

Systemd provides a standard process for controlling programs / processes on Linux hosts. In effect it has replaced `initd`. If you're using a somewhat modern Linux system (RHEL 7.x, SuSE 12, Ubuntu 15, etc), then you will encounter systemd. One of the nice things about systemd is that it is a single command that can be used to manage almost all aspects of a process. Also, systemd is easily configurable through what are known as systemd unit files.

The following explains how systemd can be used with the Docker daemon. For the most part this should not be any different from how systemd would interact with any service.

Prerequisites

- A system that uses systemd
- Docker installed, preferably the latest

Verifying the Status and Health of the Docker Daemon

The `systemctl status` command provides not only information about whether the service / process is running (`active`), but also whether it is `enabled` to start at boot of the machine. This in itself would have required multiple programs (`chkconfig` etc) to query. Additionally, it shows the default unit file loaded in the following line:

```
Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
```

Among other information, useful messages about the PID, memory consumed, arguments (`DOCKER_OPTS`) and the last 10 lines (aka `tail`) are also shown:

```
[ec2-user@ip-10-0-46-113 ~]$ sudo systemctl status docker
docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2016-02-02 18:13:42 EST; 2 days ago
     Docs: https://docs.docker.com
    Main PID: 16915 (docker)
      Memory: 14.7M
      CGroup: /system.slice/docker.service
              └─16915 /usr/bin/docker daemon -H fd://
                  └─20162 docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 3375 -container-ip
172.17.0.2 -container-port 2375

Feb 03 19:51:35 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T19:51:35.411425044-
05:00" level=info msg="GET /v1.21/conta.../json"
Feb 03 19:51:46 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T19:51:46.575906700-
05:00" level=info msg="GET /v1.21/version"
Feb 03 19:57:51 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T19:57:51.748731356-
05:00" level=info msg="GET /v1.21/conta.../json"
Feb 03 19:57:59 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T19:57:59.959082130-
05:00" level=info msg="POST /v1.21/cont...?t=10"
Feb 03 19:58:03 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T19:58:03.958291300-
05:00" level=info msg="GET /v1.21/conta...mit=1"
Feb 03 19:58:04 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T19:58:04.000435804-
05:00" level=info msg="GET /v1.21/conta.../json"
Feb 03 19:58:04 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T19:58:04.001717319-
05:00" level=info msg="GET /v1.21/conta...l=all"
Feb 03 20:08:52 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T20:08:52.811916273-
05:00" level=info msg="GET /v1.21/conta...mit=1"
Feb 03 20:08:52 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T20:08:52.855966139-
05:00" level=info msg="GET /v1.21/conta.../json"
Feb 03 20:08:52 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T20:08:52.857644962-
05:00" level=info msg="GET /v1.21/conta...l=all"
Hint: Some lines were ellipsized, use -l to show in full.
```

To enable the daemon to start at boot time or not use `sudo systemctl enable docker` and `sudo systemctl disable docker` respectively.

Overriding Defaults for the Docker Daemon

By default the systemd configuration files controlling the service are under the folder `/usr/lib/systemd/system`. This is also evident in the `Loaded:` line in the output of the `systemctl status` command.

```
[ec2-user@ip-10-0-46-113 ~]$ sudo systemctl status docker.service | grep Loaded
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
```

The `docker.service` file contains all the configuration options for the docker process. The options are organized under sections `[Unit]`, `[Service]` and `[Install]`. The format of the file is very similar to an init file, with `name=value` pairs in each section.

To override the daemon options, it is recommended not to edit this file. Instead, an override file should be used. The override file would contain just the options that need to be changed along with the section heading it is under.

Now many instructions about `systemd` describe the need to create a separate directory tree and host the override file in that path. While this will work, it is not necessary to create it by hand, nor is it necessary to remember where exactly the override file(s) need to go. Instead it is easier to just run `systemctl edit docker`. So, say we want to modify the daemon options so that the service is accessible over the network and also have it bind to the unix socket at `/var/run/docker.sock`. We would first want to see what the option is set to currently. This can be done using the `systemctl cat` like below:

```
[ec2-user@ip-10-0-46-113 ~]$ sudo systemctl cat docker | grep ExecStart
ExecStart=/usr/bin/docker daemon -H fd://
[ec2-user@ip-10-0-46-113 ~]$
```

The nice thing about the command above is that it shows a unified view of all options and takes into consideration any overrides that may have been applied.

