edfa218: Pull complete
leebe8a262e7: Pull complete
4867c32ba582: Pull complete
c0ccc037099c: Pull complete
Digest: sha256:ab861527a8485e7df91069e80cd7a94237c22995f13494c2cccc071b76e347f0
Status: Downloaded newer image for microsoft/aspnetcore-build:2.0
--> 86a6525397c2
Step 2/9 : WORKDIR /build
--> Running in 19b59c46077a
Removing intermediate container 19b59c46077a
--> e8e7ed550b17
Step 3/9 : COPY .
--> d58bee03044
Step 4/9 : RUN dotnet restore
--> Running in 6f76f929dd78
Restoring packages for /build/ConferenceApp.csproj...
Restoring packages for /build/ConferenceApp.csproj...
Installing Microsoft.DotNet.Watcher.Tools 2.0.0.
Restore completed in 3.48 sec for /build/ConferenceApp.csproj.
Restoring packages for /build/ConferenceApp.csproj...

```

```

Installing Microsoft.AspNetCore.Mvc.Formatters.Json 2.0.2.
Installing Microsoft.AspNetCore.Mvc.Localization 2.0.2.
Installing Microsoft.Extensions.Logging.Abstractions 2.0.0.
Installing Microsoft.Extensions.DependencyInjection.Binder 2.0.0.
Installing Microsoft.Extensions.Options 2.0.0.
Installing Microsoft.Extensions.WebEncoders 2.0.0.
Installing Microsoft.Extensions.Logging 2.0.0.
Installing Microsoft.Extensions.Options.ConfigurationExtensions 2.0.0.
Installing Microsoft.AspNetCore.DataProtection.Extensions 2.0.1.
Installing Microsoft.Extensions.Configuration 2.0.0.
Installing Microsoft.Extensions.Configuration.Abstractions 2.0.0.
Installing Microsoft.Extensions.FileProviders.Embedded 2.0.0.
Installing Microsoft.AspNetCore 2.0.1.
Installing Microsoft.AspNetCore.Http.Extensions 2.0.1.
Installing Microsoft.AspNetCore.DataProtection.AzureStorage 2.0.1.
Installing Microsoft.AspNetCore.Identity 2.0.1.
Installing Microsoft.AspNetCore.Html.Abstractions 2.0.0.
Installing Microsoft.EntityFrameworkCore 2.0.1.
Installing Npgsql 3.2.6.
Installing Microsoft.EntityFrameworkCore.Relational 2.0.1.
Installing Microsoft.VisualStudio.Web.CodeGeneration.Design 2.0.2.
Installing Microsoft.AspNetCore.All 2.0.5.
Installing Npgsql.EntityFrameworkCore.PostgreSQL 2.0.1.
Generating MSBuild file /build/obj/ConferenceApp.csproj.nuget.g.props.
Generating MSBuild file /build/obj/ConferenceApp.csproj.nuget.g.targets.
Restore completed in 18.04 sec for /build/ConferenceApp.csproj.
Removing intermediate container 6f76f929dd78
--> 24e9b03001e8
Step 5/9 : RUN dotnet publish -c Release -o output
--> Running in c7d51a07d64e
Microsoft (R) Build Engine version 15.7.179.6572 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

Restore completed in 45.39 ms for /build/ConferenceApp.csproj.
Restore completed in 21.62 ms for /build/ConferenceApp.csproj.
Restore completed in 25.06 ms for /build/ConferenceApp.csproj.
ConferenceApp --> /build/bin/Release/netcoreapp2.0/ConferenceApp.dll
ConferenceApp --> /build/output/
Removing intermediate container c7d51a07d64e
--> 5cc3175a82ab
Step 6/9 : FROM microsoft/aspnetcore:2.0
2.0: Pulling from microsoft/aspnetcore

```

```

Installing Microsoft.AspNetCore.All 2.0.5.
Installing Npgsql.EntityFrameworkCore.PostgreSQL 2.0.1.
Generating MSBuild file /build/obj/ConferenceApp.csproj.nuget.g.props.
Generating MSBuild file /build/obj/ConferenceApp.csproj.nuget.g.targets.
Restore completed in 18.04 sec for /build/ConferenceApp.csproj.
Removing intermediate container 6f76f929dd78
--> 24e9b03001e8
Step 5/9 : RUN dotnet publish -c Release -o output
--> Running in c7d51a07d64e
Microsoft (R) Build Engine version 15.7.179.6572 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

Restore completed in 45.39 ms for /build/ConferenceApp.csproj.
Restore completed in 21.62 ms for /build/ConferenceApp.csproj.
Restore completed in 25.06 ms for /build/ConferenceApp.csproj.
ConferenceApp --> /build/bin/Release/netcoreapp2.0/ConferenceApp.dll
ConferenceApp --> /build/output/
Removing intermediate container c7d51a07d64e
--> 5cc3175a82ab
Step 6/9 : FROM Microsoft/aspnetcore:2.0
2.0: Pulling from microsoft/aspnetcore
55cbf04beb70: Already exists
2ad9fb0b9dd3d: Pull complete
7debd121b442: Pull complete
76fbad0100d: Pull complete
1cba46fa0c00: Downloading [=====] 38.68MB/56.2MB

```

```

Installing Npgsql.EntityFrameworkCore.PostgreSQL 2.0.1.
Generating MSBuild file /build/obj/ConferenceApp.csproj.nuget.g.props.
Generating MSBuild file /build/obj/ConferenceApp.csproj.nuget.g.targets.
Restore completed in 18.04 sec for /build/ConferenceApp.csproj.
Removing intermediate container 6f76f929dd78
--> 24e9b03001e8
Step 7/9 : RUN dotnet publish -c Release -o output
--> Running in c7d51a07d64e
Microsoft (R) Build Engine version 15.7.179.6572 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

Restore completed in 45.39 ms for /build/ConferenceApp.csproj.
Restore completed in 21.62 ms for /build/ConferenceApp.csproj.
Restore completed in 25.06 ms for /build/ConferenceApp.csproj.
ConferenceApp --> /build/bin/Release/netcoreapp2.0/ConferenceApp.dll
ConferenceApp --> /build/output/
Removing intermediate container c7d51a07d64e
--> 5cc3175a82ab
Step 8/9 : FROM microsoft/aspnetcore:2.0
2.0: Pulling from microsoft/aspnetcore
55cbf04beb70: Already exists
2ad9fb0b9dd3d: Pull complete
7debd121b442: Pull complete
76fbad0100d: Pull complete
1cba46fa0c00: Pull complete
Digest: sha256:780891568a92b7f1d08a4bab07ef811e6417808847dcaf698a4a788ab11
Status: Downloaded newer image for microsoft/aspnetcore:2.0
--> dbb30c19e94b
Step 8/9 : WORKDIR /app
--> Running in 1cba46fa0c00
Removing intermediate container 1cba46fa0c00
--> 01ea57cd83dc
Step 8/9 : COPY --from=build /build/output .
--> 300832c52428
Step 8/9 : RUN dotnet publish -c Release -o output
--> Running in fe6be19d054c
Removing intermediate container fe6be19d054c
--> 9808818e0d1c
Successfully built 9808818e0d1c
Successfully tagged conference/api:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and directories added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permissions for sensitive files and directories.
PS D:\Projects\ Docker\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp>

```

The security warning..

Let's check if the image was created

```
PS D:\Projects\ Docker\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker image list
REPOSITORY TAG IMAGE ID CREATED SIZE
conference/api latest 9808318e0d1c 2 minutes ago 352MB
<none> <none> 5cc3175a82ab 2 minutes ago 2.09GB
nginx latest 9beeba249f3e 8 days ago 127MB
hello-world latest b7f56fb1ae65 4 months ago 13.3kB
microsoft/aspnetcore-build 2.0 06a6525397c2 21 months ago 2.02GB
microsoft/aspnetcore 2.0 db030c19e94b 21 months ago 347MB
PS D:\Projects\ Docker\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp>
```

Check the sizes..

```
PS D:\Projects\ Docker\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker image list
REPOSITORY TAG IMAGE ID CREATED SIZE
conference/api latest 9808318e0d1c 2 minutes ago 352MB
<none> <none> 5cc3175a82ab 2 minutes ago 2.09GB
nginx latest 9beeba249f3e 8 days ago 127MB
hello-world latest b7f56fb1ae65 4 months ago 13.3kB
microsoft/aspnetcore-build 2.0 06a6525397c2 21 months ago 2.02GB
microsoft/aspnetcore 2.0 db030c19e94b 21 months ago 347MB
PS D:\Projects\ Docker\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp>
```

Few things to notice:

- The `conference/api` is tagged with "latest" as we did not mention anything
- Size of the `conference/api` image is only 5mb larger than the `microsoft/aspnetcore` image
 - This is because all it has on top of the `microsoft/aspnetcore` image is the code
 - Since images are based on layers, **It actually is taking only 5mb of diskspace**
- Because it is sharing the 347mb from the `microsoft/aspnetcore` image

- `<none>` image with the tag `<none>`
 - This is the intermediate image that we have created while doing the build step in the Dockerfile

This way we would see many such images created.. To avoid this there is an easy command to clean this up

`<docker system prune>`

```
PS D:\Projects\ Docker\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker image list
REPOSITORY TAG IMAGE ID CREATED SIZE
conference/api latest 9808318e0d1c 2 minutes ago 352MB
<none> <none> 5cc3175a82ab 2 minutes ago 2.09GB
nginx latest 9beeba249f3e 8 days ago 127MB
hello-world latest b7f56fb1ae65 4 months ago 13.3kB
microsoft/aspnetcore-build 2.0 06a6525397c2 21 months ago 2.02GB
microsoft/aspnetcore 2.0 db030c19e94b 21 months ago 347MB
PS D:\Projects\ Docker\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker system prune
WARNING! This will remove:
- all stopped containers
- all networks not used by at least one container
- all dangling images
- all dangling build cache

Are you sure you want to continue? [y/N]
```

its going to remove

- All stopped container
- All networks not used by at least one container
- All dangling images
- All dangling build cache

