Building and Tagging Container Images in \

April 7, 2017 Posted by: Loges Category: Blog, Containers

(https://www.assi
and-tagging-c
windc

Building and Tagging Container Im in Windows 2016

In this blog, we will show how to Building and Tagging Container Images in windows 2016 using docker comr PowerShell.

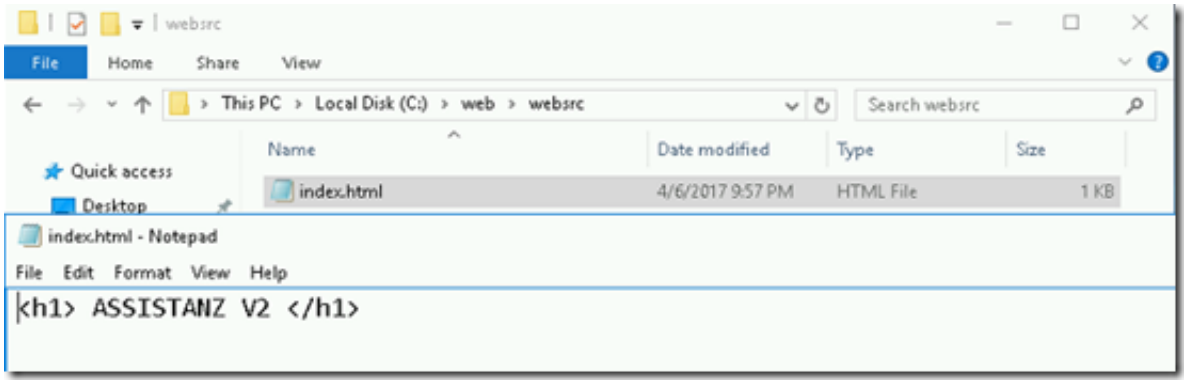
BUILDING NEW CONTAINER IMAGE USING CACHE

- You need to use **docker build** command to building and tagging the container images. If you run the help you w option.

```
PS C:\> docker build --help

Usage:  docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile

Options:
      --build-arg list             Set build-time variables (default [])
      --cache-from stringSlice     Images to consider as cache sources
      --cgroup-parent string       Optional parent cgroup for the container
      --compress                   Compress the build context using gzip
      --cpu-period int             Limit the CPU CFS (Completely Fair Scheduler) period
      --cpu-quota int              Limit the CPU CFS (Completely Fair Scheduler) quota
  -c, --cpu-shares int             CPU shares (relative weight)
      --cpuset-cpus string         CPUs in which to allow execution (0-3, 0,1)
      --cpuset-mems string         MEMs in which to allow execution (0-3, 0,1)
      --disable-content-trust      Skip image verification (default true)
  -f, --file string                Name of the Dockerfile (Default is 'PATH/Dockerfile')
      --force-rm                   Always remove intermediate containers
      --help                       Print usage
      --isolation string           Container isolation technology
      --label list                 Set metadata for an image (default [])
  -m, --memory string              Memory limit
      --memory-swap string         Swap limit equal to memory plus swap: '-1' to enable unlimited swap
      --network string             Set the networking mode for the RUN instructions during build (default
      --no-cache                   Do not use cache when building the image
      --pull                       Always attempt to pull a newer version of the image
  -q, --quiet                      Suppress the build output and print image ID on success
      --rm                         Remove intermediate containers after a successful build (default true)
      --security-opt stringSlice   Security options
      --shm-size string            Size of /dev/shm, default value is 64MB
  -t, --tag list                   Name and optionally a tag in the 'name:tag' format (default [])
      --ulimit ulimit              Ulimit options (default [])
PS C:\> _
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-65.png)

- Need to use the option –**t** to name and tag the container images.
- We have already built an image named **web** using dockerfile. Now we are building the second version of previou
  Read this blog to learn how to create customize images: https://www.assistanz.com/build-and-customize-contain
  windows-2016/ (https://www.assistanz.com/build-and-customize-container-image-in-windows-2016/)
- First, Go to **index.html** and edit the file content as **assistanz V2**.



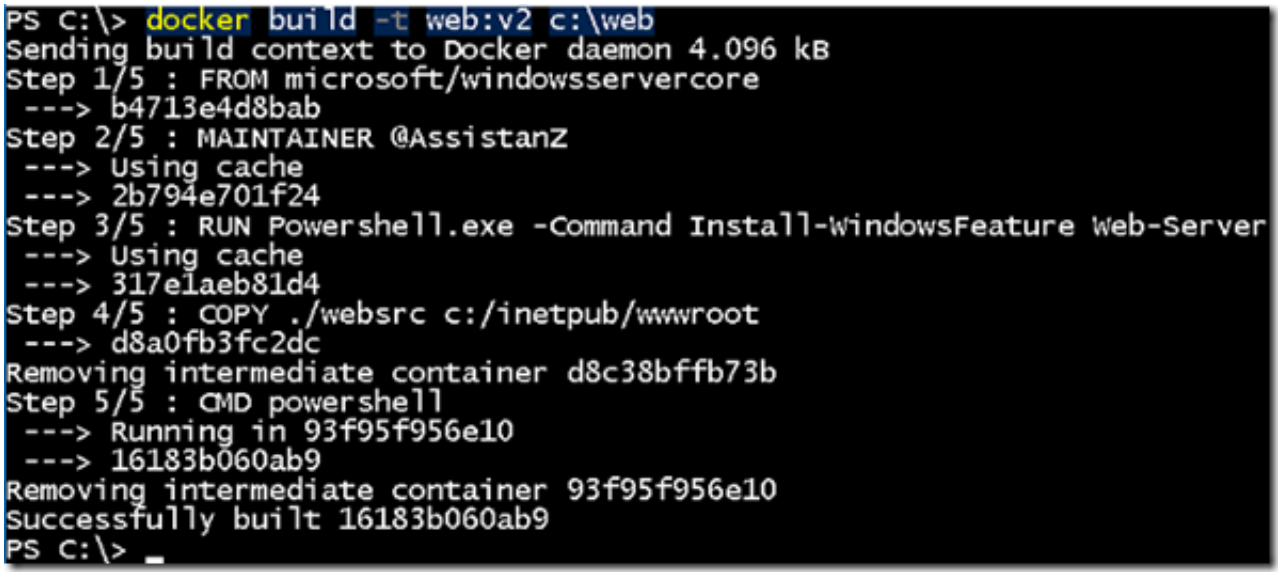
(https://www.assistanz

content/uploads/2017/04/image-66.png)

- Now build the image using the below command.

(https://www.assistanz.com/wp-content/uploads/2017/04/image-67.png)

- Build ran extremely fast and completed successfully.



(https://www.ass

content/uploads/2017/04/image-68.png)

- You can notice the output it shows **using cache** underneath **Step 2** & **Step 3.**

