

# symbol 'grub\_calloc' not found -- how to fix on AWS

An eventful day. Besides upgrading Jira and Confluence, I also do upgrades of the underlying Ubuntu operating system. Today I had 8 AWS EC2 instances to upgrade, all running Ubuntu 16.04.6 LTS.

I did the usual steps, upgrading the sandboxes first:

```
apt-get update
apt-get upgrade
reboot
```

3 of the 4 sandboxes failed to boot! They were all stuck on Grub rescue screen, showing:

```
Booting from Hard Disk 0...
error: symbol `grub_calloc' not found. Entering rescue mode...
grub rescue> _
```

## Get instance screenshot

Below is a screenshot of i- ( ) at 2020-07-30T00:44:41.531-04:00.

 Refresh

```
SeaBIOS (version 1.7.375.10-g22c3a61ce2e1.18)
Machine UUID ec2
Booting from Hard Disk 0...
error: symbol `grub_calloc' not found.
Entering rescue mode...
grub rescue> _
```

Close

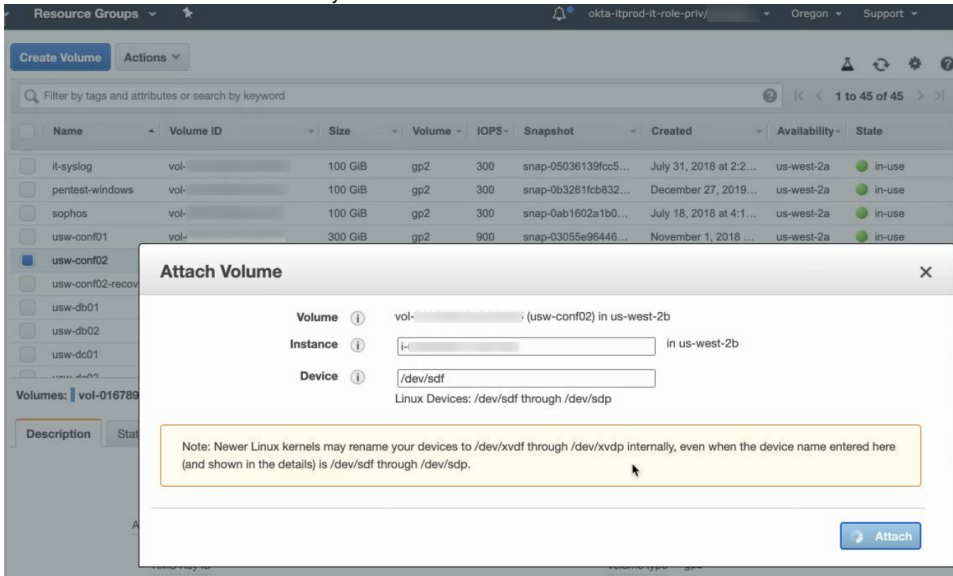
Googling yielded this [askubuntu.com](https://askubuntu.com) post, which provides a general way forward: we need to reinstall grub with `grub-install <disk>`.

## How to reinstall grub on AWS

The joyous thing about AWS is you only get a *screenshot* of the `grub rescue>` prompt. You can't actually rescue anything. (*Edit: this wasn't a MacGuyver-rescue'able situation anyway.*)

For AWS the process is:

1. Launch a recovery t2.micro in the same AZ/subnet as your borked instance(s).
2. Stop the broken instance and detach the root volume (the one containing the OS).
3. Attach the root volume to the recovery instance as /dev/sdf:



4. Now run these commands:

```
mount /dev/xvdf1 /mnt
for fs in {proc,sys,tmp,dev}; do mount -o bind /$fs /mnt/$fs; done
chroot /mnt
lsblk
grub-install /dev/xvdf
```

It looked like this:

```
[root@ip-10-36-7-171 ~]#
[root@ip-10-36-7-171 ~]#
[root@ip-10-36-7-171 ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   8G  0 disk
└─xvda1    202:1    0   8G  0 part /
xvdf        202:80   0 600G  0 disk
└─xvdf1    202:81   0 600G  0 part
[root@ip-10-36-7-171 ~]#
[root@ip-10-36-7-171 ~]#
[root@ip-10-36-7-171 ~]# mount /dev/xvdf1 /mnt
[root@ip-10-36-7-171 ~]# for fs in {proc,sys,tmp,dev}; do mount -o bind /$fs /mnt/$fs; done
[root@ip-10-36-7-171 ~]# chroot /mnt
root@ip-10-36-7-171:~#
root@ip-10-36-7-171:~#
root@ip-10-36-7-171:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   8G  0 disk
└─xvda1    202:1    0   8G  0 part /tmp
xvdf        202:80   0 600G  0 disk
└─xvdf1    202:81   0 600G  0 part /
root@ip-10-36-7-171:~# grub-install /dev/xvdf
Installing for i386-pc platform.
Installation finished. No error reported.
root@ip-10-36-7-171:~# grub-install /dev/xvdf1
Installing for i386-pc platform.
grub-install: warning: File system 'ext2' doesn't support embedding.
grub-install: warning: Embedding is not possible. GRUB can only be installed in this setup by using blocklists. However, blocklists are UNRELIABLE and their use is discouraged.
grub-install: error: will not proceed with blocklists.
root@ip-10-36-7-171:~#
```