```
PS D:\Projects\ Docker\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker image list
REPOSITORY TAG IMAGE ID CREATED SIZE
conference/api latest 9808318e0d1c 2 minutes ago 352MB
<none> <none> 5cc3175a82ab 2 minutes ago 2.09GB
nginx latest 9beeba249f3e 8 days ago 127MB
hello-world latest b7f56fb1ae65 4 months ago 13.3kB
microsoft/aspnetcore-build 2.0 06a6525397c2 21 months ago 2.02GB
microsoft/aspnetcore 2.0 db030c19e94b 21 months ago 347MB
PS D:\Projects\ Docker\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker system prune
WARNING! This will remove:
- all stopped containers
- all networks not used by at least one container
- all dangling images
- all dangling build cache

Are you sure you want to continue? [y/N]
Deleted Images:
deleted: sha256:5cc3175a82abdd17732a0b31c2f78a8be8b8184fb47e85668215fc93996895ac
deleted: sha256:b0fa9a37b34eb60b24ad02c9df8c65b9e6885b18868c935f6aeb9d0eb563df2
deleted: sha256:24e9b03001e81edc3bf4cf6f937:94371b35bb48eb17dac3202d350cdcbcc089
deleted: sha256:71af52863ead50d71720d93c6df0577752cd0c4d48c4f21f16ec1d892f9df90
deleted: sha256:d58beef03044b6cbd517d73804b316ceac2ad2a5d7a54e2577a986debc25a
deleted: sha256:9876ff5a4df2fcfd5572dc342a7bef1fa928eccaa7d0ccb87fa42f190b9e1bba
deleted: sha256:e8e7ed550b17dc7f09606587b44e35fa4cbe2aa38c1lcce404a288beaccf5e69
deleted: sha256:bd93c2e78449a901e021db08b793f67e88d55c4f5bc13b2e5fsa2a434227bf78

Total reclaimed space: 67.95MB
PS D:\Projects\ Docker\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker image list
REPOSITORY TAG IMAGE ID CREATED SIZE
conference/api latest 9808318e0d1c 13 minutes ago 352MB
nginx latest 9beeba249f3e 8 days ago 127MB
hello-world latest b7f56fb1ae65 4 months ago 13.3kB
microsoft/aspnetcore-build 2.0 06a6525397c2 21 months ago 2.02GB
microsoft/aspnetcore 2.0 db030c19e94b 21 months ago 347MB
PS D:\Projects\ Docker\ DockerTutorialProject\Starter_Project\ConferenceApp\ConferenceApp>
```

The `<none>` image is gone

Dockerfile - Run a Container from the new image

24 May 2020 19:47

Now lets run a container from this image conference/api

```
PS D:\Projects\Docker\TutorialProject\Starter_Project\ConferenceApp\ConferenceApp> docker image list
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
conference/api     latest   9808318e0d1c  14 minutes ago  352MB
nginx              latest   9beeba249f3e  8 days ago    127MB
hello-world         latest   bf756fb1ae65  4 months ago   13.3kB
microsoft/aspnetcore-build  2.0   06a6525397c2  21 months ago  2.02GB
microsoft/aspnetcore  2.0   db030c19e94b  21 months ago  347MB
PS D:\Projects\Docker\TutorialProject\Starter_Project\ConferenceApp\ConferenceApp> -
```

To run a container we use `docker run`

`docker run -d --name conference_api conference/api:latest`

-d	Detached
--name	To specify a name, if not docker will give it an auto generated name
<conference_api>	Its a name that we choose to run this container with
conference/api	The image we want to build the container from
Or	
conference/api:latest	latest: is the tag name If we don't specify a tag it would assume latest

```
PS C:\WINDOWS> docker image list
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
conference/api     latest   9808318e0d1c  22 minutes ago  352MB
nginx              latest   9beeba249f3e  8 days ago    127MB
hello-world         latest   bf756fb1ae65  4 months ago   13.3kB
microsoft/aspnetcore-build  2.0   06a6525397c2  21 months ago  2.02GB
microsoft/aspnetcore  2.0   db030c19e94b  21 months ago  347MB
PS C:\WINDOWS> docker run -d --name conference_api conference/api
e9b9f0cfa07bb96e96889aaeff3b7e154fd2e8aa37e55ac9d441d940d54cdaf4
PS C:\WINDOWS>
```

After running the command we see a hash

```
PS C:\WINDOWS> docker image list
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
conference/api     latest   9808318e0d1c  22 minutes ago  352MB
nginx              latest   9beeba249f3e  8 days ago    127MB
hello-world         latest   bf756fb1ae65  4 months ago   13.3kB
microsoft/aspnetcore-build  2.0   06a6525397c2  21 months ago  2.02GB
microsoft/aspnetcore  2.0   db030c19e94b  21 months ago  347MB
PS C:\WINDOWS> docker run -d --name conference_api conference/api
e9b9f0cfa07bb96e96889aaeff3b7e154fd2e8aa37e55ac9d441d940d54cdaf4
PS C:\WINDOWS> docker container list
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
PS C:\WINDOWS>
```

But when we try to check for the list of containers.. No containers show up. Why ?

Reason: the container failed to start

Lets try listing all the containers

```
PS C:\WINDOWS> docker container list
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
PS C:\WINDOWS> docker container list -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
e9b9f0cfa07b        conference/api     "dotnet ConferenceAp..."  2 minutes ago     Exited (139)  2 minutes ago
PS C:\WINDOWS> -
PS C:\WINDOWS> docker container list
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
PS C:\WINDOWS> docker container list -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
e9b9f0cfa07b        conference/api     "dotnet ConferenceAp..."  2 minutes ago     Exited (139)  2 minutes ago
PS C:\WINDOWS>
```

It shows that it has exited with the status code 139

Anything other than 0 as status code is an error

We can find more details by running the `docker logs` command

```
docker logs <name of the container>
```

```

PS C:\WINDOWS> docker container list
CONTAINER ID        IMAGE               COMMAND       CREATED          STATUS           PORTS     NAMES
PS C:\WINDOWS> docker container list -a
CONTAINER ID        IMAGE               COMMAND       CREATED          STATUS           PORTS     NAMES
e9b9f0cfa07b        conference/api      "dotnet ConferenceAp..."   2 minutes ago   Exited (139) 2 minutes ago
PS C:\WINDOWS> docker log conference_api
docker: 'log' is not a docker command.
See 'docker --help'
PS C:\WINDOWS> docker logs conference_api
warn: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[35]
  No XML encryptor configured. Key {00f29cff-f93f-41b4-8a7f-1900cf538d93} may be persisted to storage in unencrypted form.
fail: Microsoft.EntityFrameworkCore.Database.Connection[20004]
  An error occurred using the connection to database 'conference' on server 'tcp://postgres:5432'.
System.Net.Internals.SocketExceptionFactory+ExtendedSocketException (0x00000005): No such device or address
  at System.Net.Dns.InternalGetHostByName(String hostName, Boolean includeIPv6)
  at System.Net.Dns.GetHostAddresses(String hostNameOrAddress)
  at Npgsql.NpgsqlConnector.Connect(NpgsqlTimeout timeout)
  at Npgsql.NpgsqlConnector.<RawOpen>d__146.MoveNext()
--- End of stack trace from previous location where exception was thrown ---
  at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()
  at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)
  at Npgsql.NpgsqlConnector.<Open>d__143.MoveNext()
--- End of stack trace from previous location where exception was thrown ---
  at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()
  at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)
  at Npgsql.NpgsqlConnectorPool.<AllocateLong>d__24.MoveNext()
--- End of stack trace from previous location where exception was thrown ---
  at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()
  at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)
  at System.Runtime.CompilerServices.TaskAwaiter`1.GetResult()
  at System.Runtime.CompilerServices.ValueTaskAwaiter`1.GetResult()
  at Npgsql.NpgsqlConnection.<Open>d__28.MoveNext()
--- End of stack trace from previous location where exception was thrown ---
  at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()
  at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)
  at Npgsql.NpgsqlConnection.Open()
  at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.Open(Boolean errorsExpected)
fail: Microsoft.EntityFrameworkCore.Query[10100]
  An exception occurred in the database while iterating the results of a query for context type 'ConferenceApp.Data.ConferenceContext'.
System.Net.Internals.SocketExceptionFactory+ExtendedSocketException (0x00000005): No such device or address
  at System.Net.Dns.InternalGetHostByName(String hostName, Boolean includeIPv6)
  at System.Net.Dns.GetHostAddresses(String hostNameOrAddress)
  at Npgsql.NpgsqlConnector.Connect(NpgsqlTimeout timeout)
  at Npgsql.NpgsqlConnector.<RawOpen>d__146.MoveNext()
--- End of stack trace from previous location where exception was thrown ---
  at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()
  at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)
  at Npgsql.NpgsqlConnector.<Open>d__143.MoveNext()
  End of stack trace from previous location where exception was thrown

```

This is because we do not have a database..

What we need is to start up the database when we start up an API

This is what **Docker Compose** is made for

Dockerfile - cleanup

24 May 2020 19:48

Lets remove the container and image that we create to start with

