

How to set environment variable in systemd service?

[Ask Question](#)

▲
131
▼
★

I have an [Arch Linux](#) [systemd](#) and I've created my own service. The configuration service at

41 `/etc/systemd/system/my.service` looks like this:

```
[Unit]
Description=My Daemon

[Service]
ExecStart=/bin/myforegroundcmd

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

Now I want to have an environment variable set for the `/bin/myforegroundcmd`. How do I do that?

[arch-linux](#) [systemd](#)

edited Mar 4 '14 at 8:37



Falcon Momot

22.7k 10 48 79

asked Aug 1 '12 at 19:43



lfagundes

758 2 6 6

5 Answers

▲ Times change and so do best



The current best way to do this is to run `systemctl edit myservice`, which will create an override file for you or let you edit an existing one.

In normal installations this will create a directory `/etc/systemd/system/myservice.service.d`, and inside that directory create a file whose name ends in `.conf` (typically, `override.conf`), and in this file you can add to or override any part of the unit shipped by the distribution.

For instance, in a file `/etc/systemd/system/myservice.service.d/myenv.conf`:

```
[Service]
Environment="SECRET"
Environment="ANOTHER"
```

Also note that if the directory exists and is empty, your service will be disabled! If you don't intend to put something in the directory, ensure that it does not exist.

For reference, the old way was:

The recommended way to do this is to create a file `/etc/sysconfig/my-service` which

contains your
variables, and then
load them with
`EnvironmentFile` .

For complete
details, see
Fedora's
documentation on
[how to write a
systemd script](#).

edited Feb 27 at 12:52



Mikolasan

3 3

answered Aug 1 '12 at 20:07





Michael Hampton ♦



171k 27 312 637



4 ▲ I guess the
sysconfig
path is
specific to
Fedora but
the question is
about Arch
Linux. The
answer by
paluh is more
interesting I
think –
[Ludovic Kutý](#)
Apr 27 '13 at
8:49

1 ▲ /etc/sysco
nfig is
Fedora-
specific.
AFAIR Arch
Linux was
pushing for
having the
config files
somewhere
package-
specific rather
in /etc
rather than
that Fedora-
specific
location. Like
/etc/myser
vice.conf ,
though using
extra file
doesn't seem
the right way
here. –
[Michał Górný](#)
Apr 23 '14 at
7:13

5   No, no, no.
/etc/sysconfig
is not
recommended.
It is
discouraged,
along with
/etc/default/*
from debian,
because they
are pointless,
and the
names are
meaningless
and make
sense only for
backwards
compatibility
reasons (all of

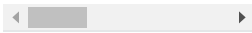
/etc is about
configuration
of the system,
not just
/etc/sysconfig,
and
/etc/defaults is
for overrides,
not the
defaults). Just
put the
definitions
directly in the
unit file, or if it
is not
possible, in an
environment
file that has a
package
specific
location (like
Michał's
comment
suggests). –
zbyszek Oct
4 '14 at 18:41

1   @FrederickNo
rd It's just
variable=valu
e pairs, such
as
DJANGO_SET
TINGS_MODUL
E=project.s
ettings ,
one per line. –
Michael Hampt
Nov 2 '15 at
17:01

1   @MichaelHa
mpton Could
you please
add
documentatio
n link for

"current best
way"? – **jb.**
Dec 31 '15 at
13:26

|



71

The answer depends on whether the variable is supposed to be constant (that is, not supposed to be modified by user getting the unit) or variable (supposed to be set by the user).

Since it's your local unit, the boundary is quite blurry and either way would work. However, if you started to distribute it and it would end up in `/usr/lib/systemd/system`, this would become important.

Constant value

If the value doesn't need to change per instance, the preferred way would be to place it as `Environment=`, directly in the unit file:

```
[Unit]
Description=My Daemon

[Service]
Environment="FOO=bar"
ExecStart=/bin/myfoo

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

The advantage of that is that the variable is kept in a

single file with the unit. Therefore, the unit file is easier to move between systems.

Variable value

However, the above solution doesn't work well when `sysadmin` is supposed to change the value of the environment variable locally. More specifically, the new value would need to be set every time the unit file is updated.

For this case, an extra file is to be used. How — usually depends on the distribution policy.

One particularly interesting solution is to use

`/etc/systemd/system/myservice.service.d` directory. Unlike other solutions, this directory is supported by `systemd` itself and therefore comes with no distribution-specific paths.