5. Then:

```
exit
for fs in {proc,sys,tmp,dev}; do umount /mnt/$fs; done
umount /mnt
```

(Edit: added fs umounts per [this LP comment](#))

6. Detach the volume from the recovery instance
7. Attach the volume back on the original instance.
8. Reboot original instance

Then you're golden.

## What caused the problem?

Edit: rewritten 31/July based on [lp#1889509](#) comments)

On the 3 servers that broke, `apt-get update` did not prompt me for anything grub-related. Checking afterwards with `debconf-get-selections` I see that debconf was pre-configured to install grub on one device:

```
grub-pc      grub-pc/install_devices_disks_changed  multiselect      /dev/disk/by-id/nvme-
Amazon_Elastic_Block_Store_vol01e713272eb256s52-part1
```

That device symlink **is wrong**. It was either pre-seeded by cloud-init or by the AMI creator (I'm not sure). My servers need grub on `/dev/xvdf`, not `/dev/xvdf1`. The grub postinst script would have encountered the same failure I saw while recovering:

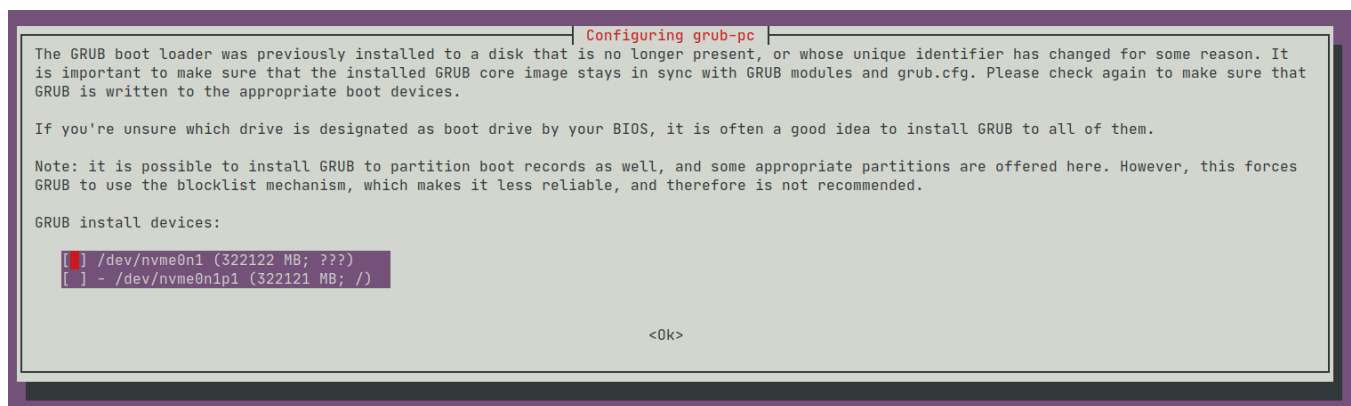
```
# grub-install /dev/xvdf1
grub-install: warning: File system `ext2` doesn't support embedding.
grub-install: warning: Embedding is not possible.  GRUB can only be installed in this setup by using
blocklists.  However, blocklists are UNRELIABLE and their use is discouraged..
grub-install: error: will not proceed with blocklists.
```

I must not have noticed this error in the wall of text scrolling past on the upgrade.

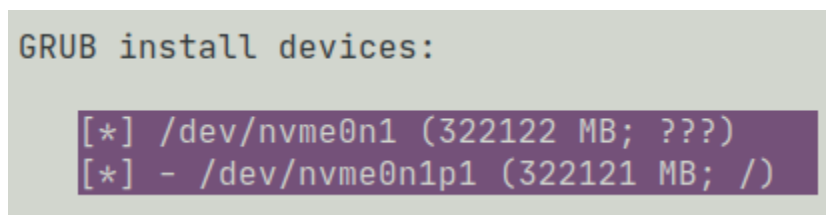
Grub should have failed hard, but didn't (now been filed as <https://bugs.launchpad.net/ubuntu/+source/grub2/+bug/1889556/>).

The result: grub is upgraded in my `/boot` partition, but the grub loader in `/dev/xvdb` is still old, and the mismatch causes the failure (hat tip to ~juliank on [lp#1889509](#)).