```
PS C:\WINDOWS> docker image list
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
conference/api     latest   9808318e0d1c  33 minutes ago  352MB
nginx              latest   9beeba249f3e  8 days ago    127MB
hello-world         latest   bf756fb1ae65  4 months ago   13.3kB
microsoft/aspnetcore-build 2.0   06a6525397c2  21 months ago  2.02GB
microsoft/aspnetcore 2.0    db030c19e94b  21 months ago  347MB
PS C:\WINDOWS> docker container list -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                 NAMES
e9b9f0cfa07b       conference/api      "dotnet ConferenceAp..."   10 minutes ago   Exited (139) 10 minutes ago
PS C:\WINDOWS> -
```

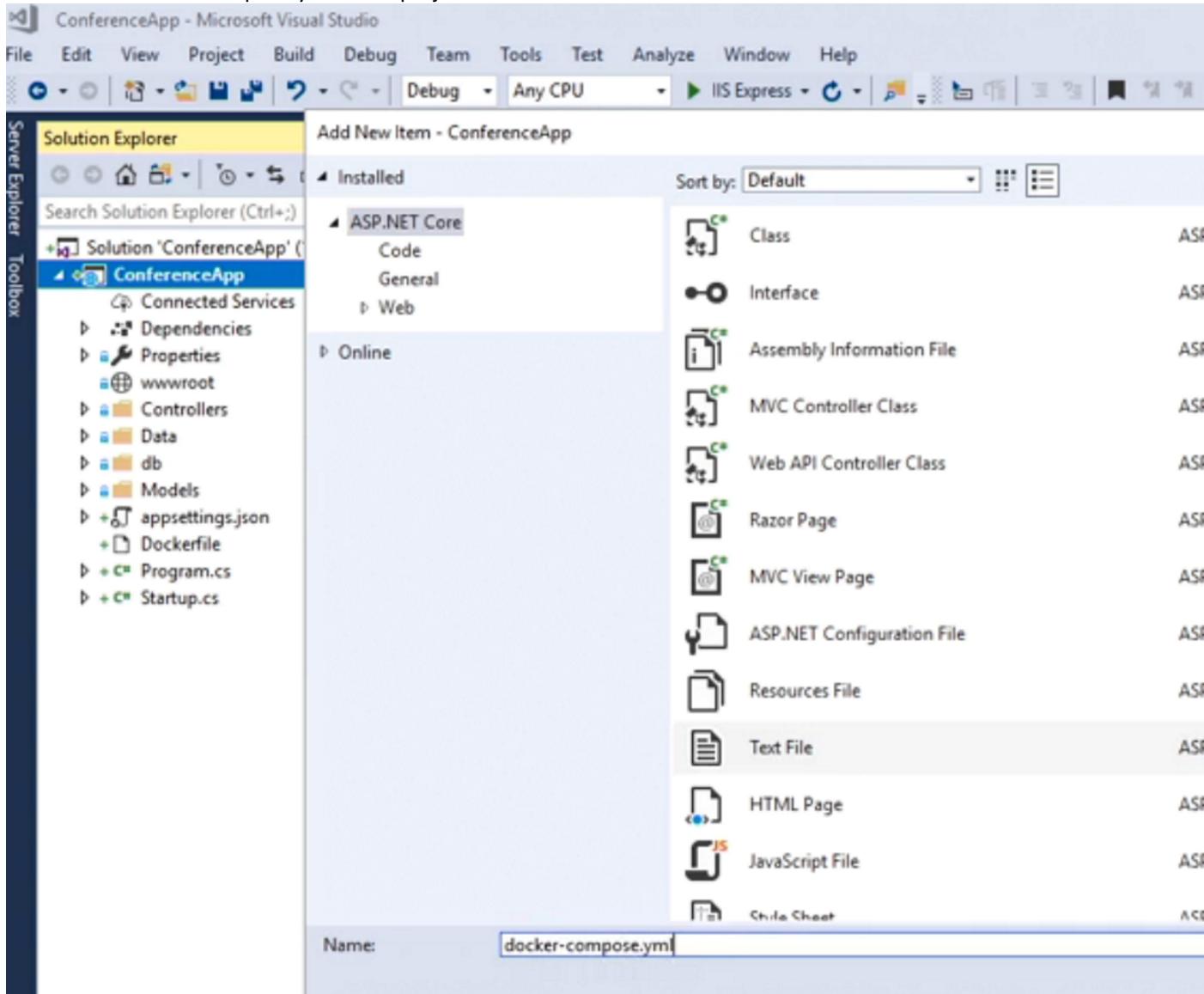
```
PS C:\WINDOWS> docker rm conference_api
conference_api
PS C:\WINDOWS> docker rmi conference/api
Untagged: conference/api:latest
Deleted: sha256:9808318e0d1c119fd4b2df1c91b2b68bfae1a67399207c32395db137872e31
Deleted: sha256:300832c5242814961998fd1cd7557640dae1ad085efd2290cc538ab75a347532
Deleted: sha256:f6d04ee860b215f0cf49e80fe57cad3b7d06a030d8510900dec7f751ff32c56a
Deleted: sha256:018a57edb3dce4f53a90b5978403f8cba3380ad0e46a4df100f78c6937d122b6
Deleted: sha256:b29817a6ca6496e5a0ee5262892b86185f79672b1b004356a5760b9db6e344c2
PS C:\WINDOWS> docker image list
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
nginx              latest   9beeba249f3e  8 days ago    127MB
hello-world         latest   bf756fb1ae65  4 months ago   13.3kB
microsoft/aspnetcore-build 2.0   06a6525397c2  21 months ago  2.02GB
microsoft/aspnetcore 2.0    db030c19e94b  21 months ago  347MB
PS C:\WINDOWS> docker container list -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                 NAMES
PS C:\WINDOWS>
```

Docker Compose - format

24 May 2020 20:44

Create a docker compose file

Create a file docker-compose.yml in the project root



Yml uses indentation

```
version: '3'  
services:  
  api:  
  postgres:
```

First line	Version line
Next section: Services:	Services that we want to create Once we get into docker container/ docker swarm - we refer to containers are services Here we have <ul style="list-style-type: none">• api service• Postgres service
Next section: Network	Will create a default network for you if we do not mention Based on the <name of the app> and the word <default> <ul style="list-style-type: none">• In this case it would be called <conferenceapp_default>• Docker compose will add the api service and the postgres service to that network so that they can communicate

Docker Compose - API service

24 May 2020 20:44

```
version: '3'
services:
  api:
  postgres:
```

Lets name the 2 services api and postgres

Configure a Service:

STEP1: Image	We need to tell docker compose what to name our image when it creates it That is done with the image key <pre>version: '3' services: api: image: conference/api postgres:</pre>
STEP2 : Container name	What to name the container once it builds a container from that image We use the container_name key for that Container_image: <conference_api>

```

1 version: '3'
2
3 services:
4   api:
5     image: conference/api
6     container_name: conference_api
7
8   postgres:

```

STEP3: Build context

How to find the docker file to build the image in the first place

We use the **build context**

build:

Context: ..

```

1 version: '3'
2
3 services:
4   api:
5     image: conference/api
6     container_name: conference_api
7     build:
8       context: .
9
10  postgres:

```

STEP4: Ports

Map some ports

We know that the application would be running on the port 80

We want to forward **port 5000 in the host machine to port 80 in the container**

To do that we use the **ports** key

The ports key is an array which is formatted with an indentation and -

```

1 version: '3'
2
3 services:
4   api:
5     image: conference/api
6     container_name: conference_api
7     build:
8       context: .
9     ports:
10    - 5000:80
11
12  postgres:

```

STEP5: Environment variables

In this case we want to set the environment to production

We use the key **environment**

```

1 version: '3'
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1190
1191
1192
1193
1194
1195
1195
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
```

```
docker-compose.yml* ✘ X
1 version: '3'
2
3 services:
4   api:
5     image: conference/api
6     container_name: conference_api
7     build:
8       context: .
9     ports:
10    - 5000:80
11   environment:
12     ASPNETCORE_ENVIRONMENT: Production
13
```

STEP6: Startup dependencies

Here we want to start postgres service before starting this container

We us the **depends_on** key

This is also an array

```
docker-compose.yml* ✘ X
1 version: '3'
2
3 services:
4   api:
5     image: conference/api
6     container_name: conference_api
7     build:
8       context: .
9     ports:
10    - 5000:80
11   environment:
12     ASPNETCORE_ENVIRONMENT: Production
13   depends_on:
14     - postgres
```

Docker Compose - Postgres service

24 May 2020 20:45

Postgres service configuration would look similar to the api service..

But in this case we would use an image from the dockerhub

```
postgres:  
  image: postgres:9.6.3
```

Container name

We would name our container conference_db

```
postgres:  
  image: postgres:9.6.3  
  container_name: conference_db
```

Environment variables

Postgres uses these environment variables to setup the database

We would use the `environment` key

POSTGRES_DB: conference	This would create a database for us called conference
POSTGRES_USER:conf_app	This is define the username
POSTGRES_PASSWORD:docker	This is the password

Ports

Next we open up our port.

This not necessary but if we want some tool like PGAdmin to administrate rather than using a commandline,

We need to open up this port to the localhost

5432 is the default port postgres runs on, we are mapping it to 5432 on the host machine

```
postgres:  
  image: postgres:9.6.3  
  container_name: conference_db  
  environment:  
    POSTGRES_DB: conference  
    POSTGRES_USER: conf_app  
    POSTGRES_PASSWORD: docker  
  ports:  
    - 5432:5432
```

Volume

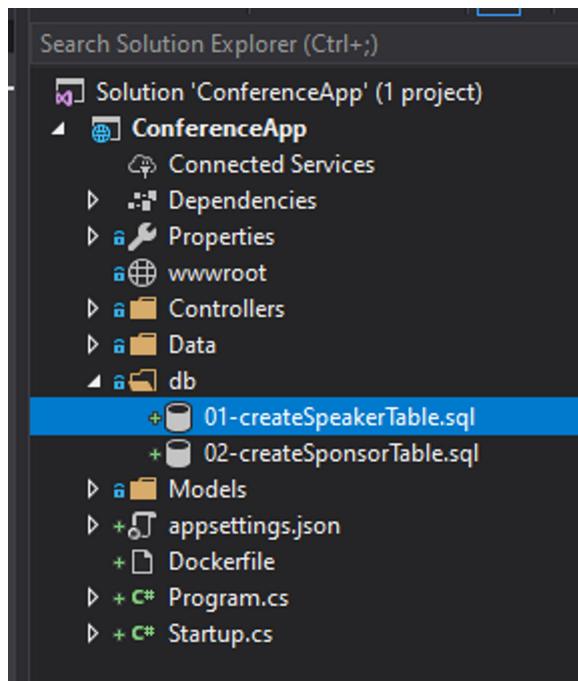
Volumes are pretty in docker compose

They are way to map the local folder to the container's folder(that is the folder in the container would actually be pointing to a folder in the host machine).

Here we want to map ./db folder in the host machine to a special folder in the postgres called docker-entrypoint-initdb.d

```
./db --> /docker-entrypoint-initdb.d
```

This sql folder in the postgres image looks for sql scripts that it can run.
In our case they would be createspeakerTable.sql and createSponserTable.sql