To change the value of an option, `ExecStart` in this case, do the following:

```
[ec2-user@ip-10-0-46-113 ~]$ sudo systemctl edit docker
```

This will create the necessary directory structure under `/etc/systemd/system/docker.service.d` and open an editor (using the default editor configured for the user) to the override file. Add the section below into the editor:

```
[Service]
ExecStart=
ExecStart=/usr/bin/docker daemon -H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sock
```

It was necessary to clear out `ExecStart` using `ExecStart=` before setting it to the override value. This is only required for some options and most options in the configuration file would not need to be cleared like this

Using `systemctl edit` also ensures that the override settings are loaded. Had you edited the file manually using `vi` etc, you would have to run `sudo systemctl daemon-reload` in order for the new settings to be loaded.

The only reason you may want to run the `systemctl daemon-reload` command is if you made a typo or error in the override file when you did `systemctl edit`. For this you would need to be aware of the path of the override file, which is `/etc/systemd/system/docker.service.d/override.conf`.

Finally, in order for the override settings to take effect, restart the daemon using `systemctl restart docker`. Again, the status of the new settings and the process health can be verified by running `systemctl status docker`.

```
[ec2-user@ip-10-0-46-113 ~]$ sudo systemctl edit docker
[ec2-user@ip-10-0-46-113 ~]$ sudo systemctl status docker
docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Drop-In: /etc/systemd/system/docker.service.d
            └─override.conf
   Active: active (running) since Fri 2016-02-05 18:49:14 EST; 1min 28s ago
     Docs: https://docs.docker.com
    Main PID: 24850 (docker)
    CGroup: /system.slice/docker.service
            └─24850 /usr/bin/docker daemon -H tcp://0.0.0.0:9999
```

```
Feb 05 18:49:14 ip-10-0-46-113.ec2.internal docker[24850]: time="2016-02-05T18:49:14.335228582-0..."
Feb 05 18:49:14 ip-10-0-46-113.ec2.internal docker[24850]: time="2016-02-05T18:49:14.378558031-0..."
Feb 05 18:49:14 ip-10-0-46-113.ec2.internal docker[24850]: time="2016-02-05T18:49:14.386429304-0...e"
Feb 05 18:49:14 ip-10-0-46-113.ec2.internal docker[24850]: time="2016-02-05T18:49:14.419513857-0...s"
Feb 05 18:49:14 ip-10-0-46-113.ec2.internal docker[24850]: time="2016-02-05T18:49:14.466389616-0..."
Feb 05 18:49:14 ip-10-0-46-113.ec2.internal docker[24850]: ...
Feb 05 18:49:14 ip-10-0-46-113.ec2.internal docker[24850]: time="2016-02-05T18:49:14.469497693-0..."
Feb 05 18:49:14 ip-10-0-46-113.ec2.internal docker[24850]: time="2016-02-05T18:49:14.469518888-0...n"
Feb 05 18:49:14 ip-10-0-46-113.ec2.internal docker[24850]: time="2016-02-05T18:49:14.469537359-0...s3
Feb 05 18:49:14 ip-10-0-46-113.ec2.internal systemd[1]: Started Docker Application Container Engine.
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-10-0-46-113 ~]$
```

Viewing Logs Related to the Docker Service

To view logs related to the Docker service, run the following command:

```
journalctl -u docker
```

The output will be similar to the following:

```
ec2-user@ip-10-0-46-113 ~]$ journalctl -u docker
-- Logs begin at Tue 2016-02-02 11:20:18 EST, end at Fri 2016-02-05 19:18:49 EST. --
Feb 02 18:13:40 ip-10-0-46-113.ec2.internal systemd[1]: Starting Docker Application Container Engine.
Feb 02 18:13:40 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-02T18:13:40.394723367-05:00"
Feb 02 18:13:40 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-02T18:13:40.464672420-05:00"
Feb 02 18:13:42 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-02T18:13:42.225872353-05:00"
Feb 02 18:13:42 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-02T18:13:42.245163371-05:00"
Feb 02 18:13:42 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-02T18:13:42.283163052-05:00"
Feb 02 18:13:42 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-02T18:13:42.373603957-05:00"
Feb 03 10:33:06 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T10:33:06.826177379-05:00"
Feb 03 10:33:07 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T10:33:07.104231280-05:00"
Feb 03 10:33:10 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T10:33:10.474069661-05:00"
Feb 03 10:33:27 ip-10-0-46-113.ec2.internal docker[16915]: time="2016-02-03T10:33:27.031864375-05:00"
lines 1-12...skipping...
```