RUNNING DOCKER CONTAINERS WITH SYSTEMD

Dec 30, 2017 · 2 minutes read

Services in Docker containers can easily be run under systemd, but I had some difficulty finding good examples of how to do so. Here's what I came up with.

This example is a container running nginx, which has access to letsencrypt certificates, but could be adapted to any service. I've annotated lines below:

```
[Unit]
Description=nginx (Docker)
# start this unit only after docker has started
After=docker.service
Requires=docker.service

[Service]
TimeoutStartSec=0
Restart=always
# The following lines start with '-' because they are allowed to fail without
# causing startup to fail.
#
# Kill the old instance, if it's still running for some reason
ExecStartPre=-/usr/bin/docker kill nginx
# Remove the old instance, if it stuck around
ExecStartPre=-/usr/bin/docker rm nginx
# Pull the latest version of the container; NOTE you should be careful to
# pull a tagged version, that way you won't accidentially pull a major-version
# upgrade and break your service!
ExecStartPre=-/usr/bin/docker pull "nginx:1.13"
# Start the actual service; note we remove the instance after it exits
ExecStart=/usr/bin/docker run --rm --name nginx -p 80:80 -p 443:443 -v /etc/se
rvice-configs/nginx/nginx.conf:/etc/nginx/nginx.conf:ro -v /var/www/letsencryp
t:/var/www/letsencrypt:z -v /etc/letsencrypt:/etc/letsencrypt:ro nginx:1.13
# On exit, stop the container
ExecStop=/usr/bin/docker stop nginx

[Install]
WantedBy=multi-user.target
```

This file is saved as `/etc/systemd/system/docker.nginx.service`. You must make it executable (`chmod +x <unit_file>`) and then do a systemd reload:

```
systemctl daemon-reload
systemctl start <unit_file>
# if everything works as expected, enable it
systemctl enable <unit_file>
```

I'm using this method to run several containers on a Digital Ocean instance, and it's worked as expected.

Back to posts