In this case, you place a file like

`/etc/systemd/system/myservice.service.d/local.conf` that adds the missing parts of unit file:

```
[Service]
Environment="FOO=bar"
```

Afterwards, `systemd` merges the two files

when starting the service (remember to `systemctl daemon-reload` after changing either of them). And since this path is used directly by `systemd`, you don't use `EnvironmentFile=` for this.

If the value is supposed to be changed only on some of the affected systems, you may combine both solutions, providing a default directly in the unit and a local override in the other file.

edited Oct 16 '18 at 18:12



The Guy with The Hat

105 5

answered Apr 23 '14 at 7:48

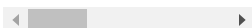


Michał Górny

951 7 6

systemctl
daemon-reload is the command to reload `systemd` –
Dmitry Buzolin
Apr 22 '18 at 23:57

Environment
`File=` is better when the values are secrets like passwords. See [my answer](#) for details. –
Don Kirkby
May 4 '18 at 0:31



<http://0pointer.de/pu>

37

[blic/systemd-man/systemd.exec.html#Environment=](#) -
you have two options (one already pointed by Michael):

```
Environment=
```

and

```
EnvironmentFile=
```

edited Aug 29 '13 at 17:25

answered Oct 16 '12 at 13:55



paluh

471 4 7

12

The answers by [Michael](#) and [Michał](#) are helpful and answer the original question of how to set an environment variable for a systemd service. However, one [common use](#) for environment variables is to configure sensitive data like passwords in a place that won't accidentally get committed to source control with your application's code.

If that's why you want to pass an environment variable to your service, **do not** use `Environment=` in the unit configuration file. Use `EnvironmentFile=` and point it to another configuration file that is only readable

by the service
account (and users
with root access).

The details of the
unit configuration file
are visible to any
user with this
command:

```
systemctl show my_
```

I put a configuration
file at

```
/etc/my_service/my_
_service.conf and
put my secrets in
there:
```

```
MY_SECRET=correctho
```

Then in my service
unit file, I used

```
EnvironmentFile= :
```

```
[Unit]
Description=my_serv
```

```
[Service]
ExecStart=/usr/bin/
EnvironmentFile=/et
User=myservice
```

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

I checked that `ps`
`auxc` can't see
those environment
variables, and other
users don't have
access to

```
/proc/*/environ .
```

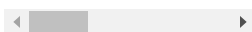
Check on your own
system, of course.

answered May 4 '18 at 0:29



Don Kirkby

375 2 5 20



8



Michael gave one
clean solution but I
wanted to get
updated env
variable from script.
Unfortunately

executing bash commands is not possible in systemd unit file. Fortunately you can trigger bash inside ExecStart:

<http://www.dsm.fordham.edu/cgi-bin/man-cgi.pl?topic=systemd.service§=5>

Note that this setting does not directly support shell command lines. If shell command lines are to be used they need to be passed explicitly to a shell implementation of some kind.

Example in our case is then:

```
[Service]
ExecStart=/bin/bash
```

answered Jul 23 '14 at 13:31



user1830432

208 2 5

-
- 6 ▲ This won't work for multiple reasons (unless it's a "one-shot" service, which is rather pointless). I managed to get the following to work:
- ```
/bin/bash -
a -c 'source
/etc/sysconf
ig/whatever
&& exec
whatever -
program' .
The -a
ensures the
environment is
```

exported to the  
sub-process  
(unless you  
want to prefix  
all variables in  
whatever  
with `export` )  
– Otheus Apr  
29 '15 at 22:42

---

▲ why it won't  
▮ work? It should  
always trigger  
entire  
command  
which includes  
executing the  
script, ain't it?  
– user1830432  
Apr 30 '15 at  
8:29

---

▲ Maybe  
▮ ExecStart=  
usr/bin/env  
ENV=script  
/bin/myforeg  
roundcmd is a  
little better  
solution in this  
case. – kstep  
Nov 26 '15 at  
6:18

---

▲ @Otheus:  
▮ Great answer,  
saved by day  
when I had to  
create a  
Tomcat 8 Unit  
file. – Daniel  
Apr 13 '16 at  
6:58

---

1 ▲ There IS a way  
▮ to execute a  
bash  
command "in"  
a systemd  
service file.  
See this link:  
[coreos.com/os/docs/latest/...](https://coreos.com/os/docs/latest/...)  
– Mark Lakata  
Jan 13 '17 at  
0:35

---