Running Multiple Services in a Docker Container via Supervisord at runtime

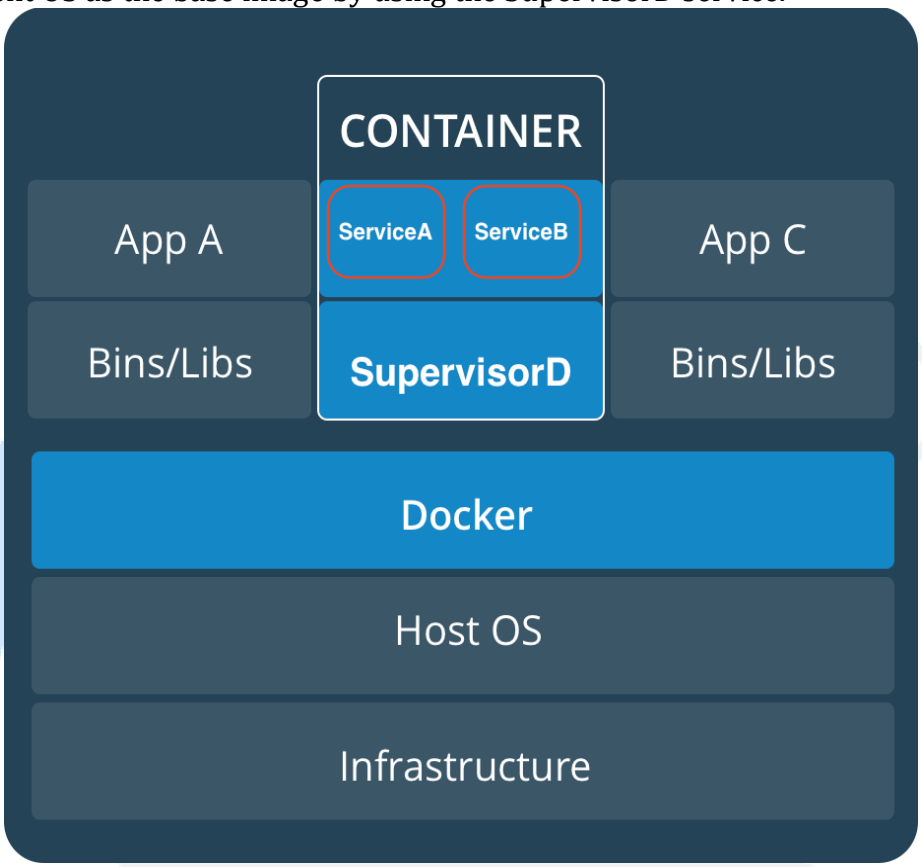
Author : Anubhav Chaturvedi

Have you ever faced this scenario where you want to run two or more lightweight services or multiple executables within the same container when the image starts running?

Although Docker provides a Docker-compose tool for building and running multi-services applications in multiple containers. Docker-compose requires a YAML file to configure your multiple services. But sometimes we want to run two or more lightweight services inside the same container.

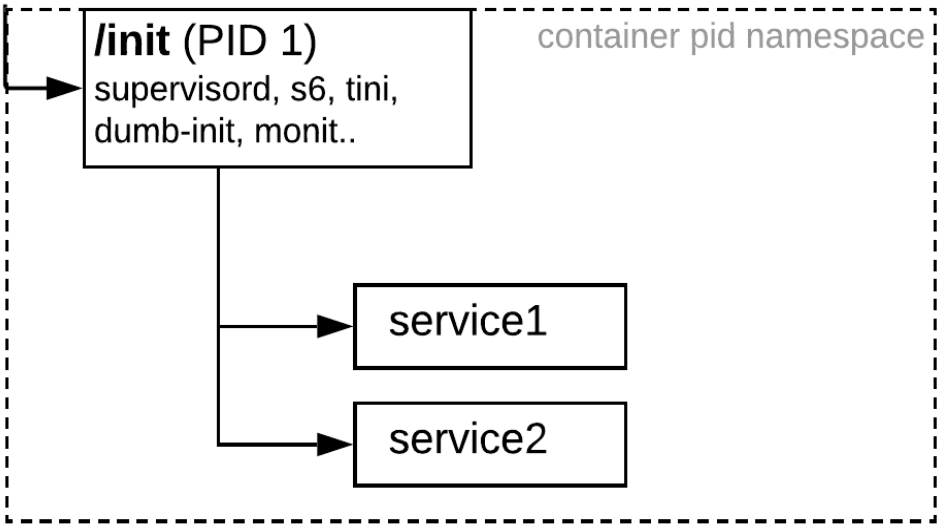
While working with Docker on a use case wherein I was supposed to implement two processes in a single docker container. Docker always had a limitation that only one CMD parameter can be provided in the Dockerfile as only one process can be run in the foreground. This use case included running Httpd and SSH on a single docker container that by far seem to be achievable only by passing shell script in CMD parameter of Dockerfile. After a rigorous search on the internet, I found this utility Supervisor by virtue of which we can run more than one process in a Docker container without having to worry about single CMD parameter or using shell scripts.