```
base.yml* ✘ X
build:
  context: .
ports:
  - 5000:80
environment:
  ASPNETCORE_ENVIRONMENT: Production
depends_on:
  - postgres
postgres:
  image: postgres:9.6.3
  container_name: conference_db
  environment:
    POSTGRES_DB: conference
    POSTGRES_USER: conf_app
    POSTGRES_PASSWORD: docker
  ports:
    - 5432:5432
  volumes:
    - ./db:/docker-entrypoint-initdb.d
```

Docker Compose - Running the stack

24 May 2020 20:45

To run our new docker compose stack..

We run

`docker-compose up`

Note: by default it would be looking for the docker-compose file in this directory.

If we want to specify the file name, we can use the -f switch

Example:

`ConferenceApp\ConferenceApp> docker-compose -f docker-compose.dev.yml`

We do not need to provide the switch.. As we have the docker-compose file in the same directory

```
Administrator: Windows PowerShell
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp> docker-compose up
Creating network "conferenceapp_default" with the default driver
Building api
Step 1/9 : FROM microsoft/aspnetcore-build:2.0 AS build
--> 6105426f13e9
Step 2/9 : WORKDIR /build
Removing intermediate container 31587ba21618
--> febef600250a
Step 3/9 : COPY . .
-
```

```
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp> docker-compose up
Creating network "conferenceapp_default" with the default driver
Building api
Step 1/9 : FROM microsoft/aspnetcore-build:2.0 AS build
--> 6105426f13e9
Step 2/9 : WORKDIR /build
Removing intermediate container 31587ba21618
--> febef600250a
Step 3/9 : COPY . .
--> da33fd14e6e1
Step 4/9 : RUN dotnet restore
--> Running in 26703c80fcf7
  Restoring packages for /build/ConferenceApp.csproj...
  Restoring packages for /build/ConferenceApp.csproj...
  Restore completed in 57.65 ms for /build/ConferenceApp.csproj.
  Restoring packages for /build/ConferenceApp.csproj...
  Restore completed in 199.22 ms for /build/ConferenceApp.csproj.
  Installing Npgsql 3.2.6.
  Installing Npgsql.EntityFrameworkCore.PostgreSQL 2.0.1.
  Generating MSBuild file /build/obj/ConferenceApp.csproj.nuget.g.props.
  Generating MSBuild file /build/obj/ConferenceApp.csproj.nuget.g.targets.
  Restore completed in 2.01 sec for /build/ConferenceApp.csproj.
Removing intermediate container 26703c80fcf7
--> 435d36b4dcc9
Step 5/9 : RUN dotnet publish -c Release -o output
--> Running in 9f619ca7df19
Microsoft (R) Build Engine version 15.5.180.51428 for .NET Core
Copyright (C) Microsoft Corporation. All rights reserved.

  Restore completed in 27.76 ms for /build/ConferenceApp.csproj.
  Restore completed in 26.63 ms for /build/ConferenceApp.csproj.
  Restore completed in 33.38 ms for /build/ConferenceApp.csproj.
```

```

Administrator: Windows PowerShell
0_0.<Configure>b__0(IApplicationBuilder builder)
conference_api |     at Microsoft.AspNetCore.Hosting.InternalWebHost.BuildApplication()
conference_api |     at Microsoft.AspNetCore.Hosting.WebHostBuilder.Build()
conference_api |     at ConferenceApp.Program.Main(String[] args) in /build/Program.cs:line 17
conference_db |     done
conference_db |     server started
conference_db |     exited with code 139
conference_db |     CREATE DATABASE
conference_db |
conference_db |     CREATE ROLE
conference_db |
conference_db |     /usr/local/bin/docker-entrypoint.sh: running /docker-entrypoint-initdb.d/01-createSpeakerTable
sql |
conference_db |     CREATE TABLE
conference_db |
conference_db |     /usr/local/bin/docker-entrypoint.sh: running /docker-entrypoint-initdb.d/02-createSponsorTable
sql |
conference_db |     CREATE TABLE
conference_db |
conference_db |     LOG: received fast shutdown request
conference_db |     waiting for server to shut down...LOG: aborting any active transactions
.conference_db |     .LOG: autovacuum launcher shutting down
conference_db |     LOG: shutting down
conference_db |     LOG: database system is shut down
done |
conference_db |     server stopped
PostgreSQL init process complete; ready for start up.

LOG: database system was shut down at 2018-02-07 21:40:00 UTC
LOG: MultiXact member wraparound protections are now enabled
LOG: database system is ready to accept connections
LOG: autovacuum launcher started

```

Docker compose is complete..

You might notice some exceptions fly by during this process

What we have was a raise condition where the **database wasn't finished starting up before the api tried to seek it**

```

Administrator: Windows PowerShell
0_0.<Configure>b__0(IApplicationBuilder builder)
conference_api |     at Microsoft.AspNetCore.Hosting.InternalWebHost.BuildApplication()
conference_api |     at Microsoft.AspNetCore.Hosting.WebHostBuilder.Build()
conference_api |     at ConferenceApp.Program.Main(String[] args) in /build/Program.cs:line 17
conference_db |     done
conference_db |     server started
conference_db |     exited with code 139 ←
conference_db |     CREATE DATABASE
conference_db |
conference_db |     CREATE ROLE
conference_db |
conference_db |     /usr/local/bin/docker-entrypoint.sh
sql |
conference_db |     CREATE TABLE
conference_db |
conference_db |     /usr/local/bin/docker-entrypoint.sh
sql |

```

Thats why we see on top that the conference_api exited with the code 139

Now we exit the console since it is in the interactive mode

- o Ctrl +C

```

conference_db |     LOG: MultiXact member wraparound protections are now enabled
conference_db |     LOG: database system is ready to accept connections
conference_db |     LOG: autovacuum launcher started
Gracefully stopping... (press Ctrl+C again to force)
Stopping conference_db ... done
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp>

```

It should gracefully stop any containers from running.

If it doesn't stop then use **docker stop <container name>**

```
\ConferenceApp> docker stop conference_db.
```

In our case the container stopped gracefully.. So docker stop command isn't required

```

Administrator: Windows PowerShell
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp> docker container list -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
PORTS              NAMES
0537a9889c1a        conference/api      "dotnet ConferenceAp..."   2 minutes ago       Exited (139) 2 minutes ago
d7faadec6747        postgres:9.6.3     "docker-entrypoint.s..."   2 minutes ago       Exited (0) 41 seconds ago
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp>

```

We saw this 139 code earlier.. Reason: it was unable to connect to the database.

We can re-check by looking into the container logs

```
docker logs <container name>
```

```
\ConferenceApp> docker logs conference_api
```

```
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp> docker logs conference_api
warn: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[35]
      No XML encryptor configured. Key {51482939-9c7e-470d-a73a-d7bca445cfe5} may be persisted to storage in un
      encrypted form.
fail: Microsoft.EntityFrameworkCore.Database.Connection[20004]
      An error occurred using the connection to database 'conference' on server 'tcp://postgres:5432'.
System.Net.Sockets.SocketException (0x80004005): Connection refused
  at Npgsql.NpgsqlConnector.Connect(NpgsqlTimeout timeout)
  at Npgsql.NpgsqlConnector.<RawOpen>d__146.MoveNext()
--- End of stack trace from previous location where exception was thrown ---
  at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()
```

So now since we are sure that the database exists (since we created the database).

The reason would be that the api tried seeking the db even before it was ready.

Lets run `docker-compose up` again

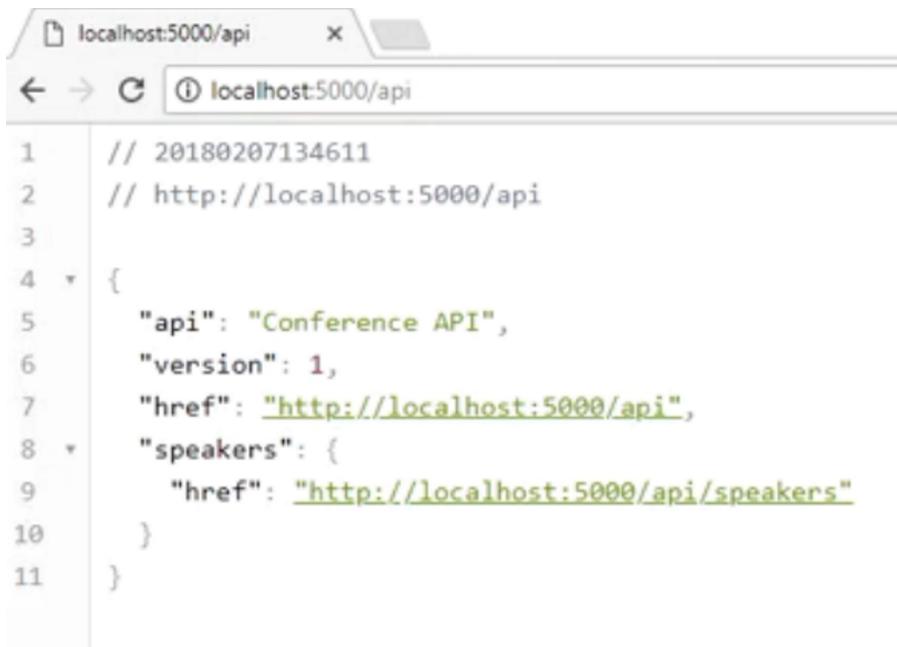
Before it had to build the images and then the container to start..

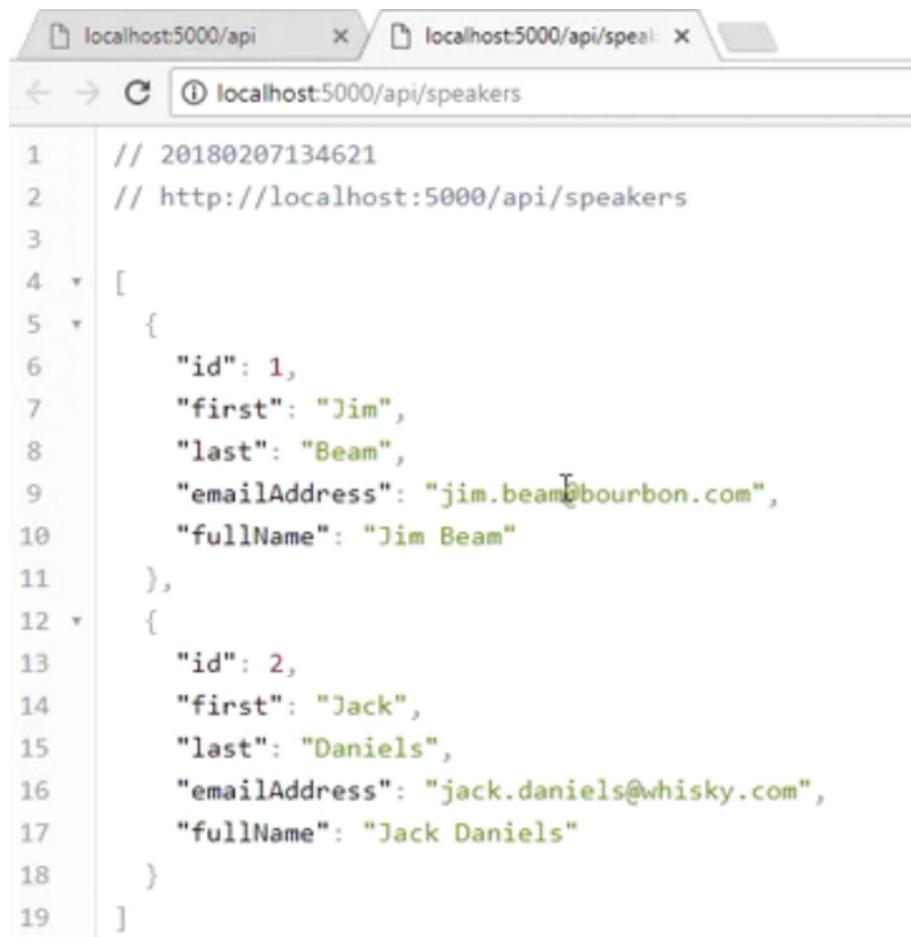
Now, all it has to do is start the containers..

It should be really quick

