QA Consulting

Docker
Bind Mounts

TRAINING MATERIALS - MODULE HANDOUT

Contacts

robert.crutchley@qa.com

team.qac.all.trainers@qa.com

www.consulting.qa.com

Contents

Overview

Bind Mounts are a very old feature in Docker and the most basic way of sharing a file or folder between a container and the host machine. Compared to Volumes in Docker, they do not have a great deal of functionality. This doesn't mean that they are worse by any means, sometimes you just want a file from the host machine mounted in the container, this is a nice and simple way of accomplishing that. Bind Mounts are very performant but you need to have the directory structure for the files and directories that you are sharing on the host in place.

There are two flags, or options that you can use with Docker for creating Bind Mounts, the **-v** and **--mount** flags. Which one you will use for creating a Bind Mount will dependent on a mixture of which one you prefer using and what functionality you need.

Creating with the --volume Flag

We have the **-v** or **--volume** way of creating a bind mount which takes three fields, separated by colons (**:**). All the fields provided must be in order. This example mounts a configuration file for NGINX, giving the container read only access.

```
docker run --volume [HOST_LOCATION]:[CONTAINER_LOCATION]:[OPTIONS] [IMAGE]
```

```
docker run --volume $(pwd)/nginx.conf:/etc/nginx/nginx.conf:ro nginx
```

First Field

The First field you provide is the file or directory that is on the host. The location that you give must be the full path to it, notice in the example, we can mount a file from the same directory easily by using a command substitution of pwd. If the file or directory does not exist on the host when you run this command then Docker will create it for you, be aware though, if Docker creates the directory for you then it will be owned by root.

Second Field

The location that the file or directory from the host is going to be mounted in the container. In the example, the default configuration for the NGINX application is being replaced.

Options

Options are usually for access rights to the Bind Mount from within the container or platform specific requirements, for instance running on MacOS machine as opposed to Linux. Multiple options can be provided by separating them with commas.

ro	Only allow read access to the file or directory that has been mounted in the container.
rw	Allow read and write access to the file or directory that has been mounted in the container.
Configuring SELinux Labels (-v/--volume only)	
On some Linux distributions like CentOS and RHEL, Security Enhanced Linux (SELinux) is enforced which effects the file permissions when the file or folder is mounted into the container. By using any of these options you will be modifying the SELinux labels for the host's file or directory.	
z	Indicate that the file or directory is going to be accessed by more than one container.
Z	Indicate that the file or directory is going to be accessed by just one container.

Creating with the --mount flag

The mount flag is a newer option that you can use as opposed to using volume flag. The mount flag can actually do nearly everything that the volume flag can, apart from using the SELinux labels. The reason for having this flag as an option is because it is intended to be more verbose and clear about what it is accomplishing. Instead of having an ordered listing of options that need to be providing, the options for the mount flag are key-value pairs and comma separated in any order.

```
docker run --mount [OPTIONS]
```

```
docker run --mount \
type=bind,source=$(pwd)/nginx.conf,target=/etc/nginx/nginx.conf nginx
```

The mount flag tends to be the preferred flag one because of how clear it is to understand as you can see from the example above.

Mount Flag Options

source	The file or directory on the host to create a bind mount from. When using the mount flag, if the file or directory doesn't exist on the host then Docker will throw an error.
target	The File or directory that is going to be mounted inside the container.
destination	Appears to have the same functionality as the **target** option.
type	Mount is also a tool for mounting volumes in docker, the type option allows us to specify that we want to use Bind Mounts, as opposed to volumes.
readonly	Only allow read access to the file or directory that has been mounted in the container. It is Read/Write by default.

Tasks

This exercise will get you to create a custom configuration for NGINX on your host machine and then override the default configuration for NGINX running in a container by using a bind mount of the configuration file.

Create a new folder for this exercise ~/docker/bind_mounts_exercise

NGINX Configuration

Create you NGINX configuration file. This file config file is going to be very simple, when you make a request to NGINX (running on port 80) just the word NGINX should be returned.

nginx.conf

```
events {}
http {
   server {
       listen 80;
       location / {
           return 200 "NGINX";
       }
   }
}
```

Create a Bind Mount

Now override the default NGINX configuration file (/etc/nginx/nginx.conf) with the one that we created on the host machine. When creating your bind mount keep in mind the following:

- You will need to provide the full path of the config file on the host.
- NGINX runs on port 80, to access it outside the container, you will need to publish that port.
- If you are using the mount flag you need to provide the bind option.

If the Bind Mount was successful then when you make a request to the NGINX server, the word NGINX should be shown.

Using Docker in a Docker Container

Stop and remove all containers. The **/var/run/docker.sock** file is the socket file for communicating with the docker service. Try to build a Jenkins image, with Docker installed and the ability to build images and run containers on the host machine.