In this article I will illustrate how you can run services like SSHd and HTTPd inside a single container at start time with Cent OS as the base image by using the SupervisorD service.

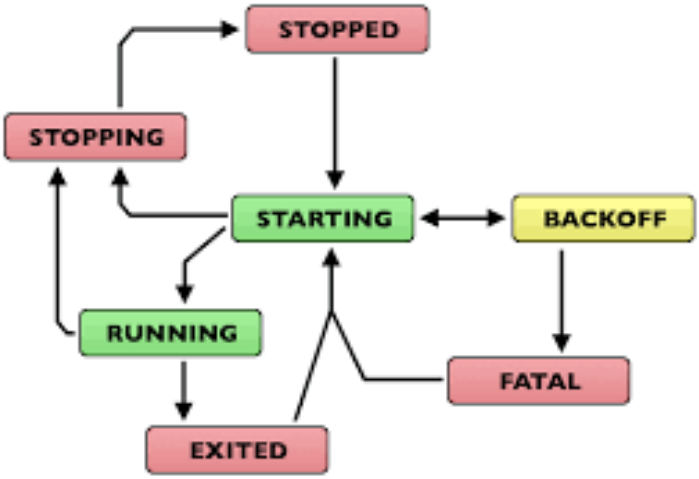


Why do we need Supervisord ?

Now, one of the problems we will face deploying multiple services like SSH ,Http is: How to control these services ? as we do not have init scripts or systemd available inside the containers and we do not even want them. Hence we will need some kind of mechanism to start, stop and monitor status of our services.



SupervisorD is a process control system, designed to monitor and control processes. It does not aim to replace init, instead it encapsulates processes inside its own framework, and can start them at boot time, just like we want.The Supervisor is a process control and monitoring tool that can be used to control multiple processes on UNIX-like OS.

