

Devops Experiment 7

Experiment 7

To share data between host and docker using Volume.

In general, Docker containers are ephemeral, running just as long as it takes for the command issued in the container to complete, this is an important feature as well as a limitation. By default, any data created inside the container is only available from within the container and only while the container is running, meaning there is close to no standard means to spawn docker container for the standard purpose of programming and development. A certain method of using volumes can enable the sharing of data. Docker volumes can be used to share files between a host system and the Docker container. For example, let's say you wanted to use the official Docker Nginx image and keep a permanent copy of Nginx's log files to analyze later. By default, the nginx Docker image will log to the `/var/log/nginx` directory inside the Docker Nginx container. Normally it's not reachable from the host filesystem.

This example shall help us understand how to achieve the same.

```
docker run --name=nginx -d \
-v ~/nginxlogs:/var/log/nginx -p 5000:80 nginx
# --name for naming, -v for mounting nginxlogs file, 5000:80 is basically
for exposing # the port 80 of container as port 5000 of machine
```

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo docker run --name=nginxContainerX -d -v ~/nginxlogs:/var/log/nginx -p 5001:80 nginx
86d197e9dbcf791af580e261d72843d19eac10820aed892762f31027294d7e05
vagrant@vagrant-ubuntu-trusty-64:~$ sudo apt install lnav
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libpcrcpp0
The following NEW packages will be installed:
  libpcrcpp0 lnav
0 upgraded, 2 newly installed, 0 to remove and 1 not upgraded.
Need to get 325 kB of archives.
After this operation, 926 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu/ trusty-updates/main libpcrcpp0 amd64 1:0.31-2ubuntu2.3 [14.5 kB]
Get:2 http://archive.ubuntu.com/ubuntu/ trusty/universe lnav amd64 0.6.2-1 [310 kB]
Fetched 325 kB in 1s (235 kB/s)
E: Couldn't make mmap of 30866462 bytes - Mmap-mmap (12: Cannot allocate memory)
E: Couldn't make mmap of 30845795 bytes - Mmap-mmap (12: Cannot allocate memory)
debconf: apt-extracttemplates failed: No such file or directory
Selecting previously unselected package libpcrcpp0:amd64.
(Reading database ... 69606 files and directories currently installed.)
Preparing to unpack .../libpcrcpp0_1%3a0.31-2ubuntu2.3_amd64.deb ...
Unpacking libpcrcpp0:amd64 (1:0.31-2ubuntu2.3) ...
Selecting previously unselected package lnav.
Preparing to unpack .../lnav_0.6.2-1_amd64.deb ...
Unpacking lnav (0.6.2-1) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Setting up libpcrcpp0:amd64 (1:0.31-2ubuntu2.3) ...
Setting up lnav (0.6.2-1) ...
Processing triggers for libc-bin (2.19-0ubuntu6.15) ...
vagrant@vagrant-ubuntu-trusty-64:~$ ls
ls: command not found
vagrant@vagrant-ubuntu-trusty-64:~$ ll
No command 'lls' found, did you mean:
Command 'hls' from package 'hfsutils' (main)
Command 'les' from package 'atm-tools' (universe)
Command 'lrs' from package 'lrslib' (universe)
Command 'llc' from package 'llvm' (universe)
Command 'jls' from package 'sleuthkit' (universe)
Command 'als' from package 'atool' (universe)
Command 'tls' from package 'python-tlsite' (universe)
Command 'ls' from package 'coreutils' (main)
Command 'ils' from package 'sleuthkit' (universe)
Command 'bls' from package 'bacula-sd' (main)
Command 'lvs' from package 'lvm2' (main)
Command 'ols' from package 'speech-tools' (universe)
Command 'lil' from package 'storebackup' (universe)
Command 'lli' from package 'llvm-runtime' (universe)
Command 'fls' from package 'sleuthkit' (universe)
lls: command not found
vagrant@vagrant-ubuntu-trusty-64:~$ ls
devops  Docker  log      nginxlogs  QuackDuck.first.pem  xyzqw.qq.pem
Docker DockerHTTPD MariaDB  pythonApp  site              Zee
vagrant@vagrant-ubuntu-trusty-64:~$
```

```
Mon Apr 20 14:12:22 UTC /home/vagrant/nginxlogs/error.log generic log 100
2020/04/20 14:11:03 [error] 6#6: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 192.168.0.109, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "192.168.0.110:50
01"

L1 100% 0W 0E 7:View Help
Press q/E to move forward/backward through error messages
```