```
PS C:\> docker build -t web:v2 c:\web
Sending build context to Docker daemon 4.096 kB
Step 1/5 : FROM microsoft/windowsservercore
 ---> b4713e4d8bab
Step 2/5 : MAINTAINER @AssistanZ
 ---> Using cache
 ---> 2b794e701f24
Step 3/5 : RUN Powershell.exe -Command Install-WindowsFeature Web-Server
 ---> Using cache
 ---> 317e1aeb81d4
Step 4/5 : COPY ./websrc c:/inetpub/wwwroot
 ---> d8a0fb3fc2dc
Removing intermediate container d8c38bffb73b
Step 5/5 : CMD powershell
 ---> Running in 93f95f956e10
 ---> 16183b060ab9
Removing intermediate container 93f95f956e10
Successfully built 16183b060ab9
PS C:\> _
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-69.png)

- This caching functionally is built-in with docker build process for building container images. It's like re-using the s done with the previous build to make things more efficient.

- In Step-4, it actually copied our modified file to wwwroot folder inside the container image.

- We can disable the cache using **nocache** option.

```
PS C:\> docker build --help

Usage:  docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile

Options:
      --build-arg list           Set build-time variables (default [])
      --cache-from stringSlice   Images to consider as cache sources
      --cgroup-parent string     Optional parent cgroup for the container
      --compress                 Compress the build context using gzip
      --cpu-period int           Limit the CPU CFS (Completely Fair Scheduler) period
      --cpu-quota int            Limit the CPU CFS (Completely Fair Scheduler) quota
  -c, --cpu-shares int           CPU shares (relative weight)
      --cpuset-cpus string       CPUs in which to allow execution (0-3, 0,1)
      --cpuset-mems string       MEMs in which to allow execution (0-3, 0,1)
      --disable-content-trust    Skip image verification (default true)
  -f, --file string              Name of the Dockerfile (Default is 'PATH/Dockerfile')
      --force-rm                 Always remove intermediate containers
      --help                     Print usage
      --isolation string         Container isolation technology
      --label list               Set metadata for an image (default [])
  -m, --memory string            Memory limit
      --memory-swap string       Swap limit equal to memory plus swap: '-1' to enable unlimited swap
      --network string           Set the networking mode for the RUN instructions during build (default "defaul
      --no-cache                 Do not use cache when building the image
      --pull                     Always attempt to pull a newer version of the image
  -q, --quiet                    Suppress the build output and print image ID on success
      --rm                       Remove intermediate containers after a successful build (default true)
      --security-opt stringSlice Security options
      --shm-size string          Size of /dev/shm, default value is 64MB
  -t, --tag list                 Name and optionally a tag in the 'name:tag' format (default [])
      --ulimit ulimit            Ulimit options (default [])
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-70.png)

TAGGING THE CONTAINER IMAGE

- When we ran **docker images** command we can find two versions of container images named **web.**

```
PS C:\> docker images
REPOSITORY                      TAG                 IMAGE ID            CREATED             SIZE
web                             v2                  16183b060ab9        About an hour ago   10.4 GB
web                             latest              8a559d00d45d        23 hours ago        10.4 GB
my-nano-image                   latest              29ffc86d86c5        23 hours ago        1.04 GB
microsoft/windowsservercore     latest              b4713e4d8bab        4 weeks ago         10.1 GB
microsoft/nanoserver            10.0.14393.953      18a0d32a4b98        4 weeks ago         1 GB
microsoft/nanoserver            latest              18a0d32a4b98        4 weeks ago         1 GB
PS C:\>
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-71.png)

- **web V2** is the latest version, so we need to change the tag from **V2** to **latest.**
- First, we change the **web latest** to **web V1** using below command.

docker tag web:latest web:V1

```
PS C:\> docker tag web:latest web:v1
PS C:\> _
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-72.png)

- Now it will show two images with same image ID.