```
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp> docker-compose up
Starting conference_db ... done
Starting conference_api ... done
Attaching to conference_db, conference_api
conference_db | LOG:  database system was shut down at 2018-02-07 21:41:49 UTC
conference_db | LOG:  MultiXact member wraparound protections are now enabled
conference_db | LOG:  database system is ready to accept connections
conference_db | LOG:  autovacuum launcher started
conference_api | Hosting environment: Production
conference_api | Content root path: /app
conference_api | Now listening on: http://[::]:80
conference_api | Application started. Press Ctrl+C to shut down.
```

Remember, we mapped port 5000 of the host machine to the port 80 in the container..





```
1 // 20180207134621
2 // http://localhost:5000/api/speakers
3
4 [
5   {
6     "id": 1,
7     "first": "Jim",
8     "last": "Beam",
9     "emailAddress": "jim.beam@bourbon.com",
10    "fullName": "Jim Beam"
11  },
12  {
13    "id": 2,
14    "first": "Jack",
15    "last": "Daniels",
16    "emailAddress": "jack.daniels@whisky.com",
17    "fullName": "Jack Daniels"
18  }
19 ]
```

So entire stack is spun up in one command

Docker push - Push images to hub

24 May 2020 21:46

Create a docker hub account

Lets list out the images

```
Administrator: Windows PowerShell
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp> docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
conference/api     latest   43564c936dee  20 hours ago  307MB
microsoft/aspnetcore-build  2.0    6105426f13e9  4 weeks ago   1.97GB
microsoft/aspnetcore     2.0    bb8bdc966bb5  4 weeks ago   302MB
docker4w/nsenter-dockerd  latest  cae870735e91  3 months ago  187kB
postgres           9.6.3   33b13ed6b80a  6 months ago  269MB
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp>
```

The problem with the image name is - unless your image name(conference/api) is unless your docker username is conference it won't work/upload

The portion before the slash before the repository name is going to be the dockerhub username

Expected Format:

<dockerhub username>/<repository name>

Solution:

1. Build an image from the docker file and tag it with another name
`docker build -t leebrandt/super:duper .`
2. Even simpler than that, we can tag our existing image with the new name
`docker tag 43564c936dee leebrandt/conference_a`

```
Administrator: Windows PowerShell
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp> docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
conference/api     latest   43564c936dee  20 hours ago  307MB
microsoft/aspnetcore-build  2.0    6105426f13e9  4 weeks ago   1.97GB
microsoft/aspnetcore     2.0    bb8bdc966bb5  4 weeks ago   302MB
docker4w/nsenter-dockerd  latest  cae870735e91  3 months ago  187kB
postgres           9.6.3   33b13ed6b80a  6 months ago  269MB
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp> docker tag 43564c936dee leebrandt/conference_a
pi:first_push_


```

Now when we list

```
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp> docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
conference/api     latest   43564c936dee  20 hours ago  307MB
leebrandt/conference_api  first_push  43564c936dee  20 hours ago  307MB
microsoft/aspnetcore-build  2.0    6105426f13e9  4 weeks ago   1.97GB
microsoft/aspnetcore     2.0    bb8bdc966bb5  4 weeks ago   302MB
docker4w/nsenter-dockerd  latest  cae870735e91  3 months ago  187kB
postgres           9.6.3   33b13ed6b80a  6 months ago  269MB
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp>
```

The latest image is just a pointer to that image

Now Login to your docker account with `docker login`

```
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp> docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username (leebrandt):
```

Enter username and password

```
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp> docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username (leebrandt): leebrandt
Password:
Login Succeeded
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp>
```

Login succeeded, we are ready to go

Now push the image using `docker push <image name>:<tag name>`

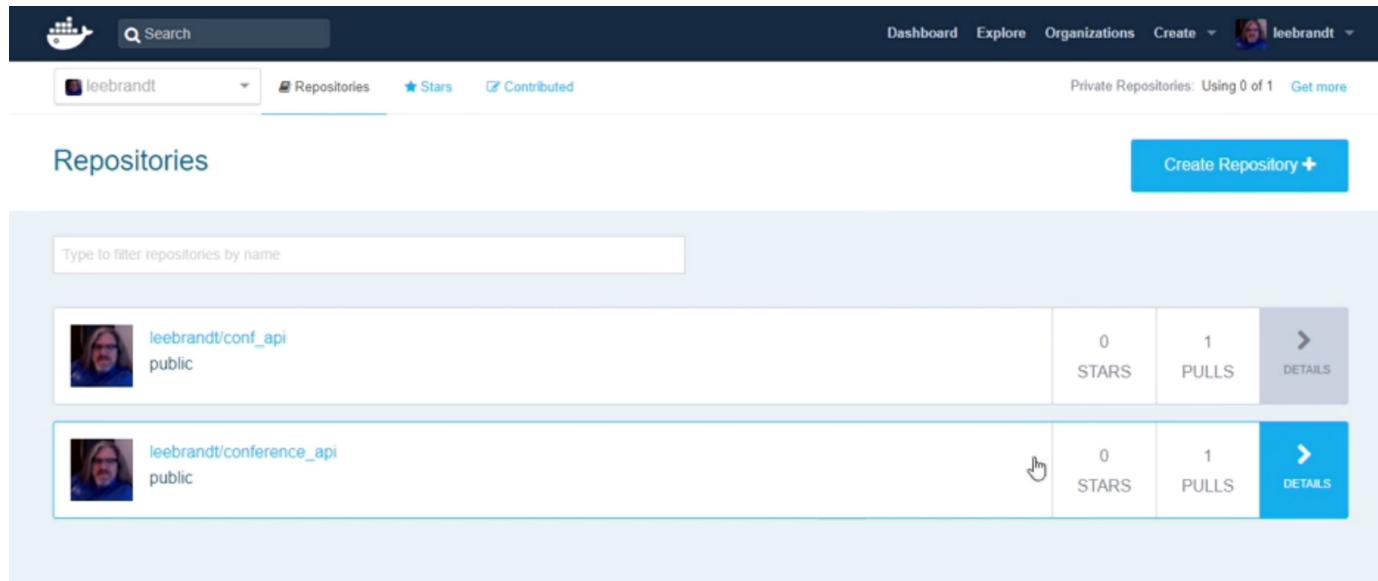
Note: here the image name is username/imagename

```
docker push leebrandt/conference_api:first_push
```

```
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp> docker push leebrandt/conference_api:first_push
The push refers to repository [docker.io/leebrandt/conference_api]
616e992fc970: Pushing [=====] 4.98MB
ade3d4c8ad14: Pushing 1.536kB
aa3d55a4a032: Pushing [=>] 3.23MB/83.46MB
f9d2d3e199b9: Preparing
9eec8925915b: Pushing [=====] 615.9kB
6c69f7ce3401: Waiting
e27a10675c56: Waiting
```