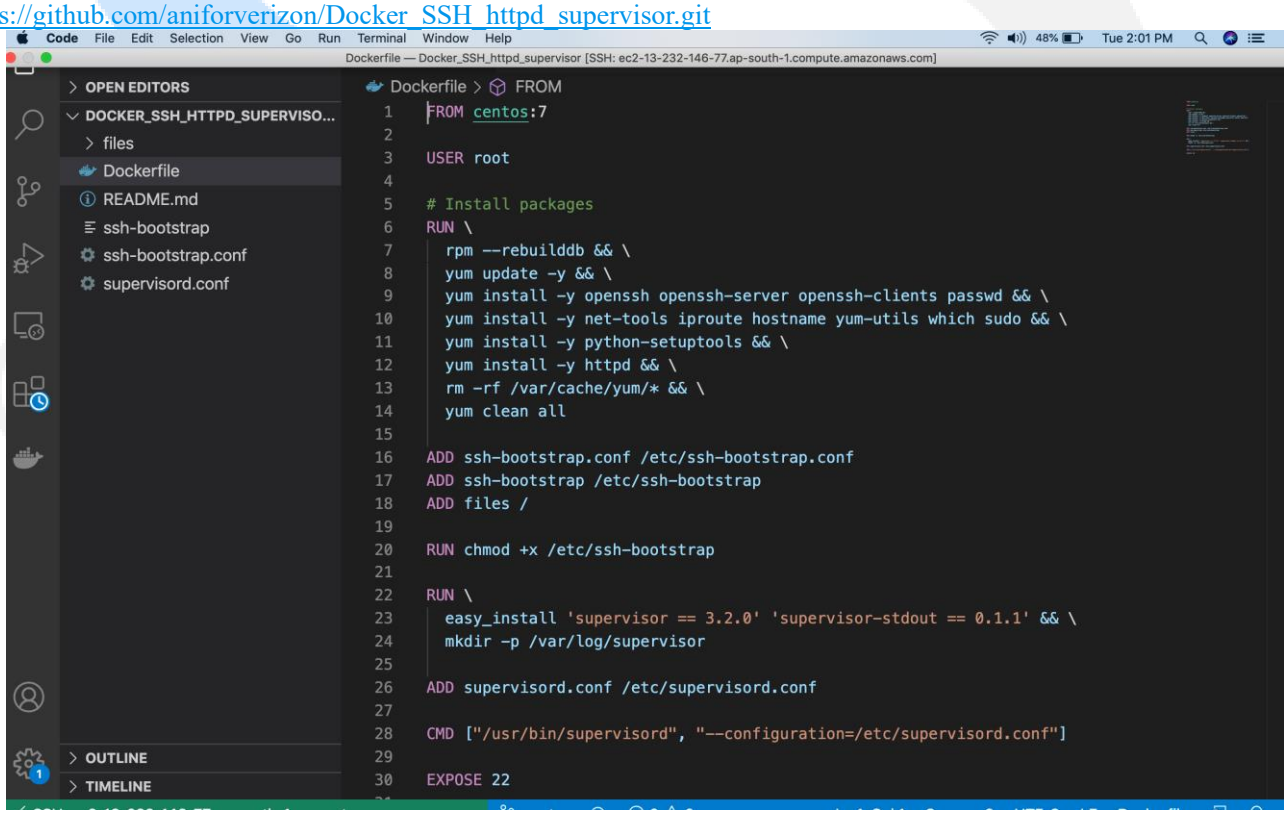


Although SupervisorD adds another layer of complexity ,it is indeed a very simple and useful way to manage multiple services .

According to some senior Docker Captains like Bret Fisher , this method is the most successful in deployment of highly scaled Web Applications running in production of the AWS cloud.

In order to install Supervisord almost all the Linux distributions have supervisor (or supervisord) in their default package repository – it just needs to be installed. SupervisorD is a python module. It can be installed using easy_install, which is a part of setuptools, which itself is an extension of the Python distutils package. Yes, it does get complicated quite at this point. Luckily, most Linux distributions ship with easy_install, including CentOS, which we will create today to make a Linux Distribution image.

Lets get started with the Dockerfile , below is the step by step screenshots of the process ,for code : https://github.com/aniforverizon/Docker_SSH_httpd_supervisor.git



You can see that I am taking CentOS as the base image, having a root as the user and installing standard packages for SSHd ,Supervisord and HTTPd as a service : Openssh ,Python setup tools ,network tools and httpd.Also adding the ssh configurations file consisting of passwords ,usernames and policies.

We will run Supervisord in the foreground and supervisord.conf file that has to be passed in the Dockerfile in which we can tell multiple processes to be run inside Docker container.

```
USER root

# Install packages
RUN \
  rpm --rebuilddb && \
  yum update -y && \
  yum install -y openssh openssh-server openssh-clients passwd && \
  yum install -y net-tools iproute hostname yum-utils which sudo && \
  yum install -y python-setuptools && \
  yum install -y httpd && \
  rm -rf /var/cache/yum/* && \
  yum clean all
```

```
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$ docker build -t anubhav_ssh_httpd
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.
Error: no context directory specified, and no dockerfile specified
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$ docker build -t anubhav_ssh_httpd .
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.
STEP 1: FROM centos:7
Getting image source signatures
Copying blob 75f829a71a1c done
Copying config 7e6257c9f8 done
Writing manifest to image destination
Storing signatures
STEP 2: USER root
9f028bba2ea049510bc1bf72413725e5a193156ad8e009e61321a5894fc1886d
STEP 3: RUN   rpm --rebuilddb &&   yum update -y &&   yum install -y openssh openssh-server openssh
-clients passwd &&   yum install -y net-tools iproute hostname yum-utils which sudo &&   yum instal
l -y python-setuptools &&   yum install -y httpd &&   rm -rf /var/cache/yum/* &&   yum clean all
```

Building the image

We are building the image for the same, lets observe the layer by layer build .

```
STEP 2: USER root
9f028bba2ea049510bc1bf72413725e5a193156ad8e009e61321a5894fc1886d
STEP 3: RUN   rpm --rebuilddb &&   yum update -y &&   yum install -y openssh openssh-server openssh
-clients passwd &&   yum install -y net-tools iproute hostname yum-utils which sudo &&   yum instal
l -y python-setuptools &&   yum install -y httpd &&   rm -rf /var/cache/yum/* &&   yum clean all
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
 * base: d36uatko69830t.cloudfront.net
 * extras: d36uatko69830t.cloudfront.net
 * updates: d36uatko69830t.cloudfront.net
base                                                     | 3.6 kB     00:00
extras                                                   | 2.9 kB     00:00
updates                                                  | 2.9 kB     00:00
(1/4): base/7/x86_64/group_gz                              | 153 kB   00:00
(2/4): extras/7/x86_64/primary_db                          | 206 kB   00:00
(3/4): updates/7/x86_64/primary_db                         | 3.8 MB   00:00
(4/4): base/7/x86_64/primary_db                            | 6.1 MB   00:02
No packages marked for update
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
 * base: d36uatko69830t.cloudfront.net
 * extras: d36uatko69830t.cloudfront.net
 * updates: d36uatko69830t.cloudfront.net
Package passwd-0.79-6.el7.x86_64 already installed and latest version
Resolving Dependencies
--> Running transaction check
---> Package openssh.x86_64 0:7.4p1-21.el7 will be installed
--> Processing Dependency: libfipscheck.so.1()(64bit) for package: openssh-7.4p1-21.el7.x86_64
---> Package openssh-clients.x86_64 0:7.4p1-21.el7 will be installed
--> Processing Dependency: libedit.so.0()(64bit) for package: openssh-clients-7.4p1-21.el7.x86_64
---> Package openssh-server.x86_64 0:7.4p1-21.el7 will be installed
--> Processing Dependency: libwrap.so.0()(64bit) for package: openssh-server-7.4p1-21.el7.x86_64
--> Running transaction check
---> Package fipscheck-lib.x86_64 0:1.4.1-6.el7 will be installed
```

