Tech Tip: Docker

In this tech tip we will

* Pull an image from DockerHub
* List images we have in the local repository
* Create a container from that container
* Attach to the an containerd execute the shell
* Create a Dockerfile
* Build a new image
* Use the new image to create a container and execute a command
* Remove the image pulled and the image we created

Further Reading for more detailed information about writing a Dockerfile and building images.

<https://docs.docker.com/develop/develop-images/dockerfile_best-practices/>

Docker commands we will use

|  |  |
| --- | --- |
| **Command** | **Notes** |
| docker image ls | List the images in the local repository |
| *docker run -it --rm alpine /bin/sh* | Create a container from the image alpine:latest.  Attach to the conainer in interactive mode with a tty and execute /bin/sh.  After you exit the container docker will automatically remove the container. |
| *docker build -t first .* | Build a docker image from the Dockerfile in the current directory.  Add a tag first with a default version latest. |
| *docker run -it --rm first /bin/sh* | Create and use the container we built. |
| *docker run --rm first* | Create a container from the image we build.  Docker will execute the command from the CMD line in the docker file. Then exit and remove the container. |
| *docker image rm alpine first* | Remove the image we pulled and the image we created. |

Follow along and type the commands below to experience creating a new docker image and running the container with automatic execution of code in the container.

The description of the commands is **bold** and what you will type are in *italics*.

Prerequisites

* Install docker
* A text editor to create and modify the Dockerfile

**Create a directory for the image and change to it**

*mkdir first*

*cd first*

**Pull the image for Alpine Linux**

*docker pull alpine*

**Validate we got the image (you should see the image alpine in the resulting list)**

*docker image ls*

**Create the container and run it, tell docker to automatically remove the container**

*docker run -it --rm alpine /bin/sh*

**Run some commands inside the container running Alpine Linux**

*whoami*

*cat /etc/\*-release*

*ls*

*exit*

**Create our first Dockerfile.   The file should contain:**

FROM alpine

RUN echo 'Create a text file in first' >test.txt

**The *FROM command tells us the base image to use.  RUN tells up the commands to execute while building the image.***

**Build our new image with the tag first:latest**

*docker build -t first .*

**Have docker create a container from our new image attach it with tty and run the shell command.  The option -it tells run to use interactive mode open a tty to the container.  The optio –rm tells docker to remove the container when it has finished executing.  The final /bin/sh tells docker to execute the shell command in the container.**

*docker run -it --rm first /bin/sh*

**Inside the container run these commands( you should see that you are now root and that the file test.txt was created in the container)**

*whoami*

*ls*

*cat test.txt*

*exit*

**Modify the Dockerfile to have docker automatically run a command when the container is created.  The file should look like this**

FROM alpine

RUN echo 'Create a text file in first' >test.txt

CMD cat test.txt

**Build the image again with the new Dockerfile**

*docker build -t first .*

**Create the container and run it.  Tell docker to automatically remove the container when done**

*docker run --rm first*

**List the images (you should see the images alpine and first)**

*docker image ls*

**Remove the image we pulled and the image we just created**

*docker image rm alpine first*

**Validate they are gone**

*docker image ls*