```
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp> docker push leebrandt/conference_api:first_push
The push refers to repository [docker.io/leebrandt/conference_api]
616e992fc970: Pushed
ade3d4c8ad14: Pushed
aa3d55a4a032: Pushed
f9d2d3e199b9: Pushed
9eec8925915b: Pushed
6c69f7ce3401: Pushed
e27a10675c56: Pushed
first_push: digest: sha256:d428f6c56bf601e7e47c96c81cce6dd0a198f7ce469ace9698ea23631df7fce4 size: 1792
PS C:\Users\Lee Brandt\source\repos\ConferenceApp\ConferenceApp>
```

Lets check the docker hub



The screenshot shows the Docker Hub interface. At the top, there is a search bar and a navigation bar with links for Dashboard, Explore, Organizations, Create, and a user profile for 'leebrandt'. Below the navigation, there are filters for 'Repositories', 'Stars', and 'Contributed'. A message indicates 'Private Repositories: Using 0 of 1' and a link to 'Get more'. On the right, there is a blue button labeled 'Create Repository +'. The main area is titled 'Repositories' and contains two entries:

Repository	Stars	Pulls	Details
leebrandt/conf_api	0 STARS	1 PULLS	DETAILS
leebrandt/conference_api	0 STARS	1 PULLS	DETAILS

Conference_api is listed in the new list of repositories
If we click on details



PUBLIC REPOSITORY

leebrandt/conference_api ☆

Last pushed: a minute ago

[Repo Info](#) [Tags](#) [Collaborators](#) [Webhooks](#) [Settings](#)

Short Description

Short description is empty for this repo.

Docker Pull Command



docker pull leebrandt/conference_api

Full Description

Full description is empty for this repo.

Owner

 leebrandt

What next : Container Orchestration

24 May 2020 22:21



Orchestrators:

Tools that allow to cluster and scale the containers

Windows	Docker Swarm
Google	Kubernetes
Apache	Mesosphere



Docker orchestrators

Docker Series

<https://developer.okta.com/blog>

What next : Exporting Container and Saving Image

26 May 2020 08:25

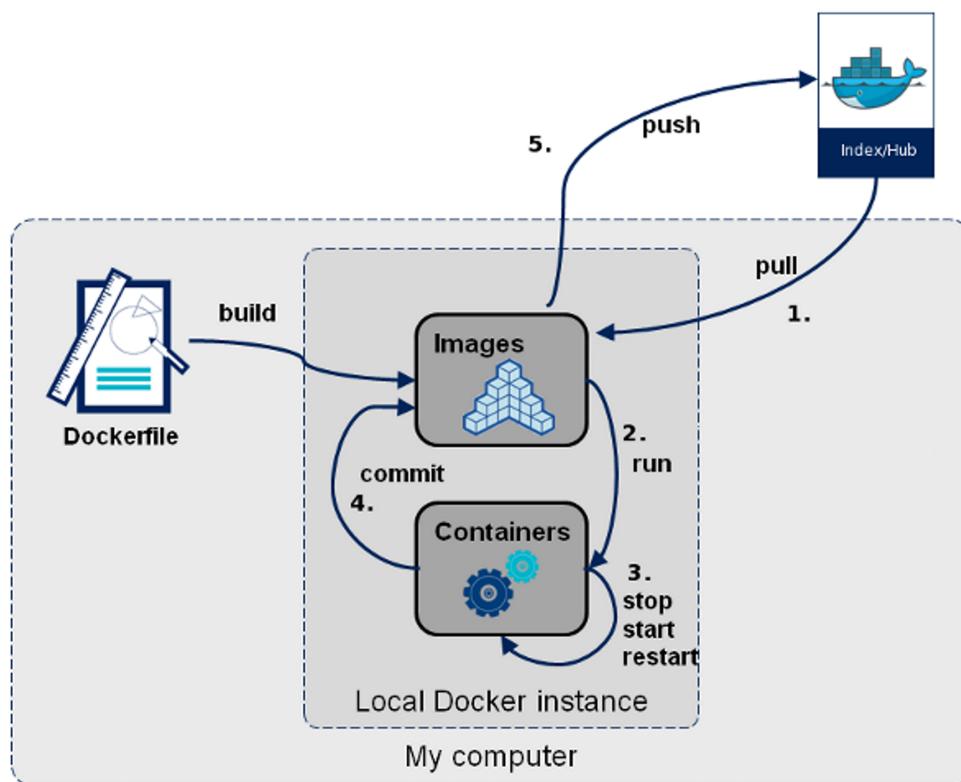
Exporting Container and Saving Image

It is important to save the change inside the container. To do this, we need to know the difference of export and save

Export: Export a container

Save: Save an image

So, let's see the difference. ([Sik-Ho Tsang @ Medium](#))



Docker flow

If we list the image, we can see that we have the `ubuntu:18.04` image already.
sudo docker images

File Edit View Search Terminal Help

cnn@cnn:~

cnn@cnn:~\$ sudo docker images

[sudo] password for cnn:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	18.04	113a43faa138	4 weeks ago	81.2MB
hello-world	latest	e38bc07ac18e	2 months ago	1.85kB

cnn@cnn:~\$ █

Let me run and go into the container.

sudo docker run -itd --name ubuntu ubuntu:1804 /bin/bash

sudo docker exec -it ubuntu bash

2. Let us mkdir to create a new directory *aaa*. We have some changes in the container. Let's exit.

mkdir aaa

ls -al

exit

root@5cc2d9614281: /

File Edit View Search Terminal Help

root@5cc2d9614281:/# mkdir aaa

root@5cc2d9614281:/# ls -al

```
total 76
drwxr-xr-x  1 root root 4096 Jul  6 08:03 .
drwxr-xr-x  1 root root 4096 Jul  6 08:03 ..
-rw-r--r--  1 root root    0 Jul  6 08:02 .dockerenv
drwxr-xr-x  2 root root 4096 Jul  6 08:03 aaa
drwxr-xr-x  2 root root 4096 May 26 00:45 bin
drwxr-xr-x  2 root root 4096 Apr 24 08:34 boot
drwxr-xr-x  5 root root 360 Jul  6 08:02 dev
drwxr-xr-x  1 root root 4096 Jul  6 08:02 etc
drwxr-xr-x  2 root root 4096 Apr 24 08:34 home
drwxr-xr-x  8 root root 4096 May 26 00:44 lib
drwxr-xr-x  2 root root 4096 May 26 00:44 lib64
drwxr-xr-x  2 root root 4096 May 26 00:44 media
drwxr-xr-x  2 root root 4096 May 26 00:44 mnt
drwxr-xr-x  2 root root 4096 May 26 00:44 opt
dr-xr-xr-x 345 root root    0 Jul  6 08:02 proc
drwx----- 2 root root 4096 May 26 00:45 root
drwxr-xr-x  1 root root 4096 Jun  5 21:20 run
drwxr-xr-x  1 root root 4096 Jun  5 21:20 sbin
drwxr-xr-x  2 root root 4096 May 26 00:44 srv
dr-xr-xr-x 13 root root    0 Jul  6 06:13 sys
drwxrwxrwt  2 root root 4096 May 26 00:45 tmp
```

```
cnn@cnn:~
```

File Edit View Search Terminal Help

```
-rwxr-xr-x 1 root root 0 Jul 6 08:02 .dockerenv
drwxr-xr-x 2 root root 4096 Jul 6 08:03 aaa
drwxr-xr-x 2 root root 4096 May 26 00:45 bin
drwxr-xr-x 2 root root 4096 Apr 24 08:34 boot
drwxr-xr-x 5 root root 360 Jul 6 08:02 dev
drwxr-xr-x 1 root root 4096 Jul 6 08:02 etc
drwxr-xr-x 2 root root 4096 Apr 24 08:34 home
drwxr-xr-x 8 root root 4096 May 26 00:44 lib
drwxr-xr-x 2 root root 4096 May 26 00:44 lib64
drwxr-xr-x 2 root root 4096 May 26 00:44 media
drwxr-xr-x 2 root root 4096 May 26 00:44 mnt
drwxr-xr-x 2 root root 4096 May 26 00:44 opt
dr-xr-xr-x 345 root root 0 Jul 6 08:02 proc
drwx----- 2 root root 4096 May 26 00:45 root
drwxr-xr-x 1 root root 4096 Jun 5 21:20 run
drwxr-xr-x 1 root root 4096 Jun 5 21:20 sbin
drwxr-xr-x 2 root root 4096 May 26 00:44 srv
dr-xr-xr-x 13 root root 0 Jul 6 06:13 sys
drwxrwxrwt 2 root root 4096 May 26 00:45 tmp
drwxr-xr-x 1 root root 4096 May 26 00:44 usr
drwxr-xr-x 1 root root 4096 May 26 00:45 var
root@5cc2d9614281:# exit
exit
cnn@cnn:~$
```

3. Then we can save the image and export the container.

```
sudo docker save ubuntu > ubuntu_save.tar
sudo docker export ubuntu > ubuntu_export.tar
```

```
cnn@cnn:~
```

File Edit View Search Terminal Help

```
cnn@cnn:~$ sudo docker save ubuntu > ubuntu_save.tar
cnn@cnn:~$ sudo docker export ubuntu > ubuntu_export.tar
cnn@cnn:~$
```

4. Let us remove the image and container before seeing the differences.

```
sudo docker stop ubuntu
sudo docker rm ubuntu
sudo docker rmi ubuntu:18.04
```

```
cnn@cnn:~$ sudo docker stop ubuntu
ubuntu
cnn@cnn:~$ sudo docker rm ubuntu
ubuntu
```

```
cnn@cnn:~$ sudo docker rmi ubuntu:18.04
Untagged: ubuntu:18.04
Untagged: ubuntu@sha256:5f4bdc3467537cbbe563e80db2c3ec95d548a9145d64453b06939c4592d67b6d
Deleted: sha256:113a43faa1382a7404681f1b9af2f0d70b182c569aab71db497e33fa59ed87e6
Deleted: sha256:a9fa410a3f1704cd9061a802b6ca6e50a0df183cb10644a3ec4cac9f6421677a
Deleted: sha256:b21f75f60422609fa79f241bf80044e6e133dd0662851afb12dacd22d199233a
Deleted: sha256:038d2d2aa4fb988c06f04e3af208cc0c1dbd9703aa04905ade206d783e7bc06a
Deleted: sha256:b904d425ea85240d6af5a6c6f145e05d5e0127f547f8eb4f68552962df846e81
Deleted: sha256:db9476e6d963ed2b6042abef1c354223148cdcbd6c7416c71a019ebcae0edb
cnn@cnn:~$
```

5. Load the image ubuntu_save.tar that we've just saved first.

```
sudo docker load < ubuntu_save.tar
```

```
cnn@cnn:~$ sudo docker load < ubuntu_save.tar
db9476e6d963: Loading layer 83.62MB/83.62MB
3a89e0d8654e: Loading layer 15.87kB/15.87kB
904d60939c36: Loading layer 10.24kB/10.24kB
a20a262b87bd: Loading layer 5.632kB/5.632kB
b6f13d447e00: Loading layer 3.072kB/3.072kB
Loaded image: ubuntu:18.04
```

6. If we go into the container, the directory *aaa* is not here.