installation of binaries

```
EXPLORER
OPEN EDITORS
DOCKER_SSH...
files
Dockerfile
README.md
ssh-bootstrap
ssh-bootstrap.conf
supervisord.conf
OUTLINE
TIMELINE

Dockerfile ×   ssh-bootstrap
Dockerfile > FROM

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   1: bash

---> Package fipscheck.x86_64 0:1.4.1-6.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch        Version                    Repository
                                                                        Size
================================================================================
Installing:
 openssh                x86_64      7.4p1-21.el7               base      510 k
 openssh-clients        x86_64      7.4p1-21.el7               base      655 k
 openssh-server         x86_64      7.4p1-21.el7               base      459 k
Installing for dependencies:
 fipscheck              x86_64      1.4.1-6.el7                base       21 k
 fipscheck-lib          x86_64      1.4.1-6.el7                base       11 k
 libedit                x86_64      3.0-12.20121213cvs.el7     base       92 k
 tcp_wrappers-libs      x86_64      7.6-77.el7                 base       66 k

Transaction Summary
================================================================================
Install  3 Packages (+4 Dependent packages)

Total download size: 1.8 M
Installed size: 5.8 M
Downloading packages:
warning: /var/cache/yum/x86_64/7/base/packages/fipscheck-1.4.1-6.el7.x86_64.rpm: Header V3 RSA/SHA2
56 Signature, key ID f4a80eb5: NOKEY
Public key for fipscheck-1.4.1-6.el7.x86_64.rpm is not installed
(1/7): fipscheck-1.4.1-6.el7.x86_64.rpm                    |  21 kB   00:00
(2/7): fipscheck-lib-1.4.1-6.el7.x86_64.rpm                |  11 kB   00:00
(3/7): libedit-3.0-12.20121213cvs.el7.x86_64.rpm           |  92 kB   00:00
(4/7): openssh-7.4p1-21.el7.x86_64.rpm                     | 510 kB   00:00
(5/7): openssh-clients-7.4p1-21.el7.x86_64.rpm             | 655 kB   00:00
```

openssh and ftp

```
EXPLORER
OPEN EDITORS
DOCKER_SSH_HTTPD_SUPERVISO...
files
Dockerfile
README.md
ssh-bootstrap
ssh-bootstrap.conf
supervisord.conf
OUTLINE
TIMELINE

Dockerfile ×   ssh-bootstrap
Dockerfile > FROM

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   1: bash

 python-ipaddress                    noarch   1.0.16-2.el7      base    34 k

Transaction Summary
================================================================================
Install  1 Package (+3 Dependent packages)

Total download size: 450 k
Installed size: 2.2 M
Downloading packages:
(1/4): python-backports-ssl_match_hostname-3.5.0.1-1.el7.n |  13 kB   00:00
(2/4): python-backports-1.0-8.el7.x86_64.rpm               | 5.8 kB   00:00
(3/4): python-ipaddress-1.0.16-2.el7.noarch.rpm            |  34 kB   00:00
(4/4): python-setuptools-0.9.8-7.el7.noarch.rpm            | 397 kB   00:00
--------------------------------------------------------------------------------
Total                                              1.6 MB/s | 450 kB  00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : python-backports-1.0-8.el7.x86_64                          1/4
  Installing : python-ipaddress-1.0.16-2.el7.noarch                       2/4
  Installing : python-backports-ssl_match_hostname-3.5.0.1-1.el7.noarch   3/4
  Installing : python-setuptools-0.9.8-7.el7.noarch                       4/4
  Verifying  : python-ipaddress-1.0.16-2.el7.noarch                       1/4
  Verifying  : python-setuptools-0.9.8-7.el7.noarch                       2/4
  Verifying  : python-backports-ssl_match_hostname-3.5.0.1-1.el7.noarch   3/4
  Verifying  : python-backports-1.0-8.el7.x86_64                          4/4

Installed:
  python-setuptools.noarch 0:0.9.8-7.el7

Dependency Installed:
  python-backports.x86_64 0:1.0-8.el7
  python-backports-ssl_match_hostname.noarch 0:3.5.0.1-1.el7
```