```
PS C:\> docker images
REPOSITORY                      TAG                 IMAGE ID            CREATED             SIZE
web                             v2                  16183b060ab9        About an hour ago   10.4
web                             v1                  8a559d00d45d        23 hours ago        10.4
web                             latest              8a559d00d45d        23 hours ago        10.4
my-nano-image                   latest              29ffc86d86c5        23 hours ago        1.04
microsoft/windowsservercore     latest              b4713e4d8bab        4 weeks ago         10.1
microsoft/nanoserver            10.0.14393.953      18a0d32a4b98        4 weeks ago         1 GB
microsoft/nanoserver            latest              18a0d32a4b98        4 weeks ago         1 GB
PS C:\> _
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-73.png)

- To remove the tag from the old image execute the below command.

docker rmi web:latest

```
PS C:\> docker rmi web:latest
Untagged: web:latest
PS C:\> _
```

(https://www.assistanz.com/wp-content/uploads/2017/04/

- Now **docker images** command will show only the two container images.

```
PS C:\> docker images
REPOSITORY                      TAG                 IMAGE ID            CREATED             SIZE
web                             v2                  16183b060ab9        About an hour ago   10.4 GB
web                             v1                  8a559d00d45d        23 hours ago        10.4 GB
my-nano-image                   latest              29ffc86d86c5        24 hours ago        1.04 GB
microsoft/windowsservercore     latest              b4713e4d8bab        4 weeks ago         10.1 GB
microsoft/nanoserver            10.0.14393.953      18a0d32a4b98        4 weeks ago         1 GB
microsoft/nanoserver            latest              18a0d32a4b98        4 weeks ago         1 GB
PS C:\> _
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-75.png)

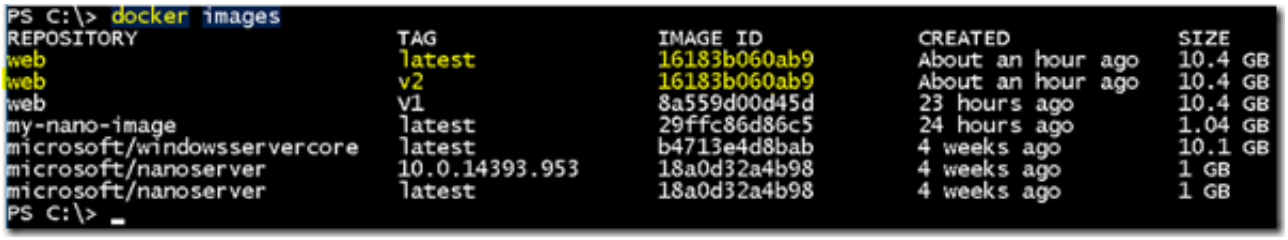
- Also, make sure that you always use the **latest** tag for most recent container images. We just need to repeat the did earlier.

- We need to change the tag of web image from v2 to latest using below command.

docker tag web:v2 web:latest

(https://www.assistanz.com/wp-content/uploads/2017/04/image-76.png)

- Now there will be two images under same image ID.



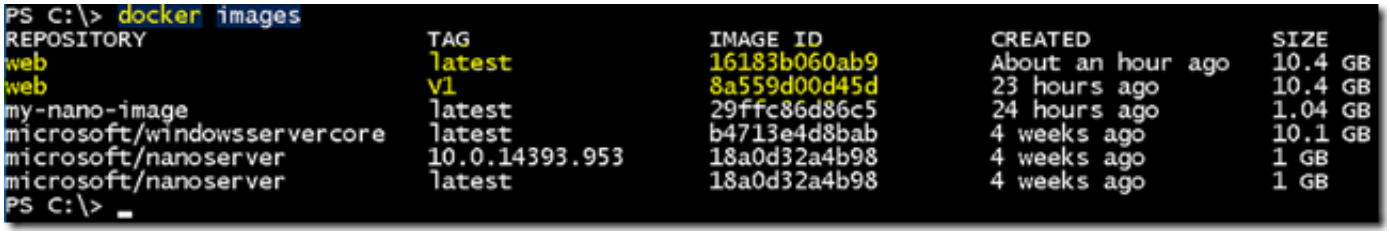
(https://www.assistanz.com/wp-content/uploads/2017/04/image-77.png)

- Now remove the tag from the web v2 version using below command.

docker rmi web:v2

(https://www.assistanz.com/wp-content/uploads/2017/04/image

- Now we have only two images with **V1** and **latest** tags.



(https://www.assistanz.com/wp-content/uploads/2017/04/image-79.png)

- These tags will be used while creating the containers. For example.

(https://www.assistanz.com/wp-content/uploads/2017/04/image-80.png)

VIDEO