To reinforce this theory, we turn to 1 of the 4 sandboxes did not break. When I did `apt-get upgrade` the surviving server had previously demanded I give it some 'GRUB install devices':



I had decided to take debconf's advice, and installed grub on all devices:



and that saved me, at least on this one server.

But why did *this* update go wrong?

I upgrade OS packages on these servers every month, but looking at my `/var/log/dpkg.log` history, grub is very rarely updated. The last update was a full year ago:

```
root@usw-jira01:/var/log# zgrep 'upgrade grub-pc:amd64' dpkg.log*
dpkg.log:2020-07-30 04:06:23 upgrade grub-pc:amd64 2.02~beta2-36ubuntu3.23 2.02~beta2-36ubuntu3.26
dpkg.log.12.gz:2019-07-15 02:04:55 upgrade grub-pc:amd64 2.02~beta2-36ubuntu3.20 2.02~beta2-36ubuntu3.22
dpkg.log.7.gz:2019-12-09 03:06:58 upgrade grub-pc:amd64 2.02~beta2-36ubuntu3.22 2.02~beta2-36ubuntu3.23
```

So I posit that there is nothing directly wrong with this grub update specifically, but rather I (and a lot of other people) are hitting a **problem in the general Debian grub update process**; specifically, when grub fails to update with an error:

```
grub-install: warning: File system `ext2` doesn't support embedding.
grub-install: warning: Embedding is not possible. GRUB can only be installed in this setup by using blocklists.
However, blocklists are UNRELIABLE and their use is discouraged..
grub-install: error: will not proceed with blocklists
```

it *should* fail hard, but instead proceeds.

## How do I know if my server will break?

If your system boots with UEFI, you're fine.

If you're on Linode with default settings, you're fine.

For BIOS users, including AWS and other VPS hosters, run the following:

```
cd /tmp
curl -LOJ https://gist.github.com/jefft/76cf6c5f6605eee55df6079223d8babc/raw/
/bf0985bdbl1e2ef5fc74e2aee7ebf29c4eaf7199f/grubvalidator.sh
chmod +x grubvalidator.sh
./grubvalidator.sh
```

This script checks if your grub version includes a fix for [lp #1889556](#) and if not, checks if you are likely to experience boot problems.

I just upgraded a 20.04 Ubuntu laptop without drama, but those all use BIOS, not EFI. So far I've only seen this affect BIOS users.

*Edit: that is correct. Per ~juliank on [lp #1889509](#):*

This is not a problem on UEFI systems fwiw, as they do not use a small image in MBR and load the rest from /boot, but a single monolithic grub image in the ESP.

So here's my litmus test: run:

```
sudo apt-get install debconf-utils
sudo debconf-get-selections | awk '$1=="grub-pc" && $2 == "grub-pc/install_devices" {print $4}'
```

If you see nothing, you're fine: either:

- you're on EFI and unaffected
- you're on BIOS, but will be forcefully prompted to pick a device, as I was.

If you get a single device back:

```
sudo debconf-get-selections | awk '$1=="grub-pc" && $2 == "grub-pc/install_devices" {print $4}'
/dev/disk/by-id/nvme-Amazon_Elastic_Block_Store_vol01e713272eb256s52-part1
```

then you *may* be in trouble (not always - see below re. Linode). You'd best run `dpkg-reconfigure grub-pc` before rebooting, and install Grub on some other devices to be safe.

## Linode

~mnordhoff on [lp #1889509](#) says:

*On Linode, I believe you should be safe unless the kernel is set to "Direct Disk". It doesn't matter if the GRUB installation works if it's not being used.*

This is correct in my experience. My own servers are on linode, where the OS disk is /dev/sda and the above command prints:

```
root@radish-linode:~# sudo debconf-get-selections | awk '$1=="grub-pc" && $2 == "grub-pc/install_devices"
{print $4}'
/dev/disk/by-id/scsi-0QEMU_QEMU_HARDDISK_drive-scsi-disk-0,
```

When I run `dpkg-reconfigure grub-pc` and pick /dev/sda I get a scary warning, replicable from the command-line:

```
root@radish-linode:~# grub-install /dev/sda
Installing for i386-pc platform.
grub-install: warning: File system `ext2' doesn't support embedding.
grub-install: warning: Embedding is not possible. GRUB can only be installed in this setup by using
blocklists. However, blocklists are UNRELIABLE and their use is discouraged..
grub-install: error: will not proceed with blocklists.
```

However despite all the warning signs, my VM rebooted fine.

Edit: I filed a launchpad bug: <https://bugs.launchpad.net/ubuntu/+source/grub2/+bug/1889509>

Edit: The essential problem is that the `grub-pc` package doesn't fail in the presence of bad debconf data. That has been fixed per <https://bugs.launchpad.net/ubuntu/+source/grub2/+bug/1889556>.

Edit2: Updated 'What caused the problem' section with info from comments on the bug.