python and supervisord install on container

```
Dependencies Resolved

================================================================================
 Package             Arch          Version                     Repository   Size
================================================================================
Installing:
 httpd               x86_64        2.4.6-93.el7.centos         base        2.7 M
Installing for dependencies:
 apr                 x86_64        1.4.8-5.el7                 base        103 k
 apr-util            x86_64        1.5.2-6.el7                 base         92 k
 centos-logos        noarch        70.0.6-3.el7.centos         base         21 M
 httpd-tools         x86_64        2.4.6-93.el7.centos         base         92 k
 mailcap             noarch        2.1.41-2.el7                base         31 k

Transaction Summary
================================================================================
Install  1 Package (+5 Dependent packages)

Total download size: 24 M
Installed size: 32 M
Downloading packages:
(1/6): apr-1.4.8-5.el7.x86_64.rpm                          | 103 kB   00:00
(2/6): apr-util-1.5.2-6.el7.x86_64.rpm                     |  92 kB   00:00
(3/6): httpd-2.4.6-93.el7.centos.x86_64.rpm                | 2.7 MB   00:00
(4/6): httpd-tools-2.4.6-93.el7.centos.x86_64.rpm          |  92 kB   00:00
(5/6): mailcap-2.1.41-2.el7.noarch.rpm                     |  31 kB   00:00
(6/6): centos-logos-70.0.6-3.el7.centos.noarch.rpm         |  21 MB   00:00
--------------------------------------------------------------------------------
Total                                              32 MB/s |  24 MB  00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : apr-1.4.8-5.el7.x86_64                                      1/6
```

httpd install on the container

```
Complete!
Loaded plugins: fastestmirror, ovl
Cleaning repos: base extras updates
e09160bd21b995521923dfc1ad871491a28c4d023a017dd6f31683d4412860b5
STEP 4: ADD ssh-bootstrap.conf /etc/ssh-bootstrap.conf
1db50319d4fb2ffae5b7dd68b7a3cdd8ae8e3c16e93654f8b9e66a85603990a6
STEP 5: ADD ssh-bootstrap /etc/ssh-bootstrap
802269e09345efa9c64e0632f6491e346f2725c8d816e7e4157f770200b4de78
STEP 6: ADD files /
33d693f001d4366fab647334a99d720a04cb7eafd9abff238bb9714be2004b52
STEP 7: RUN chmod +x /etc/ssh-bootstrap
aad552cf4b24564d5414d87c801abfb5626ef32f3deeb2fdc7d9d49517cbe89c
STEP 8: RUN   easy_install 'supervisor == 3.2.0' 'supervisor-stdout == 0.1.1' &&   mkdir -p /var/lo
g/supervisor
Searching for supervisor==3.2.0
Reading https://pypi.python.org/simple/supervisor/
Best match: supervisor 3.2.0
Downloading https://files.pythonhosted.org/packages/c1/5d/f2badebeb0d40ec6a6d3e76c4cc5116cb4a839947
90d361c2ccae8a78f44/supervisor-3.2.0.tar.gz#sha256=522d7dad95c4b26a448bf39430a5404542000c15b45a2cf1
88c2fe05821965ce
Processing supervisor-3.2.0.tar.gz
Writing /tmp/easy_install-eDWZnE/supervisor-3.2.0/setup.cfg
Running supervisor-3.2.0/setup.py -q bdist_egg --dist-dir /tmp/easy_install-eDWZnE/supervisor-3.2.0
/egg-dist-tmp-OmTGxA
warning: no previously-included files matching '*' found under directory 'docs/.build'
Adding supervisor 3.2.0 to easy-install.pth file
Installing echo_supervisord_conf script to /usr/bin
Installing pidproxy script to /usr/bin
Installing supervisorctl script to /usr/bin
Installing supervisord script to /usr/bin

Installed /usr/lib/python2.7/site-packages/supervisor-3.2.0-py2.7.egg
Processing dependencies for supervisor==3.2.0
```