```
cnn@cnn: ~
File Edit View Search Terminal Help
cnn@cnn:~$ sudo docker run -itd --name ubuntu ubuntu:18.04 /bin/bash
e4fb3aa51ec7b20949a74cc256dd72cff535e2519dd4719e38413185049235a
cnn@cnn:~$ sudo docker exec -it ubuntu bash
root@e4fb3aa51ec:/# ls -al
total 72
drwxr-xr-x  1 root root 4096 Jul  6 08:11 .
drwxr-xr-x  1 root root 4096 Jul  6 08:11 ..
-rw-rxr-xr-x  1 root root    0 Jul  6 08:11 .dockerenv
drwxr-xr-x  2 root root 4096 May 26 00:45 bin
drwxr-xr-x  2 root root 4096 Apr 24 08:34 boot
drwxr-xr-x  5 root root 360 Jul  6 08:11 dev
drwxr-xr-x  1 root root 4096 Jul  6 08:11 etc
```

7. Exit first.

```
root@e4fb3aa51ec:/# exit
exit
cnn@cnn:~$
```

8. This time, we import the container *ubuntu_export.tar* that we've just export as well.

```
cat ubuntu_export.tar | sudo docker import - ubuntu:18.04
sudo docker run -itd --name ubuntu ubuntu:18.04 /bin/bash
sudo docker exec -it ubuntu bash
```

```
root@46655826dbae: /
File Edit View Search Terminal Help
cnn@cnn:~$ cat ubuntu_export.tar | sudo docker import - ubuntu:18.04
sha256:9392f5ce467af25d98190604fee30c2e3752722193b33d4499f11b2789abdfa8
cnn@cnn:~$ sudo docker run -itd --name ubuntu ubuntu:18.04 /bin/bash
46655826dbae390452d55055daca8a602aea7aa33da3a074738608e0bf466ba8
cnn@cnn:~$ sudo docker exec -it ubuntu bash
root@46655826dbae:/# ls -al
total 80
drwxr-xr-x  1 root root 4096 Jul  6 08:15 .
drwxr-xr-x  1 root root 4096 Jul  6 08:15 ..
-rw-rxr-xr-x  1 root root    0 Jul  6 08:15 .dockerenv
drwxr-xr-x  2 root root 4096 Jul  6 08:07 aaa
drwxr-xr-x  2 root root 4096 May 26 00:45 bin
drwxr-xr-x  2 root root 4096 Apr 24 08:34 boot
drwxr-xr-x  5 root root 360 Jul  6 08:15 dev
drwxr-xr-x  1 root root 4096 Jul  6 08:15 etc
```

We can see, the directory *aaa* has been back!!!

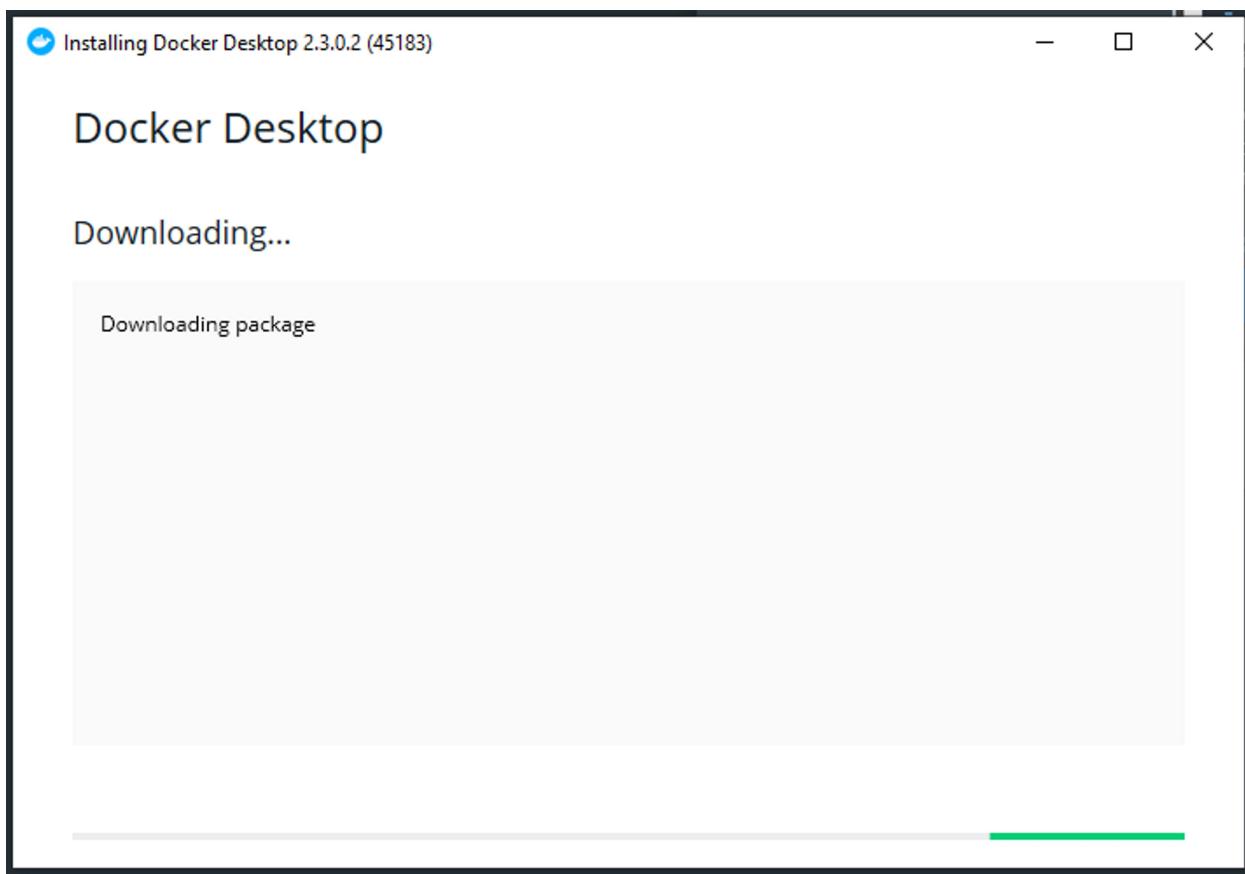
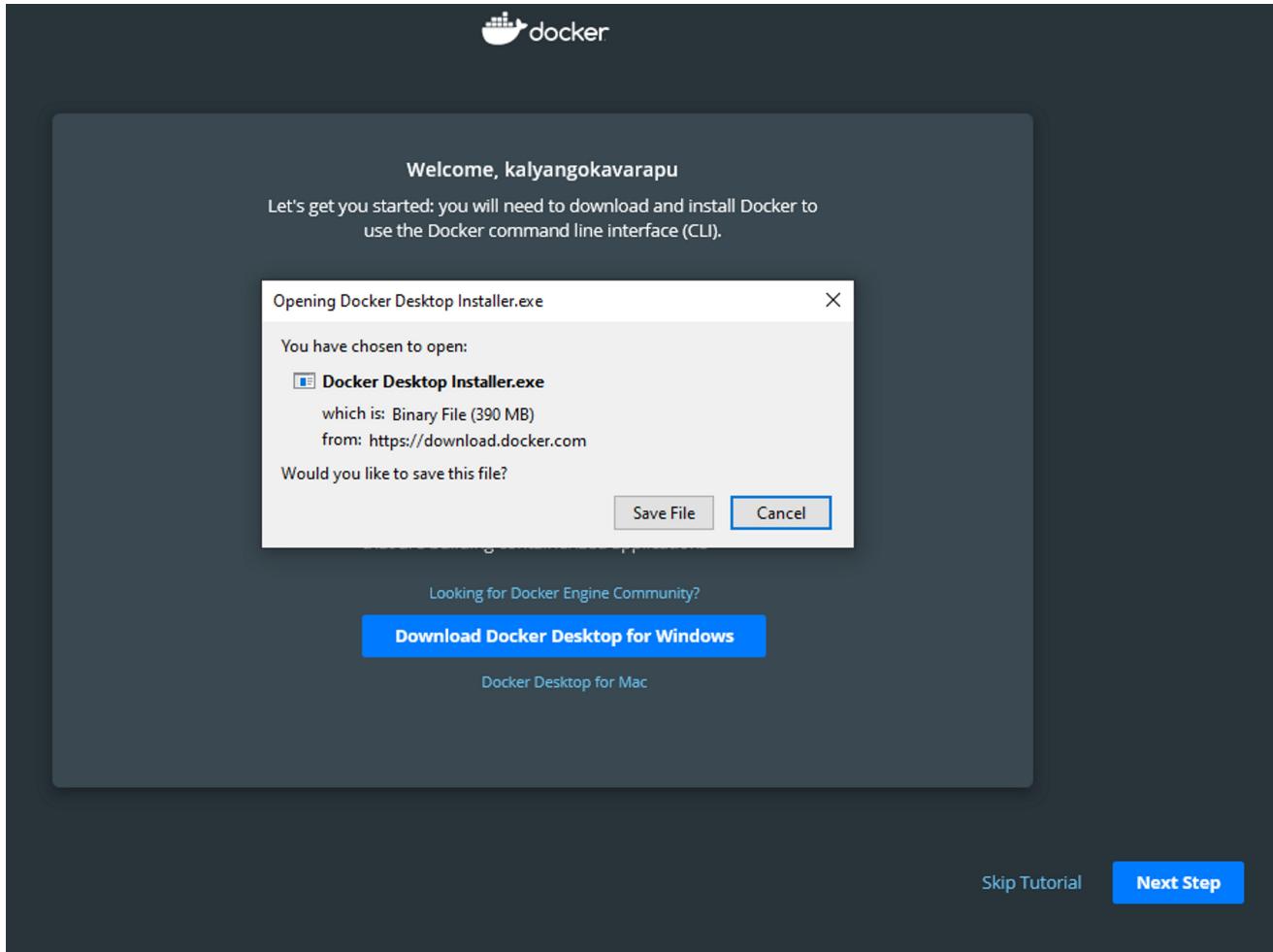
From <<https://medium.com/@sh.tsang/docker-tutorial-4-exporting-container-and-saving-image-c3a7d792cfb6>>

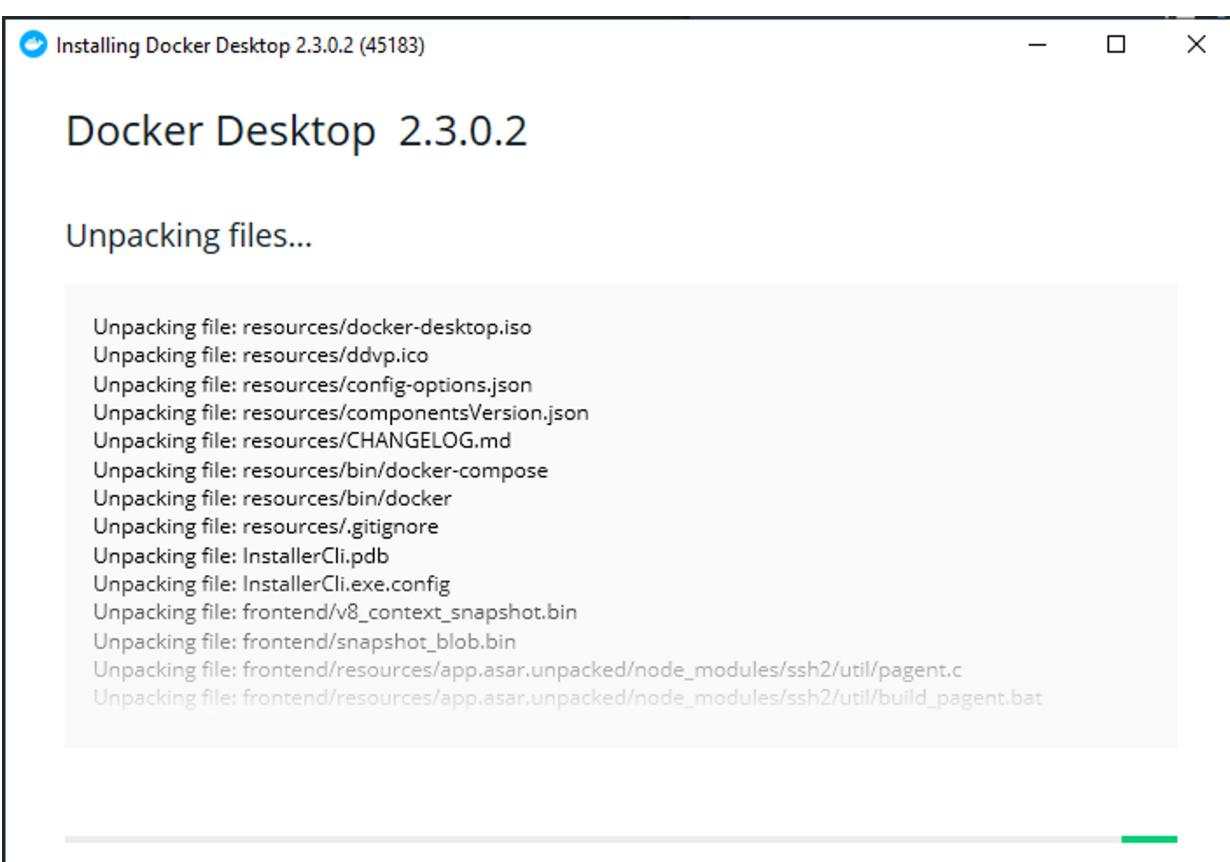
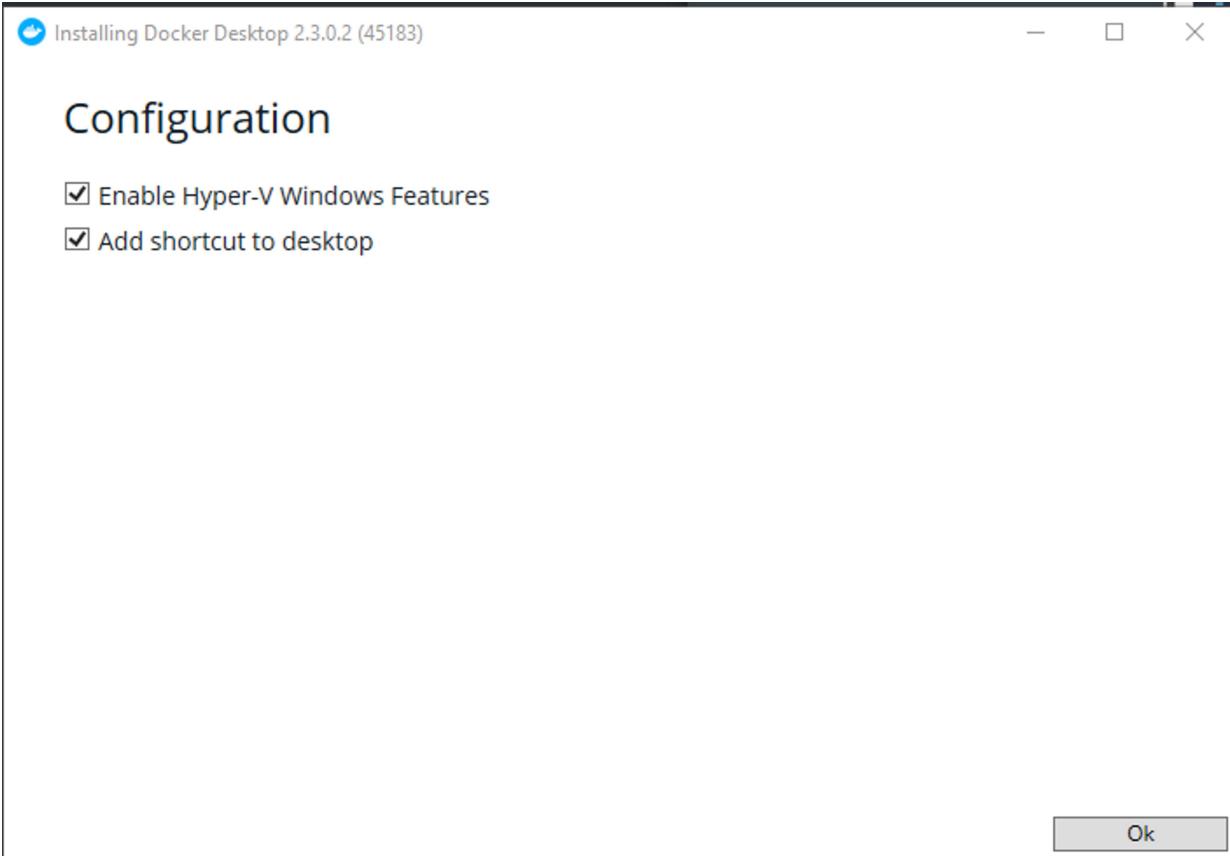
Help: Download Docker Desktop

18 May 2020 13:22

The screenshot shows the Docker Hub homepage at https://hub.docker.com. At the top, there's a banner for DockerCon LIVE on May 28th with a 'Register for free!' button. Below the banner, the Docker Hub logo is on the left, followed by a search bar with placeholder text 'Search for great content (e.g., mysql)'. To the right of the search bar are links for 'Explore', 'Repositories', 'Organizations', 'Get Help', and a user profile for 'kalyangokavarapu'. A blue header bar contains the user's name 'kalyangokavarapu' and a dropdown arrow, along with a search bar for 'repository name...'. A 'Create Repository' button is also in this bar. The main content area shows a repository card for 'kalyangokavarapu / cheers2019', which was updated 8 months ago. It has 0 stars, 6 downloads, and is marked as 'PUBLIC'. Below the repository card is a tip: 'Tip: Not finding your repository? Try switching namespace via the top left dropdown.' To the right of the repository card is a sidebar for creating organizations, with a 'Create an Organization' button and a link to 'Manage Docker Hub repositories with your team'. At the bottom right of the page is a large blue button with white text: 'Download Docker Desktop' and an icon of a Docker ship.

The screenshot shows the Docker Hub onboarding tutorial at https://hub.docker.com/?overlay=onboarding. The title 'Welcome, kalyangokavarapu' is at the top. Below it, a message says: 'Let's get you started: you will need to download and install Docker to use the Docker command line interface (CLI).'. There's an image of a laptop displaying the Docker desktop interface. To the left, a 'Quick Start' sidebar lists five steps: 1. Download (which is highlighted with a blue circle), 2. Clone, 3. Build, 4. Run, and 5. Ship. Below the sidebar is a section for 'Docker Desktop' with the text: 'The preferred choice for millions of developers that are building containerized applications'. A 'Download Docker Desktop for Windows' button is available. At the bottom right, there are 'Skip Tutorial' and 'Next Step' buttons.







Installing Docker Desktop 2.3.0.2 (45183)



Docker Desktop 2.3.0.2

Unpacking files...