```
vagrant@vagrant-ubuntu-trusty-64:~$ cd nginxlogs/
vagrant@vagrant-ubuntu-trusty-64:~/nginxlogs$ ls
access.log error.log
vagrant@vagrant-ubuntu-trusty-64:~/nginxlogs$ ls
access.log error.log
vagrant@vagrant-ubuntu-trusty-64:~/nginxlogs$ ln -s access.log
vagrant@vagrant-ubuntu-trusty-64:~/nginxlogs$ ln -s error.log
vagrant@vagrant-ubuntu-trusty-64:~/nginxlogs$ ls
access.log error.log
vagrant@vagrant-ubuntu-trusty-64:~/nginxlogs$ cat access.log
192.168.0.109 - - [19/Apr/2020:14:27:50 +0000] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/80.0.3987.149 Chrome/80.0.3987.149 Safari/537.36" "-"
192.168.0.109 - - [19/Apr/2020:14:27:51 +0000] "GET /favicon.ico HTTP/1.1" 404 556 "http://192.168.0.110:5000/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/80.0.3987.149 Chrome/80.0.3987.149 Safari/537.36" "-"
192.168.0.109 - - [20/Apr/2020:14:11:02 +0000] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:73.0) Gecko/20100101 Firefox/73.0" "-"
192.168.0.109 - - [20/Apr/2020:14:11:03 +0000] "GET /favicon.ico HTTP/1.1" 404 154 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:73.0) Gecko/20100101 Firefox/73.0" "-"
vagrant@vagrant-ubuntu-trusty-64:~/nginxlogs$ cat error.log
2020/04/19 14:27:51 [error] 6#6: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 192.168.0.109, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "192.168.0.110:5000", referer: "http://192.168.0.110:5000/"
2020/04/20 14:11:03 [error] 6#6: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 192.168.0.109, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "192.168.0.110:5001"
vagrant@vagrant-ubuntu-trusty-64:~/nginxlogs$
```

With docker 1.9, there comes the capability of creating volumes which allows easy sharing of data between containers

To make use of the volume, we'll create a new container from the Ubuntu image, using the `--rm` flag to automatically delete it when we exit. We'll also use `-v` to mount the new volume. `-v` requires the name of the volume, a colon, then the absolute path to where the volume should appear inside the container. If the directories in the path don't exist as part of the image, they'll be created when the command runs. If they do exist, the mounted volume will hide the existing content:

```
docker run -ti --rm -v DataVolume1:/datavolume1 ubuntu
```

```

attach  create  export  import  load  pause  push  rmi  start  top
build   diff    help    info    login  port   rename run  stats unpaus
commit  events  history inspect logout ps      restart save stop  versio
cp      exec   images  kill    logs   pull   rm      search tag   wait
vagrant@vagrant-ubuntu-trusty-64:~$ docker
attach  create  export  import  load  pause  push  rmi  start  top
build   diff    help    info    login  port   rename run  stats unpaus
commit  events  history inspect logout ps      restart save stop  versio
cp      exec   images  kill    logs   pull   rm      search tag   wait
vagrant@vagrant-ubuntu-trusty-64:~$ docker volume

Usage:  docker volume COMMAND

Manage volumes

Commands:
  create      Create a volume
  inspect     Display detailed information on one or more volumes
  ls          List volumes
  prune       Remove all unused local volumes
  rm          Remove one or more volumes

Run 'docker volume COMMAND --help' for more information on a command.
vagrant@vagrant-ubuntu-trusty-64:~$ docker volume create --name DataVolume1
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.38/volumes/create: dial unix /var/run/docker.sock: connect: permission de
vagrant@vagrant-ubuntu-trusty-64:~$ sudo docker volume create --name DataVolume1
DataVolume1
vagrant@vagrant-ubuntu-trusty-64:~$ docker run -ti --rm -v DataVolume1:/datavolume1 ubuntu
docker: Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.38/containers/create: dial unix /var/run/docker.sock: connect: pe
mission denied.
See 'docker run --help'.
vagrant@vagrant-ubuntu-trusty-64:~$ sudo docker run -ti --rm -v DataVolume1:/datavolume1 ubuntu
docker: Error response from daemon: OCI runtime create failed: container_linux.go:348: starting container process caused "process_linux.go:297: copying bootstrap data to pipe caused \"write init-p: broken pipe\": unknown
vagrant@vagrant-ubuntu-trusty-64:~$ sudo docker run -ti --rm -v DataVolume1:/datavolume1 ubuntu
docker: Error response from daemon: OCI runtime create failed: container_linux.go:348: starting container process caused "process_linux.go:297: copying bootstrap data to pipe caused \"write init-p: broken pipe\": unknown
vagrant@vagrant-ubuntu-trusty-64:~$ docker volume inspect DataVolume1
[
  {
    "CreatedAt": "2020-04-20T14:55:34Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/DataVolume1/_data",
    "Name": "DataVolume1",
    "Options": {},
    "Scope": "local"
  }
]
vagrant@vagrant-ubuntu-trusty-64:~$

```

We can also verify the DataVolume using following

```

vagrant@vagrant-ubuntu-trusty-64:~$ sudo docker volume inspect DataVolume1
[
  {
    "CreatedAt": "2020-04-20T14:55:34Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/DataVolume1/_data",
    "Name": "DataVolume1",
    "Options": {},
    "Scope": "local"
  }
]

```

the output is in json which can be basically parsed using any known language and then can be used for Automation