SSH configurations files mounted on the container

```
Writing /tmp/easy_install-jXymoH/meld3-2.0.1/setup.cfg
Running meld3-2.0.1/setup.py -q bdist_egg --dist-dir /tmp/easy_install-jXymoH/meld3-2.0.1/egg-dist-
tmp-I97DRS
zip_safe flag not set; analyzing archive contents...
Adding meld3 2.0.1 to easy-install.pth file

Installed /usr/lib/python2.7/site-packages/meld3-2.0.1-py2.7.egg
Finished processing dependencies for supervisor==3.2.0
Searching for supervisor-stdout==0.1.1
Reading https://pypi.python.org/simple/supervisor-stdout/
Best match: supervisor-stdout 0.1.1
Downloading https://files.pythonhosted.org/packages/ef/98/557ea85b26753c990a00159e32ead242be617754b
0f9f2683b0d4350a1b2/supervisor-stdout-0.1.1.tar.gz#sha256=44d39f634ca8d1876c3414422b5a070826cd3d22e
5003c482c3451eaffa45267
Processing supervisor-stdout-0.1.1.tar.gz
Writing /tmp/easy_install-ouSgz9/supervisor-stdout-0.1.1/setup.cfg
Running supervisor-stdout-0.1.1/setup.py -q bdist_egg --dist-dir /tmp/easy_install-ouSgz9/superviso
r-stdout-0.1.1/egg-dist-tmp-geUTPn
zip_safe flag not set; analyzing archive contents...
Adding supervisor-stdout 0.1.1 to easy-install.pth file
Installing supervisor_stdout script to /usr/bin

Installed /usr/lib/python2.7/site-packages/supervisor_stdout-0.1.1-py2.7.egg
Processing dependencies for supervisor-stdout==0.1.1
Finished processing dependencies for supervisor-stdout==0.1.1
e06dffa35da53f515e213240ee25e09b7f1ad1fef2ba0903498b5f1933463246
STEP 9: ADD supervisord.conf /etc/supervisord.conf
a46fc1fc7e288496ea441ffbb84ba14ab1db911bdf5df6fe6562e5f56b483df9
STEP 10: CMD ["/usr/bin/supervisord", "--configuration=/etc/supervisord.conf"]
c458284f80a4123b7b0e3f2541986a124a83f6b4f1cb105631fd3caddff55d77
STEP 11: EXPOSE 22
STEP 12: COMMIT anubhav_ssh_httpd
ffdbce61b3434897a528338c3268b55889024b4014f1f2bb1f7190d677d045c8
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$
```

Supervisord running via Docker CMD and SSH port 22 exposed

Finally when we spawn the container , we see httpd and sshd coming up on runtime:

```
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$ docker images
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.
REPOSITORY                    TAG      IMAGE ID       CREATED         SIZE
localhost/anubhav_ssh_httpd   latest   ffdbce61b343   2 minutes ago   272 MB
docker.io/library/centos      7        7e6257c9f8d8   7 days ago      211 MB
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$ docker run -it anubhav_ssh_httpd
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.
2020-08-18 08:38:36,919 CRIT Supervisor running as root (no user in config file)
2020-08-18 08:38:36,919 WARN Included extra file "/etc/supervisor.d/sshd.conf" during parsing
2020-08-18 08:38:36,923 INFO supervisord started with pid 1
2020-08-18 08:38:37,925 INFO spawned: 'supervisor_stdout' with pid 8
2020-08-18 08:38:37,927 INFO spawned: 'sshd-bootstrap' with pid 9
2020-08-18 08:38:37,928 INFO spawned: 'sshd' with pid 10
2020-08-18 08:38:37,930 INFO spawned: 'httpd' with pid 11
2020-08-18 08:38:37,969 INFO success: sshd-bootstrap entered RUNNING state, process has stayed up f
or > than 0 seconds (startsecs)
2020-08-18 08:38:38,163 INFO exited: sshd (exit status 1; not expected)
2020-08-18 08:38:38,461 INFO exited: sshd-bootstrap (exit status 0; expected)
2020-08-18 08:38:39,463 INFO success: supervisor_stdout entered RUNNING state, process has stayed u
p for > than 1 seconds (startsecs)
2020-08-18 08:38:39,464 INFO spawned: 'sshd' with pid 45
2020-08-18 08:38:39,464 INFO success: httpd entered RUNNING state, process has stayed up for > than
 1 seconds (startsecs)
2020-08-18 08:38:40,478 INFO success: sshd entered RUNNING state, process has stayed up for > than
1 seconds (startsecs)
```

both services coming up on runtime in the logs

When we look at all the ports and services open via : "ss -tln" command in Linux we can see some ports coming up:

```
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$ docker run -dit anubhav_ssh_httpd
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.
0792c4c388434698317f696c0f7c898ad739c233f028b374f42c1c6c86b1811c
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$ docker exec -it 079 bash
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.
[root@0792c4c38843 /]# ss -tln
State       Recv-Q Send-Q     Local Address:Port                 Peer Address:Port
LISTEN      0      0                  *:22                              *:*
LISTEN      0      0               [::]:80                           [::]:*
LISTEN      0      0               [::]:22                           [::]:*
[root@0792c4c38843 /]#
```