```
Unpacking file: frontend/Docker Desktop.exe  
Unpacking file: frontend/d3dcompiler_47.dll  
Unpacking file: DockerCli.exe  
Unpacking file: Docker.WPF.dll  
Unpacking file: Docker.HttpApi.dll  
Unpacking file: Docker.Core.dll  
Unpacking file: Docker.Backend.dll  
Unpacking file: Docker.ApiServices.dll  
Unpacking file: Docker Desktop.exe  
Unpacking file: Bugsnag.dll  
Unpacking file: resources/wsl/wsl-data.tar  
Unpacking file: resources/wsl/wsl-bootstrap.tar  
Unpacking file: resources/wsl/docker-wsl-cli.iso  
Unpacking file: resources/wsl/docker-for-wsl.iso
```



Installing Docker Desktop 2.3.0.2 (45183)



Docker Desktop 2.3.0.2

Installation succeeded

[Close](#)

If docker doesn't start

You might need to reduce the resources under settings.

Settings

General

Resources

ADVANCED

FILE SHARING

PROXIES

NETWORK

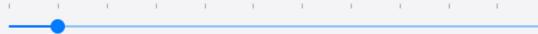
Docker Engine

Command Line

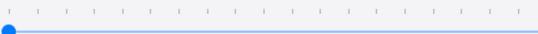
Kubernetes

Resources Advanced

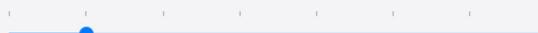
CPUs: 2



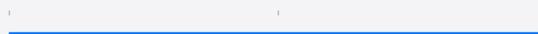
Memory: 1.00 GB



Swap: 1 GB



Disk image size: 64 GB (4 MB used)



Docker running

Cancel

Apply & Restart