Running commands to see ports open in container

When I say that SSHd service is up and running on the container at runtime along with spawn & boot, then we need to confirm this by taking ssh access to this container, lets see what port is up and how to take access:

```
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$ docker run  -P -dit anubhav_ssh_httpd
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.
e2a31d89bd1cf4b9e68b63752d4916f3d535db920f8304ec4fb8c8fb729ae946
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$ docker port e2a
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.
22/tcp -> 0.0.0.0:43797
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$ ssh root@13.232.146.77 -p 43797
```

getting the port allocated and later getting access

We can get the SSH access and get started by using it :

```
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$ ssh root@13.232.146.77 -p 43797
^C
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:82:c9:19:fe:9a brd ff:ff:ff:ff:ff:ff
    inet 172.31.45.18/20 brd 172.31.47.255 scope global dynamic noprefixroute eth0
       valid_lft 2405sec preferred_lft 2405sec
    inet6 fe80::82:c9ff:fe19:fe9a/64 scope link
       valid_lft forever preferred_lft forever
3: cni0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether fa:f4:d2:0c:67:5c brd ff:ff:ff:ff:ff:ff
    inet 10.88.0.1/16 brd 10.88.255.255 scope global cni0
       valid_lft forever preferred_lft forever
    inet6 fe80::f8f4:d2ff:fe0c:675c/64 scope link
       valid_lft forever preferred_lft forever
4: veth5ab0ce7e@if3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master cni0 state UP
group default
    link/ether 52:e8:fd:b2:94:63 brd ff:ff:ff:ff:ff:ff link-netns cni-15ae91e6-296b-ffbd-4bfb-2655a
11661d9
    inet6 fe80::50e8:fdff:feb2:9463/64 scope link
       valid_lft forever preferred_lft forever
[ec2-user@ip-172-31-45-18 Docker_SSH_httpd_supervisor]$ ssh root@172.31.45.18 -p 43797
The authenticity of host '[172.31.45.18]:43797 ([172.31.45.18]:43797)' can't be established.
RSA key fingerprint is SHA256:8AsOheEPEqhOtB4hidxZiMV3UPHqHwAAhauxTJOKrwc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[172.31.45.18]:43797' (RSA) to the list of known hosts.
root@172.31.45.18's password:
[root@e2a31d89bd1c ~]#
```

To see the configurations sent in Supervisord for SSH service:

```
SSH_ROOT_PASSWORD="anubhav_ril"
SSH_USER="${SSH_USER:-admin}"
SSH_USER_PASSWORD="${SSH_USER_PASSWORD:-}"
SSH_USER_HOME_DIR="/home/admin"
SSH_USER_SHELL="/bin/bash"
SSH_SUDO="${SSH_SUDO:-ALL=(ALL) ALL}"
```

bootstrap config for ssh

There are supervisord.conf standard configurations that have to be made, this is what the file looks like:

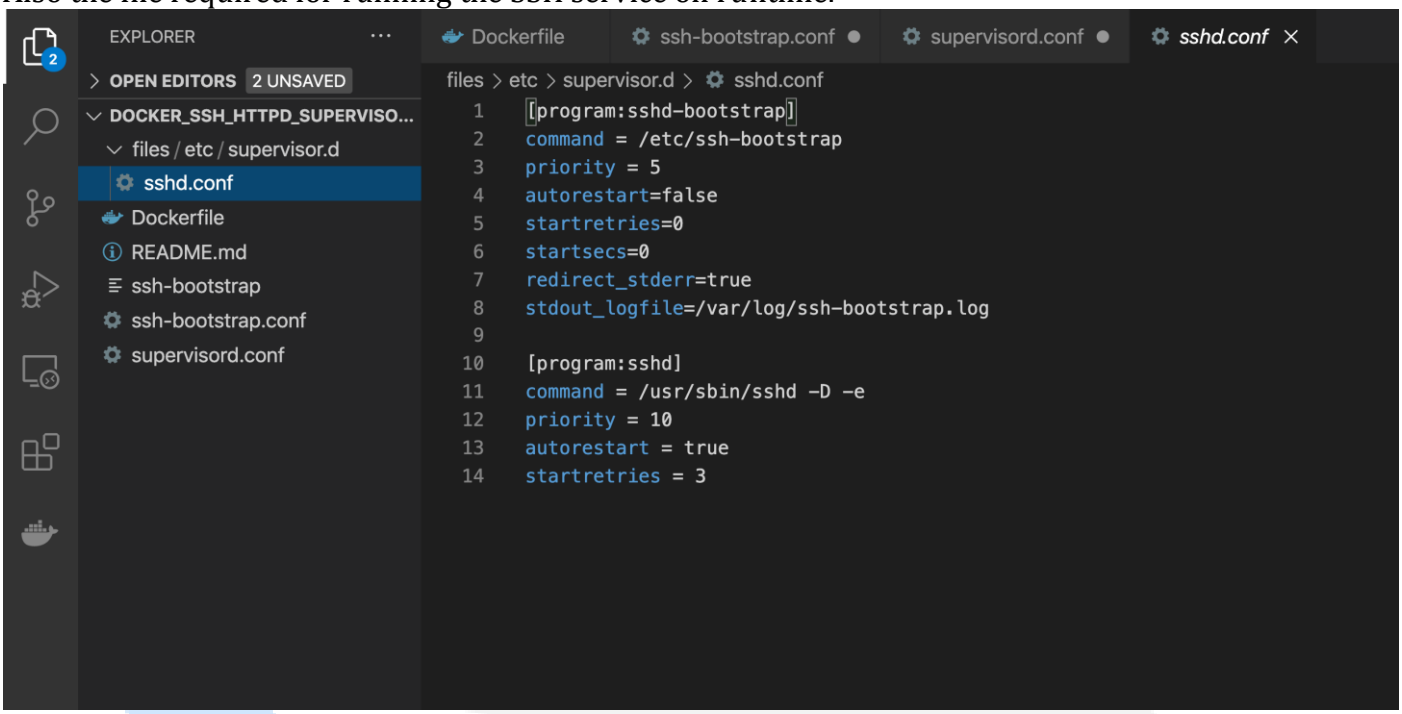
```
[supervisord]
logfile=/var/log/supervisor/supervisord.log  ; (main log file;default $CWD/supe
logfile_maxbytes=50MB        ; (max main logfile bytes b4 rotation;default 50MB)
logfile_backups=10           ; (num of main logfile rotation backups;default 10)
loglevel=info                ; (log level;default info; others: debug,warn,trace
pidfile=/var/run/supervisord.pid ; (supervisord pidfile;default supervisord.pid
nodaemon=true                ; (start in foreground if true;default false)
minfds=1024                  ; (min. avail startup file descriptors;default 1024
minprocs=200                 ; (min. avail process descriptors;default 200)


[eventlistener:supervisor_stdout]
command = /usr/bin/supervisor_stdout
priority = -1
buffer_size = 100
events = PROCESS_LOG
result_handler = supervisor_stdout:event_handler

[include]
files = /etc/supervisor.d/*.conf

[program:httpd]
command=/usr/sbin/httpd -DFOREGROUND
autostart=true
autorestart=true
stderr_logfile=/var/log/httpd.err.log
stdout_logfile=/var/log/httpd.out.log
```

Supervisord.conf file

What are we doing here? Each square bracket pair defines a section. For supervisord itself, we define that it should start in the foreground rather than daemonize itself, which would mean becoming a background service.

For the program named sshd, we execute the relevant command, essentially running SSHD in the background. For the program named httpd, we start the server in the foreground, in a separate shell. There are many other options available, but at the moment, this is the bare minimum we need to get underway with supervisord.

Also the file required for running the SSH service on runtime.



I took SSHD service in this example because its a popular service and wanted to demonstrate how containers can be accessed via SSH, but ideally it is not a good practice to run SSHd inside your containers , for further reference :

follow this link .

For accessing code of this article

click on this.

For any further Queries or anything related to Blockchain or DevOps or specialized Docker queries you can

DM me on Linkedin or instagram id=acanubhav94.

About Cilans Systems:

We are a rapidly growing Corporate consultancy, and software development company based in Ahmedabad, India. We specialize in Python/R/SPSS, Data Science, Artificial Intelligence-Machine Learning (AI-ML), Data Visualization (Power BI, Excel), Blockchain-Hyperledger, and DevOps. We also develop Mobile/Web/UX/UI applications to support earlier domains and relevant assignments. Our services portfolio include Corporate Consulting/Mentorship, Corporate Training (executive and mid management for Future Technologies and G-T-M /Go-To-Market strategies and also Hands-on foe engineering level), develop POC (Proof of concept) for complex solutions/designs, custom software development and QA/Software Testing services. Our clients are from diverse sectors including Finance/Banking, Supply chain, Govt, Healthcare, E-Commerce, Hospitality, Industrial automation, auto-fleet etc. For more details, please visit **www.cilans.net**